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

Last Updated: 2024-02-08

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

Florian Delehaye
Red Hat Customer Content Services
fdelehay@redhat.com

Marc Muehlfeld
Red Hat Customer Content Services

Filip Hanzelka
Red Hat Customer Content Services

Lucie Maňásková
Red Hat Customer Content Services

Aneta Šteflová Petrová
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

キーワード

1. FreeIPA. 2. Identity Management. 3. IdM. 4. IPA.

概要

ユーザーとマシン両方の ID およびポリシー管理は、ほとんどのエンタープライズ環境の中核となる機能です。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となる 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。 このガイドに加えて、Red Hat Enterprise Linux Identity Management に関するその他の機能およびサービスについては、以下のガイドを参照してください。 システムレベルの認証ガイド では、authconfig ユーティリティー、System Security Services Daemon (SSSD) サービス、Pluggable Authentication Module(PAM) フレームワーク、Kerberos、certmonger ユーティリティー、およびアプリケーションのシングルサインオン (SSO) など、ローカルシステムにおける認証の設定に使用できるアプリケーションおよびサービスについて説明します。 Windows 統合ガイド では、Identity Management を使用して Linux ドメインを Microsoft Windows Active Directory (AD) と統合する方法を説明します。このガイドでは、特に、直接的および間接的な AD 統合、SSSD を使用した Common Internet File System (CIFS) へのアクセス、および realmd システムのさまざまな側面について説明します。

目次

パート I. RED HAT IDENTITY MANAGEMENT の概要

本章では、Red Hat **Identity Management** の目的を説明します。また、**Identity Management** ドメイン (およびこのドメインに含まれるクライアントおよびサーバーのマシン) に関する基本的な情報も提供します。

第1章 RED HAT IDENTITY MANAGEMENT の概要

1.1. RED HAT IDENTITY MANAGEMENT の目的

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、および認可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、以下に対応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 とのネイティブな統合

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盤とします。独自のプロセスと設定がありますが、その基盤となる技術は Linux システムで十分に確立されおり、Linux 管理者から信頼されています。
- IdM サーバーおよびクライアントは Red Hat Enterprise Linux マシンです。ただし、IdM は Windows クライアントを直接サポートしていない場合でも、Active Directory 環境との統合が可能です。

注記

本ガイドでは、Linux 環境のみを対象とした IdM の使用方法を説明します。Active Directory との統合に関する詳細は、『Windows Integration Guide』を参照してください。

Linux マシンの Active Directory 環境への統合を可能にする Samba スイートの詳細は、『Windows 統合ガイド』のUsing Samba for Active Directory Integrationを参照してください。Samba をサーバーとして使用する場合は、サーバーを IdM ドメインに統合し、IdM または信頼された Active Directory ドメインに対して Samba サーバーに接続するユーザーを認証できないことに注意してください。

1.1.1. IdM による利点の例

複数の Linux サーバーによる ID およびポリシーの管理

IdM を使用しない場合 - 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、すべてのマシンでユーザーを管理し、認証ポリシーおよび認可ポリシーを別々に設定し、ローカルパスワードを維持します。

IdM を使用する場合 - IT 管理者は以下が可能になります。

- ID を一か所で管理 - IdM サーバー
- 複数のマシンで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、委譲など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 を使用しない場合 - 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異なる場合もあるため、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 を使用する場合 - 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。

- ユーザービリティーの向上
- 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ことによるセキュリティーリスクの低減
- ユーザーの生産性向上

Linux と Windows の混合環境の管理

IdM を使用しない場合 - Windows システムは Active Directory フォレストで管理されますが、開発、実稼働環境などのチームには多くの Linux システムがあります。Linux システムは、Active Directory 環境から除外されます。

IdM を使用する場合 - IT 管理者は以下が可能になります。

- ネイティブの Linux ツールを使用して Linux システムを管理する
- Linux システムを Windows システムと統合して、一元化されたユーザーストアを確保する
- Linux ベースを容易に拡張する
- Linux および Active Directory マシンの別々に管理し、Linux および Windows の管理者が環境を直接制御できる

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

Red Hat Directory Server などの標準 LDAP ディレクトリーは汎用ディレクトリーで、幅広いユースケースに適用するようにカスタマイズできます。

- スキーマー - ユーザー、マシン、ネットワークエンティティー、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ー
- 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

Identity Management (IdM) には、ID とその ID に関連する認証および認可ポリシーを管理するという特定の目的があります。

- スキーマー - 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
- 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ーを管理する ID および認証サーバー

Red Hat Directory Server と IdM では、基礎となるディレクトリーサーバーの技術は同じです。ただし、IdM は ID を管理するよう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の改善、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。

関連情報

- 『Red Hat Enterprise Linux Blog のブログ』投稿Identity Management or Red Hat Directory Server – Which One Should I Use?

1.2. IDENTITY MANAGEMENT ドメイン

Identity Management (IdM) ドメインは、同じ設定、ポリシー、およびアイデンティティーストアを共有するマシンのグループで設定されます。この共有プロパティーにより、ドメイン内のマシンは相互に認識、連携できます。

IdM の観点では、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動作する。
- サーバーに登録されている IdM クライアント

IdM サーバーは、それ自体に登録している IdM クライアントでもあるため、サーバーマシンはクライアントと同じ機能を提供します。

IdM は、IdM サーバーおよびクライアントとしての Red Hat Enterprise Linux マシンに対応します。

注記

本ガイドでは、Linux 環境で IdM を使用する方法を説明します。Active Directory との統合に関する詳細は、『Windows Integration Guide』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 情報およびポリシー情報の中央リポジトリーとして機能します。また、ドメインメンバーが使用するサービスをホストします。IdM は、IdM 関連の全サービスを一元的に管理する管理ツールを提供します (IdM Web UI およびコマンドラインユーティリティー)。

IdM サーバーのインストールの詳細は、*2章Identity Management サーバー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。

冗長性と負荷分散をサポートするため、データと設定を IdM サーバーから別のサーバーに複製できます (初期サーバーの *レプリカ*)。サーバーとそのレプリカを設定して、各種サービスをクライアントに提供できます。IdM レプリカの詳細は、*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 サーバーがホストするサービス

以下のサービスの多くは、IdM サーバーへのインストールが必須というわけではありません。たとえば、認証局 (CA)、DNS サーバー、または Network Time Protocol (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できます。

Kerberos: krb5kdc および kadmin

IdM は、シングルサインオンに対応する Kerberos プロトコルを使用します。Kerberos では、正しいユーザー名とパスワードを一度提示するだけで済み、システムから認証情報を再度求められることなく IdM サービスにアクセスできます。

- **Kerberos** は 2 つの部分に分類されます。
  - **krb5kdc** サービス。Kerberos 認証サービスおよびキー配布センター (KDC) デーモンです。
  - **kadmin** サービス。Kerberos V5 データベース管理プログラムです。

  Kerberos の仕組みの詳細については、『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。
- IdM で Kerberos を使用して認証する方法は、 「Kerberos を使用した IdM へのログイン」 を参照してください。
- IdM で Kerberos を管理する方法は、29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: dirsrv

IdM の内部 **LDAP ディレクトリーサーバー** インスタンスは、 Kerberos、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 などの情報をはじめとした、IdM 情報をすべて保存します。

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にしています。ただし、IdM 固有のタスクに合わせて調整されます。

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: pki-tomcatd

統合 **認証局 (CA)** は、Red Hat Certificate System と同じテクノロジーをベースにしています。**pki** は、証明書システムサービスにアクセスするコマンドラインインターフェイスです。

- さまざまな CA 設定を備えた IdM サーバーをインストールする方法は、 「使用する CA 設定の決定」 を参照してください。

注記

本書では、実装について言及するときはこのコンポーネントを証明書システムと、実装が提供するサービスを言及するときは証明局と呼びます。

Red Hat Certificate System、スタンドアロン Red Hat 製品の詳細は、Product Documentation for Red Hat Certificate Systemを参照してください。

DNS (Domain Name System): 名前

IdM は、動的サービス検出に **DNS** を使用します。IdM クライアントのインストールユーティリティーは、DNS からの情報を使用して、クライアントマシンを自動的に設定できます。クライアントを IdM ドメインに登録したら、クライアントは DNS を使用してドメイン内の IdM サーバーおよびサービスを検索します。

Red Hat Enterprise Linux の DNS (Domain Name System) プロトコルの **BIND** (Berkeley Internet

Name Domain) 実装には、**名前付き** の DNS サーバーが含まれています。**named-pkcs11** は、PKCS#11 暗号化標準に対するネイティブサポートありで構築された **BIND** DNS サーバーのバージョンです。

- サービス検出の詳細は、『System-Level Authentication Guide』のConfiguring DNS Service Discoveryを参照してください。
- DNS サーバーの詳細は、『Red Hat Enterprise Linux ネットワークガイド』のBINDを参照してください。
- IdM で DNS を使用する方法と重要な前提条件については、「ホスト名および DNS 設定」を参照してください。
- 統合 DNS の有無に関わらず、IdM サーバーをインストールする方法は、「統合 DNS を使用するかどうかの決定」 を参照してください。

ネットワークタイムプロトコル: ntpd

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用して有効性を判断し、リプレイ攻撃を防ぎます。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 **Network Time Protocol (NTP)** を使用し、**ntpd** サービスを介してネットワークでクロックを同期します。NTP では、中央サーバーが権威クロックとして機能し、クライアントはサーバークロックと同じ時刻を使用するように同期します。IdM サーバーは、サーバーのインストールプロセス時に IdM ドメインの NTP サーバーとして設定されます。

注記

仮想マシンにインストールされる IdM サーバーで NTP サーバーを実行すると、一部の環境で時間同期が不正確になることがあります。問題が発生する可能性を回避するには、仮想マシンにインストールされている IdM サーバーで NTP を実行しないでください。仮想マシンで NTP サーバーの信頼性に関する詳細は、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。

Apache HTTP Server: httpd

Apache HTTP Web サーバー には、IdM Web UI があり、認証局とその他の IdM サービスの間の通信も管理します。

- 詳細は、『System Administrator's Guide』のThe Apache HTTP Serverを参照してください。

Samba / Winbind:smb、winbind

Samba は、Red Hat Enterprise Linux に、Common Internet File System (CIFS) プロトコルとも呼ばれる Server Message Block (SMB) プロトコルを実装します。**smb** サービス経由で SMB プロトコルを使用すると、ファイル共有や共有プリンターなどのサーバーのリソースにアクセスできます。Active Directory (AD) 環境で信頼を設定している場合には、**Winbind** サービスが IdM サーバーと AD サーバー間の通信を管理します。

- 詳細は、『System Administrator's Guide』のSambaを参照してください。
- 詳細は、『System-Level Authentication Guide』のWinbindを参照してください。

ワンタイムパスワード (OTP) 認証:ipa-otpd

ワンタイムパスワード (OTP) は、2 要素認証の一部として、認証トークンがセッション 1 回だけ使用できるように生成するパスワードです。OTP 認証は、**ipa-otpd** サービスを介して Red Hat Enterprise Linux に実装されています。

- OTP 認証の詳細は、「ワンタイムパスワード」 を参照してください。

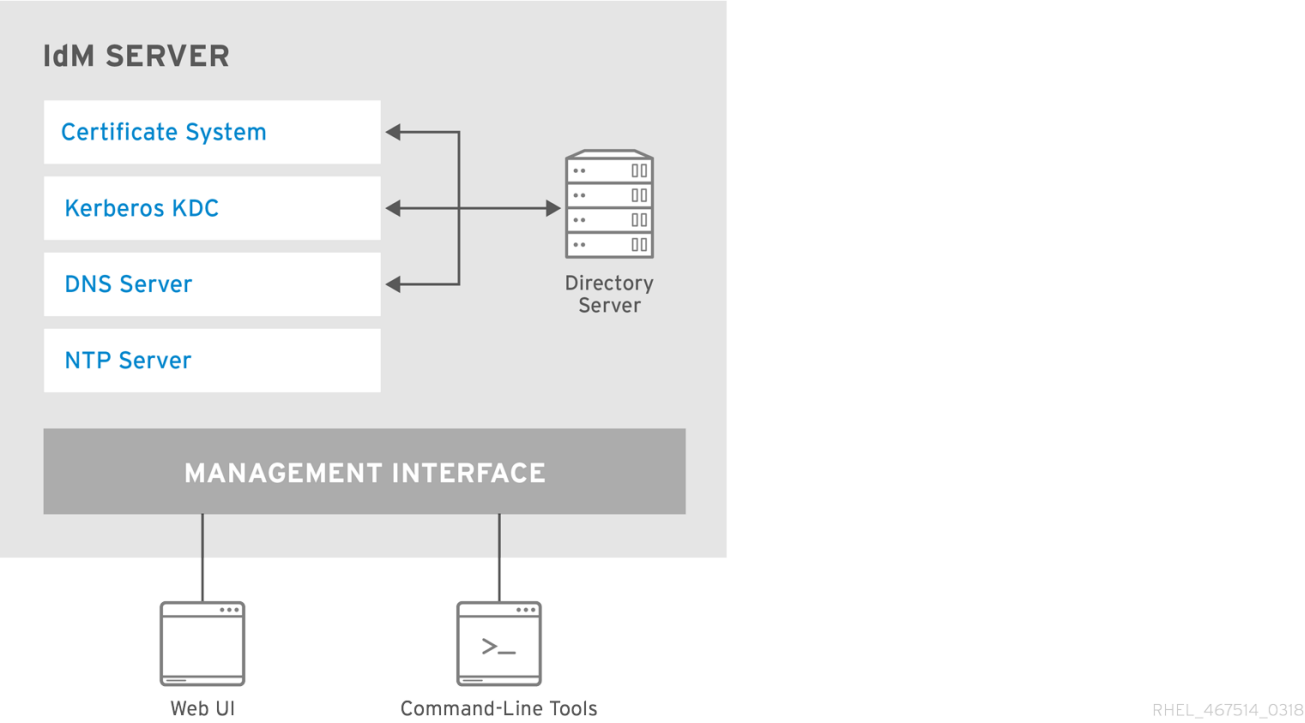
custodia: ipa-custodia

Custodia はシークレットサービスプロバイダーで、パスワード、キー、トークン、証明書などのシークレット資料へのアクセスを保存し、共有します。

OpenDNSSEC: ipa-dnskeysyncd

OpenDNSSEC は、DNSSEC (DNS Security Extensions) キーおよびゾーンの署名の記録プロセスを自動化する DNS マネージャーです。**ipa-dnskeysyncd serv** サービスは、IdM Directory Server と OpenDNSSEC との間の同期を管理します。

図1.1 Identity Management サーバー: サービスの統合



1.2.2. Identity Management クライアント

IdM クライアントは、IdM ドメイン内で動作するように設定されたマシンです。ドメインリソースにアクセスするために IdM サーバーと対話します。たとえば、クライアントは、サーバーに設定した Kerberos ドメインに属し、サーバーが発行する証明書およびチケットを受け取り、その他の一元管理サービスを使用して認証および認可を行います。

IdM クライアントでは、ドメインの一部として操作するための専用のクライアントソフトウェアは必要ありません。必要なのは、Kerberos や DNS など、特定のサービスおよびライブラリーのシステム設定を適切に指定するだけです。この設定では、クライアントマシンが IdM サービスを使用するように指定します。

IdM クライアントのインストールは、*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: sssd

SSSD (**System Security Services Daemon**) は、ユーザー認証およびキャッシュ認証情報を管理するクライアント側のアプリケーションです。

キャッシュを使用すると、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りした場合に、ローカルシステムが通常の認証操作を継続できるようになります。

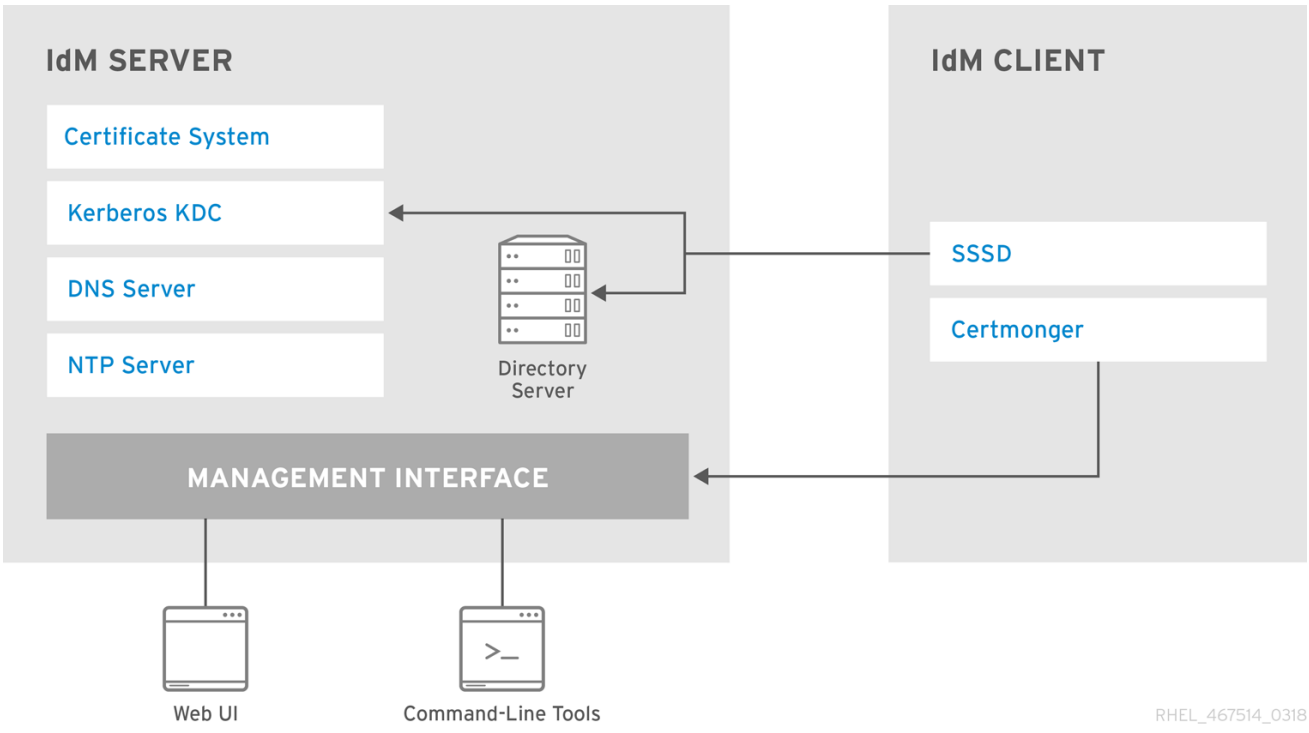
詳細は、『System-Level Authentication Guide』のConfiguring SSSDを参照してください。SSSD は、Windows Active Directory (AD) にも対応しています。AD で SSSD を使用する方法は、『Windows Integration Guide』のUsing Active Directory as an Identity Provider for SSSD を参照してください。

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。このサービスは、システム上のサービスに対して新しい証明書を要求できます。

詳細は、『System-Level Authentication Guide』のWorking with certmongerを参照してください。

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

このパートでは、**ID 管理** デプロイメントのプランニング方法と、**ID 管理** サーバー、クライアント、およびレプリカのインストール方法を説明します。

第2章 IDENTITY MANAGEMENT サーバーのインストールおよびアンインストール

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、管理します。IdM サーバーを設定するには、以下を行う必要があります。

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

Red Hat は、ドメイン内に複数のドメインコントローラーを設定して、負荷分散と冗長性を確保することを強く推奨します。これらの追加サーバーは、初期マスター IdM サーバーの *レプリカ* です。

本章では、最初に初期の IdM サーバーをインストールする方法を説明します。初期サーバーからレプリカをインストールする方法は、*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。

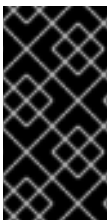
2.1. サーバーのインストールの前提条件

2.1.1. 最小ハードウェア要件

Identity Management (IdM) を実行するには、サーバーに少なくとも以下のハードウェア設定が必要です。

- 1 x (仮想) CPU コア
- 2 GB RAM

  少ないメモリーで IdM をインストールできる場合でも、IdM の更新などの一部の操作には 4 GB 以上の RAM が必要です。
- 10 GB のハードディスク



重要

データベースに保存されているデータ量によっては、IdM にはより多くのリソースが必要になります (特に RAM)。詳細は、「ハードウェア推奨事項」を参照してください。必要なハードウェアリソースは、サーバーの実稼働環境のワークロード、または Active Directory で信頼が設定されている場合など、他の要素に依存します。

2.1.2. ハードウェア推奨事項

ハードウェアでは、RAM の容量を適切に確保することが最も重要になります。必要な RAM 容量を判断するには、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 3 GB の RAM と 1 GB のスワップ領域を割り当てます。
- 100,000 ユーザーおよび 50,000 グループには、最低 16 GB の RAM と 4 GB のスワップ領域を割り当てます。

注記

基本的なユーザーエントリーまたは証明書のあるシンプルなホストエントリーのサイズは、約 5 - 10 KiB です。

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ために、基礎となる Directory Server を調整してパフォーマンスを向上させることができます。詳細は、『Red Hat Directory Server パフォーマンスチューニングガイド』 を参照してください。

2.1.3. システム要件

Identity Management は、Red Hat Enterprise Linux 7 でサポートされています。DNS、Kerberos、Directory Server などのサービスのカスタム設定を行わずに、クリーンなシステムに IdM サーバーをインストールします。

重要

パフォーマンスおよび安定性の理由から、Red Hat は、IdM サーバーに他のアプリケーションやサービスをインストールしないことを推奨します。たとえば、特に LDAP オブジェクトの数が多くなると、IdM サーバーはシステムからなくなる可能性があります。また、IdM はシステムに統合され、サードパーティーのアプリケーションが IdM に依存する設定ファイルを変更すると、IdM が破損される可能性があります。

IdM サーバーのインストールは、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を **/var/lib/ipa/sysrestore/** にバックアップします。

Name Service Cache Daemon (NSCD) 要件

Red Hat は、Identity Management マシンで NSCD を無効にすることを推奨します。または、NSCD を無効にできない場合は、SSSD でキャッシュされないマップの NSCD のみを有効にします。

NSCD と SSSD サービスはいずれもキャッシュを実行し、システムが両方のサービスを同時に使用すると問題が発生する可能性があります。NSCD と SSSD 間の競合を回避する方法については、System-Level Authentication Guide を参照してください。

システムで IPv6 を有効にする必要がある

IdM サーバーで、カーネルで IPv6 プロトコルが有効になっている必要があります。IPv6 は、Red Hat Enterprise Linux 7 システムでデフォルトで有効になっている点に注意してください。

IPv6 を無効にする場合は、Red Hat ナレッジベースの How do I disable or enable the IPv6 protocol in Red Hat Enterprise Linux? で説明されているように、IPv6 プロトコルを再度有効にします。

注記

IdM では、クライアントとして登録するホストのカーネルで IPv6 プロトコルを有効にする必要はありません。たとえば、内部ネットワークで IPv4 プロトコルのみを使用する場合には、System Security Services Daemon (SSSD) が IPv4 だけを使用して IdM サーバーと通信するように設定できます。**/etc/sssd/sssd.conf** ファイルの **[domain/_NAME_]** セクションに次の行を追加して、これを設定できます。

```
lookup_family_order = ipv4_only
```

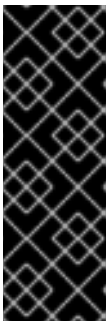
lookup_family_order の詳細は、**sssd.conf(5)** man ページを参照してください。

2.1.4. FIPS 環境にサーバーをインストールする場合の前提条件

Red Hat Enterprise Linux 7.4 以降を使用して環境を設定する環境では、以下を行います。

- 連邦情報処理規格 (FIPS) モードが有効になっているシステムに、新しい IdM サーバーまたはレプリカを設定できます。インストールスクリプトは、FIPS が有効になっているシステムを自動的に検出し、管理者の介入なしに IdM を設定します。

  オペレーティングシステムで FIPS を有効にするには、『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

以下を行うことはできません。

- FIPS モードを無効にしてから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。
- FIPS モードを無効にして既存の IdM サーバーを使用する場合に FIPS モードでレプリカをインストールする。

Red Hat Enterprise Linux 7.3 以前を使用して設定された環境では、以下を行います。

- IdM では FIPS モードをサポートされません。IdM サーバーまたはレプリカをインストールする前に FIPS を無効にしてインストール後に有効にしないでください。

FIPS モードの詳細は、『セキュリティーガイド』の連邦情報処理標準 (FIPS)を参照してください。

2.1.5. ホスト名および DNS 設定

警告

以下の点を確認し、十分注意してください。

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

この要件は、統合 DNS サービスがある IdM サーバーと、DNS なしでインストールした IdM サーバーに適用されます。DNS レコードは、稼働中の LDAP ディレクトリーサービス、Kerberos、Active Directory 統合など、ほぼすべての IdM ドメイン機能で必須となります。

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

.company など、単一ラベルのドメイン名を使用しないでください。IdM ドメインは、トップレベルドメインと、1 つ以上のサブドメイン (**example.com** や **company.example.com** など) で設定する必要があります。

サーバーホストは、DNS サーバーが IdM 内に統合されているか、外部でホストされるかに関係なく、DNS を適切に設定する必要があります。

Identity Management では、サービスレコードに別の DNS ドメインを使用する必要があります。DNS レベルの競合を回避するため、*プライマリー IdM DNS ドメイン*(IdM Kerberos 名の小文字バージョン) では、他の IdM や AD ドメインなどの他のシステムと共有できません。

プライマリー IdM DNS ドメインには、標準の IdM サービス用の独自の SRV レコードを含める必要があります。必要なレコードは以下のとおりです。

- _kerberos._tcp.*domain_name* と _kerberos._udp.*domain_name* 両方の SRV レコード
- _ldap._tcp.*domain_name* の SRV レコード
- _kerberos.*domain_name* の TXT レコード

登録済みのクライアントで **ipa** コマンドラインツール経由で提供されるサービスを検索すると、**/etc/ipa/default.conf** ファイルの **xmlrpc_uri** パラメーターで指定したサーバーを検索します。必要な場合には、同じファイルにある **domain** パラメーターに指定している IdM DNS ドメイン名も検索し、そのドメインの _ldap._tcp.*domain_name* SRV レコードを確認して、検索しているサーバーを特定します。**/etc/ipa/default.conf** ファイルにドメインがない場合に、クライアントはファイルの **xmlrpc_uri** パラメーターに設定したサーバーとのみ通信します。

IdM クライアントおよびサーバーのホスト名は、プライマリー DNS ドメインの一部にする必要はありません。ただし、Active Directory (AD) を使用する信頼環境では、IdM サーバーのホスト名は IdM 所有ドメイン、IdM レルムに関連付けられたドメインに所属する必要があり、AD 所有ドメイン、信頼された AD レルムに関連付けられたドメインには含めないようにする必要があります。信頼の観点から見ると、この関連付けは レルムドメイン を使用して管理されます。

Active Directory DNS ドメインからのホスト名を使用して IdM クライアントにアクセスするようにユーザーを設定し、クライアント自体が IdM に参加するように設定する方法は、『Windows 統合ガイド』のActive Directory DNS ドメインの IdM クライアント を参照してください。

サーバーのホスト名の確認

ホスト名は、完全修飾ドメイン名 (例: **server.example.com**) である必要があります。

重要

.company など、単一ラベルのドメイン名を使用しないでください。IdM ドメインは、トップレベルドメインと、1 つ以上のサブドメイン (example.com や company.example.com など) で設定する必要があります。

完全修飾ドメイン名は、以下の条件を満たす必要があります。

- 数字、アルファベット文字、およびハイフン (-) のみが使用される有効な DNS 名である。ホスト名でアンダーライン (_) を使用すると DNS が正常に動作しません。
- すべてが小文字である。大文字は使用できません。
- 完全修飾ドメイン名は、ループバックアドレスを解決できません。**127.0.0.1** ではなく、マシンの公開 IP アドレスを解決する必要があります。

その他の推奨命名プラクティスは『Red Hat Enterprise Linux Security Guide』のRecommended Naming Practicesを参照してください。

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** 以外である必要があります。

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** コマンドを実行すると、IPv4 アドレスと IPv6 アドレスの両方が表示されます。
   - 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、設定した IPv4 アドレスは **192.0.2.1** です。
   - IPv6 アドレスは、**inet6** で始まる行に表示されます。この手順は、**scope global** の IPv6 アドレスのみが対象です。以下の例では、返される IPv6 アドレスは **2001:DB8::1111** です。

   ```
   [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** ユーティリティーを使用して、正引き DNS 設定を確認し、ホスト名を追加します。

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** により返される 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。このコマンドにアドレスを返されるアドレスは、**ip addr show** で返される 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からといって、設定が間違っているわけではありません。出力されないのは、サーバーのマシンの DNS に IPv6 アドレスが設定されていないことを意味します。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** ユーティリティーを使用して、逆引き DNS 設定 (PTR レコード) を確認し、IP アドレスを追加します。

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

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

2. 前の手順で **dig +short -x *server.example.com* AAAA** コマンドにより IPv6 アドレスが返されていた場合は、**dig** を使用して、IPv6 アドレスのクエリーを行います。ここでも、サーバーのホスト名がコマンド出力に表示される必要があります。以下に例を示します。

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

注記

前の手順で **dig +short *server.example.com* AAAA** コマンドにより IPv6 アドレスが返されなかった場合は、AAAA レコードのクエリーを実行しても、何も出力されません。この場合、これは正常な動作で、誤った設定を示すものではありません。

前の手順の **dig +short *server.example.com*** で IP アドレスが返されても異なる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 設定が正しくありません。

DNS フォワーダーの標準コンプライアンスの確認

統合 DNS で IdM を設定する場合は、DNSSEC (DNS Security Extensions) レコード検証の使用を推奨します。他のサーバーから署名済み DNS レコードを検証することで、偽装アドレスから IdM インストールを保護します。ただし、DNSSEC の検証は、IdM を正常にインストールするためのハード要件ではありません。

IdM インストーラーは、デフォルトで DNSSEC レコードの検証を有効にします。DNSSEC の検証に成功すると、DNSSEC が適切に設定されているフォワーダーが必要です。インストール時に、IdM はグローバルフォワーダーを確認し、フォワーダーが DNSSEC に対応していない場合は、フォワーダーで DNSSEC 検証が無効になります。

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) および DNSSEC の規格に準拠していることを確認します。

```
$ 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ます。

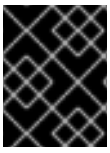
- 状態 - **NOERROR**
- フラグ - **ra**
- EDNS フラグ - **do**
- **ANSWER** セクションには **RRSIG** レコードが必要です。

出力に上記のいずれかの項目がない場合は、使用している DNS フォワーダーのドキュメントに従い、EDNS0 と DNSSEC に対応し、ともに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、**dnssec-enable yes;** オプションが **/etc/named.conf** ファイルに設定されている必要があります。

たとえば、想定される出力は次のようになります。

```
;; ->>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。以前に **/etc/hosts** を変更したことがある場合は、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、適切に設定された **/etc/hosts** ファイルの例になります。ホストの IPv4 および IPv6 の localhost エントリーを適切にリスト表示し、その後に IdM サーバーの IP アドレスとホスト名を最初のエントリーとしてリスト表示します。IdM サーバーのホスト名は、**localhost** エントリーには追加できないこと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
```

2.1.6. ポートの要件

IdM はサービスとの通信に多くのポートを使用します。IdM を機能させるには、これらのポートを開放して利用できるようにしておく必要があります。別のサービスを使用したり、ファイアウォールでブロックしたりしないようにしてください。

- 必要なポートのリストは、「必須ポートのリスト」 を参照してください。
- 必要なポートに対応する **firewalld** サービスのリストは、「firewalld サービスのリスト」 を参照してください。

必須ポートのリスト

表2.1 Identity Management ポート

サービス	ポート	プロトコル
HTTP/HTTPS	80、443	TCP
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 はポート 80 および 389 を使用しますが問題ありません。

- ポート 80 (HTTP) は、Online Certificate Status Protocol (OCSP) 応答および証明書失効リスト (CRL) の提供に使用されます。いずれもデジタル署名されているため、中間者攻撃に対してセキュリティーが保護されます。
- ポート 389 (LDAP) は、暗号化に STARTTLS および GSSAPI を使用します。

さらに、IdM はポート 8080 でリッスンでき、一部のインストールはポート 8443 および 749 でもリッスンできます。ただし、これらの 3 つのポートは内部でのみ使用されます。IdM が開放したままであっても、外部からアクセスできません。ポート 8080、8443、および 749 を開放せず、ファイアウォールでブロックした状態にすることが推奨されます。

firewalld サービスのリスト

表2.2 firewalld サービス

サービス名	詳細は、次を参照してください。
freeipa-ldap	**/usr/lib/firewalld/services/freeipa-ldap.xml**
freeipa-ldaps	**/usr/lib/firewalld/services/freeipa-ldaps.xml**

サービス名	詳細は、次を参照してください。
dns	**/usr/lib/firewalld/services/dns.xml**

必要なポートの開放

1. **firewalld** サービスが実行中である必要があります。
   - **firewalld** が実行中であることを確認するには、次のコマンドを実行します。

     ```
     # 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、次のコマンドを実行します。

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

2. **firewall-cmd** ユーティリティーを使用して必要なポートを開きます。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のポートをファイアウォールに追加します。たとえば、デフォルトゾーンでポートを開くには、次のコマンドを実行します。

      ```
      # 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、デフォルトゾーンでポートを開くには、次のコマンドを実行します。

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

   firewall-cmd を使用してシステムでポートを開く方法は、『Security Guide』のModifying Settings in Runtime and Permanent Configuration using CLIか、firewall-cmd(1)の man ページを参照してください。

3. **firewall-cmd** 設定を再ロードして、変更が即座に反映されるようにします。

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

   実稼働システムで **firewalld** を再ロードすると、DNS の接続がタイムアウトになる可能性があることに注意してください。『Security Guide』のModifying Settings in Runtime and Permanent Configuration using CLIも参照してください。必要な場合は、以下の例のように **firewall-cmd** コマンドで **--runtime-to-permanent** オプションを指定して、タイムアウトが発生しないようにし、変更を永続化します。

   ```
   # firewall-cmd --runtime-to-permanent --add-port=
   {80/tcp,443/tcp,389/tcp,636/tcp,88/tcp,88/udp,464/tcp,464/udp,53/tcp,53/udp,123/udp}
   ```

4. **オプション:**ポートが現在利用可能であるかを確認するには、**nc**、**telnet** または **nmap** ユーティリティーを使用して、ポートへの接続またはポートスキャンの実行を行います。

注記

さらに、着信および送信トラフィックの両方でネットワークベースのファイアウォールを開く必要があることに注意してください。

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 サービスのないサーバーに必要な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 サービスを使用するサーバーに必要な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
```

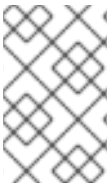
注記

DNS がユースケースに適しているかどうかを確認するには、「統合 DNS を使用するかどうかの決定」 を参照してください。

ipa-server パッケージは、以下のように、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。

- Directory Server LDAP サービスの 389-ds-base
- Kerberos サービスの krb5-server パッケージ
- IdM 固有の各種ツール

2.3. IDM サーバーのインストール: 概要



注記

以下のセクションのインストール手順と例は合わせて使用できます。手順と例を組み合わせて、必要な結果を得ることができます。たとえば、統合 DNS と、外部でホストされるルート CA のあるサーバーをインストールできます。

ipa-server-install ユーティリティーは、IdM サーバーをインストールし、設定します。

サーバーをインストールする前に、以下のセクションを参照してください。

- 「統合 DNS を使用するかどうかの決定」
- 「使用する CA 設定の決定」

ipa-server-install ユーティリティーは、非対話型インストールモードで、自動化および無人サーバーの設定が可能です。詳細は、「非対話形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 インストールスクリプトにより、**/var/log/ipaserver-install.log** にログファイルが作成されます。ログは、インストールに失敗した時の問題特定に役立ちます。

2.3.1. 統合 DNS を使用するかどうかの決定

IdM は、統合 DNS があるサーバーまたは統合 DNS のないサーバーのインストールに対応します。

統合 DNS サービスを備えた IdM サーバー

IdM が提供する統合 DNS サーバーは、汎用 DNS サーバーとして使用するように設計されていません。IdM のデプロイメントとメンテナンスに関連する機能のみに対応します。高度な DNS 機能の一部には対応していません。

Red Hat では、IdM デプロイメントにおける基本的な使用のために IdM 統合 DNS を強く推奨します。IdM サーバーが DNS も管理する場合には、DNS とネイティブの IdM ツールが密接に統合されるため、DNS レコード管理の一部が自動化できます。

IdM サーバーがマスター DNS サーバーとして使用されている場合でも、その他の外部 DNS サーバーはスレーブサーバーとしても使用できます。

たとえば、環境がすでに Active Directory 統合 DNS サーバーなどの別の DNS サーバーを使用している場合には、IdM のプライマリードメインのみを IdM 統合 DNS に委譲できます。DNS ゾーンを IdM 統合 DNS に移行する必要はありません。

注記

SAN (Subject Alternative Name) 拡張機能の IP アドレスを使用して IdM クライアントの証明書を発行する必要がある場合は、IdM 統合 DNS サービスを使用する必要があります。

統合 DNS のあるサーバーをインストールするには、「統合 DNS を使用した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

外部 DNS サーバーは、DNS サービスを提供するために使用されます。以下の状況では、DNS を使用せずに 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 のスコープを超える高度な DNS 機能が必要な場合
- 外部の DNS サーバーを使用できるようにする、適切に確立された 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には、「統合 DNS を使用しないサーバーのインストール」 を参照してください。

重要

お使いのシステムが 「ホスト名および DNS 設定」 に記載されている DNS 要件を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には、ほとんどの DNS レコードのメンテナンスは自動化されます。必要な作業は以下のとおりです。

- 親ドメインから IdM サーバーに正しい委譲を設定する

たとえば、IdM ドメイン名が **ipa.example.com** の場合は、**example.com** ドメインから適切に委譲する必要があります。

注記

以下のコマンドを使用して委譲を確認できます。

```
# 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。委譲が正しい場合は、このコマンドは DNS サーバーがインストールされている IdM サーバーを表示します。

外部 DNS サーバーを使用する場合は、以下を行う必要があります。

- DNS サーバーに新規ドメインを手動で作成する
- IdM インストーラーが生成したゾーンファイルのレコードをもとに新しいドメインを手動で入力する
- レプリカのインストールまたは削除後にレコードを手動で更新する。また、Active Directory 信頼が設定された後など、サービス設定の変更後にレコードを手動で更新する。

DNS アンプ攻撃の防止

IdM 統合 DNS サーバーのデフォルト設定により、すべてのクライアントが DNS サーバーに再帰クエリーを発行できます。信頼できないクライアントが含まれるネットワークにサーバーがデプロイされている場合は、サーバー設定を変更して、承認されたクライアントにのみ再帰を制限します。[1]

承認された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、サーバーの **/etc/named.conf** ファイルに適切なアクセス制御リスト (ACL) ステートメントを追加します。以下に例を示します。

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

2.3.2. 使用する CA 設定の決定

IdM は、統合 IdM 認証局 (CA) を使用した、または使用しないサーバーのインストールに対応します。

統合 IdM CA を使用するサーバー

これは、ほとんどのデプロイメントに適したデフォルト設定です。証明書システムは、*CA 署名* 証明書を使用して、IdM ドメインで証明書を作成して署名します。

警告

Red Hat は、複数のサーバーに CA サービスをインストールすることを強く推奨します。CA サービスを含む初期サーバーのレプリカのインストールは、「CA のあるレプリカのインストール」 を参照してください。

CA を 1 台のサーバーにのみインストールすると、CA サーバーが失敗した場合に CA 設定を復元できる機会なしに CA 設定が失われるリスクがあります。詳細は「失われた CA サーバーの復旧」を参照してください。

IdM CA 署名証明書は、**自己署名** と呼ばれるルート CA か、外部 CA で署名することもできます。

IdM CA はルート CA です。

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には、「統合 DNS を使用したサーバーのインストール」 および 「統合 DNS を使用しないサーバーのインストール」 を参照してください。

外部 CA はルート CA です。

IdM CA は、外部 CA の下位局になります。ただし、IdM ドメインのすべての証明書は、証明書システムインスタンスにより引き続き発行されます。

外部 CA は、Verisign や Thawte などの企業 CA またはサードパーティーの CA です。外部 CA は、ルート CA または下位 CA です。IdM ドメイン内で発行された証明書には、証明書を発行できるドメインなど、外部ルート CA 証明書または中間 CA 証明書が指定する制限が適用される可能性があります。

外部ホストのルート CA を備えたサーバーをインストールするには、「外部 CA をルート CA として使用するサーバーのインストール」 を参照してください。

CA を使用しないサーバー

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がサーバーに証明書サービスをインストールできない場合など、非常にまれなケースに適しています。

インストール前に、サードパーティーの認証局からこれらの証明書を要求する必要があります。

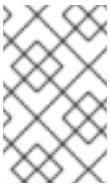
- LDAP サーバー証明書および秘密鍵
- Apache サーバー証明書および秘密鍵
- LDAP および Apache のサーバー証明書を発行した CA の完全な CA 証明書チェーン

警告

統合 IdM CA を使用せずに証明書を管理すると、メンテナンスの負担が大きくなります。たとえば、IdM サーバーの Apache Web サーバーおよび LDAP サーバー証明書を手動で管理する必要があります。これには、以下のものが含まれます。

- 証明書を作成してアップロードする。
- 証明書の有効期限を監視する。**certmonger** サービスでは、統合 CA を使用せずに IdM がインストールされている場合には証明書は追跡されない点に注意してください。
- 有効期限が切れる前に証明書を更新してサービスが停止されないようにする。

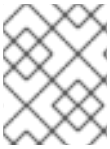
統合 CA のないサーバーをインストールするには、「CA なしでのインストール」を参照してください。



注記

CA を使用せずに IdM ドメインをインストールする場合は、後で CA サービスをインストールできます。既存の IdM ドメインに CA をインストールするには、「既存の IdM ドメインへの CA のインストール」 を参照してください。

2.3.3. 統合 DNS を使用したサーバーのインストール



注記

DNS または CA 設定が適切かどうかが不明な場合は、「統合 DNS を使用するかどうかの決定」 および 「使用する 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時に以下の情報を指定します。

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ます。

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

DNS 転送を使用するかどうか不明な場合は、「DNS 転送の管理」 を参照してください。

逆引き DNS ゾーン

以下の逆引き DNS ゾーン設定がサポートされます。

- IdM DNS で作成する必要がある逆引きゾーンの自動検出 (対話型インストールの場合は **--auto-reverse** オプションのデフォルト設定)
- 逆引きゾーンの自動検出なし (対話型インストールの **--no-reverse** オプション)

--auto-reverse オプションが設定されている場合には、**--allow-zone-overlap** オプションは無視されます。オプションの組み合わせの使用:

```
$ ipa-server-install --auto-reverse --allow-zone-overlap
```

そのため、別の DNS サーバーなどで、既存の DNS ゾーンと重複する **逆引きゾーン** は作成されません。

非対話的なインストールでは **--setup-dns** オプションも追加します。

例2.1統合 DNS を使用した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
- 統合 IdM CA をルート CA とするサーバー (デフォルトの CA 設定)

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

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

2. スクリプトにより、統合 DNS サービスの設定が求められます。**yes** を入力します。

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

3. このスクリプトで、必要な設定を入力するように求められます。
   - 括弧内のデフォルト値をそのまま使用するには、**Enter** を押します。
   - 提案されたデフォルト値とは異なる値を指定するには、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 レルム名がプライマリー DNS ドメイン名と同じで、すべて大文字にすることを強く推奨します。たとえば、プライマリー DNS ドメインが **ipa.example.com** の場合は、Kerberos レルム名として **IPA.EXAMPLE.COM** を使用します。

異なる命名プラクティスを使用すると、Active Directory の信頼を使用できなくなるなど、悪影響をもたらします。

4. Directory Server のスーパーユーザー (**cn=Directory Manager**)、および IdM システムユーザーアカウント (**admin**) 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スクリプトプロンプトが表示されます。

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

- DNS フォワーダーを設定するには、**yes** を入力して表示された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により、インストールした IdM サーバーの **/etc/named.conf** ファイルに、フォワーダーの IP アドレスが追加されます。

  - 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** の説明を参照してください。
  - 詳細は、「フォワードポリシー」 も参照してください。

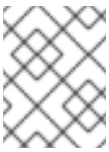
- DNS 転送を使用しない場合は、**no** と入力します。

6. その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スクリプトプロンプトが出されます。

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

検索を実行して欠落している逆引きゾーンが見つかると、PTR レコードの逆引きゾーンを作成するかどうかが尋ねられます。

```
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

オプションで、逆引きゾーンの管理に IdM を使用できます。代わりに、この目的で外部 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. インストールスクリプトにより、サーバーが設定されます。動作が完了するまで待ちます。

9. 親ドメインから IdM DNS ドメインに DNS 委譲を追加します。たとえば、IdM DNS ドメインが **ipa.example.com** の場合は、ネームサーバー (NS) レコードを親ドメイン **example.com** に追加します。

重要

この手順は、IdM DNS サーバーをインストールするたびに繰り返す必要があります。

このスクリプトでは、CA 証明書をバックアップして必要なネットワークポートが開いていることの確認が推奨されます。IdM ポートの要件と、これらのポートを開く方法は、「ポートの要件」 を参照してください。

新しいサーバーをテストするには、以下を行います。

1. admin 認証情報を使用して Kerberos レルムに対して認証します。これにより、**admin** が適切に設定され、Kerberos レルムにアクセスできることを確認します。

```
# kinit admin
```

2. **ipa user-find** などのコマンドを実行します。新しいサーバーでは、このコマンドは、設定したユーザー (**admin**) のみを出力します。

```
# 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 を使用しないサーバーのインストール

注記

DNS または CA 設定が適切かどうかが不明な場合は、「統合 DNS を使用するかどうかの決定」 および 「使用する CA 設定の決定」 を参照してください。

統合 DNS を使用せずにサーバーをインストールするには、DNS 関連のオプションを指定せずに **ipa-server-install** ユーティリティーを実行します。

例2.2 統合 DNS を使用しない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
- 統合 IdM CA をルート CA とするサーバー (デフォルトの CA 設定)

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

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

2. スクリプトにより、統合 DNS サービスの設定が求められます。**Enter** を押して、**no** オプションを選択します。

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

3. このスクリプトで、必要な設定を入力するように求められます。
   - 括弧内のデフォルト値をそのまま使用するには、**Enter** を押します。
   - 提案されたデフォルト値とは異なる値を指定するには、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 レルム名がプライマリー DNS ドメイン名と同じで、すべて大文字にすることを強く推奨します。たとえば、プライマリー DNS ドメインが **ipa.example.com** の場合は、Kerberos レルム名として **IPA.EXAMPLE.COM** を使用します。

異なる命名プラクティスを使用すると、Active Directory の信頼を使用できなくなるなど、悪影響をもたらします。

4. Directory Server のスーパーユーザー (**cn=Directory Manager**)、および IdM システムユーザーアカウント (**admin**) 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. インストールスクリプトにより、サーバーが設定されます。動作が完了するまで待ちます。

7. インストールスクリプトは、以下の出力例の DNS リソースレコード でファイル (**/tmp/ipa.system.records.UFRPto.db**) を生成します。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って異なります。

```
...
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 レコードを追加するまで、サーバーのインストールは完了しません。

このスクリプトでは、CA 証明書をバックアップして必要なネットワークポートが開いていることの確認が推奨されます。IdM ポートの要件と、これらのポートを開く方法は、「ポートの要件」 を参照してください。

新しいサーバーをテストするには、以下を行います。

1. admin 認証情報を使用して Kerberos レルムに対して認証します。これにより、**admin** が適切に設定され、Kerberos レルムにアクセスできることを確認します。

```
# kinit admin
```

2. **ipa user-find** などのコマンドを実行します。新しいサーバーでは、このコマンドは、設定したユーザー (**admin**) のみを出力します。

```
# 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 をルート CA として使用するサーバーのインストール

注記

DNS または CA 設定が適切かどうかが不明な場合は、「統合 DNS を使用するかどうかの決定」 および 「使用する 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、外部 CA をルート CA として連携させるには、**ipa-server-install** ユーティリティーでこのオプションを指定します。

- **--external-ca** は、外部 CA を使用するように指定します。
- **--external-ca-type** は、外部 CA のタイプを指定します。詳細は ipa-server-install(1) の man ページを参照してください。

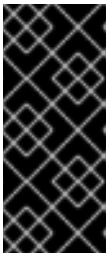
それ以外のインストール手順はほぼ、「統合 DNS を使用したサーバーのインストール」 または 「統合 DNS を使用しないサーバーのインストール」 と同じです。

Certificate System インスタンスの設定時、このユーティリティーが証明書署名要求 (CSR) の場所 (**/root/ipa.csr**) を出力します。

```
...

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
```

この場合は、以下を行います。

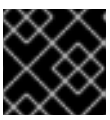
1. **/root/ipa.csr** にある CSR を外部 CA に提出します。このプロセスは、外部 CA として使用するサービスにより異なります。



重要

証明書の適切な拡張を要求する必要があります。Identity Management 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。そのためには、基本的な制約エクステンションの ***CA*** パラメーターを **true** に設定する必要があります。詳細は、『RFC 5280』 の『基本的な制約』 のセクションを参照してください。

2. 発行した証明書と、Base64 エンコードされたブロブ (PEM ファイルか Windows CA からの Base_64 証明書) で CA を発行する CA 証明書チェーンを取得します。繰り返しになりますが、プロセスは各証明書サービスによって異なります。通常は Web ページか通知メールにダウンロードリンクがあり、管理者が必要なすべての証明書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. 新たに発行された CA 証明書と CA チェーンファイルの場所と名前を指定して **ipa-server-install** を再度実行します。以下に例を示します。

```
# 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、次のエラーにより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
```

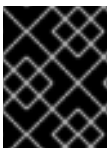
この失敗は、***_proxy** 環境変数が設定されていると発生します。この問題を解決するには、「外部 CA のインストールに失敗する」 を参照してください。

2.3.6. CA なしでのインストール

注記

DNS または CA 設定が適切かどうかが不明な場合は、「統合 DNS を使用するかどうかの決定」 および 「使用する CA 設定の決定」 を参照してください。

CA を使用せずにサーバーをインストールするには、**ipa-server-install** ユーティリティーにオプションを追加して、必要な証明書を手動で指定する必要があります。これ以外のインストール手順はほぼ、「統合 DNS を使用したサーバーのインストール」 または 「統合 DNS を使用しないサーバーのインストール」 と同じです。



重要

自己署名のサードパーティー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することはできません。

CA なしで IdM サーバーをインストールするために必要な証明書

CA なしで IdM サーバーをインストールするには、以下の証明書を指定する必要があります。

- 以下のオプションを使用して LDAP サーバー証明書および秘密鍵を指定します。
  - **--dirsrv-cert-file** - LDAP サーバー証明書の証明書ファイルおよび秘密鍵ファイル。
  - **--dirsrv-pin** - **--dirsrv-cert-file** に指定されたファイルにある秘密鍵にアクセスするパスワード。
- 以下のオプションを使用して Apache サーバー証明書および秘密鍵を指定します。
  - **--http-cert-file** - Apache サーバー証明書の証明書および秘密鍵ファイル。
  - **--http-pin** - **--http-cert-file** に指定したファイルにある秘密鍵にアクセスするパスワード。
- 以下のオプションを使用して、LDAP および Apache サーバー証明書を発行した CA の完全な CA 証明書チェーンを指定します。
  - **--dirsrv-cert-file** および **--http-cert-file** - 完全な CA 証明書チェーンまたはその一部が含まれる証明書ファイル。

以下の形式の **--dirsrv-cert-file** オプションおよび **--http-cert-file** オプションを指定して、ファイルを指定できます。

- PEM (Privacy-Enhanced Mail) がエンコードした証明書 (RFC 7468)。IdM インストーラーでは、連結した PEM エンコードオブジェクトを使用できることに注意してください。
- 識別名エンコーディングルール (DER)
- PKCS #7 証明書チェーンオブジェクト
- PKCS #8 秘密鍵オブジェクト
- PKCS #12 アーカイブ

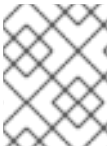
--dirsrv-cert-file オプションおよび **--http-cert-file** オプションを複数回指定して、複数のファイルを指定できます。

- 必要に応じて、このオプションを使用して完全な CA 証明書チェーンを完了するための証明書ファイルを指定します。
  - **--ca-cert-file** - このオプションは複数回追加できます。
- オプションで、このオプションを使用して外部 Kerberos 鍵配布センター (KDC) の PKINIT 証明書を提供する証明書ファイルを指定します。
  - **--pkinit-cert-file** - Kerberos KDC SSL の証明書および秘密鍵を提供します。
  - **--pkinit-pin** - Kerberos KDC の秘密鍵のロックを解除するパスワード。

  PKINIT 証明書を提供しないと、**ipa-server-install** は自己署名証明書を使用するローカル KDC で IdM サーバーを設定します。詳細は、*27章IdM の Kerberos PKINIT 認証*を参照してください。

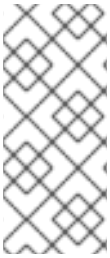
--ca-cert-file を使用して提供されるファイルと、**--dirsrv-cert-file** と **--http-cert-file** を使用して提供されるファイルには、LDAP および Apache のサーバー証明書を発行した CA の完全 CA 証明書チェーンが含まれる必要があります。

このオプションで使用できる証明書ファイル形式の詳細は、ipa-server-install(1) man ページを参照してください。



注記

表示された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ありません。



注記

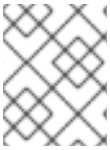
Identity Management の以前のバージョンでは、**--root-ca-file** オプションを使用してルート CA 証明書の PEM ファイルを指定していました。信頼される CA は常に DS および HTTP サーバー証明書の発行者であるため、これは不要になりました。IdM は、**--dirsrv-cert-file**、**--http-cert-file** および **--ca-cert-file** で指定された証明書をもとに、ルート CA 証明書を自動的に認識するようになりました。

例2.3 CA なしで IdM サーバーをインストールするコマンドの例

-

```
[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
```

2.3.7. 非対話形式でのサーバーのインストール



注記

DNS または CA 設定が適切かどうかが不明な場合は、「統合 DNS を使用するかどうかの決定」 および 「使用する CA 設定の決定」 を参照してください。

非対話型インストールで最低限必要なオプションは次のとおりです。

- **--ds-password** - 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** - IdM 管理者である **admin** のパスワードを指定します。
- **--realm** - Kerberos レルム名を指定します。
- **--unattended** - インストールプロセスでホスト名およびドメイン名のデフォルトオプションを選択するようにします。

必要に応じて、これらの設定にカスタム値を指定できます。

- **--hostname**: サーバーホスト名
- **--domain**: ドメイン名

--dirsrv-config-file パラメーターを使用して、カスタム値を持つ LDIF ファイルへのパスを指定すると、デフォルトの Directory Server 設定を変更することもできます。詳細は、『Release Notes for Red Hat Enterprise Linux 7.3』のIdM now supports setting individual Directory Server options during server or replica installationを参照してください。

警告

Red Hat では、Kerberos レルム名がプライマリー DNS ドメイン名と同じで、すべて大文字にすることを強く推奨します。たとえば、プライマリー DNS ドメインが **ipa.example.com** の場合は、Kerberos レルム名として **IPA.EXAMPLE.COM** を使用します。

異なる命名プラクティスを使用すると、Active Directory の信頼を使用できなくなるなど、悪影響をもたらします。

ipa-server-install で使用できるオプションの完全リストを表示するには、**ipa-server-install --help** コマンドを実行します。

例2.4 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、必要な設定を指定します。たとえば、以下は、統合 DNS あり、統合 CA なしで、サーバーをインストールします。

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

2. 設定スクリプトで、サーバーが設定されるようになりました。動作が完了するまで待ちます。

3. インストールスクリプトは、以下の出力例の DNS リソースレコード でファイル (**/tmp/ipa.system.records.UFRPto.db**) を生成します。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って異なります。

```
...
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 レコードを追加するまで、サーバーのインストールは完了しません。

このスクリプトでは、CA 証明書をバックアップして必要なネットワークポートが開いていることの確認が推奨されます。IdM ポートの要件と、これらのポートを開く方法は、「ポートの要件」 を参照してください。

新しいサーバーをテストするには、以下を行います。

1. admin 認証情報を使用して Kerberos レルムに対して認証します。これにより、**admin** が適切に設定され、Kerberos レルムにアクセスできることを確認します。

```
# kinit admin
```

2. **ipa user-find** などのコマンドを実行します。新しいサーバーでは、このコマンドは、設定したユーザー (**admin**) のみを出力します。

```
# 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。

前提条件

- 認証局 (CA)、鍵回復機関 (KRA)、または DNS Security Extensions (DNSSEC) サーバーとして機能するサーバーをアンインストールする前に、これらのサービスがドメインの別のサーバーで実行していることを確認している。

警告

CA サーバー、KRA サーバー、または DNSSEC サーバーとして機能する唯一のレプリカを削除すると、Identity Management 機能に深刻な不具合が発生する可能性があります。

手順
server.example.com をアンインストールするには、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)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 により管理される統合 DNS であるか、外部 DNS であるかに関わらず、確認を行なってください。

2.5. サーバーの名前変更

IdM サーバーのホスト名は、設定後に変更できません。ただし、サーバーを別の名前でレプリカに置き換えることができます。

1. CA および必要な新しいホスト名または IP アドレスを使用して、サーバーの新しいレプリカを作成します。この操作は、*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* に説明があります。

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. 他のすべてのレプリカおよびクライアントが以前と同じように機能していることを確認します。

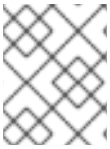
4. 「IdM サーバーのアンインストール」 の説明に従って、初期 IdM サーバーをアンインストールします。

---

[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. クライアントのインストールの前提条件

DNS 要件

適切な DNS 委譲を採用します。IdM の DNS 要件に関する詳細は、「ホスト名および DNS 設定」を参照してください。

クライアントの **resolv.conf** ファイルは変更しないでください。

ポートの要件

IdM クライアントは、IdM サーバーの複数のポートに接続し、サービスと通信します。このポートは、受信方向の **IdM サーバー** で開放する必要があります。IdM が必要とするポートの詳細は、「ポートの要件」 を参照してください。

クライアントで、これらのポートを送信方向で開きます。**firewalld** などの、送信パケットにフィルターを設定しないファイアウォールを使用している場合は、ポートを送信方向で使用できます。

Name Service Cache Daemon (NSCD) 要件

Red Hat は、Identity Management マシンで NSCD を無効にすることを推奨します。または、NSCD を無効にできない場合は、SSSD でキャッシュされないマップの NSCD のみを有効にします。

NSCD と SSSD サービスはいずれもキャッシュを実行し、システムが両方のサービスを同時に使用すると問題が発生する可能性があります。NSCD と SSSD 間の競合を回避する方法については、System-Level Authentication Guide を参照してください。

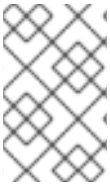
3.1.1. IdM クライアントのインストールをサポートする RHEL のバージョン

IdM サーバーが RHEL 7 の最新のマイナーバージョンで実行されている Identity Management (IdM) デプロイメントでは、以下のバージョンの最新のマイナーバージョンで実行されているクライアントがサポートされます。

- RHEL 7

  RHEL 8

  RHEL 9



注記

IdM デプロイメントを FIPS 準拠にする予定の場合、{RH} は環境を RHEL 9 に移行することを強く推奨します。RHEL 9 は、FIPS 140-3 で認定された最初の RHEL メジャーバージョンです。

3.1.2. FIPS 環境にクライアントをインストールする場合の前提条件

Red Hat Enterprise Linux 7.4 以降を使用して環境を設定する環境では、以下を行います。

- 連邦情報処理規格 (FIPS) モードが有効になっているシステムに、新しいクライアントを設定できます。インストールスクリプトは、FIPS が有効になっているシステムを自動的に検出し、管理者の介入なしに IdM を設定します。

  オペレーティングシステムで FIPS を有効にするには、『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。

Red Hat Enterprise Linux 7.3 以前を使用して設定された環境では、以下を行います。

- IdM では FIPS モードをサポートされません。IdM クライアントをインストールする前にシステムで FIPS を無効にし、インストール後に有効にしないでください。

FIPS モードの詳細は、『セキュリティーガイド』の連邦情報処理標準 (FIPS)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。

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し、設定します。インストールプロセスには、クライアントの登録に使用できる認証情報を指定する必要があります。以下の認証方法が、サポートの対象となります。

クライアントを登録する権限のあるユーザーの認証情報 (例:admin)

デフォルトでは、**ipa-client-install** にはこのオプションが必要です。例は、「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。

ユーザーの認証情報を直接 **ipa-client-install** に指定するには、**--principal** オプションおよび **--password** オプションを使用します。

サーバーで無作為に事前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** オプションに追加します。例3.1「無作為のパスワードを使用したクライアントの非対話型インストール」を参照してください。

前回登録時のプリンシパル

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。

詳細は 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シナリオを説明します。**ipa-client-install** の使用と、許可されるオプションの完全なリストの詳細は、ipa-client-install(1) 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、クライアントをインストールしますが、必要に応じてユーザー入力が求められます。ユーザーは、クライアントをドメインに登録する権限のあるユーザーの認証情報 (**admin** ユーザーなど) を提供します。

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

   以下のいずれかに該当する場合は **--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 とともにインストールされた場合。
   - ネットワーク上の DNS サーバーが、GSS-TSIG プロトコルを用いた DNS エントリー更新を受け入れる場合。

   --no-krb5-offline-passwords オプションを追加して、SSSD キャッシュに Kerberos パスワードの保存を無効にします。

2. インストールスクリプトは、すべての必要な設定を自動的に取得しようとします。

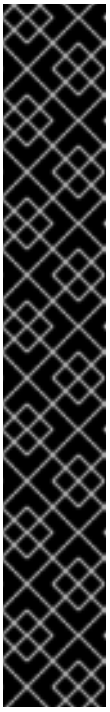
   a. お使いのシステムで DNS ゾーンと SRV レコードが正しく設定されていれば、スクリプトは必要な値をすべて自動的に検出して出力します。**yes** を入力して確定します。

   ```
   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
   ```

   別の値を使用してシステムをインストールする場合は、現在のインストールをキャンセルします。その後、**ipa-client-install** を再度実行し、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) man ページの **DNS Autodiscovery** セクションを参照してください。

   b. スクリプトが一部の設定を自動的に取得できなかった場合は、値を入力するように求められます。



重要

.company など、単一ラベルのドメイン名を使用しないでください。IdM ドメインは、トップレベルドメインと、1つ以上のサブドメイン (example.com や company.example.com など) で設定する必要があります。

完全修飾ドメイン名は、以下の条件を満たす必要があります。

- 数字、アルファベット文字、およびハイフン (-) のみが使用される有効な DNS 名である。ホスト名でアンダーライン (_) を使用すると DNS が正常に動作しません。
- すべてが小文字である。大文字は使用できません。
- 完全修飾ドメイン名は、ループバックアドレスを解決できません。**127.0.0.1** ではなく、マシンの公開 IP アドレスを解決する必要があります。

その他の推奨命名プラクティスは『Red Hat Enterprise Linux Security Guide』のRecommended Naming Practices を参照してください。

3. スクリプトにより、アイデンティティーがクライアントの登録に使用されるユーザーの入力が求められます。デフォルトでは、これは **admin** ユーザーです。

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

4. インストールスクリプトにより、クライアントが設定されます。動作が完了するまで待ちます。

```
Client configuration complete.
```

5. **ipa-client-automount** ユーティリティーを実行します。このユーティリティーは、IdM に NFS を自動的に設定します。詳細は「NFS の自動設定」を参照してください。

3.3.2. クライアントの非対話型インストール

非対話的なインストールでは、コマンドラインオプションを使用して、**ipa-client-install** ユーティリティーに必要な情報をすべて指定します。非対話型インストールで最低限必要なオプションは次のとおりです。

- クライアントの登録に使用され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** - ユーザー確認を必要とせずにインストールを実行できるようにします。

お使いのシステムで DNS ゾーンと SRV レコードが正しく設定されていれば、スクリプトは他に必要となる値をすべて自動的に検出します。スクリプトが自動的に値を検出できない場合は、コマンドラインオプションを使用して指定します。

- **--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

.company など、単一ラベルのドメイン名を使用しないでください。IdM ドメインは、トップレベルドメインと、1 つ以上のサブドメイン (example.com や company.example.com など) で設定する必要があります。

完全修飾ドメイン名は、以下の条件を満たす必要があります。

- 数字、アルファベット文字、およびハイフン (-) のみが使用される有効な DNS 名である。ホスト名でアンダーライン (_) を使用すると DNS が正常に動作しません。
- すべてが小文字である。大文字は使用できません。
- 完全修飾ドメイン名は、ループバックアドレスを解決できません。**127.0.0.1** ではなく、マシンの公開 IP アドレスを解決する必要があります。

その他の推奨命名プラクティスは『Red Hat Enterprise Linux Security Guide』のRecommended Naming Practices を参照してください。

- **--server** - クライアントが登録される IdM サーバーのホスト名を指定します。
- **--domain** - クライアントが登録される IdM サーバーの DNS ドメイン名を指定します。
- **--realm** - Kerberos レルム名を指定します。

以下のいずれかに該当する場合は **--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 とともにインストールされた場合。
- ネットワーク上の DNS サーバーが、GSS-TSIG プロトコルを用いた DNS エントリー更新を受け入れる場合。

--no-krb5-offline-passwords オプションを追加して、SSSD キャッシュに Kerberos パスワードの保存を無効にします。

ipa-client-install により許可されるオプションの完全リストは、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たクライアントの非対話型インストール

この手順では、ユーザーに入力を要求せ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、サーバー上で無作為に、登録の認証に使用するワンタイムパスワードを事前生成します。

1. 既存のサーバーの場合:
   a. 管理者としてログインします。

```
$ kinit admin
```

   b. 新しいマシンを IdM ホストとして追加します。**ipa host-add** コマンドに **--random** オプションを使用して、無作為にパスワードを生成します。

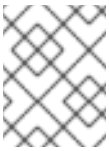
```
$ ipa host-add client.example.com --random
--------------------------------------------------
```

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

生成されたパスワードは、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。登録の完了後、このパスワードは適切なホストキータブ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、以下のオプションを使用します。

   - **--password** - **ipa host-add** の出力の無作為なパスワード

     注記

     このパスワードには通常、特殊文字が含まれています。そのため、特殊文字は一重引用符 (') で囲みます。

   - **--unattended** - ユーザー確認を必要とせずにインストールを実行できるようにします。

   お使いのシステムで DNS ゾーンと SRV レコードが正しく設定されていれば、スクリプトは他に必要となる値をすべて自動的に検出します。スクリプトが自動的に値を検出できない場合は、コマンドラインオプションを使用して指定します。

   以下に例を示します。

   ```
   # 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。このユーティリティーは、IdM に NFS を自動的に設定します。詳細は「NFS の自動設定」を参照してください。

3.4. キックスタートでの IDM クライアントの設定

キックスタートの登録により、Red Hat Enterprise Linux のインストール時に新しいシステムが自動的に IdM ドメインに追加されます。キックスタートの詳細は、『インストールガイド』のキックスタートを使用したインストールを参照してください。

クライアントのキックスタートインストールの準備には、以下の手順が含まれます。

1. 「IdM サーバーでクライアントホストエントリーの事前作成」
2. 「クライアントのキックスタートファイルの作成」

3.4.1. IdM サーバーでクライアントホストエントリーの事前作成

1. admin としてログインします。

   ```
   $ kinit admin
   ```

2. IdM サーバー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ると、キータブを使用して認証が行われます。

3.4.2. クライアント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リストに含まれる 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:
  - 登録前に SSH キーが生成されていることを確認します。
  - 以下を指定して **ipa-client-install** ユーティリティーを実行します。
    - IdM ドメインサービスのアクセスおよび設定に必要なすべての情報
    - 「IdM サーバーでクライアントホストエントリーの事前作成」に従って、IdM サーバーにクライアントホストを事前作成する際に設定するパスワード

以下に例を示します。

```
%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
```

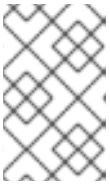
非対話的なインストールでは **--unattended** オプションも追加します。

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、**getcert** および **ipa-client-install** ユーティリティーの両方に対して **/dev/null** にシステムバスのアドレスを設定します。これには、**ipa-client-install** 手順の前に、インストール後の手順ファイル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** を起動すると、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。

キックスタートの使用方法は、『インストールガイド』のキックスタートインストールの実行を参照してください。キックスタートファイルの例は、Sample Kickstart Configurations 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考慮事項

3.5.1. Identity Management 設定の削除

ipa-client-install スクリプトは、**/etc/openldap/ldap.conf** ファイルおよび **/etc/sssd/sssd.conf** ファイルから、以前の LDAP 設定および SSSD 設定を削除します。クライアントをインストールする前に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しい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
```

Identity Management の新しい設定値を適用するには、以下を行います。

1. **/etc/openldap/ldap.conf** および **/etc/sssd/sssd.conf** を開きます。
2. 以前の設定を削除します。
3. 新しい 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の中には、再起動しないと変更が適用されない場合があります。**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

クライアントが、サーバーで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** ユーザーを確認するには、次のコマンドを実行します。

```
[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、クライアントが IdM ドメインから削除され、SSSD などのシステムサービス用の IdM 固有の設定がすべて削除されます。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。

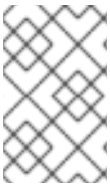
```
# ipa-client-install --uninstall
```

2. クライアントホストの DNS エントリーを、手動でサーバーから削除します。「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

クライアント仮想マシンが破棄され、そのキータブがある場合は、クライアントを再登録できます。

- 対話型 (管理者の認証情報を使用)。「管理者アカウントを使用したクライアントの対話的な再登録」を参照してください。
- 非対話型 (以前にバックアップした keytab ファイルを使用する)。「クライアントキータブを使用したクライアントの非対話的な再登録」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クライアントのみを再登録できます。クライアントをアンインストール (**ipa-client-install --uninstall** を使用) した場合や、ホストエントリーを無効 (**ipa host-disable** を使用) にした場合は再登録できません。

再登録中に IdM は以下を実行します。

- 元のホスト証明書を破棄する。
- 新しいホスト証明書を生成する。
- 新規の SSH 鍵を作成する。
- 新規のキータブを生成する。

3.8.1. 管理者アカウントを使用したクライアントの対話的な再登録

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. スクリプトにより、アイデンティティーがクライアントの登録に使用されるユーザーの入力が求められます。デフォルトでは、これは **admin** ユーザーです。

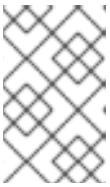
```
User authorized to enroll computers: admin
Password for admin@EXAMPLE.COM
```

3.8.2. クライアントキータブを使用したクライアントの非対話的な再登録

クライアントキータブを使用した再登録は、自動インストールや管理者パスワードを使用できない場合などの他の状況で適しています。

1. **/tmp** や **/root** などのディレクトリーに元のクライアントキータブファイルを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、**--keytab** オプションを使用してキータブ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は、**--keytab** オプションで指定するキータブのみが使用されます。再登録中、IdM はクライアントに対して新しいキータブを生成します。

3.9. クライアントマシンの名前変更

本セクションでは、IdM クライアントの名前を変更する方法を説明します。このプロセスでは、以下の操作を行います。

- 「現在のサービスとキータブ設定の特定」
- 「IdM ドメインからのクライアントマシンの削除」
- 「新規ホスト名での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、絶対に必要な場合を除き、ホスト名の変更は推奨しません。

現在のサービスとキータブ設定の特定
現在のクライアントをアンインストールする前に、クライアントの設定を書き留めます。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. マシンで実行し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 **host service** があります。ホストサービスのサービスプリンシパルは *ホストプリンシパル* とも呼ばれ、**host/client.example.com** になります。

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** で表示されるすべてのサービスプリンシパルは、**client.example.com** で対応するキータブの場所を決定します。

   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** など *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** 以外の各キータブについては、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
```

   「キータブの削除」を参照してください。

3. IdM サーバーで、ホストエントリーを削除します。これにより、すべてのサービスが削除され、そのホストに発行されたすべての証明書が無効になります。

```
[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。

新規ホスト名での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. マシンを IdM クライアントとして再登録します。「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、「現在のサービスとキータブ設定の特定」 で特定された各サービスに新しいキータブを追加します。

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

4. 「現在のサービスとキータブ設定の特定」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。これには、以下を行います。

   - IdM 管理ツールの使用24章 *ユーザー、ホスト、およびサービスの証明書の管理*を参照してください。

   - **certmonger** ユーティリティーの使用『System-Level Authentication Guide』 または certmonger(8) man ページの Working with certmonger を参照してください。

5. 「現在のサービスとキータブ設定の特定」で特定されたホストグループにクライアントを再追加します。「ユーザーまたはホストグループメンバーの追加と削除」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 サーバーの設定をクローンすることで、作成されます。そのため、サーバーとそのレプリカは同一の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では、既存のサーバー設定をコピーし、その設定に基づいてレプリカをインストールします。

ナレッジベースソリューションの Backup and Restore in IdM/IPA で説明されているように、複数のサーバーレプリカを確保することが推奨されるバックアップソリューションです。

注記

もう 1 つのバックアップソリューションは、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していますが、9章*Identity Management のバックアップおよび復元* で記載されているように **ipa-backup** ユーティリティーになります。

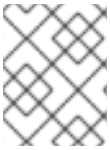
4.1. IDM レプリカの説明

多数のクライアントにサービスの可用性や冗長性を確保するために、**レプリカ** と呼ばれる複数の IdM サーバーを 1 つのドメインにデプロイできます。レプリカは、各 IdM サーバーに機能的に同じである最初の IdM サーバーのクローンで、ユーザー、マシン、証明書、および設定されたポリシーについて同じ内部情報を共有します。

ただし、一度に、環境内のサーバー 1 台のみが対応できる一意のサーバーロールは 2 つあります。

- *CA Renewalation Server*: このサーバーは、認証局 (CA) サブシステム証明書の更新を管理します。
- *CRL Generation Server*: このサーバーは、証明書失効リスト (CRL) を生成します。

デフォルトでは、最初にインストールした CA サーバーは CA Renewal Server ロールと CRL Generation Server ロールの両方に対応します。たとえば、最初にインストールしたサーバーの使用を停止する必要がある場合などこのロールをトポロジー内の他の CA サーバーに移行できます。どちらのロールも同じサーバーで対応する必要があります。

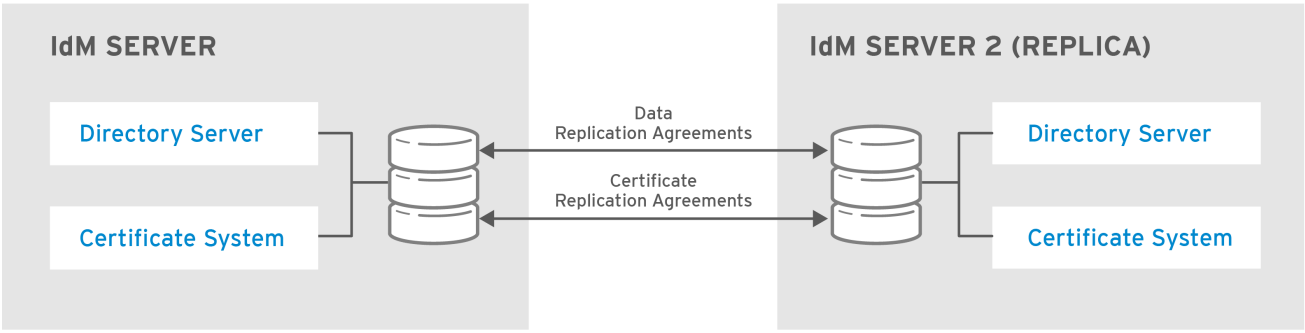


注記

IdM トポロジーのマシンの種類の詳細は、「Identity Management ドメイン」 を参照してください。

レプリケーション は、レプリカ間でデータをコピーするプロセスです。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。レプリカ合意に参加しているレプリカはすべて、更新を受信するので、データマスターとみなされます。

図4.1 サーバーとレプリカの合意



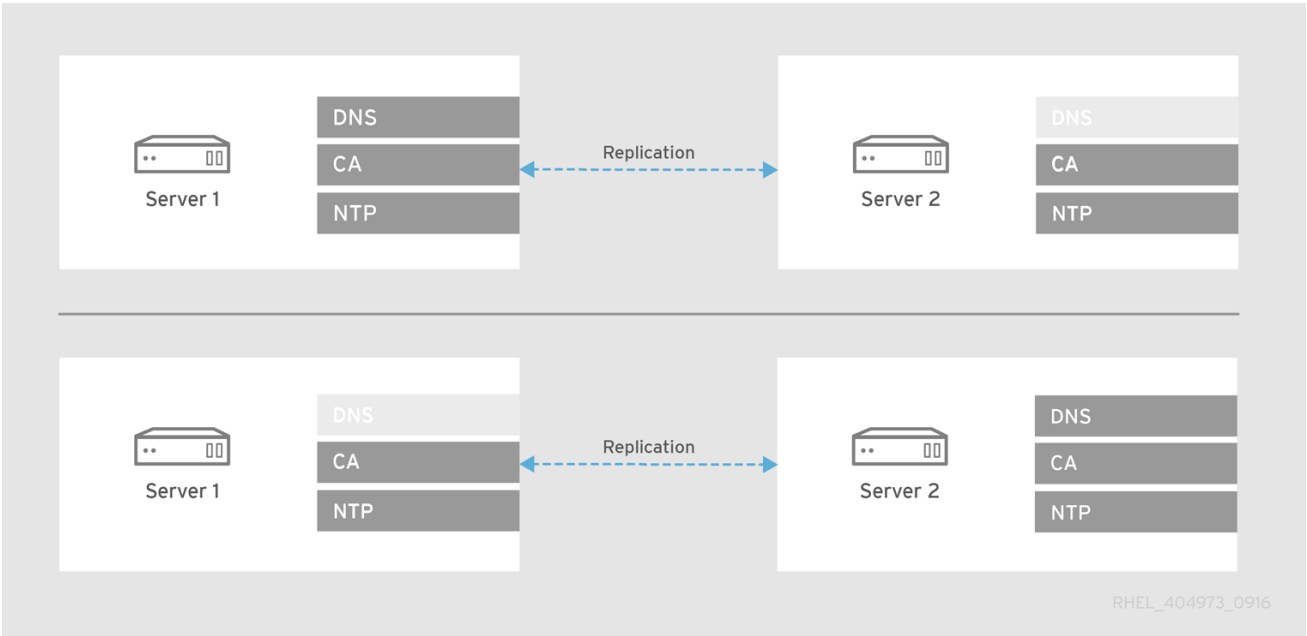
4.2. レプリカのデプロイメントに関する考慮事項

4.2.1. トポロジーでのサーバーサービスの分散

IdM サーバーは、認証局 (CA) や DNS など、多数のサービスを実行できます。レプリカは、作成したサーバーと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 を実行する場合でも、DNS サービスなしでレプリカをインストールできます。同様に、DNS を使用せずに最初のサーバーがインストールされた場合でも、レプリカを DNS サーバーとして設定できます。

図4.2 サービスが異なるレプリカ



レプリカ上の CA サービス
CA なしでレプリカを設定すると、証明書操作の全要求が、トポロジー内の CA サーバーに転送されます。

警告

Red Hat は、複数のサーバーに CA サービスをインストールすることを強く推奨します。CA サービスを含む初期サーバーのレプリカのインストールは、「CA のあるレプリカのインストール」 を参照してください。

CA を 1 台のサーバーにのみインストールすると、CA サーバーが失敗した場合に CA 設定を復元できる機会なしに CA 設定が失われるリスクがあります。詳細は「失われた CA サーバーの復旧」を参照してください。

レプリカに CA を設定する場合は、その設定が最初のサーバーの CA 設定を反映する必要があります。

- たとえば、サーバーに統合された IdM CA がルート CA として含まれている場合は、レプリカも統合 CA をルート CA としてインストールする必要があります。
- サポートされている CA 設定オプションは、「使用する 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 では、以下のガイドラインに従うことを推奨します。

1つの IdM ドメインに 60 を超えるレプリカを設定する

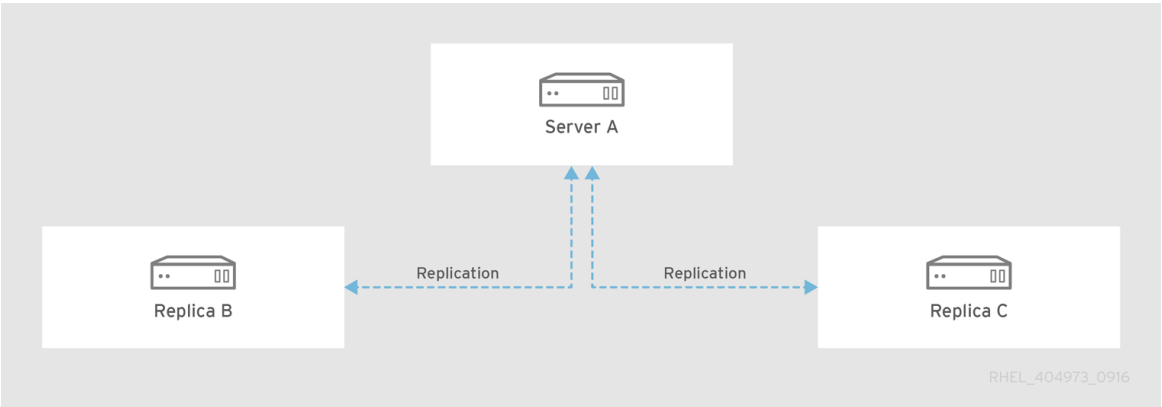
Red Hat は、レプリカが 60 台以下の環境をサポートすることを保証します。

レプリカごとに 少なくても 2 つ、多くても 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、初期レプリカとマスターサーバーとの間だけでなく、他のレプリカ間でも情報が複製されます。

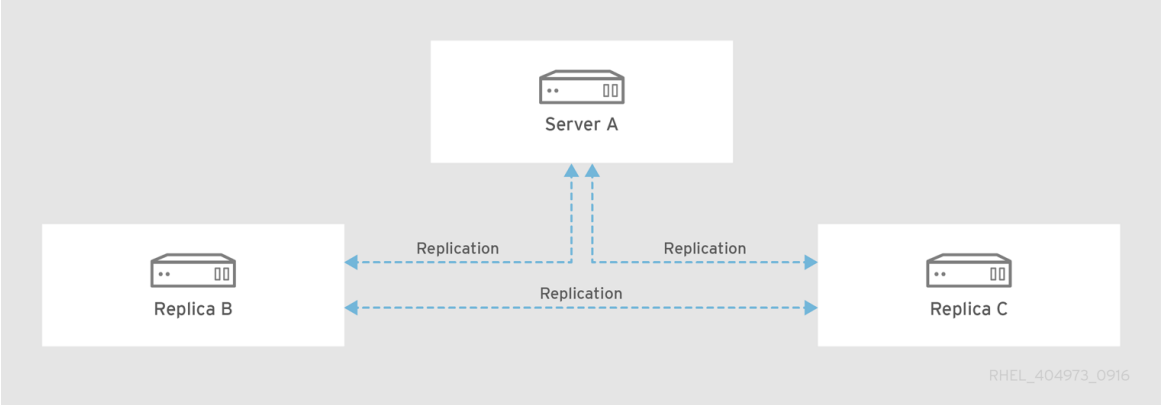
- サーバー A からレプリカ B を作成し、サーバー A からレプリカ C を作成する場合に、レプリカ B と C は直接連結されていないので、レプリカ B からのデータが先にサーバー A に複製されてから、レプリカ C に伝播する必要があります。

図4.3 レプリカ B および C はレプリカ合意には参加しない



レプリカ B とレプリカ C の間に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、フェイルオーバーの耐性、およびパフォーマンスが改善されます。

図4.4 レプリカ B および C はレプリカ合意に参加している



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。

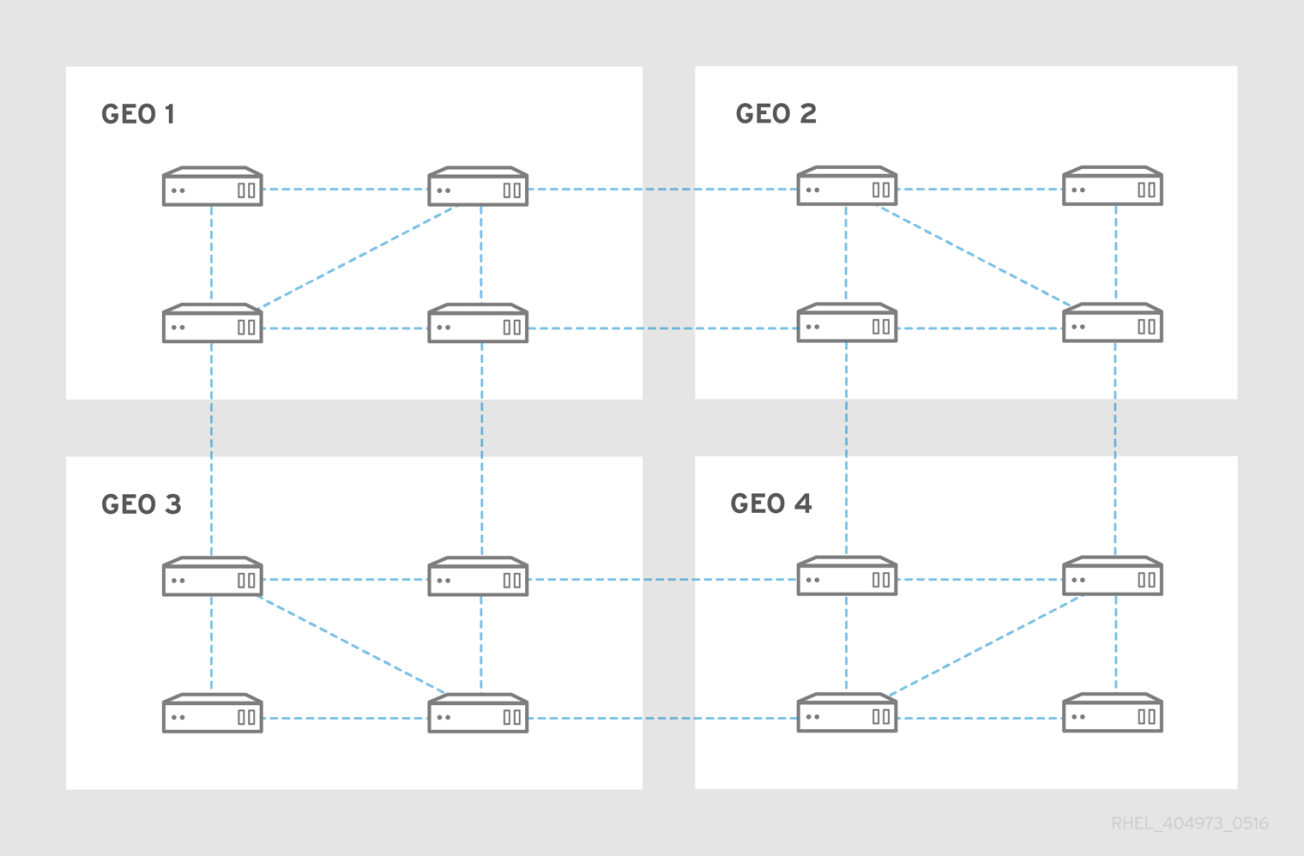
レプリカごとに 4 つ以上のレプリカ合意を設定する必要はありません。サーバーごとに多数のレプリカ合意を行っても、1 つのマスターでは、一度に 1 つのコンシューマーサーバーしか更新できないので、その他の合意はアイドル状態で、待機していることになります。また、レプリカ合意を設定しすぎると、全体的なパフォーマンスに影響を及ぼす可能性があります。

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を満たしているかどうかを確認します。詳細は、**ipa topologysuffix-verify --help** を実行します。

このコマンドには、トポロジーの接尾辞を指定する必要があります。詳細は「レプリカ合意、トポロジー接尾辞、およびトポロジーセグメントの説明」を参照してください。

図4.5 トポロジーの例



4.2.2.1. 密接なセルトポロジー

最も回復性の高いトポロジーの 1 つとして、セル内にサーバーが少数あるサーバーとレプリカのセル設定を作成します。

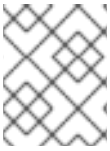
- 各セルは *密接なセル* です。密接なセルでは、すべてのサーバーにはレプリカ合意があります。
- 各サーバーには、セル **外の** 別のサーバーとレプリカ合意があります。これにより、すべてのセルがドメイン内の他のすべてのセルに疎結合されるようになります。

密接なセルトポロジーを実現するには、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。可能であれは、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代わりに、SSSD を使用して認証情報をキャッシュし、オフサイトの IdM サーバーをデータバックエンドとしてキャッシュします。

4.2.3. 非表示のレプリカモード

デフォルトでは、新しいレプリカを設定すると、インストーラーは DNS にサービス (**SRV**) リソースレコードを自動的に作成します。このレコードにより、クライアントはレプリカとそのサービスを自動検出できます。非表示のレプリカは、稼働中および利用できるすべてのサービスを持つ IdM サーバーです。ただし、DNS に **SRV** レコードがなく、LDAP サーバーロールが有効になっていません。そのため、クライアントはサービス検出を使用して非表示のレプリカを検出することができません。



注記

非表示のレプリカ機能は、テクノロジープレビューとして Red Hat Enterprise Linux 7.7 以降で利用でき、サポート対象外となります。

非表示のレプリカは、主にクライアントを中断できる専用のサービス用に設計されています。たとえば、IdM の完全バックアップは、マスターまたはレプリカ上のすべての IdM サービスをシャットダウンする必要があります。非表示のレプリカを使用するクライアントはないため、管理者はクライアントに影響を与えることなく、このホスト上のサービスを一時的にシャットダウンできます。その他のユースケースには、大量インポートや詳細なクエリーなど、IdM API または LDAP サーバーの高負荷操作が含まれます。

レプリカを非表示としてインストールするには、***--hidden-replica*** パラメーターを **ipa-replica-install** コマンドに渡します。レプリカのインストールに関する詳細は、「レプリカの作成: 概要」 を参照してください。

または、既存のレプリカの状態を変更することもできます。詳細は、「非表示のレプリカのデモートおよびプロモート」を参照してください。

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 サーバーの場合と同じです。レプリカマシンが 「サーバーのインストールの前提条件」 に記載の前提条件をすべて満たしていることを確認してください。

一般的なサーバー要件に加えて、以下の条件を満たしている必要があります。

レプリカは、同じまたはそれ以降のバージョンの IdM を実行している必要があります。

たとえば、マスターサーバーが Red Hat Enterprise Linux 7 で実行され、IdM 4.4 パッケージを使用する場合は、レプリカが Red Hat Enterprise Linux 7 以降で実行され、IdM 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、設定が適切にサーバーからレプリカにコピーされます。

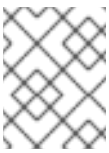
重要

IdM は、以前のバージョンのマスター以外のレプリカ作成に対応していません。以前のバージョンを使用してレプリカを作成しようとすると、インストールに失敗します。

レプリカには、追加でポートを開放する必要があります。

「ポートの要件」 に記載されている標準の IdM サーバーポート要件に加えて、以下の条件にも対応してください。

- ドメインレベル 0 で、レプリカのセットアッププロセスで **TCP ポート 22** をマスターサーバーで開いたままにします。このポートは、マスターサーバーへの接続に SSH を使用するのに必要です。

  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 1 つが Red Hat Enterprise Linux 6 で実行され、CA がインストールされている場合は、レプリカの設定中および後に **TCP ポート 7389** を開放したままにします。Red Hat Enterprise Linux 7 だけの環境では、ポート 7389 は必要ありません。

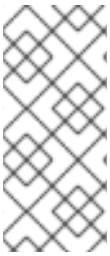
firewall-cmd ユーティリティーを使用してポートを開く方法は、「ポートの要件」 を参照してください。

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 ユーティリティーを使用して、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。Identity Management レプリカは一度に 1 つずつインストールしてください。同時に複数のレプリカをインストールすることはサポートされません。



注記

本章では、Red Hat Enterprise Linux 7.3 で導入された、レプリカの簡素化インストールについて説明します。この手順で必要なドメインレベルは 1 です (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 でレプリカをインストールする方法は、??? を参照してください。

新しいレプリカをインストールできます。

- クライアントをレプリカに *プロモート*します。「既存のクライアントのレプリカへのプロモート」 を参照してください。
- IdM ドメインに登録されていないマシンで、「クライアントではないマシンへのレプリカのインストール」を参照してください。

このいずれの状況でも、**ipa-replica-install** にオプションを追加すると、レプリカをカスタマイズできます。「ipa-replica-install を使用したユースケースに対するレプリカの設定」 を参照してください。

レプリカを非表示としてインストールするには、***--hidden-replica*** パラメーターを **ipa-replica-install** に渡します。非表示のレプリカの詳細は、「非表示のレプリカモード」 を参照してください。

重要

複製中の IdM サーバーが Active Directory と信頼関係がある場合は、**ipa-replica-install** を実行した後にレプリカを信頼エージェントとして設定します。『Windows Integration Guide』のTrust Controllers and Trust Agents を参照してください。

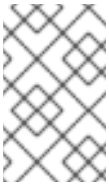
既存のクライアントのレプリカへのプロモート

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許可されていることを確認する必要があります。これを行うには、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する

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指定する方法は複数あります。これにより、以下が可能になります。

- IdM により、対話的に認証情報を取得するようにプロンプトが表示されます。



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利用できない場合には、インストールで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** に直接追加します。

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

クライアントを ipaservers ホストグループに追加する

ipaservers のメンバーシップは、マシンの権限を特権ユーザーの認証情報と同様の権限に昇格します。ユーザーの認証情報を指定する必要はありません。

例: 「ホストキータブを使用したクライアントのレプリカへのプロモート」

クライアントでは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で実行すると、**ipa-replica-install** は最初にマシンをクライアントとして登録してから、レプリカ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、以下のいずれかを選択します。

特権ユーザーの認証情報を指定する

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
```

クライアントのパスワードを無作為に指定する

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、レプリカはクライアントインストーラーで検出された最初の IdM サーバーに対してインストールされます。特定のサーバーに対して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

ipa-replica-install を使用したユースケースに対するレプリカの設定

オプションなしで実行すると、**ipa-replica-install** は基本的なサーバーサービスのみを設定します。DNS や認証局 (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 は、複数のサーバーに CA サービスをインストールすることを強く推奨します。CA サービスを含む初期サーバーのレプリカのインストールは、「CA のあるレプリカのインストール」 を参照してください。

CA を 1 台のサーバーにのみインストールすると、CA サーバーが失敗した場合に CA 設定を復元できる機会なしに CA 設定が失われるリスクがあります。詳細は「失われた CA サーバーの復旧」を参照してください。

最も主要なオプションを使用したレプリカのインストールに関するシナリオ例については以下を参照してください。

- 「DNS のあるレプリカのインストール」、**--setup-dns** および **--forwarder**の使用
- 「CA のあるレプリカのインストール」、**--setup-ca**の使用
- 「CA の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**

--dirsrv-config-file パラメーターを使用して、カスタム値を持つ LDIF ファイルへのパスを指定すると、デフォルトの Directory Server 設定を変更することもできます。詳細は、『Release Notes for Red Hat Enterprise Linux 7.3』のIdM now supports setting individual Directory Server options during server or replica installationを参照してください。

レプリカの設定に使用するオプションの完全リストは、ipa-replica-install(1) の man ページを参照してください。

4.5.1. ホストキータブを使用したクライアントのレプリカへのプロモート

この手順では、独自のホストキータブを使用して既存の IdM クライアントがレプリカにプロモートされ、プロモーションが認可されます。

この手順では、管理者または Directory Manager (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で公開されないため、安全性が向上します。

1. 既存のサーバーの場合:

   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** 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 のメンバーシップは、マシンの権限を管理者の認証情報と同様の権限に昇格します。

2. クライアントで **ipa-replica-install** ユーティリティーを実行します。

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

3. オプションで、複製する IdM サーバーに Active Directory と信頼関係がある場合は、信頼エージェントまたは信頼コントローラーとしてレプリカを設定します。詳細は、『Windows Integration Guide』のTrust Controllers and Trust Agents を参照してください。

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、IdM クライアントではないマシンにレプリカをゼロからインストールします。登録を承認するには、クライアント登録 1 回だけに有効なクライアント固有に作成せれた無作為のパスワードを使用します。

この手順では、管理者または Directory Manager (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で公開されないため、安全性が向上します。

1. 既存のサーバーの場合:

   a. 管理者としてログインします。

```
$ kinit admin
```

   b. 新しいマシンを IdM ホストとして追加します。**ipa host-add** コマンドに **--random** オプションを使用して、レプリカのインストールに使用される無作為な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
```

   生成されたパスワードは、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。登録の完了後、このパスワードは適切なホストキータブ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. レプリカをインストールするマシンで、**ipa-replica-install** を実行し、**--password** オプションを使用して無作為のパスワードを指定します。特殊文字が含まれるため、パスワードを一重引用符 (') で囲みます。

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

3. オプションで、複製する IdM サーバーに Active Directory と信頼関係がある場合は、信頼エージェントまたは信頼コントローラーとしてレプリカを設定します。詳細は、『Windows Integration Guide』のTrust Controllers and Trust Agents を参照してください。

4.5.3. DNS のあるレプリカのインストール

この手順は、IdM ドメインに所属していないマシン、およびクライアントでレプリカをインストールする場合に使用します。詳細は「レプリカの作成: 概要」を参照してください。

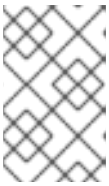
1. 以下のオプションを使用して、**ipa-replica-install** を実行します。

- **--setup-dns** - DNS ゾーンが存在しない場合は作成し、レプリカを DNS サーバーとして設定します。
- **--forwarder** - 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は **--no-forwarder** を指定します。

  フェイルオーバーの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。

以下に例を示します。

```
# 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** や **--no-host-dns** などの DNS 設定に関する複数のオプションを受け入れます。詳細は 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が作成した場合には、適切な DNS エントリーでレプリカが自動的に作成されます。これらのエントリーにより、IdM クライアントが新しいサーバーを検出できるようになります。

   初期サーバーで DNS が有効になっていない場合は、DNS レコードを手動で追加します。ドメインサービスには、以下の DNS レコードが必要です。

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

- **_ntp._udp**
- **_kpasswd._tcp**
- **_kpasswd._udp**

この例では、エントリーが存在することを確認する方法を説明します。

a. DOMAIN 変数および 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. 親ドメインから IdM DNS ドメインに DNS 委譲を追加します。たとえば、IdM DNS ドメインが **ipa.example.com** の場合は、ネームサーバー (NS) レコードを親ドメイン **example.com** に追加します。

重要

この手順は、IdM DNS サーバーをインストールするたびに繰り返す必要があります。

4. **任意ですが、推奨されます。**レプリカが利用できなくなった場合に、他の DNS サーバーをバックアップサーバーとして手動で追加します。「ネームサーバーの追加設定」を参照してください。これは、新しいレプリカが IdM ドメインの最初の DNS サーバーになる場合に特に推奨されます。

5. オプションで、複製する IdM サーバーに Active Directory と信頼関係がある場合は、信頼エージェントまたは信頼コントローラーとしてレプリカを設定します。詳細は、『Windows Integration Guide』のTrust Controllers and Trust Agents を参照してください。

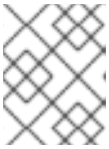
4.5.4. CA のあるレプリカのインストール

この手順は、IdM ドメインに所属していないマシン、およびクライアントでレプリカをインストールする場合に使用します。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. **--setup-ca** オプションは、サーバーの IdM CA がルート CA であるか、外部 CA に従属しているかに関係なく、最初のサーバーの設定から CA 設定をコピーします。



注記

サポート対象の CA 設定の詳細は、「使用する CA 設定の決定」 を参照してください。

3. オプションで、複製する IdM サーバーに Active Directory と信頼関係がある場合は、信頼エージェントまたは信頼コントローラーとしてレプリカを設定します。詳細は、『Windows Integration Guide』のTrust Controllers and Trust Agents を参照してください。

4.5.5. CA のないサーバーからのレプリカのインストール

この手順は、IdM ドメインに所属していないマシン、およびクライアントでレプリカをインストールする場合に使用します。詳細は「レプリカの作成: 概要」を参照してください。

重要

自己署名のサードパーティー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することはできません。

1. **ipa-replica-install** を実行して、次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**

このようなオプションを使用して提供され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。

以下に例を示します。

```
[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。

2. オプションで、複製する IdM サーバーに Active Directory と信頼関係がある場合は、信頼エージェントまたは信頼コントローラーとしてレプリカを設定します。詳細は、『Windows Integration Guide』のTrust Controllers and Trust Agents を参照してください。

4.6. 新規レプリカのテスト

レプリカの作成後にレプリケーションが想定どおりに機能するかどうかを確認するには、以下を実行します。

1. サーバーのいずれか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で表示されることを確認します。

```
[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. 管理: サーバーの管理

本パートでは、**アイデンティティー管理** サーバーとサービスの管理、**アイデンティティー管理** ドメインのサーバー間のレプリケーションなどの管理関連のトピックと、**アイデンティティー管理** トポロジーの詳細を説明し、システムの **アイデンティティー管理** パッケージを更新する方法を説明します。さらに、このパートでは、**アイデンティティー管理**のデプロイメントに影響を与える障害が発生した場合に、**アイデンティティー管理** システムのバックアップを手動で作成して復元する方法を説明します。最後の章では、さまざまな内部アクセス制御メカニズムの詳細を説明します。

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 に対して認証する方法など、IdM サーバーおよびサービスの管理に使用できる Identity Management のコマンドラインおよび UI ツールを説明します。

5.1. IDM サーバーの起動と停止

Directory Server、認証局 (CA)、DNS、Kerberos など、さまざまなサービスが IdM サーバーとともにインストールされます。**ipactl** ユーティリティーを使用して、IdM サーバー全体と、インストールしたサービスをすべて停止、起動 (開始)、または再起動 (再開) します。

IdM サーバー全体を起動するには、次のコマンドを実行します。

```
# ipactl start
```

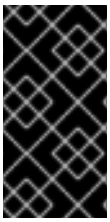
IdM サーバー全体を停止するには、次のコマンドを実行します。

```
# ipactl stop
```

IdM サーバー全体を再起動するには、次のコマンドを実行します。

```
# ipactl restart
```

個々のサービスを停止、起動、または再起動するだけの場合は、システム管理者のガイドで説明されている **systemctl** ユーティリティーを使用します。たとえば、**systemctl** を使用した個別のサービスの管理は、Directory Server の動作をカスタマイズする場合に便利です。設定の変更には Directory Server インスタンスを再起動する必要がありますが、すべての 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には、常に **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、サービスを開始および停止する順番は極めて重要です。**ipactl** ユーティリティーは、サービスが適切な順番で開始および停止するようにします。

5.2. KERBEROS を使用した IDM へのログイン

IdM は、シングルサインオンに対応する Kerberos プロトコルを使用します。Kerberos では、正しいユーザー名とパスワードを一度提示するだけで済み、システムから認証情報を再度求められることなく IdM サービスにアクセスできます。

デフォルトでは、IdM ドメインのメンバーであるマシンのみが、Kerberos を使用して IdM に対して認証できます。ただし、Kerberos 認証用に外部システムを設定することもできます。詳細は、「Web UI への Kerberos 認証のための外部システムの設定」 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** ユーティリティーを使用します。

注記

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

ユーザー名を指定せずに実行する場合は、ローカルシステムに現在ログインしているユーザーのユーザー名で **kinit** を使用して IdM にログインします。たとえば、ローカルシステムで **local_user** としてログインしている場合は、**kinit** を実行すると、**local_user** IdM ユーザーとして認証を試みます。

```
[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ーが一致しないと、認証に失敗します。

別の IdM ユーザーとしてログインするには、必要なユーザー名をパラメーターとして **kinit** ユーティリティーに渡します。たとえば、**admin** ユーザーとしてログインするには、次のコマンドを実行します。

```
[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** の実行は必要ありません。

ID および認証プロバイダーとして SSSD で IdM を設定した IdM システムでは、ユーザーが対応する Kerberos プリンシパル名でログインした後に TGT を自動的に取得します。

pam_krb5 の設定に関する詳細は、pam_krb5(8) man ページを参照してください。PAM に関する一般的な情報は、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 チケットの保存

デフォルトでは、Kerberos は、ログインしたユーザーごとに認証情報キャッシュにチケットを 1 つだけ保存します。ユーザーが **kinit** を実行すると、Kerberos は、現在保存されているチケットを新しいチケットで上書きします。たとえば、**kinit** を使用して **user_A** として認証すると、**user_B** として再度認証した後に **user_A** のチケットが失われます。

ユーザーの別の TGT を取得して保存するには、異なる認証情報キャッシュを設定します。これにより、以前のキャッシュの内容が上書きされないようにします。これは、以下のいずれかの方法で実行できます。

- **export KRB5CCNAME=*path_to_different_cache*** コマンドを実行してから、**kinit** を使用して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** 変数をリセットします。

デフォルトの認証情報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

現在保存されている TGT が、認証に使用されることを確認するには、**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リスト表示します。以下の例では、キャッシュに **user_A** のチケットが含まれています。これは、現在 IdM サービスにアクセスすることができる **user_A** のみになります。

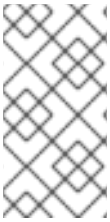
```
$ klist
Ticket cache: KEYRING:persistent:0:0
Default principal: user_A@EXAMPLE.COM

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スクリプトの名前は **ipa** です。**ipa** スクリプトは、多数のサブコマンドの親スクリプトです。これらのサブコマンドは、IdM の管理に使用されます。たとえば、**ipa user-add** コマンドは、新しいユーザーを追加します。

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

コマンドライン管理には、UI での管理に比べていくつかの利点があります。たとえば、コマンドラインユーティリティーを使用すると、手動で介入することなく、管理タスクを自動化して、一貫した方法で繰り返し実行することができます。さらに、ほとんどの管理操作はコマンドラインと Web UI の両方で利用できますが、一部のタスクはコマンドラインからのみ実行できます。



注記

本セクションでは、**ipa** サブコマンドの概要のみを説明します。詳細は、IdM 管理の特定の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てユーザーエントリーを管理する方法は、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 コマンドのヘルプの取得

ipa スクリプトは、特定のサブコマンドのヘルプ (*トピック*) を表示できます。利用可能なトピックのリストを表示するには、**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** コマンドのリストを表示することもできます。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.
...
```

各 **ipa** コマンドの詳細は、コマンドに **--help** オプションを追加します。以下に例を示します。

```
$ 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 ユーティリティーの詳細は、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、エントリー属性をリストに保存します。以下に例を示します。

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

属性のリストへの更新は、前のリストを上書きします。たとえば、1 つの属性だけを指定してその属性を追加しようとすると、以前に定義したリストがすべて新しい 1 つの属性に置き換えられます。したがって、属性のリストを変更する場合は、更新されたリスト全体を指定する必要があります。

IdM は、属性のリストを指定する以下の方法に対応します。

- 同じコマンド呼び出しで、同じコマンドライン引数を複数回指定します。以下に例を示します。

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

- リストを中括弧で囲むことで、シェルが拡張を行うことができます。以下に例を示します。

  ```
  $ 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 (&)、アステリスク (*)、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。たとえば、アスタリスク (*) をエスケープするには、次のコマンドを実行します。

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

シェルが特殊文字を正しく解析できないため、エスケープしていない特殊文字をコマンドに含めると、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 エントリーのリスト表示

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。

- 全ユーザーをリスト表示するには、以下を実行します。

```
$ 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のリストを表示するには、次のコマンドを実行します。

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

  IdM がユーザーおよびグループを検索する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
```

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の表示

ipa *-show コマンドを使用して、特定の IdM エントリーの詳細を表示します。以下に例を示します。

```
$ 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

ユーザーリストの表示など、検索結果によっては、非常に多くのエントリーを返す場合があります。この検索操作を調整して、**ipa user-find** などの **ipa *-find** コマンドの実行時や、Web UI で対応するリストを表示する際に、全体的なサーバーのパフォーマンスを向上できます。

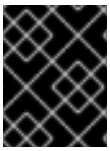
検索サイズの制限

- クライアント、Idm コマンドラインツール、または IdM Web UI からサーバーに送信されるリクエストで返される最大エントリー数を定義します。
- デフォルト値:100 エントリー

検索時間の制限

- サーバーが検索の実行を待つ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

この値が **-1** に設定されていると、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影響を及ぼすことがあります。

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、以下を行います。

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整するには、**ipa config-mod** コマンドを使用して、**--searchrecordslimit** オプションおよび **--searchtimelimit** オプションを指定します。以下に例を示します。

```
$ 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してのみ制限を調整することもできます。これを行うには、**--sizelimit** オプションまたは **--timelimit** オプションをコマンドに追加します。以下に例を示します。

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

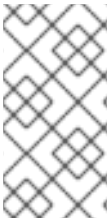
重要

ipa config-mod コマンドを使用して、**--searchrecordslimit** オプションまたは **--searchtimelimit** オプションを指定してサイズまたは時間制限を調整すると、**ipa user-find** などの **ipa** コマンドによって返されたエントリーの数に影響することに注意してください。

これらの制限に加えて、Directory Server レベルで設定される設定も考慮され、より厳しい制限が適用される場合があります。Directory Server の制限に関する詳細は、『Red Hat Directory Server 管理ガイド』 を参照してください。

5.4. IDM WEB UI

Identity Management の Web UI は、IdM 管理用の Web アプリケーションです。これには、**ipa** コマンドラインユーティリティーの機能の大部分が含まれます。そのため、UI またはコマンドラインのどちらから IdM を管理するかを選択できます。



注記

ログインしているユーザーが利用できる管理操作は、ユーザーのアクセス権限によって異なります。管理者権限を持つ **admin** ユーザーおよびその他のユーザーの場合は、すべての管理タスクが利用可能になります。通常ユーザーは、自身のユーザーアカウントに関連する限定された一連の操作のみを利用できます。

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、および IdM ドメイン外のマシンからもアクセスできます。ただし、ドメイン以外のマシンから UI にアクセスするには、IdM 以外のシステムを IdM Kerberos ドメインに接続できるように設定する必要があります。詳細は、「Web UI への Kerberos 認証のための外部システムの設定」 を参照してください。

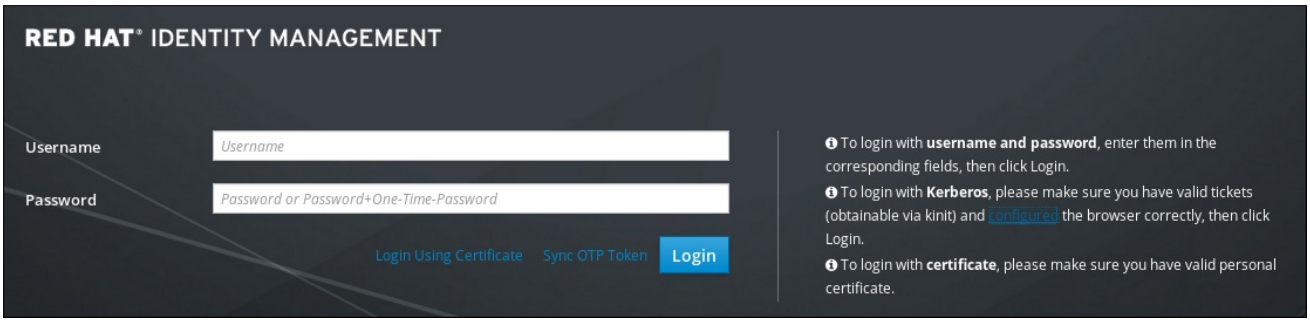
5.4.2.1. Web UI へのアクセス

Web UI にアクセスするには、ブラウザーのアドレスバーに 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 IdM Web UI ログイン画面が開きます。

図5.1 Web UI のログイン画面



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 に対して認証できます。

アクティブな Kerberos チケットの使用

kinit ユーティリティーで有効な TGT を取得した場合は、**ログイン** をクリックすると自動的にユーザーを認証します。Kerberos 認証に対応するようにブラウザーを適切に設定する必要があります。

Kerberos TGT を取得する方法は、「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詳細は、「Kerberos 認証用のブラウザーの設定」 を参照してください。

ユーザー名とパスワードの指定

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

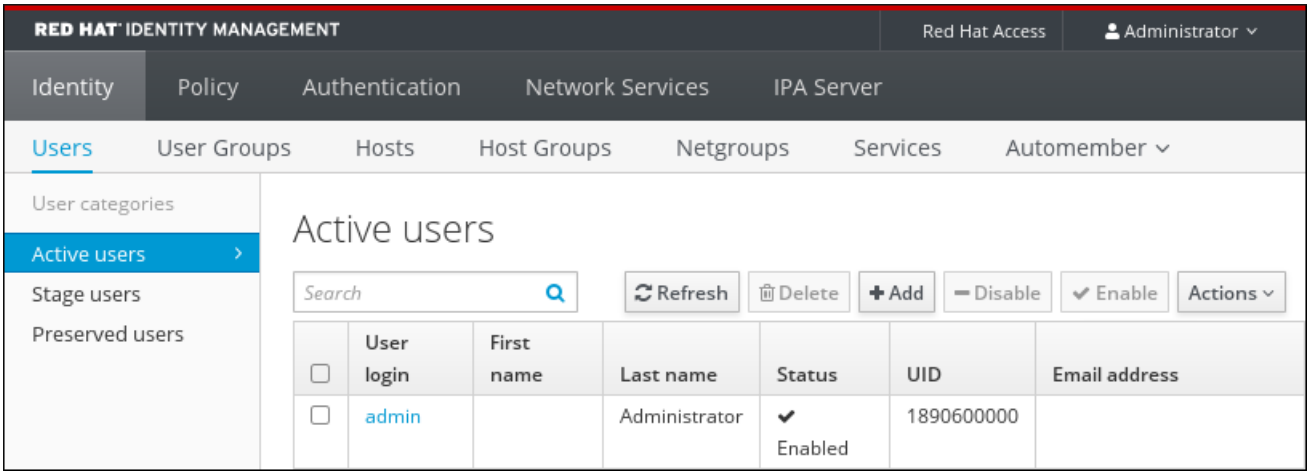
IdM は、ワンタイムパスワード (OTP) 認証にも対応しています。詳細な情報は、「ワンタイムパスワード」 を参照してください。

スマートカードの使用

詳細な情報は、「スマートカードを使用した Identity Management Web UI への認証」 を参照してください。

ユーザーが正常に認証されると、IdM 管理ウィンドウが開きます。

図5.2 IdM Web UI レイアウト



5.4.2.3. Web UI セッションの長さ

ユーザー名とパスワードを使用して IdM Web UI にログインすると、セッションの長さはログイン操作時に取得した Kerberos チケットの有効期限と同じになります。

5.4.2.4. 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、管理者権限に関連する管理操作を実行できる IdM ユーザーとは異なり、自分のユーザーアカウントに関連する限定された操作のみを実行できます。

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

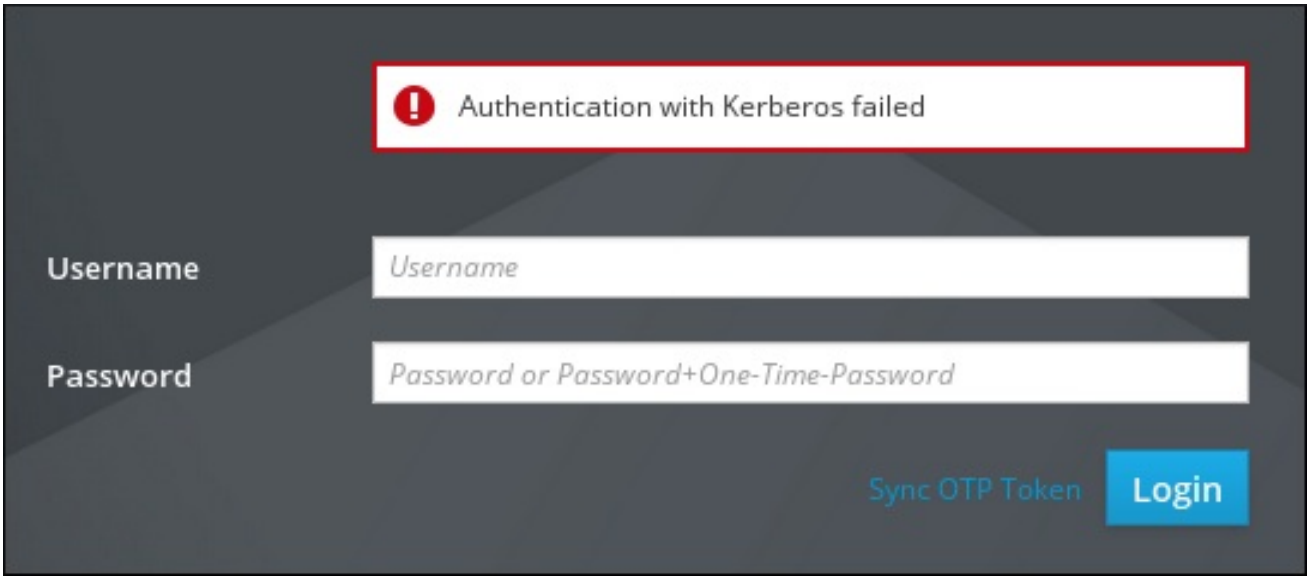
```
[admin@server ~]$ ipa idoverrideuser-add 'Default Trust View' ad_user@ad.example.com
```

AD の ID ビューの詳細は、『Windows Integration Guide』のUsing ID Views in Active Directory Environments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

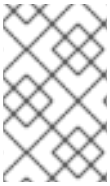
Kerberos 認証情報での認証を有効にするには、IdM ドメインにアクセスするための Kerberos ネゴシエーションに対応するようにブラウザーを設定する必要があります。ブラウザーが Kerberos 認証に対して適切に設定されていない場合は、IdM Web UI のログイン画面で **Login** をクリックするとエラーメッセージが表示されます。

図5.3 Kerberos 認証エラーメッセージ



Kerberos 認証用に、ブラウザーを以下の 3 つの方法で設定できます。

- IdM Web UI から自動的に設定する。このオプションは Firefox でのみ利用できます。詳細は「Web UI での Firefox 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時にコマンドラインから自動的に設定する。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 の自動設定」 を参照してください。
- Firefox 設定で手動で設定する。このオプションは、サポートされるすべてのブラウザーで利用できます。詳細は「手動のブラウザー設定」を参照してください。



注記

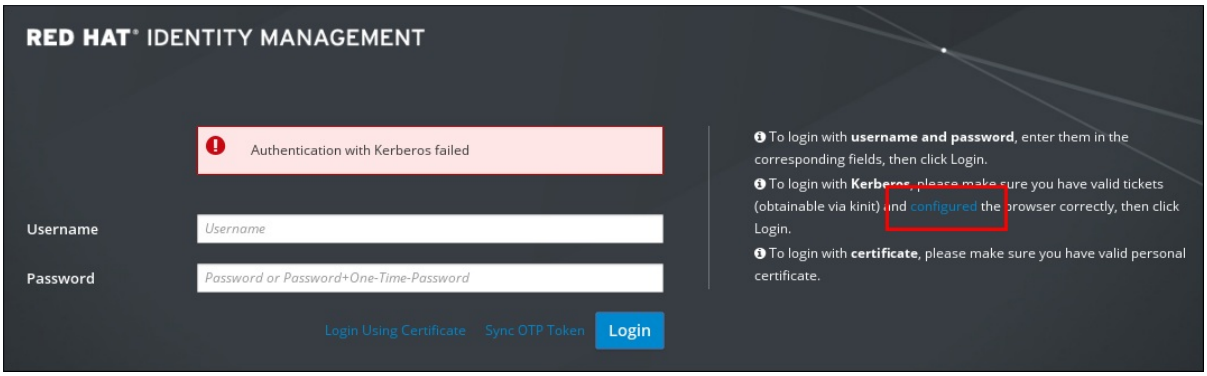
『System-Level Authentication Guide』には、Troubleshooting Firefox Kerberos Configurationが含まれます。Kerberos 認証が想定どおりに機能していない場合は、トラブルシューティングガイドで、他のアドバイスを参照してください。

Web UI での Firefox の自動設定

IdM Web UI から Firefox を自動的に設定するには、以下を実施します。

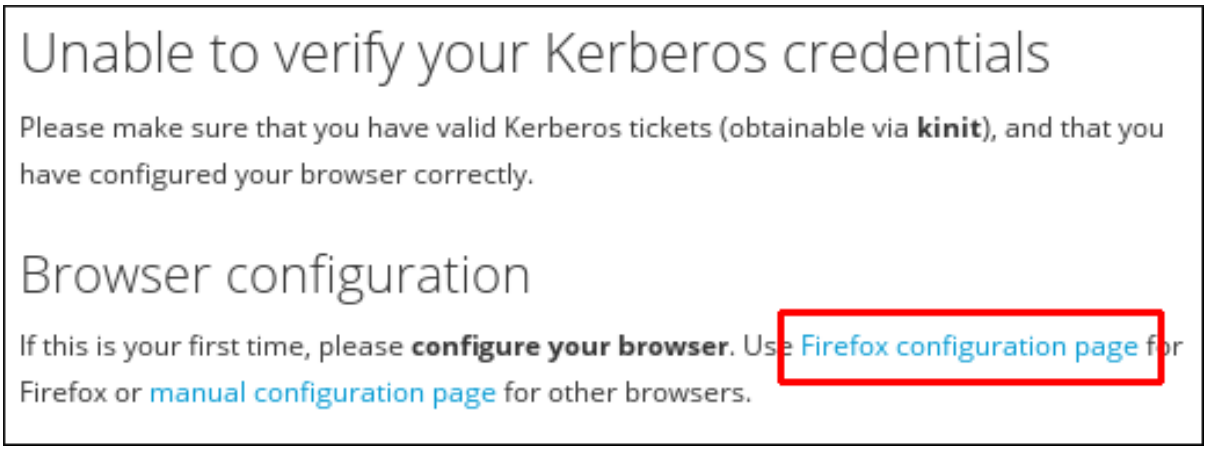
1. Web UI のログイン画面で、ブラウザー設定のリンクをクリックします。

図5.4 Web UI でのブラウザー設定へのリンク



2. Firefox 設定のリンクを選択して、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 の自動設定

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** ユーティリティーを使用して IdM クライアントをインストールする場合は **--configure-firefox** オプションを使用します。

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

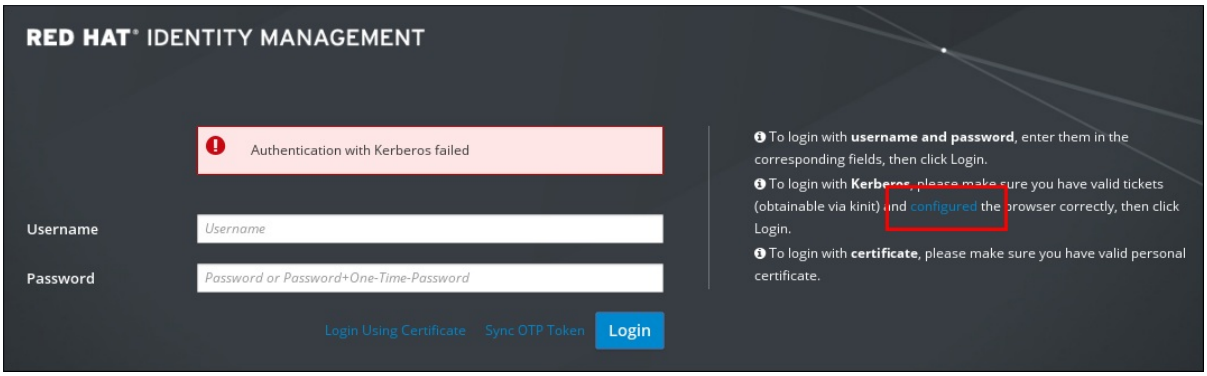
--configure-firefox オプションは、シングルサインオン (SSO) で Kerberos を有効にするデフォルトの Firefox 設定でグローバル設定ファイルを作成します。

手動のブラウザー設定

ブラウザーを手動で設定するには、以下を実行します。

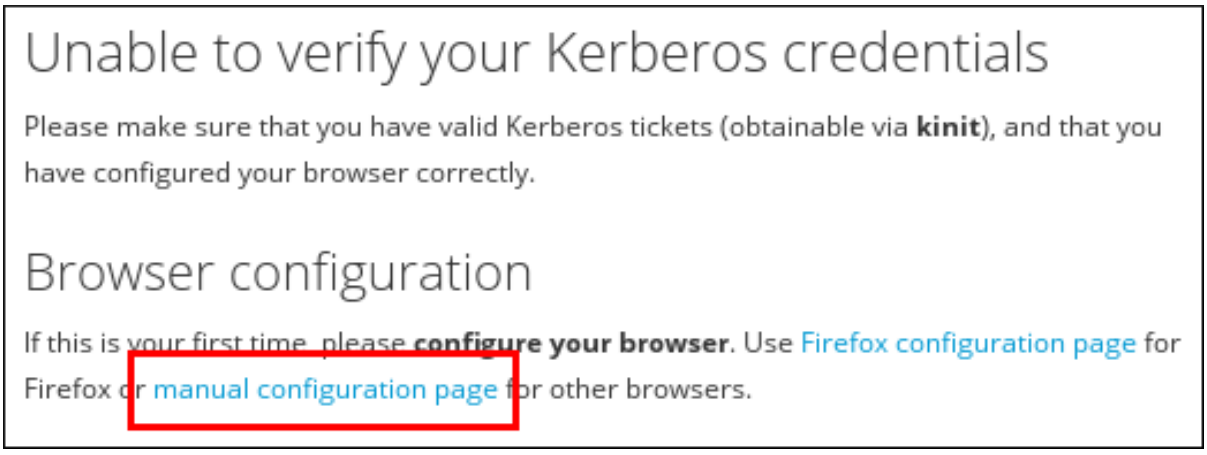
1. Web UI のログイン画面で、ブラウザー設定のリンクをクリックします。

図5.6 Web UI でのブラウザー設定へのリンク



2. 手動のブラウザー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. ブラウザーを設定する手順を探し、手順に従ってください。

5.4.4. Web UI への Kerberos 認証のための外部システムの設定

IdM ドメインのメンバーではないシステムから Web UI への Kerberos 認証を有効にするには、IdM 固有の Kerberos 設定ファイルを外部マシンに定義する必要があります。外部システムの Kerberos 認証を有効にすることは、インフラストラクチャーに、複数のレルムまたは重複ドメインが含まれている場合に特に便利です。

Kerberos 設定ファイルを作成するには、以下を実行します。

1. 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。以下に例を示します。

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

警告

外部マシンにある既存の **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」 の説明に従って、外部マシンにブラウザーを設定します。

外部システムのユーザーが、**kinit** ユーティリティーを使用して IdM サーバードメインで認証できるようになりました。

5.4.5. プロキシーサーバーおよび Web UI でのポート転送

Web UI にアクセスするためにプロキシーサーバーを使用する場合は、IdM で追加の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 ではプロキシーサーバーを使用できるので、OpenSSH と SOCKS オプションでプロキシー転送を使用して、ポート転送に似た操作を設定できます。ただし、プロキシーサーバーを使用できるため、OpenSSH および SOCKS オプションで、プロキシー転送を使用して、ポート転送に似た操作を設定できます。これは、**ssh** ユーティリティーの **-D** オプションで設定できます。**-D** の使用方法は、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) ドメイン内のサーバー間のレプリケーションを管理する方法を説明します。

注記

本章では、Red Hat Enterprise Linux 7.3 で導入された簡素化されたトポロジー管理について説明します。この手順で必要なドメインレベルは 1 です (*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 でのトポロジーの管理に関するドキュメントは、「レプリカおよびレプリカ合意の管理」 を参照してください。

初期レプリカのインストールとレプリケーションに関する基本情報は、*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、トポロジー接尾辞、およびトポロジーセグメントの説明

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 台のサーバーでレプリカ合意が設定されている場合は、データを共有します。

レプリカ合意は常に双方向のものです。最初のレプリカからサーバーから別のレプリカにデータが複製されるだけでなく、別ののレプリカから最初のレプリカにもデータが複製されます。

注記

詳細は、「IdM レプリカの説明」 を参照してください。

トポロジー接尾辞
*トポロジーの接尾辞*は、レプリケートされるデータを保存します。IdM は、**domain** と **ca** の 2 種類のトポロジー接尾辞に対応します。それぞれの接尾辞は、個別のバックエンドである個別のレプリケーショントポロジーを表します。

レプリカ合意が設定されると、同じタイプのトポロジー接尾辞を 2 つの異なるサーバーに結合します。

domain 接尾辞: dc=*example*,dc=*com*

domain 接尾辞には、ドメイン関連のデータがすべて含まれています。

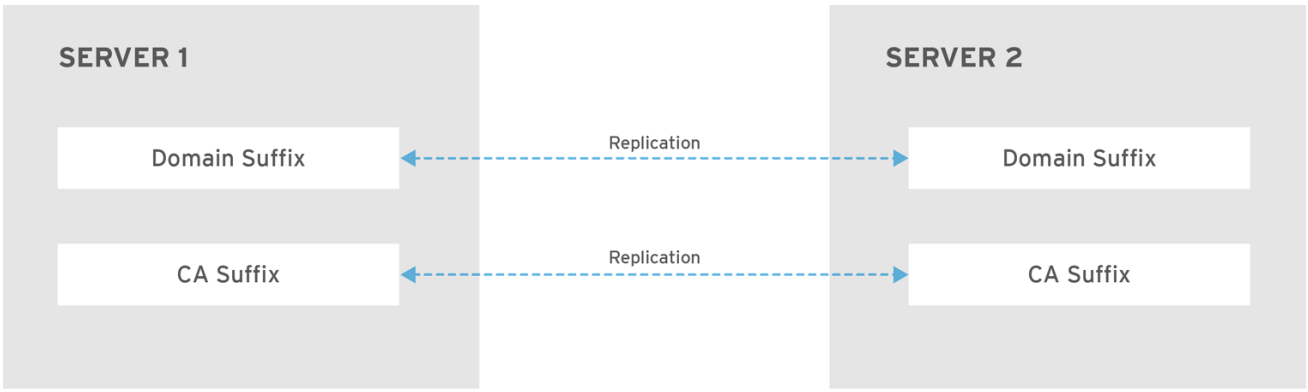
2 つのレプリカの **domain** 接尾辞間でレプリカ合意が設定されると、ユーザー、グループ、およびポリシーなどのディレクトリーデータが共有されます。

ca 接尾辞: o=ipaca

ca 接尾辞には、Certificate System コンポーネントのデータが含まれます。これは認証局 (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** 接尾辞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 トポロジー接尾辞



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 トポロジー接尾辞の表示

ipa topologysuffix-find コマンドでトポロジー接尾辞のリストが表示されます。

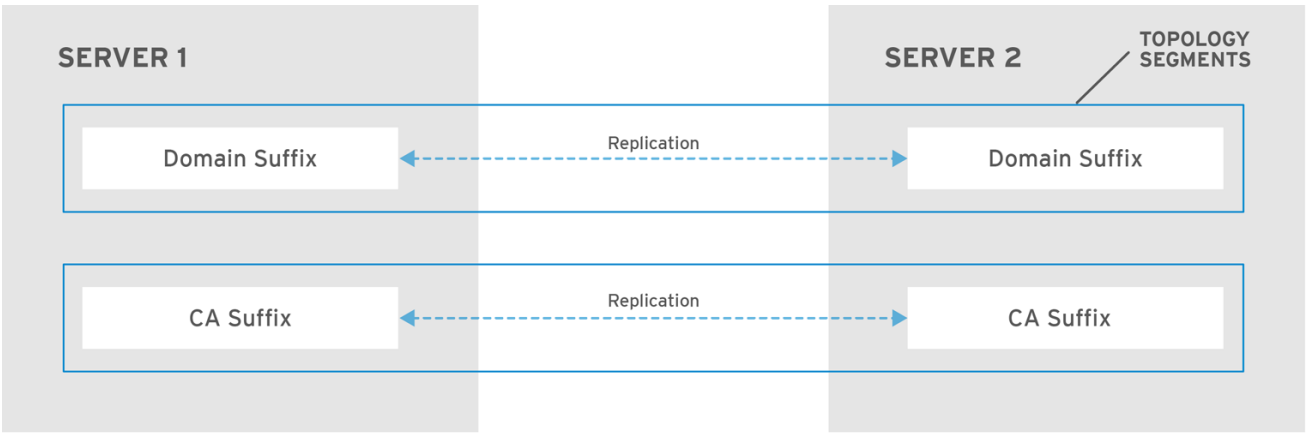
```
$ ipa topologysuffix-find
----------------------------
2 topology suffixes matched
----------------------------
  Suffix name: ca
  Managed LDAP suffix DN: o=ipaca

  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 つのレプリカの接尾辞間でレプリカ合意があると、接尾辞は *topology segment* を形成します。各トポロジーセグメントは、*左ノード*と*右ノード*で設定されます。ノードは、レプリカ合意に参加しているサーバーを表します。

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、およびサーバー B からサーバー A への 2 つのレプリカ合意を表します。そのため、データは両方の方向でレプリケートされます。

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 接尾辞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、ドメイン接尾辞の場合は、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 **server1.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ケーショントポロジーの管理

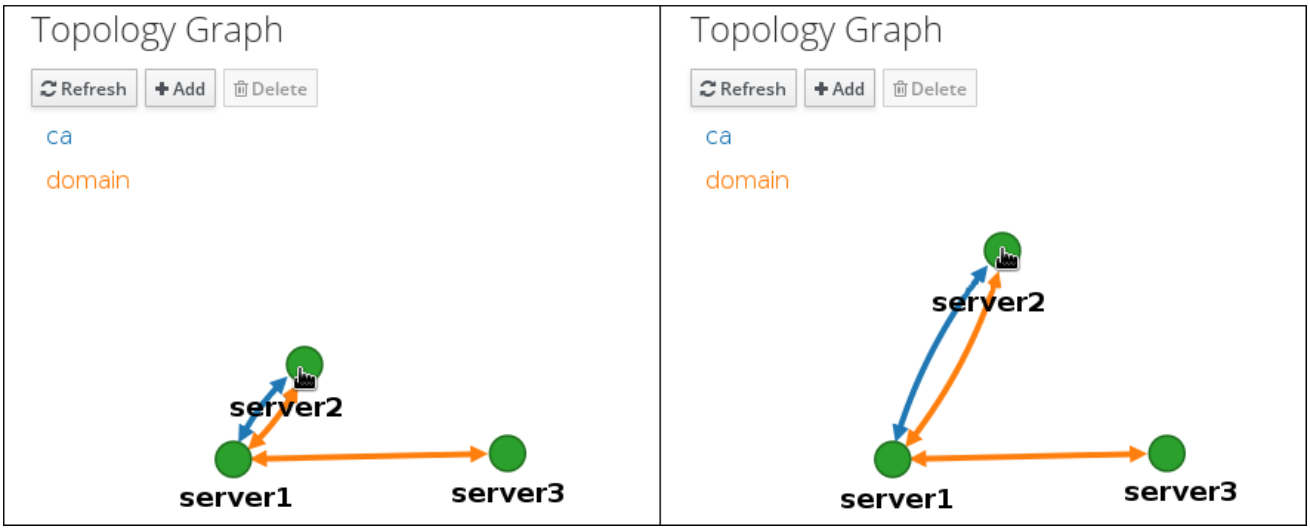
トポロジーグラフへのアクセス
Web UI のトポロジーグラフは、ドメイン内のサーバー間の関係を表示します。

1. **IPA Server** → **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

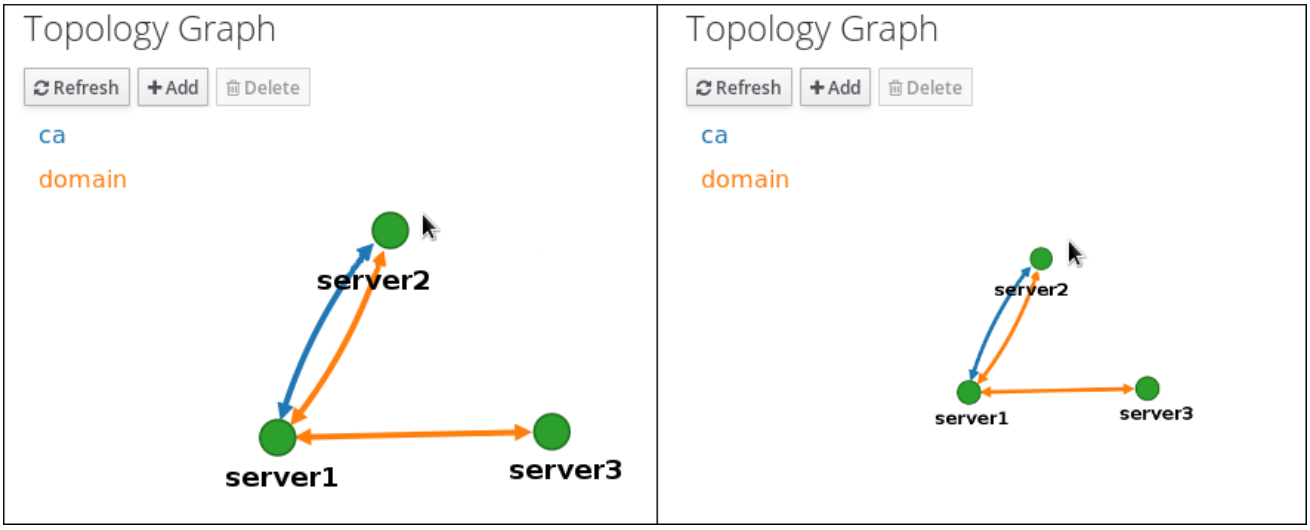
マウスをドラッグして、個別のトポロジーノードを移動できます。

図6.3 トポロジーグラフのノードの移動



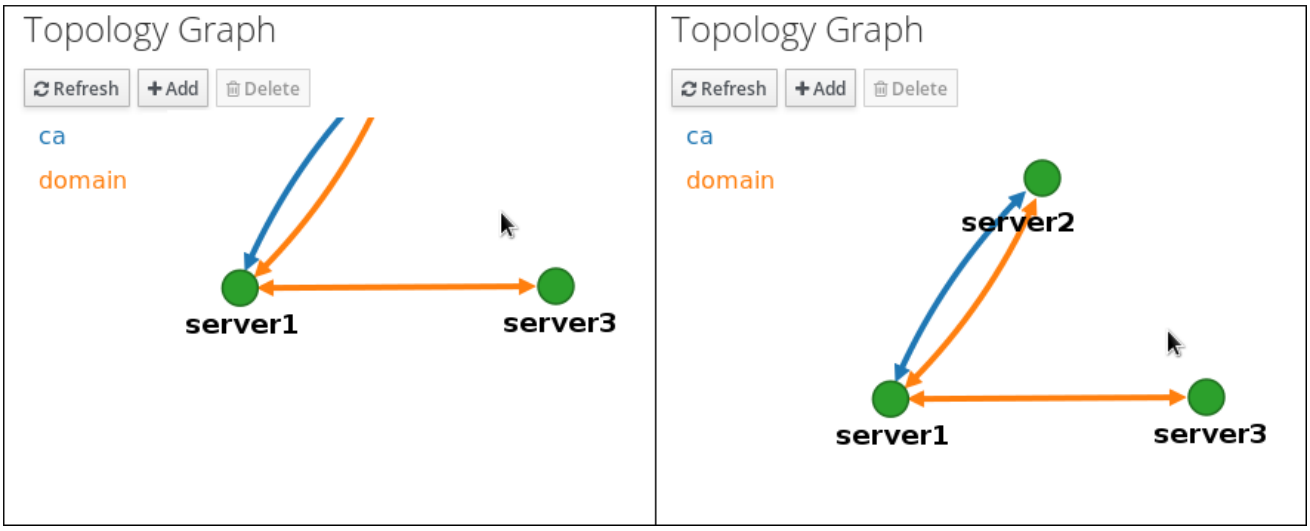
マウスのホイールを使用して、トポロジーグラフを拡大および縮小できます。

図6.4 トポロジーグラフのズーム



マウスの左ボタンを保持することで、トポロジーグラフのキャンバスを移動できます。

図6.5 トポロジーグラフのキャンバスの移動



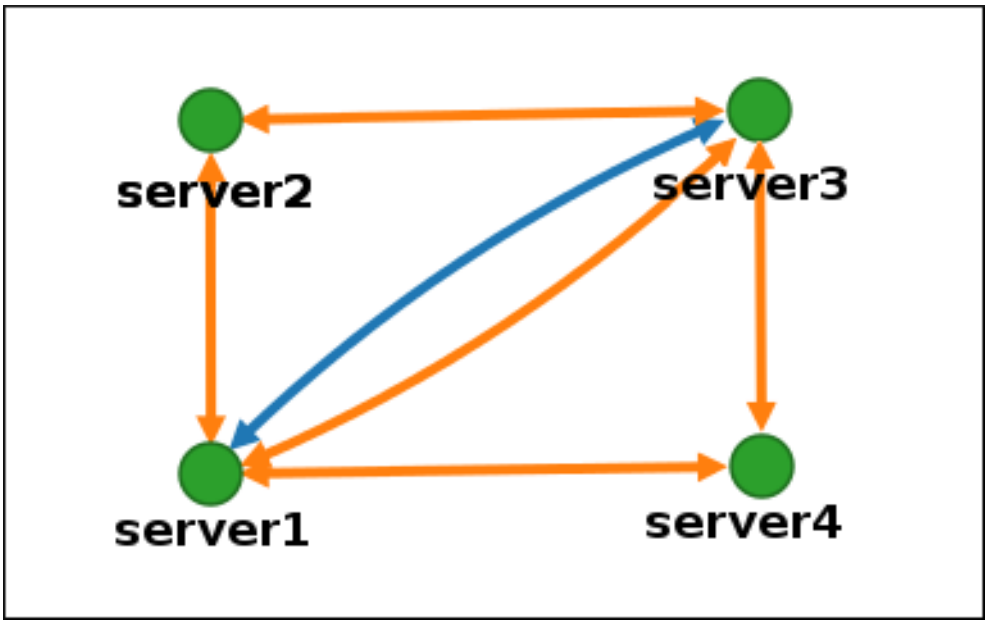
トポロジーグラフの解釈

ドメインのレプリカ合意に参加しているサーバーは、オレンジ色の矢印によって接続されます。CA のレプリカ合意に参加しているサーバーは、青色の矢印によって接続されます。

トポロジーグラフの例: 推奨されるトポロジー

図6.6「推奨されるトポロジーの例」 は、4 つのサーバーで推奨されるトポロジーの例を 1 つ示しています。各サーバーは少なくとも 2 台のサーバーに接続されており、複数のサーバーが CA マスターになります。

図6.6 推奨されるトポロジーの例

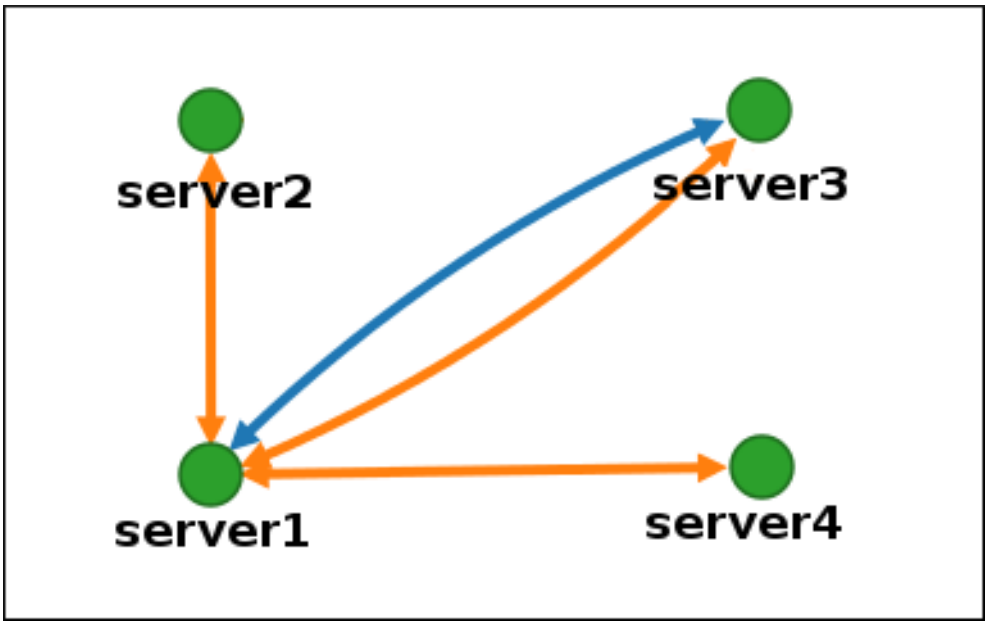


トポロジーグラフの例: 推奨されないトポロジー

図6.7「推奨されないトポロジーの例: 単一障害点」 の **server1** は単一障害点になります。その他のすべてのサーバーは、このサーバーとのレプリカ合意がありますが、他のサーバーとは合意がありません。したがって、**server1** が失敗すると、他のすべてのサーバーは分離されます。

このようなトポロジーの作成は避けてください。

図6.7 推奨されないトポロジーの例: 単一障害点

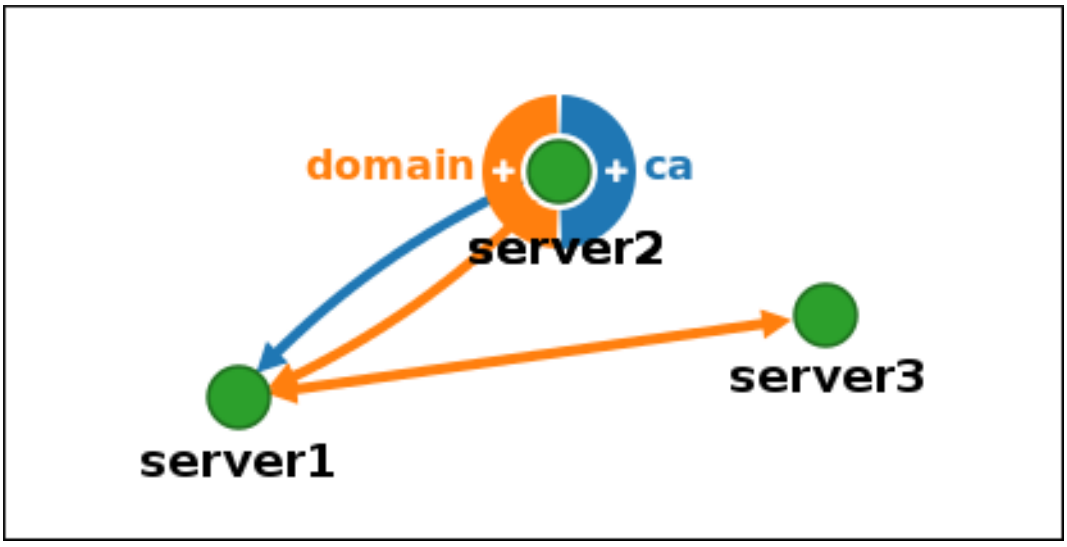


トポロジーの推奨事項の詳細は、「レプリカのデプロイメントに関する考慮事項」 を参照してください。

6.2.1. 2 台のサーバー間のレプリケーションの設定

1. トポロジーグラフで、サーバーノードの 1 つにマウスを合わせます。

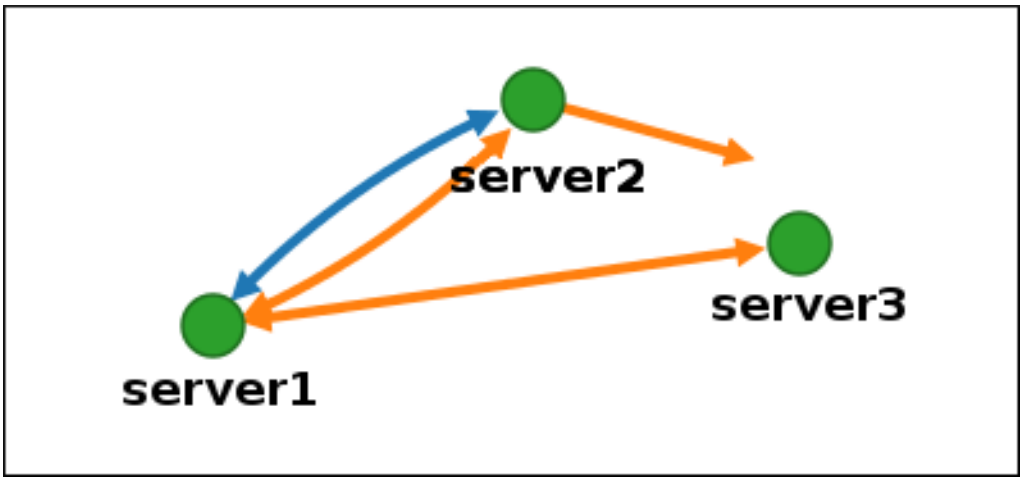
   図6.8 ドメインまたは CA オプション

2. 作成するトポロジーセグメントのタイプに応じて、**domain**または円の**ca**部分をクリックします。

3. 新しいレプリカ合意を表す新しい矢印が、マウスポインターの下に表示されます。マウスを他のサーバーノードに移動し、そこでクリックします。

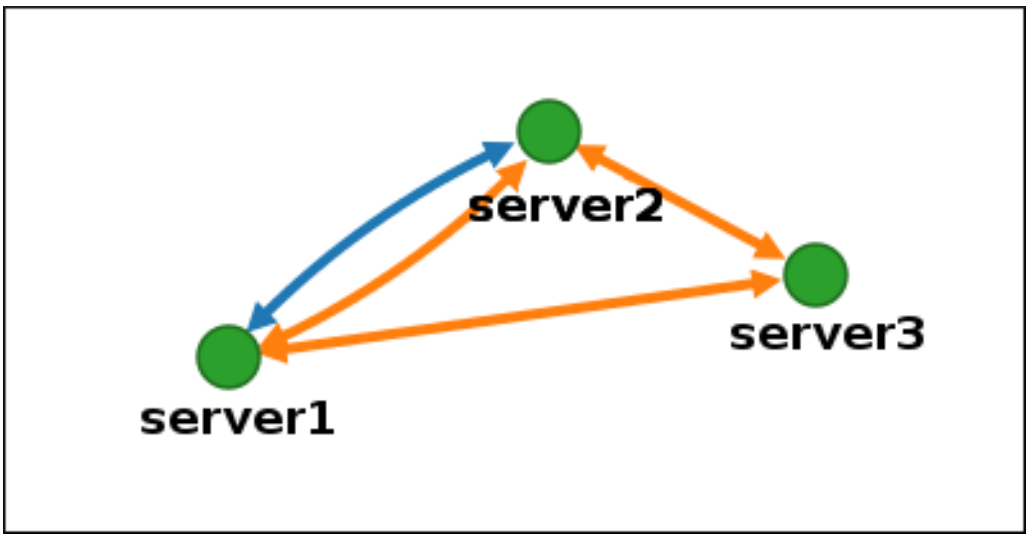
図6.9 新規セグメントの作成



4. **Add Topology Segment** ウィンドウで **Add** をクリックして、新規セグメントのプロパティーを確認します。

IdM は、2 台のサーバーの間に新しいトポロジーセグメントを作成します。これにより、サーバーをレプリカ合意に参加させます。トポロジーグラフには、更新されたレプリケーショントポロジーが表示されるようになりました。

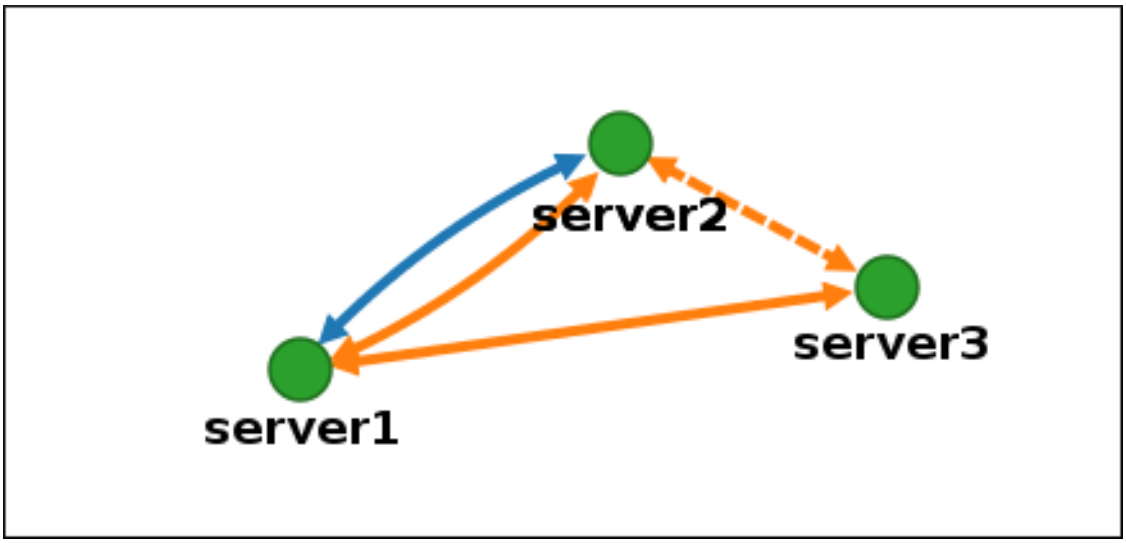
図6.10 作成された新規セグメント



6.2.2. 2 台のサーバー間のレプリケーションの停止

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

図6.11 ハイライト表示されたトポロジーセグメント

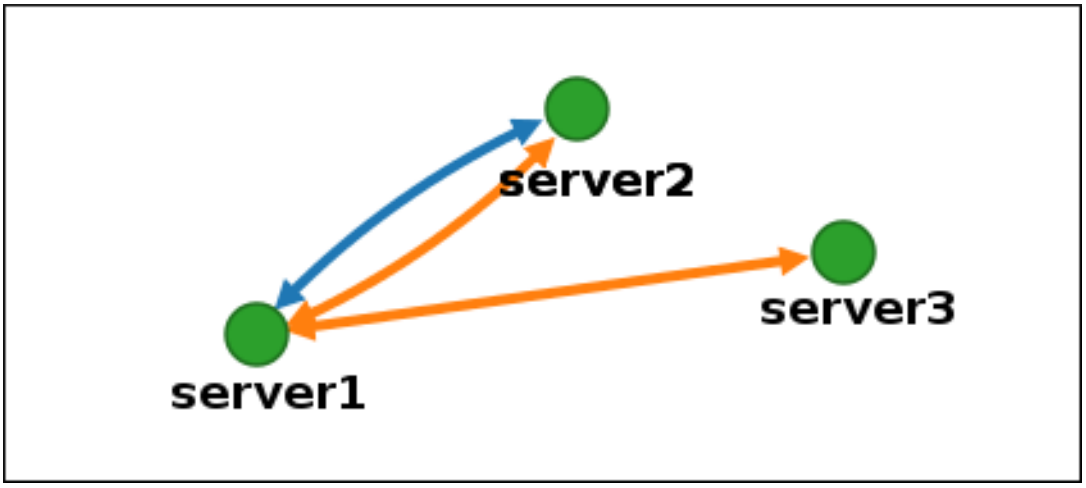


2. **Delete** をクリックします。

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

IdM は、2 台のサーバー間のトポロジーセグメントを削除します。これにより、そのレプリカ合意が削除されます。トポロジーグラフには、更新されたレプリケーショントポロジーが表示されるようになりました。

図6.12 削除されたトポロジーセグメント



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の取得

レプリケーショントポロジーの管理に使用するすべてのコマンドを表示するには、次のコマンドを実行します。

```
$ ipa help topology
```

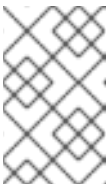
特定のコマンドの詳細なヘルプを表示するには、それを **--help** オプションを指定して実行します。

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

6.3.2. 2 台のサーバー間のレプリケーションの設定

1. **ipa topologysegment-add** コマンドを使用して、2 つのサーバーのトポロジーセグメントを作成します。プロンプトが表示されたら、以下を指定します。

   - 必要なトポロジー接尾辞: **domain** または **ca**

     注記

     ca 接尾辞間の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。「既存の IdM ドメインへの CA のインストール」を参照してください。

   - 2 つのサーバーを表す、左ノードと右のノード
   - 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
   ```

   新しいセグメントを追加すると、サーバーをレプリカ合意に参加させます。

2. **オプション:ipa topologysegment-show** コマンドを使用して、新しいセグメントが設定されたことを確認します。

   ```
   $ 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 台のサーバー間のレプリケーションの停止

1. レプリケーションを停止するには、サーバー間の対応するレプリケーション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、セグメント名を知っている必要があります。

   名前が分からない場合は、**ipa topologysegment-find** コマンドを使用してすべてのセグメントを表示し、出力で必要なセグメントを見つけます。プロンプトが表示されたら、必要なトポロジー接尾辞 (**domain** または **ca**) を指定します。以下に例を示します。

```
$ 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:ipa topologysegment-find** コマンドを使用して、セグメントが表示されなくなったことを確認します。

```
$ 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. トポロジーからのサーバーの削除

以下のいずれかが該当する場合、IdM ではトポロジーからサーバーを削除できません。

- 削除するサーバーが、残りのトポロジーと他のサーバーに接続する唯一のサーバーである。この場合、他のサーバーが分離され、これは許可されません。

- 削除するサーバーが、最後の CA または DNS サーバーである。

このような状況では、エラーで試行に失敗します。たとえば、コマンドラインで以下を行います。

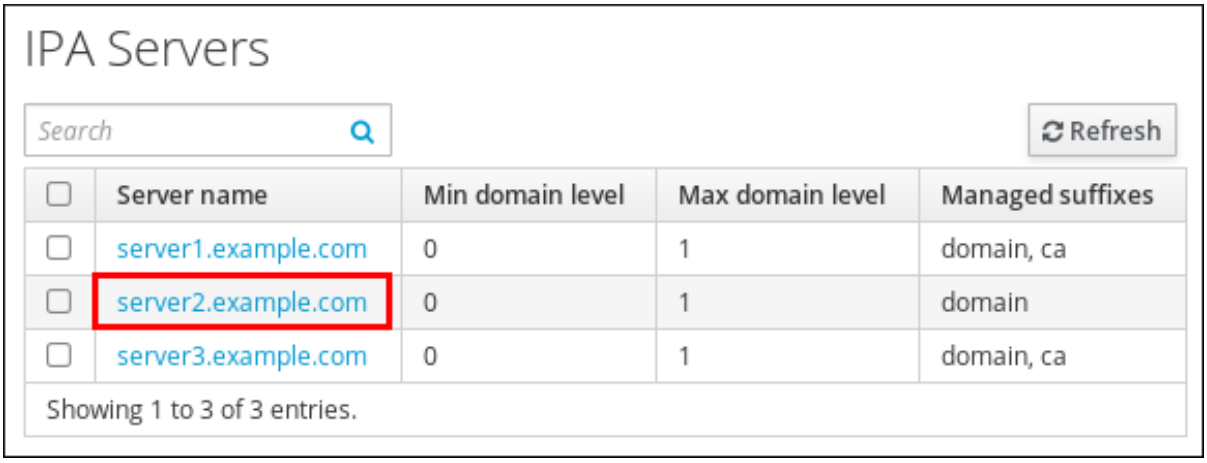
```
$ ipa server-del
Server name: server1.example.com
Removing server1.example.com from replication topology, please wait...
ipa: ERROR: Server removal aborted:

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: トポロジーからのサーバーの削除

サーバーコンポーネントをマシンから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、以下を実行します。

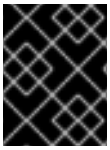
1. **IPA Server** → **Topology** → **IPA Server** を選択します。
2. 削除するサーバーの名前をクリックします。

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: トポロジーからのサーバーの削除



重要

サーバーの削除は元に戻せないアクションです。サーバーを削除すると、トポロジーに戻す唯一の方法は、マシンに新しいレプリカをインストールすることです。

server1.example.com を削除するには、次のコマンドを実行します。

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。このコマンドは、サーバーを参照するすべてのトポロジーセグメントを削除します。

```
[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

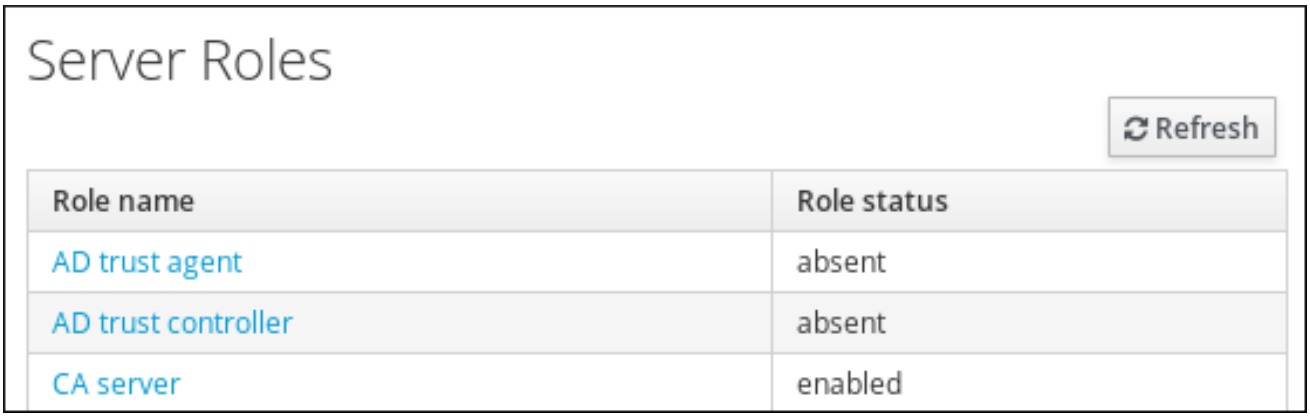
IdM サーバーにインストールされるサービスに基づいて、さまざまな *サーバーロール* を実行できます。例:CA サーバー、DNS サーバー、またはキーリカバリー認証局 (KRA) サーバーなどです。

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完全なリストは、**IPA Server → Topology → Server Roles** を参照してください。

- 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。

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
```

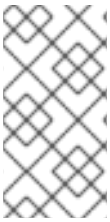
ipa server-show コマンドは、特定のサーバーで有効なロールのリストを表示します。たとえば、*server.example.com* で有効にしたロールのリストは、以下のようになります。

```
$ 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すべてのサーバーを検索します。たとえば、すべての 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 で CA Renewal Master を変更する方法を説明します (*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 CA Renewal Master を変更する方法は、「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。

IdM デプロイメントで組み込み認証局 (CA) を使用する場合は、IdM CA サーバーの 1 つがマスター CA として機能します。これは、CA サブシステム証明書の更新を管理し、証明書失効リスト (CRL) を生成します。デフォルトでは、マスター CA は、システム管理者が **ipa-server-install** コマンドまたは **ipa-ca-install** コマンドを使用して CA ロールをインストールした最初のサーバーです。

マスター CA サーバーをオフラインに移行または廃止する予定の場合には、別の CA サーバーをプロモートして、新しい CA 更新マスターとして置き換えます。

1. CA サブシステム証明書の更新を処理するようにレプリカを設定します。
   - ドメインレベル 1 の場合は 「現在の 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 の場合は 「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. CRL を生成するようにレプリカを設定します。「CRL を生成するサーバーの変更」を参照してください。
3. 以前のマスター CA サーバーの使用を終了する前に、新規マスターが適切に機能していることを確認します。「新規マスター CA サーバーが正しく設定されたことの確認」を参照してください。

6.5.2.1. 現在の CA 更新マスターの変更

Web UI: 現在の 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、新しい CA Renewal Master を選択します。

コマンドライン: 現在の 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

証明書失効リスト (CRL) を生成するサーバーを変更するには、以下を実行します。

1. 現在の CRL 生成マスターが分からない場合は、各 IdM 認証局 (CA) で **ipa-crlgen-manage status** コマンドを使用して、CRL 生成が有効になっているかどうかを確認します。

```
# ipa-crlgen-manage status
CRL generation: enabled
```

2. 現在の CRL 生成マスターで、この機能を無効にします。

```
# ipa-crlgen-manage disable
```

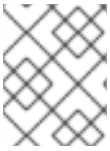
3. 新しい CRL 生成マスターとして設定する他の CA ホストで、この機能を有効にします。

```
# ipa-crlgen-manage enable
```

6.5.2.3. 新規マスター CA サーバーが正しく設定されたことの確認

/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、***ca.crl.MasterCRL.autoUpdateInterval*** パラメーターを使用して **/etc/pki/pki-tomcat/ca/CS.cfg** ファイルで定義した間隔に基づいて生成されます。デフォルト値は 240 分 (4 時間) です。



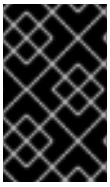
注記

ca.crl.MasterCRL.autoUpdateInterval パラメーターを更新すると、すでにスケジュールされている CRL の次回更新後に変更が有効になります。

ファイルが存在する場合には、新規マスター CA サーバーが正しく設定され、以前の CA マスターシステムを安全に廃止できます。

6.5.3. IdM サーバーからの IdM CA サービスのアンインストール

Red Hat では、トポロジーに CA ロールが割り当てられた Identity Management (IdM)レプリカの最大数を用意することを推奨します。したがって、4 つ以上のレプリカがあり、冗長な証明書のレプリケーションによりパフォーマンスの問題が発生する場合は、IdM レプリカから冗長な CA サービスインスタンスを削除します。そのためには、当該 IdM レプリカの使用を完全に停止してから、CA サービスを使用せずに IdM を再インストールする必要があります。



重要

IdM レプリカに CA ロールを **追加** することはできますが、IdM では、IdM レプリカから CA ロールのみを **削除** する方法はありません。**ipa-ca-install** コマンドには ***--uninstall*** オプションがありません。

1. 冗長 CA サービスを特定し、このサービスをホストする IdM レプリカ上の 「IdM サーバーのアンインストール」 の手順に従います。
2. 同じホストで、ユースケースに応じて 「外部 CA をルート CA として使用するサーバーのインストール」 または 「CA なしでのインストール」 の手順に従います。

6.5.4. 非表示のレプリカのデモートおよびプロモート

レプリカのインストール後、レプリカの表示状態を変更できます。

- 表示されるレプリカを非表示のレプリカにデモートするには、以下を実施します。
  1. レプリカが CA Renewal Master である場合は、サービスを別のレプリカに移動します。詳細は、「現在の CA 更新マスターの変更」を参照してください。
  2. レプリカの状態を **hidden** に変更します。

```
# ipa server-state replica.idm.example.com --state=hidden
```

- 非表示のレプリカを表示されるレプリカにプロモートするには、次のコマンドを入力します。

```
# ipa server-state replica.idm.example.com --state=enabled
```

注記

非表示のレプリカ機能は、テクノロジープレビューとして Red Hat Enterprise Linux 7.7 以降で利用でき、サポート対象外となります。

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 トポロジーで利用可能な操作および機能を示します。

ドメインレベル 1

利用可能な機能の例:

- **ipa-replica-install** の簡素化し (「レプリカの作成: 概要」を参照)
- トポロジー管理の強化 (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、IdM バージョン 4.4 で Red Hat Enterprise Linux 7.3 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、すべてのレプリカが Red Hat Enterprise Linux 7.3 以降を実行している必要があります。

最初のサーバーが Red Hat Enterprise Linux 7.3 でインストールされている場合、ドメインの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以前のバージョンから IdM バージョン 4.4 にアップグレードすると、ドメインレベルは自動的に引き上げられません。ドメインレベル 1 の機能を使用する場合は、「ドメインレベルの引き上げ」 の説明に従ってドメインレベルを手動で増やします。

ドメインレベル 0

利用可能な機能の例:

- **ipa-replica-install** では、初期サーバーでレプリカ情報ファイルを作成してレプリカにコピーするというより複雑なプロセスが必要です (「レプリカの作成」を参照)。
- **ipa-replica-manage** と **ipa-csreplica-manage** を使用したより複雑なトポロジー管理 (「レプリカおよびレプリカ合意の管理」を参照)

7.1. 現在のドメインレベルの表示

コマンドライン: 現在の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: 現在のドメインレベルの表示

IPA Server → Topology → Domain Level を選択します。

7.2. ドメインレベルの引き上げ

重要

これは元に戻すことができない操作です。ドメインレベルを **0** から **1** に増やしたら、**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** コマンドを実行して、必要なレベルを指定します。

```
$ 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 → Topology → 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 の更新および移行

8.1. IDENTITY MANAGEMENT の更新

yum ユーティリティーを使用して、システムの Identity Management パッケージを更新できます。

警告

更新をインストールする前に、RHEL システムに関連するこれまでにリリース済みのエラータをすべて適用していることを確認します。詳細は、RHEL システムにパッケージの更新を適用する方法 を参照してください。KCS の記事。

さらに、7.3 などの新しい Red Hat Enterprise Linux バージョンが利用可能になると、**yum** は Identity Management サーバーまたは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、Identity Management を Red Hat Enterprise Linux 6 から Red Hat Enterprise Linux 7 に移行することは説明しません。移行する場合は、「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 の更新に関する考慮事項

- 少なくとも 1 台のサーバーで Identity Management パッケージを更新すると、トポロジー内のその他のすべてのサーバーでパッケージを更新しなくても、更新されたスキーマを受け取ります。これは、新しいスキーマを使用する新しいエントリーを、その他のサーバー間で確実に複製できます。
- 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-* パッケージで **yum downgrade** コマンドを実行しないでください。

- Red Hat は、次のバージョンにアップグレードすることのみを推奨します。たとえば、Red Hat Enterprise Linux 7.4 の Identity Management にアップグレードする場合は、Red Hat Enterprise Linux 7.3 の Identity Management からアップグレードすることを推奨します。以前のバージョンからアップグレードすると、問題が発生する可能性があります。

8.1.2. yum を使用した Identity Management パッケージの更新

サーバーまたはクライアントの Identity Management パッケージをすべて更新するには、次のコマンドを実行します。

```
# 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、各アップグレードの間隔は少なくとも 10 分あけてください。

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** ユーティリティーの使用方法は、『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、***NSSProtocol*** パラメーターを **TLSv1.0** (後方互換性用)、**TLSv1.1**、および **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。

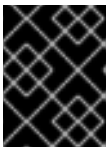
8.2. RED HAT ENTERPRISE LINUX 6 からバージョン 7 への IDENTITY MANAGEMENT の移行

この手順では、すべてのデータおよび設定を Red Hat Enterprise Linux 6 Identity Management から Red Hat Enterprise Linux 7 サーバーに移行する方法を説明します。移行手順には、以下が含まれます。

- Red Hat Enterprise Linux 6 ベースの認証局 (CA) マスターサーバーを Red Hat Enterprise Linux 7 に移行する。
- すべてのサービスを新しい Red Hat Enterprise Linux 7 サーバーに移行する。これらのサービスには、CRL および証明書の作成、DNS 管理、または Kerberos KDC の管理が含まれます。
- 元の Red Hat Enterprise Linux 6 CA マスターの使用を終了する。

手順では、以下を前提としています。

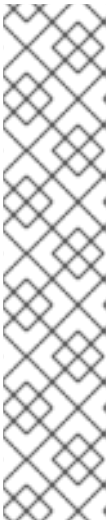
- **rhel7.example.com** は、新しい CA マスターとなる Red Hat Enterprise Linux 7 システムです。



重要

現在サポートされているマイナーバージョンは RHEL 7.9 のみです。システムに RHEL 7.9 がインストールされていることを確認してください。

- **rhel6.example.com** は、元の Red Hat Enterprise Linux 6 CA マスターです。



注記

マスター CA サーバーである Red Hat Enterprise Linux 6 サーバーを特定するには、**certmonger** サービスが **renew_ca_cert** コマンドを追跡するサーバーを決定します。すべての Red Hat Enterprise Linux 6 サーバーでこのコマンドを実行します。

```
[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 を実行する保存後アクションは CA マスターに対してのみ定義されます。

8.2.1. Red Hat Enterprise Linux 6 から 7 への Identity Management の移行の前提条件

- **rhel6.example.com** システムを最新の Red Hat Enterprise Linux 6 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-*
```

この手順では、RHBA-2015:0231-2アドバイザリーが適用されていることも確認します。このアドバイザリーは、**2.3-6.el6_6** バージョンのbind-dyndb-ldapパッケージを提供し、Red Hat Enterprise Linux 6.6 拡張更新サポート (EUS) で使用できます。

警告

以前のバージョンの bind-dyndb-ldap を使用すると、Red Hat Enterprise Linux 6.6 DNS サーバーおよび Red Hat Enterprise Linux 7 DNS サーバー間の DNS 正引きゾーンに一貫性がない動作が生じます。

- **rhel7.example.com** システムが 「サーバーの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」 の要件を満たしていることを確認します。
- **rhel7.example.com** システムで、必要なパッケージをインストールします。「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。

8.2.2. Red Hat Enterprise Linux 6 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は、**rhel7.example.com** レプリカのインストールに **rhel6.example.com** を準備します。Identity Management バージョン 3.1 とそれ以降のバージョン間のスキーマの変更により、スキーマを更新する必要があります。

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。以下に例を示します。

```
[root@rhel7 ~]# scp /usr/share/ipa/copy-schema-to-ca.py root@rhel6:/root/
```

2. **rhel6.example.com** で更新された **copy-schema-to-ca.py** スクリプトを実行します。

```
[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
```

3. Red Hat Enterprise Linux 7 レプリカに接続する前に、認証局を実行するすべての Red Hat Enterprise Linux 6 IdM レプリカで手順を繰り返します。

8.2.3. Red Hat Enterprise Linux 7 レプリカのインストール

1. **rhel6.example.com** システムで、**rhel7.example.com** レプリカをインストールするために使用するレプリカファイルを作成します。たとえば、IP アドレスが **192.0.2.1** である **rhel7.example.com** のレプリカファイルを作成するには、次のコマンドを実行します。

```
[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」 および 「レプリカの作成」 も参照してください。

2. **rhel6.example.com** から **rhel7.example.com** に、レプリカ情報ファイルをコピーします。

```
[root@rhel6 ~]# scp /var/lib/ipa/replica-info-replica.example.com.gpg root@rhel7:/var/lib/ipa/
```

3. 統合 CA のある新しいレプリカを Red Hat Enterprise Linux 7.6 以降にインストールする場合は、**/etc/httpd/conf.d/nss.conf** ファイルの ***NSSCipherSuite*** パラメーターの最後に以下のエントリーを追加します。

```
+ecdhe_rsa_aes_128_sha,+ecdhe_rsa_aes_256_sha
```

Red Hat Enterprise Linux 7.6 以降では、特定の暗号は IdM CA ではデフォルトで有効ではなくなりました。このエントリーを設定に追加せずに、Red Hat Enterprise Linux 6 で実行しているマスターのレプリカとして、統合 CA のある IdM サーバーを Red Hat Enterprise Linux 7.6 でセットアップすると、**CRITICAL Failed to configure CA instance** エラーが発生して失敗します。

4. 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、次のコマンドでは、以下のオプションを使用しています。

   - Certificate System コンポーネントを設定する **--setup-ca**

- 統合 DNS サーバーを設定し、フォワーダーを設定する **--setup-dns** および **--forwarder**
- **--ip-address** - **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」: レプリカ情報ファイルを使用してレプリカを作成する方法を説明
- 「統合 DNS を使用するかどうかの決定」 および 「使用する CA 設定の決定」

5. 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

作業を開始する前に:

- **rhel6.example.com** および **rhel7.example.com** の CA がいずれも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 サブシステム証明書の更新を停止します。

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 "20201127184547" removed.
[root@rhel6 ~]# getcert stop-tracking -d /var/lib/pki-ca/alias -n "ocspSigningCert cert-pki-ca"
```

```
Request "20201127184548" removed.
[root@rhel6 ~]# getcert stop-tracking -d /var/lib/pki-ca/alias -n "subsystemCert cert-pki-ca"
Request "20201127184549" removed.
[root@rhel6 ~]# getcert stop-tracking -d /etc/httpd/alias -n ipaCert
Request "20201127184550" removed.
```

2. **rhel6.example.com** を再設定し、新しいマスター CA から更新された証明書を取得します。

   a. 更新ヘルパースクリプトを **certmonger** サービスディレクトリーにコピーし、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 が証明書を取得するようになっ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の PIN を取得します。

   ```
   [root@rhel6 ~]# grep internal= /var/lib/pki-ca/conf/password.conf
   ```

   f. 外部更新の証明書 **certmonger** 追跡を設定します。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 "20201127184743" 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 "20201127184744" 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 "20201127184745" 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 "20201127184746" 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** で CRL 生成を無効にします。**/var/lib/pki-ca/conf/CS.cfg** ファイルを開き、***ca.crl.MasterCRL.enableCRLCache*** および **ca.crl.MasterCRL.enableCRLUpdates** 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、CRL 要求をリダイレクトするように Apache を設定します。

   a. **/etc/httpd/conf.d/ipa-pki-proxy.conf** ファイルを開いて、**RewriteRule** エントリーをコメントアウトします。

   ```
   RewriteRule ^/ipa/crl/MasterCRL.bin https://rhel6.example.com/ca/ee/ca/getCRL?
   op=getCRL&crlIssuingPoint=MasterCRL [L,R=301,NC]
   ```

注記

URL のサーバーのホスト名を置き換えないでください。URL はローカルホスト名を参照する必要があります。

b. Apache を再起動します。

```
[root@rhel6 ~]# service httpd restart
```

IdM は、ローカルファイルではなく、ローカル CA から証明書失効リスト (CRL) を取得するようになりました。

3. **rhel7.example.com** で、新しい CA マスターとして **rhel7.example.com** を設定します。

a. 「証明書更新を処理するサーバーの変更」 の説明に従って、CA サブシステム証明書の更新を処理するように **rhel7.example.com** を設定します。

b. 「CRL を生成するサーバーの変更」 で説明されているように、**rhel7.example.com** を一般的な証明書失効リスト (CRL) に設定します。

関連情報

- CA サブシステム証明書の更新および CRL 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、新しい **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 サーバーの移行後の次のステップ

トポロジーの各 Red Hat Enterprise Linux 6 サーバーの場合:

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すると、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、レプリカのインストールや管理が容易になりました。これらの機能を使用するには、トポロジーはドメインレベル 1 にする必要があります。*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 システムに新しい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 サーバーの使用を終了するには、以下を実行します。

- Red Hat Enterprise Linux 7 サーバー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。

重要

クライアント設定は自動的に更新されません。IDM サーバーの使用を終了し、新しいサーバーを異なる名前で設定した場合は、全体的なクライアント設定を確認する必要があります。特に、以下のファイルを手動で更新する必要があります。

- **/etc/openldap/ldap.conf**
- **/etc/ipa/default.conf**
- **/etc/sssd/sssd.conf**

第9章 IDENTITY MANAGEMENT のバックアップおよび復元

Red Hat Enterprise Linux Identity Management は、たとえばサーバーが正常に動作しなくなった場合やデータの喪失が発生した場合に、IdM システムを手動でバックアップして復元するソリューションを提供します。バックアップ時に、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、このバックアップディレクトリーを使用して、元の IdM セットアップを復元できます。

重要

失われたレプリカを残りのレプリカとして再インストールすることで、デプロイメント内の残りのサーバーから IdM サーバーグループの失われた部分を再構築できない場合に限り、本章で説明するバックアップおよび復元手順を使用します。

ナレッジベースソリューションBackup and Restore in IdM/IPA では、複数のサーバーレプリカを維持して損失を回避する方法を説明します。バックアップバージョンには古い使用できない情報が含まれるため、同じデータを持つ既存のレプリカから再構築することが推奨されます。

バックアップおよび復元が回避できる可能性のある脅威シナリオには、以下が含まれます。

- マシンでの致命的なハードウェア障害が発生し、マシンはそれ以降機能しなくなる。この場合、以下を実施します。

  1. オペレーティングシステムをゼロから再インストールします。
  2. マシンには、同じホスト名、完全修飾ドメイン名 (FQDN)、および IP アドレスを設定します。
  3. IdM パッケージと、元のシステムに存在していた IdM に関連するその他のオプションパッケージをすべてインストールします。
  4. IdM サーバーの完全バックアップを復元します。

- 分離されたマシンでのアップグレードに失敗する。オペレーティングシステムは機能し続けますが、IdM データが破損するため、IdM システムを既知の正常な状態に復元したい理由になります。

重要

上記の 2 項目など、ハードウェア障害またはアップグレードが失敗した場合は、すべてのレプリカまたは特別なロールを持つレプリカ (唯一の認証局 (CA) など) が失われた場合にのみバックアップから復元します。同じデータを持つレプリカがまだ存在する場合は、失われたレプリカを削除してから残りのレプリカから再構築することが推奨されます。

- LDAP コンテンツに望ましくない変更 (エントリーが削除された等) が加えられたため、それを元に戻したい。バックアップされた LDAP データを復元すると、IdM システム自体に影響を与えずに LDAP エントリーが以前の状態に戻ります。

復元されたサーバーは、IdM の唯一の情報源になります。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後に作成されたデータはすべて失われます。したがって、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ソリューションを使用しないでください。可能な場合は、レプリカとして再インストールすることで、常に失われたサーバーを再構築します。

バックアップ機能および復元機能はコマンドラインからのみ管理でき、IdM Web UI では使用できません。

9.1. サーバーのフルバックアップおよびデータのみのバックアップ

IdM には、以下の 2 つのバックアップオプションがあります。

IdM サーバーの完全なバックアップ

サーバーのフルバックアップは、すべての IdM サーバーファイルと LDAP データのバックアップコピーを作成します。これにより、スタンドアロンバックアップが作成されます。IdM は数百のファイルに影響します。バックアッププロセスのコピーするファイルは、ディレクトリー全体と特定のファイルが混在したもので (例: 設定ファイルやログファイル)、IdM に直接関係したり、IdM が依存するさまざまなサービスに関連したりします。サーバーのフルバックアップは raw ファイルのバックアップであるため、これはオフラインで実行されます。サーバーのフルバックアップを実行するスクリプトは、すべての IdM サービスを停止し、バックアッププロセスの安全を確保します。

完全なサーバーバックアップコピーの全リストについては、「バックアップ中にコピーされたディレクトリーおよびファイルのリスト」 を参照してください。

データのみのバックアップ

データのみのバックアップは、LDAP データおよび変更ログのバックアップコピーのみを作成します。このプロセスは **IPA-REALM** インスタンスをバックアップし、複数のバックエンドのバックアップを作成することも、単一のバックエンドのみをバックアップします。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アップは、LDIF(LDAP データ交換形式) に保存されている LDAP コンテンツのレコードもバックアップします。データのみのバックアップは、オンラインとオフラインの両方で実行できます。

デフォルトでは、IdM は作成したバックアップを **/var/lib/ipa/backup/** ディレクトリーに保存します。バックアップを含む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。
- データのみ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みのバックアップも、**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。

サーバーのフルバックアップを作成するには、**ipa-backup** を実行します。

重要

プロセスをオフラインで実行する必要があるため、サーバーのフルバックアップを実行すると、すべての IdM サービスが停止します。バックアップが完了すると、IdM サービスが再起動します。

データのみ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 にいくつかのオプションを追加できます。

- **--online** はオンラインバックアップを実行します。このオプションはデータのみのバックアップでのみ利用できます。

- **--logs** は、バックアップに IdM サービスログファイルを追加します、

ipa-backup の使用方法は、ipa-backup(1) の man ページを参照してください。

9.1.1.1. バックアップ中ボリューム上に十分な領域がない場合の回避策

本セクションでは、IdM バックアッププロセスに関与するディレクトリーが空き領域が不十分なボリュームに保存されている場合に問題に対応する方法を説明します。

/var/lib/ipa/backup/ が含まれるボリューム上の領域が不十分
空き容量が不足しているボリュームに**/var/lib/ipa/backup/**ディレクトリーが保存されている場合は、バックアップを作成することはできません。この問題に対処するには、以下の回避策のいずれかを使用します。

- 別のボリュームにディレクトリーを作成し、**/var/lib/ipa/backup/** にリンクします。たとえば、**/home** が十分な空き領域を持つ別のボリュームに保存されている場合は、次のコマンドを実行します。

  1. **/home/idm/backup/** などのディレクトリーを作成します。

     ```
     # mkdir -p /home/idm/backup/
     ```

  2. 以下のパーミッションをディレクトリーに設定します。

     ```
     # chown root:root /home/idm/backup/
     # chmod 700 /home/idm/backup/
     ```

  3. **/var/lib/ipa/backup/** に既存のバックアップが含まれている場合は、新しいディレクトリーに移動します。

     ```
     # mv /var/lib/ipa/backup/* /home/idm/backup/
     ```

  4. **/var/lib/ipa/backup/** ディレクトリーを削除します。

     ```
     # rm -rf /var/lib/ipa/backup/
     ```

  5. **/var/lib/ipa/backup/** リンクを **/home/idm/backup/** ディレクトリーに作成します。

     ```
     # ln -s /home/idm/backup/ /var/lib/ipa/backup/
     ```

- 別のボリュームに保存されているディレクトリーを **/var/lib/ipa/backup/** にマウントします。たとえば、**/home** が十分な空き領域がある別のボリュームに保存されている場合は、**/home/idm/backup/** を作成し、**var/lib/ipa/backup/** にマウントします。

  1. **/home/idm/backup/** ディレクトリーを作成します。

     ```
     # mkdir -p /home/idm/backup/
     ```

  2. 以下のパーミッションをディレクトリーに設定します。

```
# chown root:root /home/idm/backup/
# chmod 700 /home/idm/backup/
```

3. **/var/lib/ipa/backup/** に既存のバックアップが含まれている場合は、新しいディレクトリーに移動します。

```
# mv /var/lib/ipa/backup/* /home/idm/backup/
```

4. **/home/idm/backup/** を **/var/lib/ipa/backup/** にマウントします。

```
# mount -o bind /home/idm/backup/ /var/lib/ipa/backup/
```

5. システムの起動時に、**/home/idm/backup/** を **/var/lib/ipa/backup/** に自動的にマウントするには、以下を **/etc/fstab** ファイルに追加します。

```
/home/idm/backup/     /var/lib/ipa/backup/     none     bind     0 0
```

/tmp が含まれるボリューム上の領域が不十分
/tmp ディレクトリーに十分な領域がないためにバックアップが失敗する場合は、**TMPDIR** 環境変数を使用して、バックアップ中に作成されるステージファイルの場所を変更します。

```
# TMPDIR=/path/to/backup ipa-backup
```

詳細は、ナレッジベースソリューションipa-backup command fails to finish を参照してください。

9.1.2. バックアップの暗号化

GPG(GNU Privacy Guard) 暗号化を使用して IdM バックアップを暗号化できます。

GPG キーを作成するには、以下を実行します。

1. 鍵の詳細を含む **keygen** ファイルを作成します。たとえば**cat >keygen <<EOF**を実行して、コマンドラインからファイルに必要な暗号化の詳細を指定します。

```
[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** と呼ばれる新しいキーペアを生成し、**keygen** の内容をコマンドに入力します。以下の例では、パス名 **/root/backup.sec** および **/root/backup.pub** でキーペアを生成します。

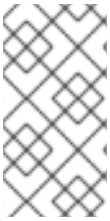
```
[root@server ~]# gpg --batch --gen-key keygen
[root@server ~]# gpg --no-default-keyring --secret-keyring /root/backup.sec \
      --keyring /root/backup.pub --list-secret-keys
```

GPG で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指定して生成された **バックアップ** キーを **ipa-backup** に渡します。

- **--GPG** - 暗号化されたバックアップを実行するために **ipa-backup** に指示します。
- **--gpg-keyring=GPG_KEYRING**: ファイル拡張子なしで GPG キーリングへの完全パスを提供します。

以下に例を示します。

```
[root@server ~]# ipa-backup --gpg --gpg-keyring=/root/backup
```



注記

gpg2 が機能するには外部プログラムが必要なため、システムに **gpg2** ユーティリティーを使用して GPG キーを生成すると問題が発生する可能性があります。この場合は、コンソールから鍵を純粋に生成するには、キーを生成する前に **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たディレクトリーおよびファイルのリスト

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/sssd/sssd.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/sssd.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/opendnssec/conf.xml
/etc/opendnssec/kasp.xml
/etc/ipa/dnssec/softhsm2.conf
/etc/ipa/dnssec/softhsm_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
```

ログファイルとディレクトリー:

```
/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を使用して作成されたバックアップのあるディレクトリーがある場合は、IdM サーバーまたは LDAP コンテンツをバックアップが実行されたときの状態に復元できます。バックアップが作成されたホストとは異なるホストでバックアップを復元することはできません。

注記

IdM サーバーをアンインストールしても、このサーバーのバックアップは自動的に削除されません。

9.2.1. サーバーのフルバックアップまたはデータのみのバックアップからの復元

重要

サーバーをアンインストールしてからフルサーバーの復元を行うことが推奨されます。

完全なサーバーおよびデータのみのバックアップは、**ipa-restore** ユーティリティーを使用して復元します。これは、常に root として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。

- デフォルトの **/var/lib/ipa/backup/** ディレクトリーにある場合に、バックアップでディレクトリーの名前のみを渡します。
- バックアップを含むディレクトリーがデフォルトディレクトリーにない場合は、バックアップへのフルパスを渡します。以下に例を示します。

```
[root@server ~]# ipa-restore /path/to/backup
```

ipa-restore ユーティリティーは、バックアップディレクトリーに含まれるバックアップタイプを自動的に検出し、デフォルトでは同じタイプの復元を実行します。

ipa-restore に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の復元を実行します。つまり、サーバーのフルバックアップを含むバックアップディレクトリーから LDAP データコンポーネントのみを復元します。
- **--online** は、オンラインでデータのみの復元で LDAP データを復元します。
- **--instance** は、どの 389 DS インスタンスを復元するかを指定します。Red Hat Enterprise Linux 7 の IdM は **IPA-REALM** インスタンスのみを使用しますが、たとえば別のインスタンス

を持つシステムでバックアップを作成することは可能です。このような場合は、**--instance** では **IPA-REALM** のみを復元することができます。以下に例を示します。

```
[root@server ~]# ipa-restore --instance=IPA-REALM /path/to/backup
```

このオプションは、データのみの復元を実行する場合にのみ使用できます。

- **--backend** は、どのバックエンドが復元されるかを指定します。このオプションがないと、**ipa-restore** は検出するすべてのバックエンドを復元します。可能なバックエンドを定義する引数は **userRoot** で、IPA データバックエンドを復元し、CA バックエンドを復元する **ipaca** です。

  このオプションは、データのみの復元を実行する場合にのみ使用できます。

- **--no-logs** は、ログファイルを復元せずにバックアップを復元します。

IdM マスターで認証の問題を回避するには、復元後に SSSD キャッシュを消去します。

1. SSSD サービスを停止します。

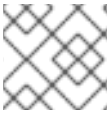
```
[root@server ~]# systemctl stop sssd
```

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

バックアップからの復元後に、システムを再起動することが推奨されます。

ipa-restore の使用方法は、ipa-restore(1) の man ページを参照してください。

9.2.2. 複数のマスターサーバーでの復元

マルチマスターレプリケーション環境で IdM を復元する方法は、「IdM のバックアップおよびリストア」を参照してください。

9.2.3. 暗号化された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復元する場合は、**--gpg-keyring** オプションを使用して、秘密鍵と公開鍵への完全パスを指定します。以下に例を示します。

```
[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、マシン、サービス、エントリーなどの特定のリソースにアクセスできるユーザーや、実行可能な操作の種類を定義するセキュリティー機能のセットです。Identity Management は複数のアクセス制御機能を提供し、付与されているアクセスの種類と、誰に付与されているかが明らかになります。この一環として、Identity Management は、ドメイン内のリソースへのアクセス制御と、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 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、その他の IdM エンティティー (Directory Server インスタンスにも LDAP エントリーとして保存されている) へのアクセスが許可されている IdM ユーザー (バックエンド Directory Server インスタンスに保存されている) に基づいています。

アクセス制御指示 (ACI) には、以下の 3 つの部分があります。

アクター

これは、何かを実行するためのパーミッションが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。

操作タイプ

操作タイプ – 最後の部分は、ユーザーが実行できるアクションの種類を判断します。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。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 のアクセス制御メソッド

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

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。

委譲ルール

委譲ルールでは、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の属性に対して書き込み (編集) 操作を実行できます。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。

Identity Management ですでに作成され、利用可能なロールもあります。ホストや自動マウント設定、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

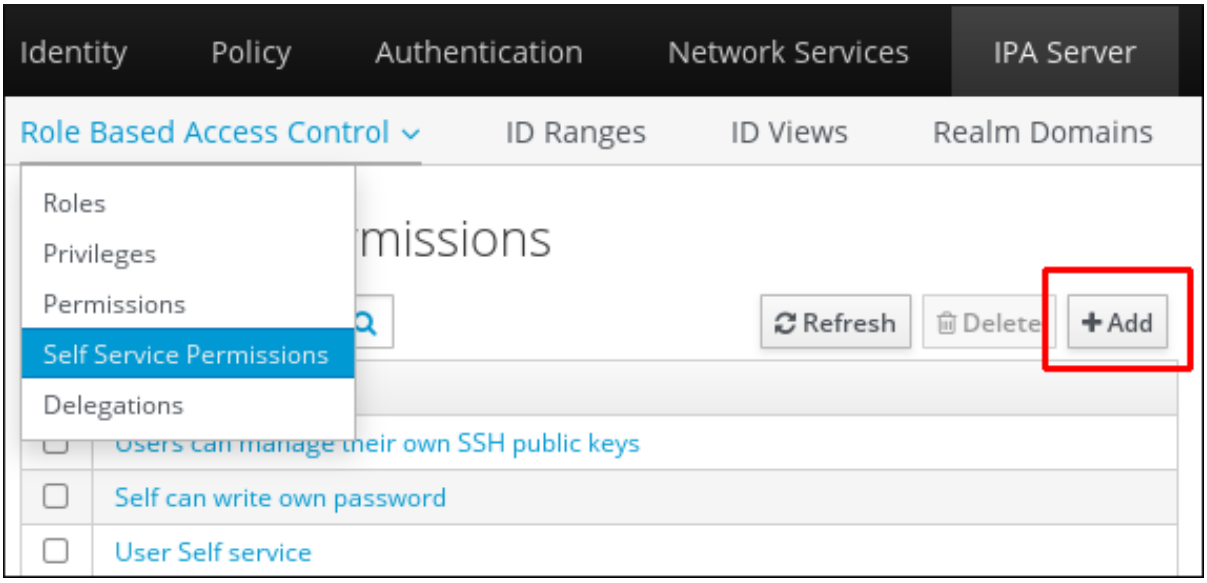
セルフサービスのアクセス制御ルールでは、エントリーがそれ自体で実行可能な操作を定義します。このルールでは、ユーザー (または他の IdM エンティティー) が自身のパーソナルエントリーで編集可能な属性のみを定義します。

10.2.1. Web UI でのセルフサービスルールの作成

1. トップメニューの **IPA Server** タブで、**Role-Based Access Control → Self Service Permissions** サブタブを選択します。

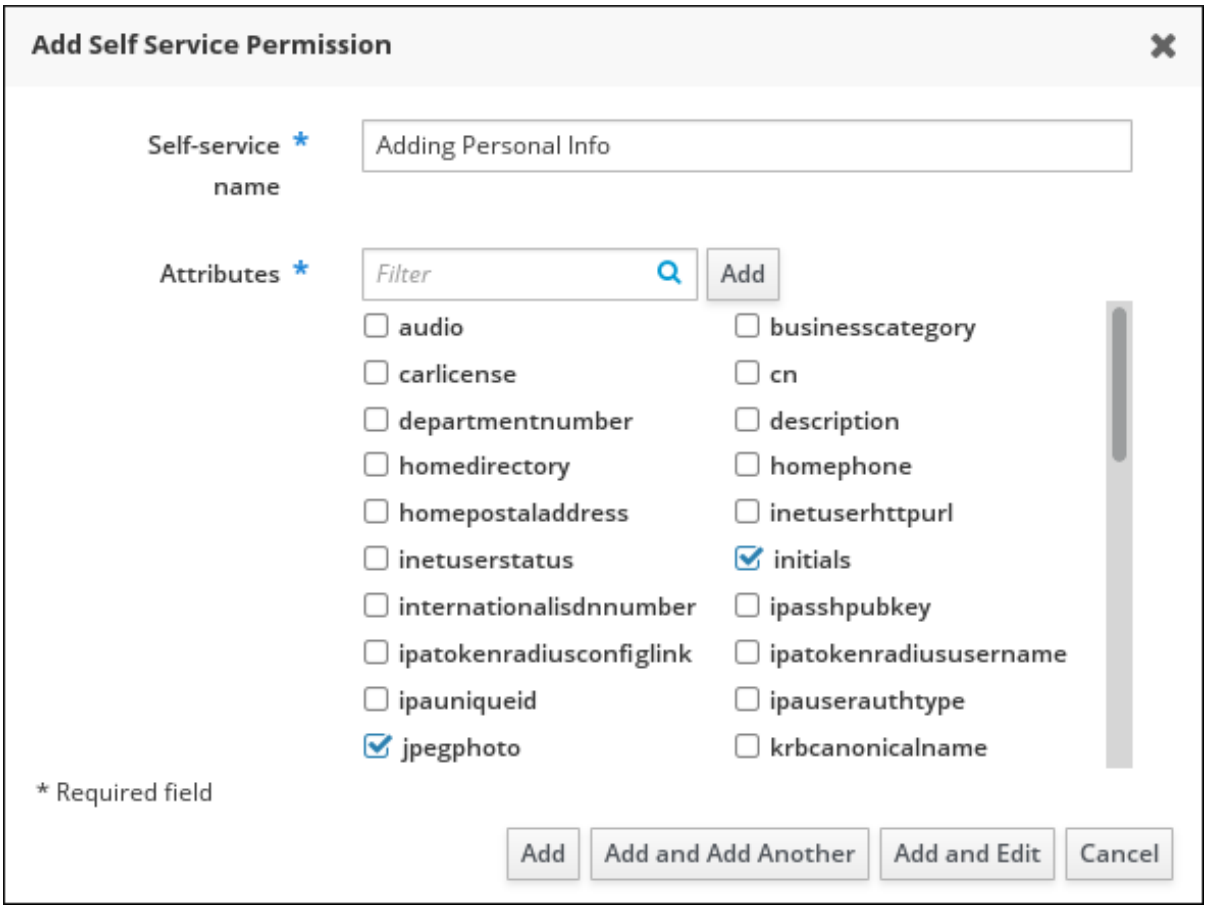
2. セルフサービスのアクセス制御手順のリストの上部にある **Add** をクリックします。

図10.1 現在の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 セルフサービスルールを追加するためのフォーム



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

selfservice-add コマンドを使用すると、新しいセルフサービスルールを追加できます。これらの 2 つのオプションが必要です。

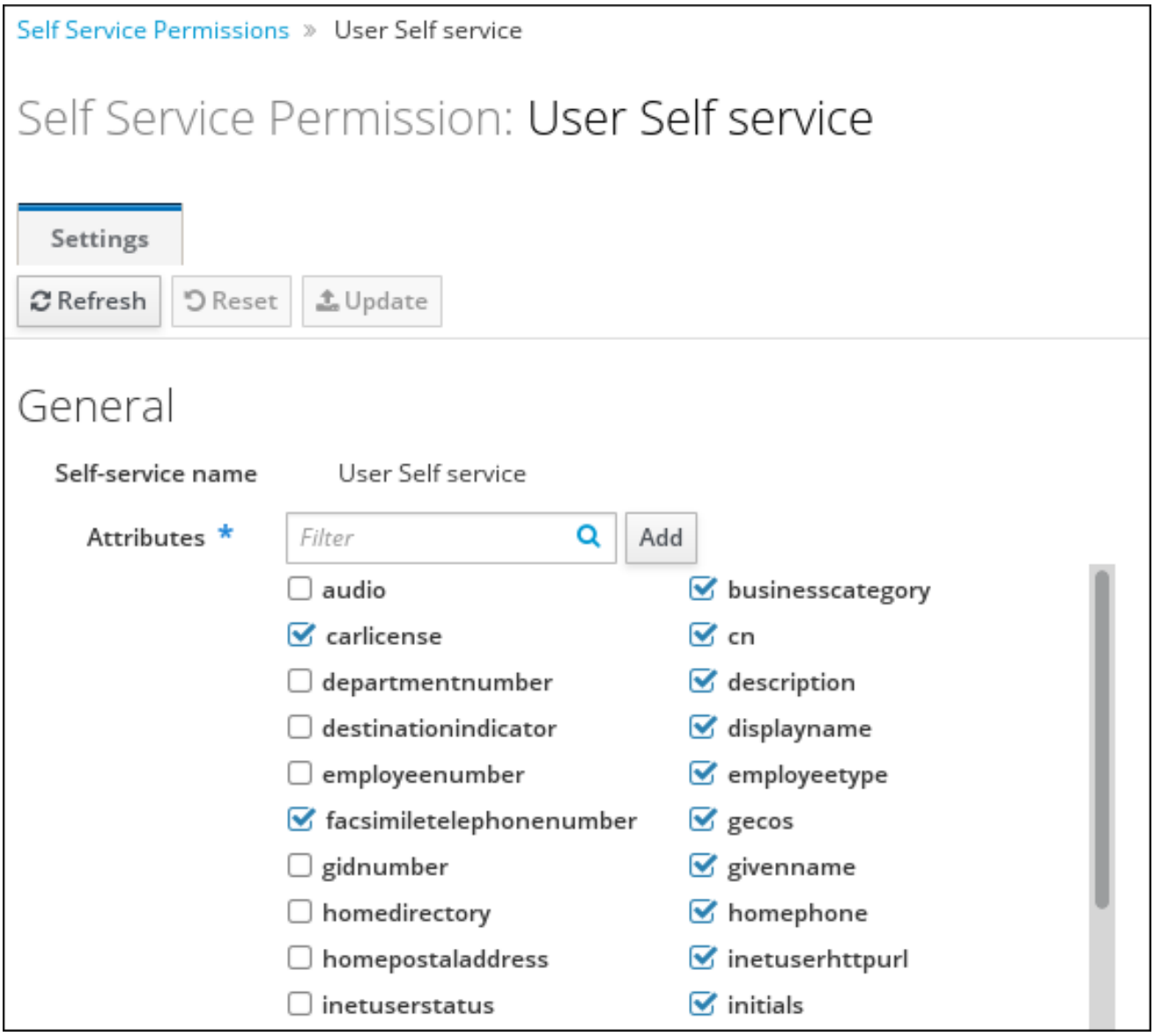
- **--permissions**: ACI 付与への書き込み、追加、または削除などのパーミッションを設定します。
- **--attrs**: この ACI がパーミッションを付与する属性の完全なリストを設定します。

```
[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リストのみが編集可能な要素です。チェックボックスは選択または選択解除できます。

図10.3 セルフサービス編集ページ



コマンドラインで、セルフサービスのルールが **ipa selfservice-mod** コマンドを使用して編集されます。**--attrs** オプションは、サポートされる属性のリストをすべて上書きするため、属性の完全なリストと新しい属性を常に含めます。

```
[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

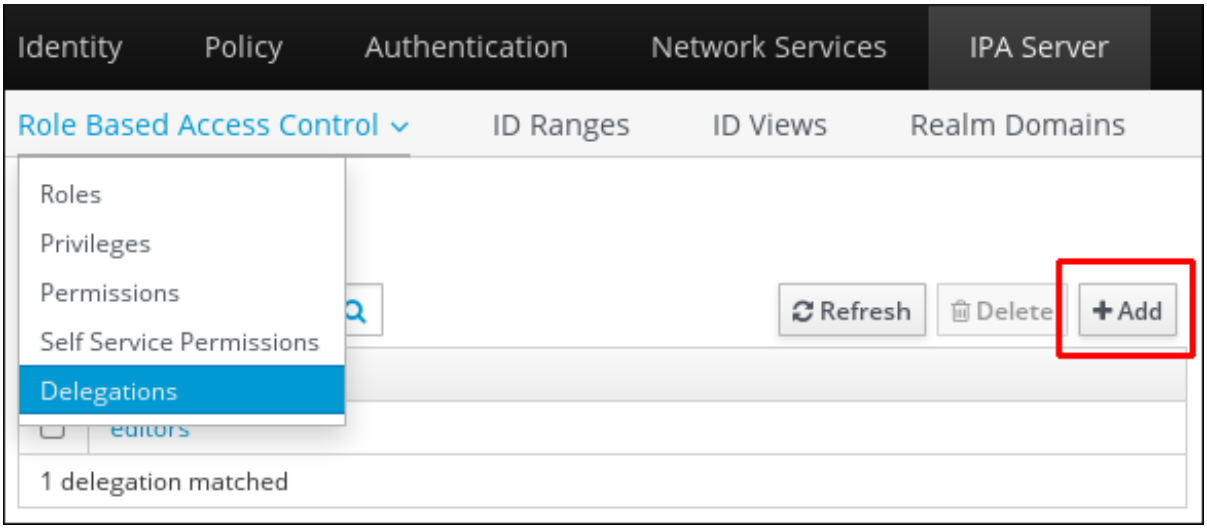
ユーザーのあるグループが別のユーザーのグループのエントリーを管理するパーミッションを割り当てられるという意味で、委任はロールにとてもよく似ています。ただし、付与される完全なアクセスがエントリー全体に対してではなく、特定のユーザー属性のみに対してであるという意味で、委任される権限はセルフサービスルールにより似ています。また、委任された権限内のグループは、アクセス制御のために特別に作成されたロールではなく、既存の IdM ユーザーグループになります。

10.3.1. Web UI でのユーザーグループへのアクセス委任

1. トップメニューの **IPA Server** タブで、**Role-Based Access Control → Delegations** サブタブを選択します。

2. 委譲アクセス制御手順のリストの上部にある **Add** リンクをクリックします。

   図10.4 新規委譲の追加



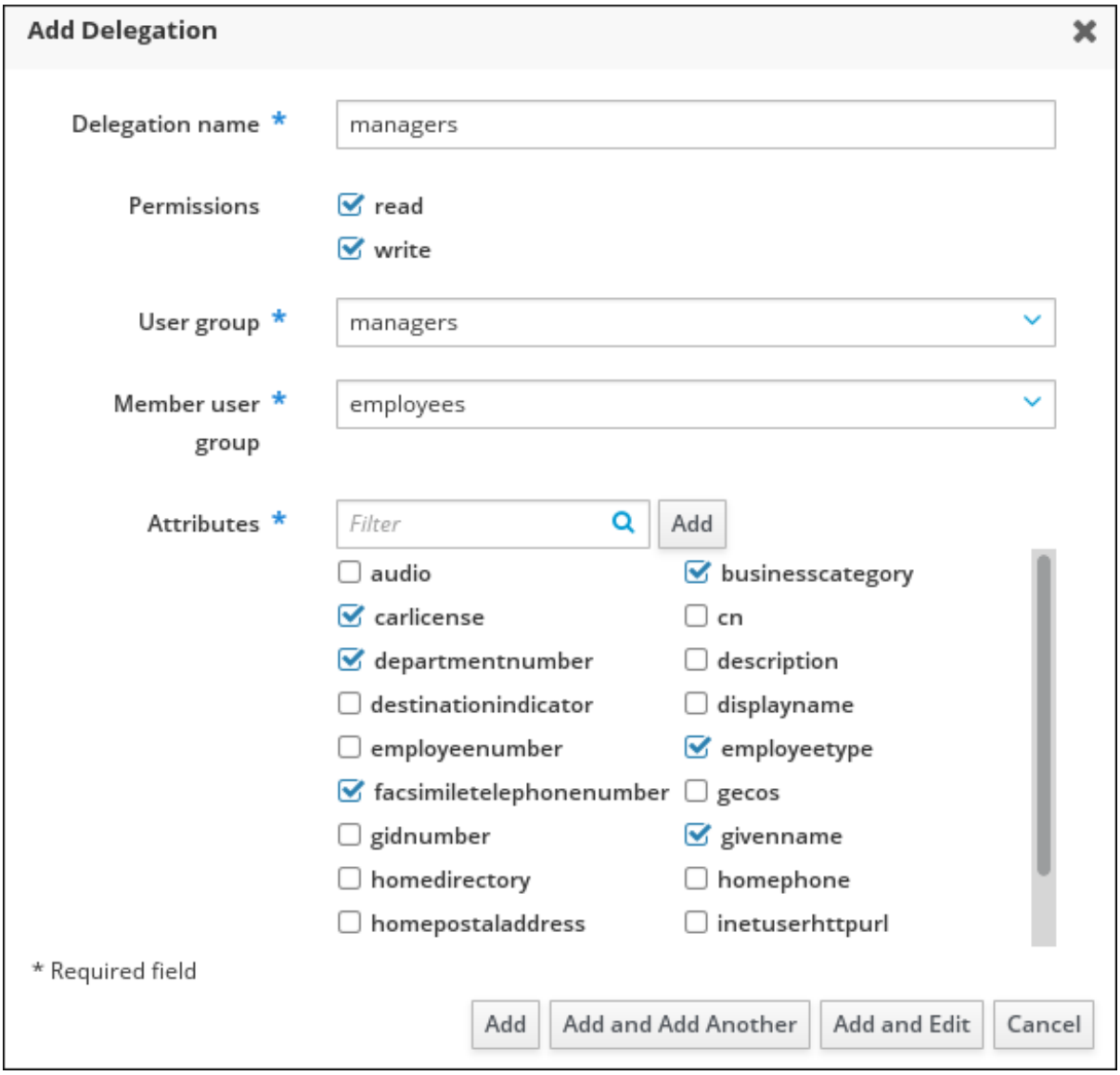
3. 新規委任に名前を付けます。

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。

図10.5 委譲を追加するためのフォーム



6. **Member user group** ドロップダウンメニューで、委譲グループのメンバーが **エントリーを編集できる** 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

delegation-add コマンドを使用して、新しい委譲アクセス制御ルールが追加されます。以下の 3 つのオプションが必須になります。

- **--group** - ユーザーグループ内のユーザーのエントリーに **パーミッションを付与されている** グループです。
- **--membergroup** - 委任グループのメンバーが **エントリーを編集できる** グループです。
- **--attrs** - メンバーグループのユーザーが表示または編集できる属性です。

以下に例を示します。

```
$ ipa delegation-add "basic manager attrs" --attrs=manager --attrs=title --attrs=employeetype --
attrs=employeenumber --group=engineering_managers --membergroup=engineering
-------------------------------------
Added delegation "basic manager attrs"
-------------------------------------
  Delegation name: basic manager attrs
  Permissions: write
  Attributes: manager, title, employeetype, employeenumber
  Member user group: engineering
  User group: engineering_managers
```

委任ルールは、**delegation-mod** コマンドを使用して編集します。**--attrs** オプションは、サポートされる属性のリストをすべて上書きするため、属性の完全なリストと新しい属性を常に含めます。

```
[jsmith@server ~]$ ipa delegation-mod "basic manager attrs" --attrs=manager --attrs=title --
attrs=employeetype --attrs=employeenumber --attrs=displayname
-----------------------------------------
Modified delegation "basic manager attrs"
-----------------------------------------
  Delegation name: basic manager attrs
  Permissions: write
  Attributes: manager, title, employeetype, employeenumber, 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管理されており、エントリーを変更する機能を提供します。

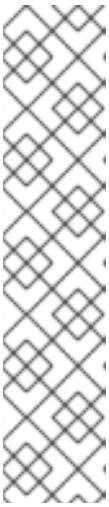
ロールベースのアクセス制御には、**パーミッション**、**特権**、および **ロール** の 3 つの部分があります。権限は 1 つ以上のパーミッションで設定され、ロールは 1 つ以上の権限で設定されます。

- *パーミッション*は、(読み取り、書き込み、追加、または削除) 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特権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、さらにこのようなオブジェクトの属性にアクセスできるかどうかを制御できます。IdM を使用すると、個々の属性をホワイトリストまたはブラックリストに登録したり、特定の IdM 機能 (ユーザー、グループ、sudo など) の全体の可視性を、すべての匿名ユーザー、すべての認証済みユーザー、または特権ユーザーの特定のグループに変更したりできます。パーミッションに対するこの柔軟なアプローチは、管理者がアクセスが必要な特定のセクションにのみユーザーまたはグループのアクセスを制限し、他のセクションをこれらのユーザーまたはグループに対して完全に非表示にする場合に便利です。

- *特権* は、ロールに適用できるパーミッションのグループです。例えば、パーミッションは自動マウントの場所の追加、編集、削除を行うために作成できます。そして、そのパーミッションは FTP サーバーの管理に関連する別のパーミッションと組み合わせることができます。これら

は、ファイルシステムの管理に関連する単一の特権を作成するために作成できます。

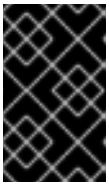


注記

Red Hat Identity Management のコンテキストでは、権限とは、パーミッションおよびそれに続いてロールが作成されるアクセス制御の Atomic 単位の非常に特殊な意味を持ちます。通常のユーザーが一時的に追加の特権を取得するという概念としての**特権昇格**は、Red Hat Identity Management には存在しません。ロールベースアクセス制御 (RBAC) を使用して、権限がユーザーに割り当てられます。アクセス権を付与するロールを持つユーザーと、持たないユーザーがいます。

ユーザーのほかに、権限はユーザーグループ、ホスト、ホストグループ、およびネットワークサービスにも割り当てられます。これにより、特定のネットワークサービスを使用するホストセットのユーザーセットによって、操作をきめ細かく制御できます。

- *ロール*は、ロールに指定したユーザーの権限のリストです。



重要

ロールは、許可されたアクションを分類するために使用されます。ロールは、特権昇格されないようにしたり、特権の分離を実装するツールとしては使用しません。

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。Red Hat Identity Management は、以下の事前定義済みのロールを提供します。

表10.1 Red Hat Identity Management の定義済みロール

ロール	特権	説明
Helpdesk	Modify Users、Reset passwords、Modify Group membership	簡単なユーザー管理タスクを実行します。
IT Security Specialist	Netgroups Administrators、HBAC Administrator、Sudo Administrator	ホストベースのアクセス制御、sudo ルールなどのセキュリティーポリシーを管理します。
IT Specialist	Host Administrators、Host Group Administrators、Service Administrators、Automount Administrators	ホストの管理を行います
Security Architect	Delegation Administrator、Replication Administrators、Write IPA Configuration、Password Policy Administrator	Identity Management 環境の管理、信頼の作成、レプリカ合意を作成します。

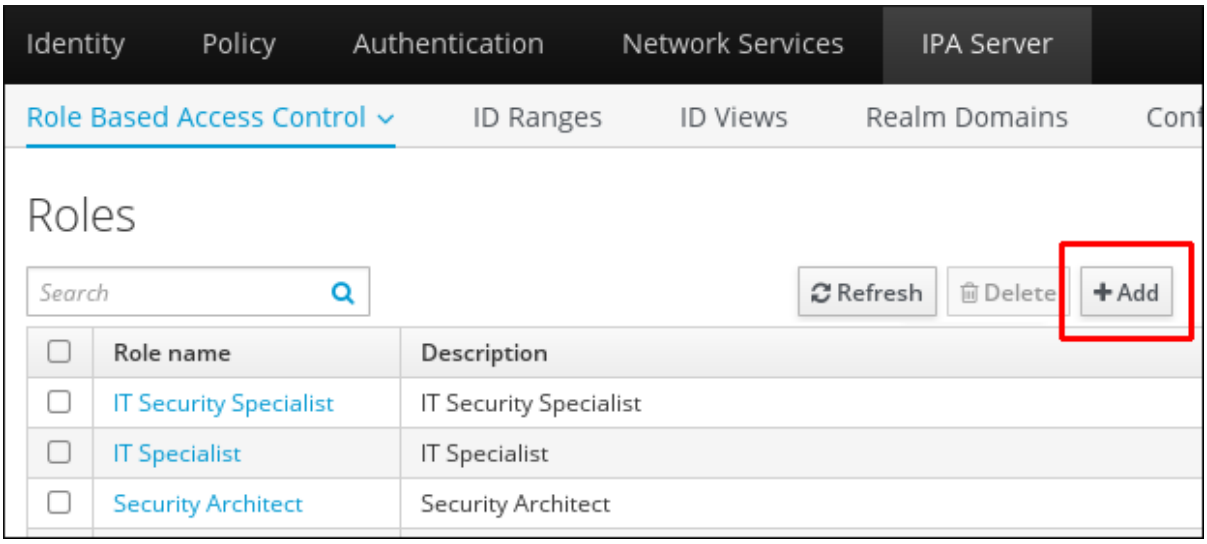
ロール	特権	説明
User Administrator	User Administrators、Group Administrators、Stage User Administrators	ユーザーおよびグループの作成を行います

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. ロールベースのアクセス制御手順のリストの上部にある **Add** リンクをクリックします。

図10.6 新規ロールの追加

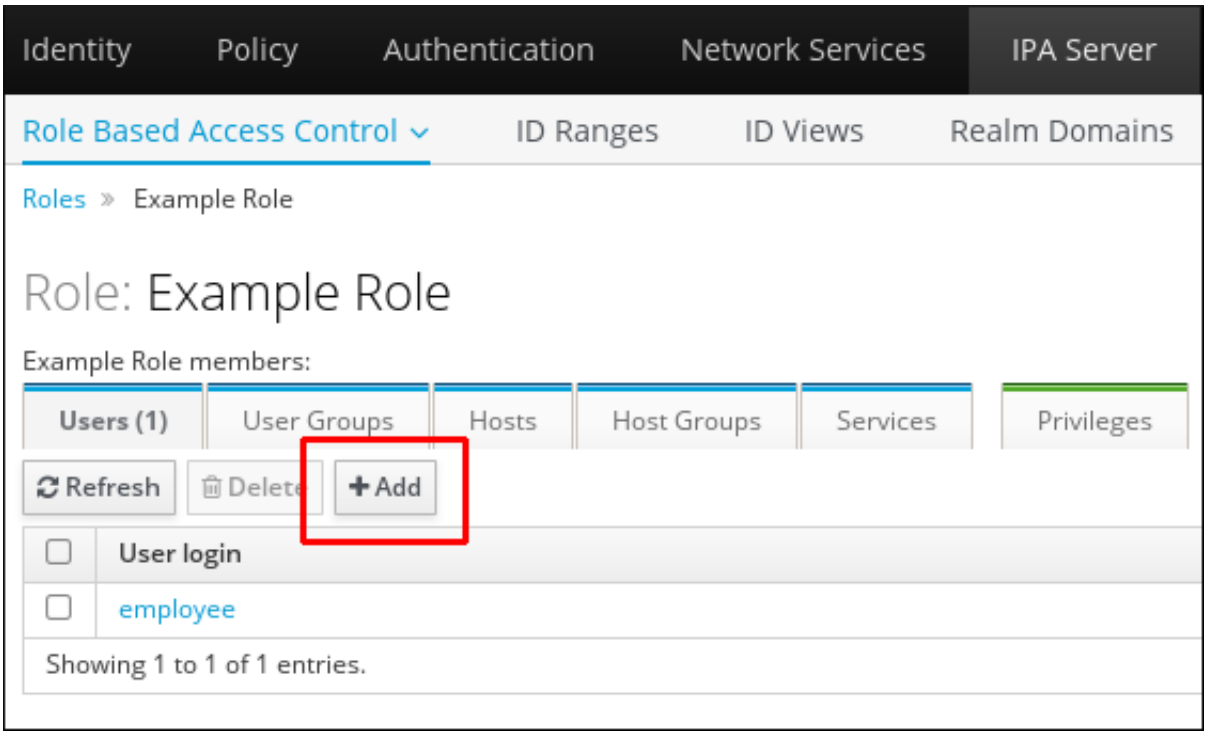


3. ロール名と説明を入力します。

図10.7 ロールを追加するためのフォーム

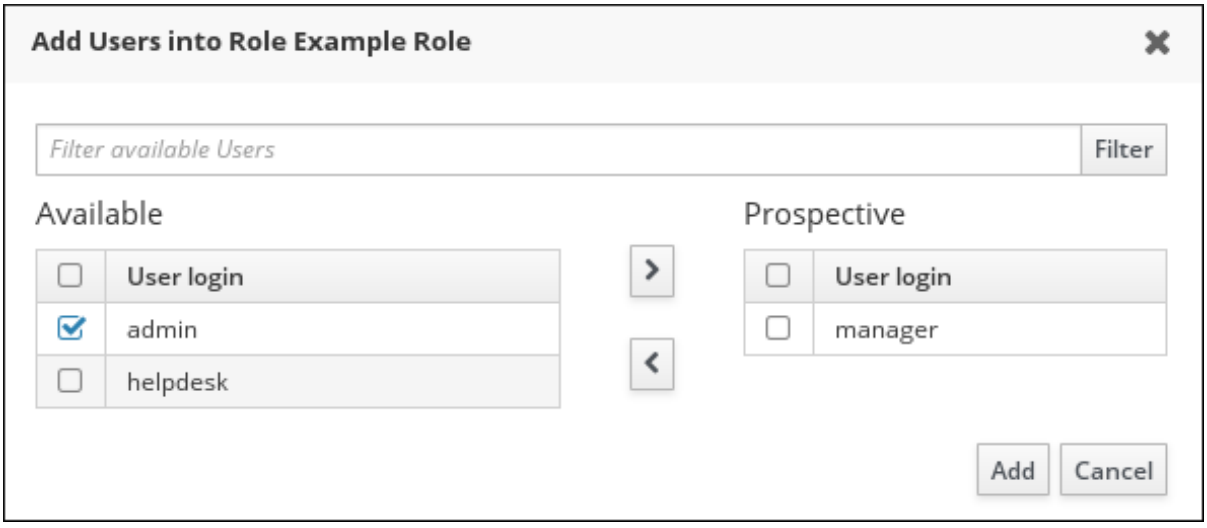
4. **Add and Edit** ボタンをクリックして新規ロールを保存し、設定ページに移動します。
5. **Users** タブの上部で、グループ追加の場合は **Users Groups** タブで、**Add** をクリックします。

図10.8 ユーザーの追加



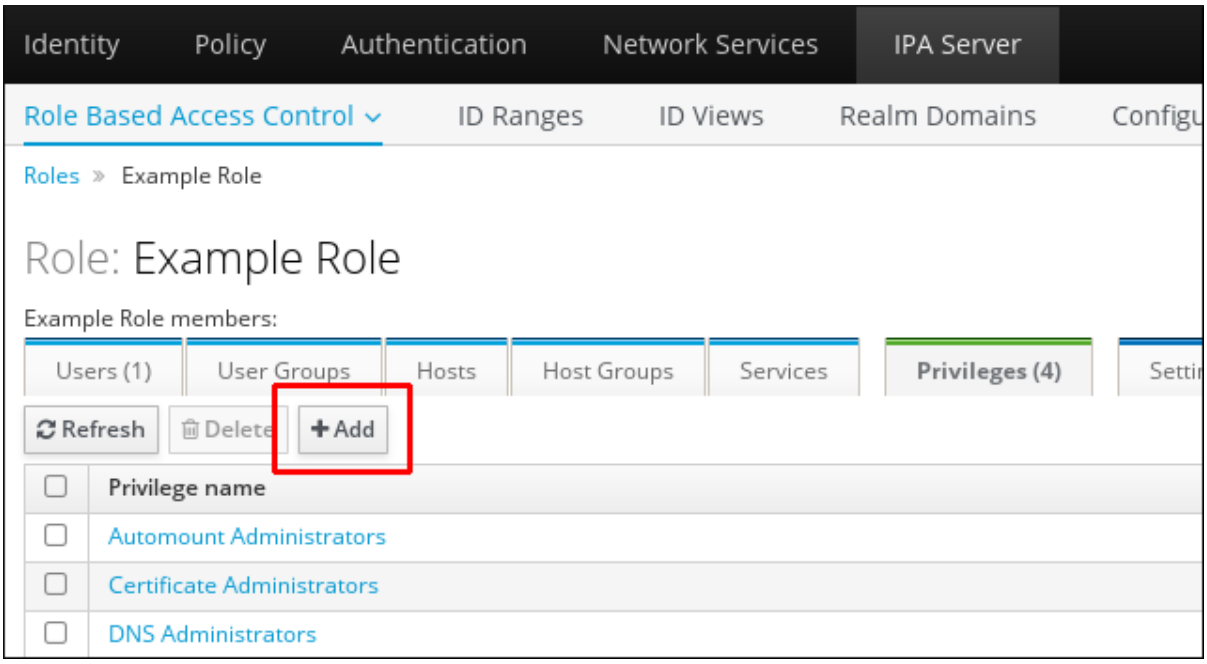
6. 左側のユーザーを選択し、> ボタンを使用して、**Prospective** 列に移動します。

図10.9 ユーザーの選択



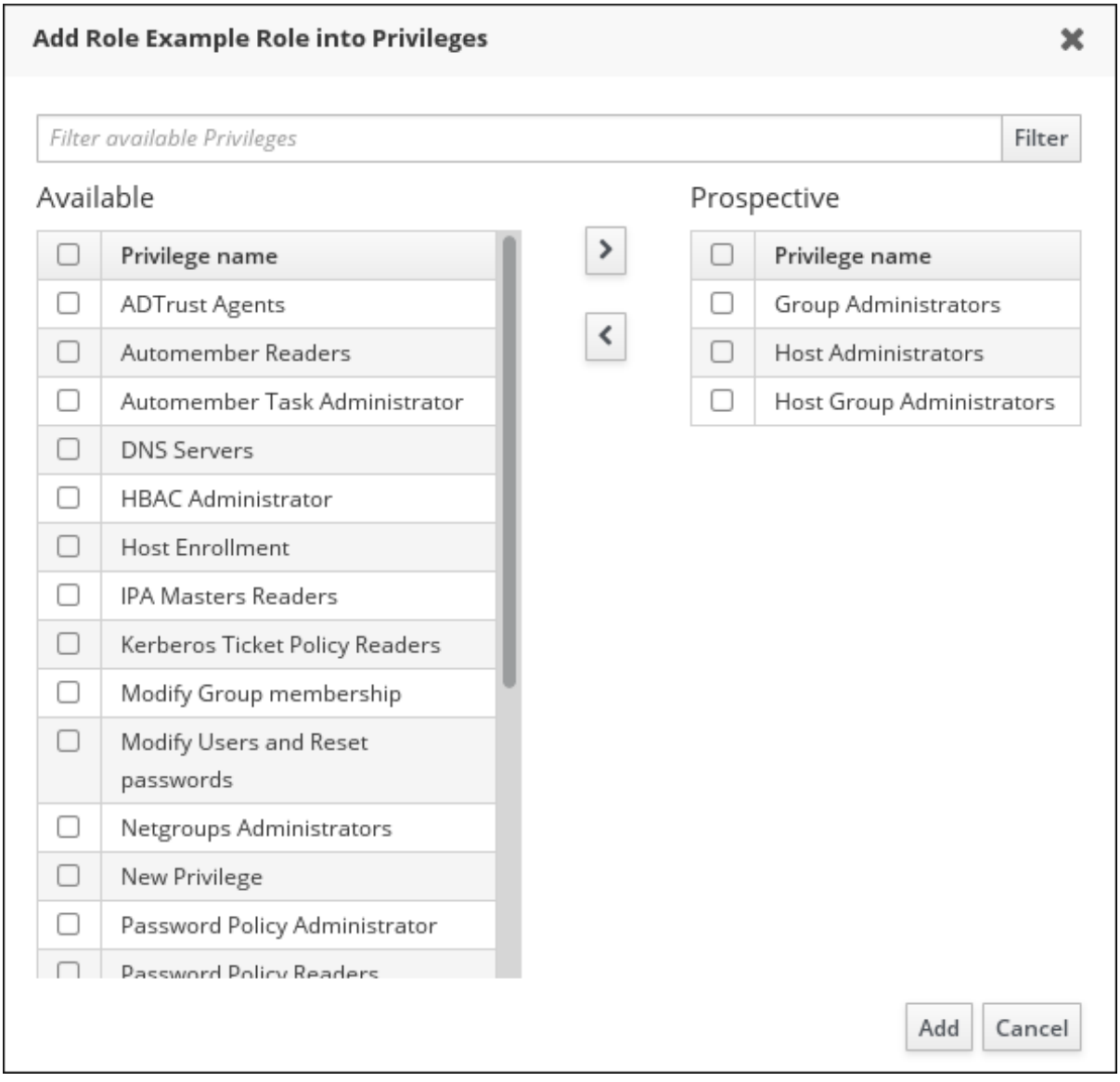
7. **Privileges** タブの上部にある **Add** をクリックします。

図10.10 権限の追加



8. 左側の権限を選択し、> ボタンを使用してそれらを **Prospective** 列に移動します。

図10.11 権限の選択



9. **Add** ボタン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. 必要なグループをロールに追加します。この場合、すでに存在する 1 つのグループ **useradmin** のみを追加することになります。

```
[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
-------------------------
```

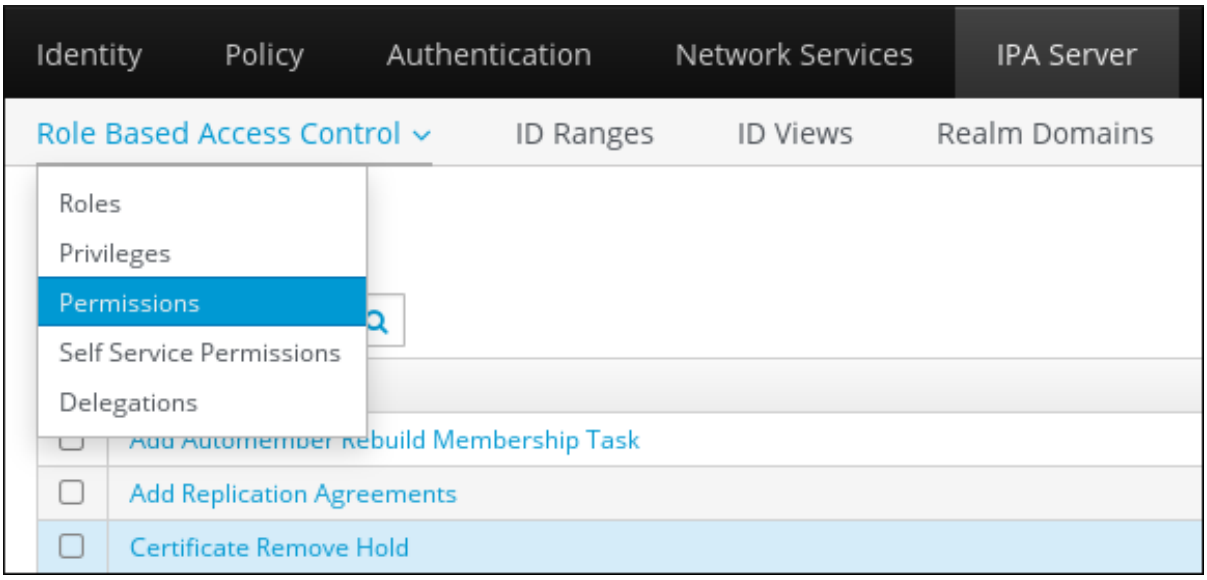
10.4.2. パーミッション

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。

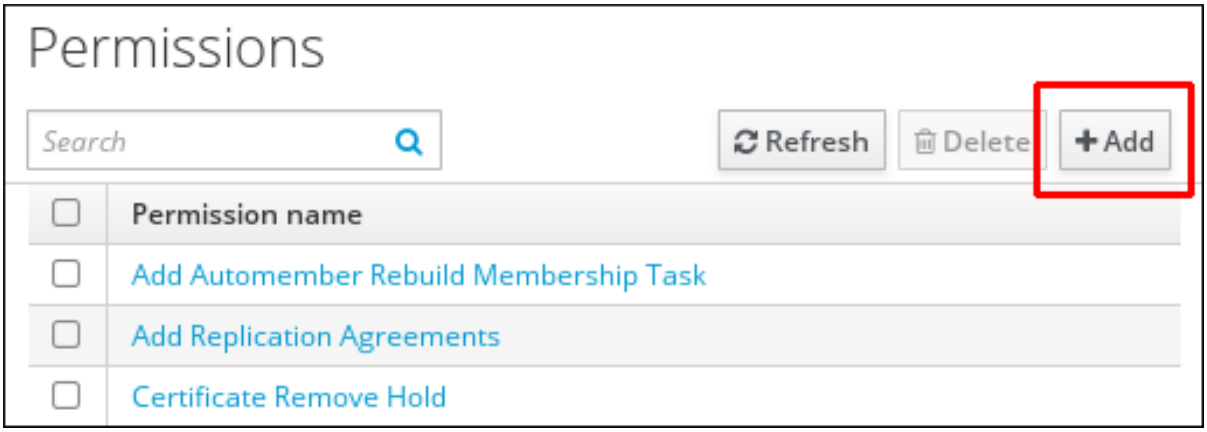
2. **Permissions** タスクリンクを選択します。

図10.12 パーミッションタスク



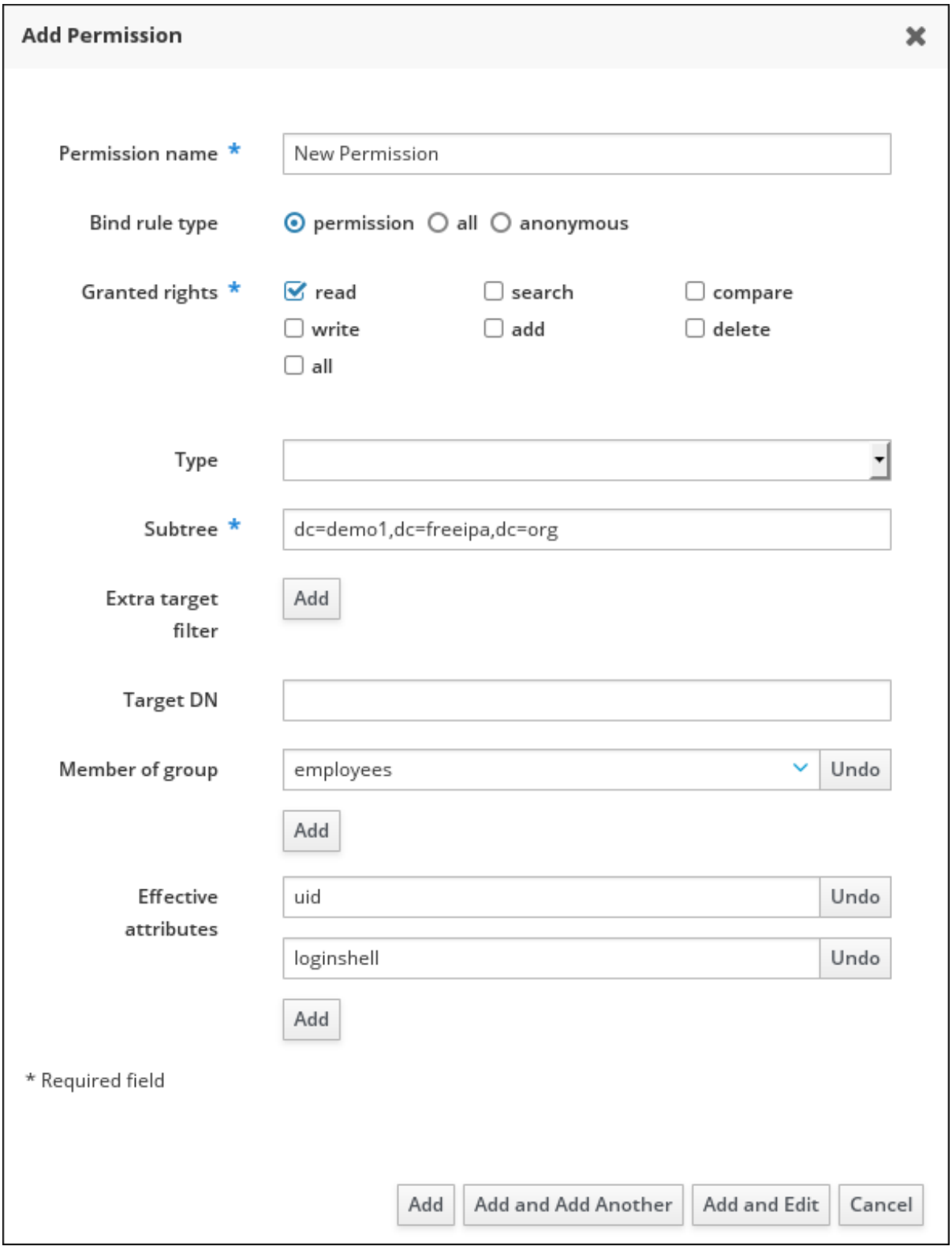
3. パーミッションのリストの上部にある **Add** ボタンをクリックします。

図10.13 新規パーミッションの追加



4. 表示される形式で、新しいパーミッションのプロパティーを定義します。

図10.14 パーミッションを追加するためのフォーム



5. フォームの下にある **Add** ボタンをクリックして、パーミッションを保存します。

以下のパーミッションプロパティーを指定できます。

1. 新規パーミッションの名前を入力します。
2. 適切な**バインドルールタイプ**を選択します。
   - **permission** はデフォルトのパーミッションタイプで、権限およびロール経由でアクセスを付与します。

- **all** は、パーミッションをすべての認証ユーザーに適用することを指定します。
- **anonymous** は、認証されていないユーザーを含む、パーミッションがすべてのユーザーに適用されることを指定します。

注記

特権には、デフォルト以外のバインドルールタイプが指定されたパーミッションを追加できません。特権に既存のパーミッションは、デフォルト以外のバインドルールタイプには設定できません。

3. **Granted rights**で、パーミッションを付与する権利を選択してください。
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。
   - **タイプ** 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** 設定の値を選択すると、そのエントリータイプのこの ACI からアクセス可能なすべての属性のリストが **Effective 属性** に表示されます。

     Type を定義すると、**Subtree** および **Target DN** が事前定義された値のいずれかに設定されます。
   - **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の下にある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) を許可しないため、既存のサブツリーエントリーを指定します。以下に例を示します。

     ```
     cn=automount,dc=example,dc=com
     ```
   - **追加のターゲットフィルター** は LDAP フィルターを使用して、パーミッションが適用されるエントリーを特定します。このフィルターには、有効な LDAP フィルターを使用できます。以下に例を示します。

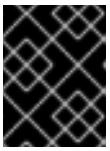
     ```
     (!(objectclass=posixgroup))
     ```

     IdM は、指定のフィルターの有効性を自動的に確認します。無効なフィルターを入力すると、パーミッションを保存しようとすると、IdM はこれについて警告します。
   - **ターゲット DN** はドメイン名 (DN) を指定し、ワイルドカードを受け入れます。以下に例を示します。

     ```
     uid=*,cn=users,cn=accounts,dc=com
     ```
   - **グループのメンバー** は、指定した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。

   フィルター設定を入力し、**Add** をクリックすると、IdM がフィルターを検証します。すべてのパーミッション設定が正しい場合は、IdM により検索が実行されます。パーミッション設定の一部が正しくない場合には、IdM により、どの設定が正しく設定されているかを示すメッセージが表示されます。
5. **Type** を設定する場合は、利用可能な ACI 属性のリストから **有効な属性** を選択します。**Type** を使用しない場合は、**有効な属性** フィールドに属性を手動で書き込みます。一度に 1 つの属性を追加します。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と、デフォルトですべての属性が含まれます。

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

新しいパーミッションを追加するには、**ipa permission-add** コマンドを実行します。対応するオプションを指定して、パーミッションのプロパティーを指定します。

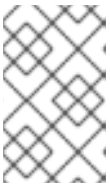
- パーミッションの名前を指定します。以下に例を示します。

```
[root@server ~]# ipa permission-add "dns admin permission"
```

- **--bindtype** は、バインドルールの種別を指定します。このオプションは、**all** 引数、**anonymous** 引数、および **permission** 引数を受け入れます。以下に例を示します。

```
--bindtype=all
```

--bindtype を使用しない場合、タイプは自動的にデフォルトの **permission** 値に設定されます。



注記

特権には、デフォルト以外のバインドルールタイプが指定されたパーミッションを追加できません。特権に既存のパーミッションは、デフォルト以外のバインドルールタイプには設定できません。

- **--permissions** は、パーミッションが付与する権限をリスト表示します。複数の属性を設定するには、複数の **--permissions** オプションを使用するか、オプションを中括弧内にコンマ区切りリストでリスト表示します。以下に例を示します。

```
--permissions=read --permissions=write
--permissions={read,write}
```

- **--attrs** は、パーミッションが付与される属性のリストを提供します。複数の属性を設定するには、複数の **--attrs** オプションを使用するか、オプションを中括弧内にコンマ区切りリストでリスト表示します。以下に例を示します。

```
--attrs=description --attrs=automountKey
--attrs={description,automountKey}
```

--attrs で提供される属性が存在し、指定のオブジェクトタイプに許可される属性である必要があります。指定しないと、コマンドがスキーマ構文エラーで失敗します。

- **--type** は、ユーザー、ホスト、またはサービスなどのエントリーオブジェクトタイプを定義します。各タイプには、許可された属性の独自のセットがあります。以下に例を示します。

```
[root@server ~]# ipa permission-add "manage service" --permissions=all --type=service --
attrs=krbprincipalkey --attrs=krbprincipalname --attrs=managedby
```

- **--subtree** はサブツリーエントリーを提供します。フィルターはこのサブツリーエントリーの下にあるすべてのエントリーをターゲットにします。既存のサブツリーエントリーを指定します。**--subtree** はワイルドカードや存在しないドメイン名 (DN) を受け入れません。ディレクトリーに DN を追加します。

  IdM は簡素化されたフラットディレクトリーツリー構造を使用しているため、**--subtree** を使用して自動マウントの場所 (他の設定のコンテナーまたは親エントリー) などの一部のエントリーをターゲットにすることができます。以下に例を示します。

  ```
  [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 オプションおよび **--subtree** オプションは相互に排他的です。

- **--filter** は LDAP フィルターを使用して、パーミッションが適用されるエントリーを特定します。IdM は、指定のフィルターの有効性を自動的に確認します。このフィルターには、有効な LDAP フィルターを使用できます。以下に例を示します。

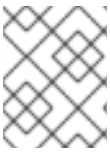
  ```
  [root@server ~]# ipa permission-add "manage Windows groups" --filter="(!
  (objectclass=posixgroup))" --permissions=write --attrs=description
  ```

- **--memberof** は、グループが存在することを確認した後に、指定した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。以下に例を示します。

  ```
  [root@server ~]# ipa permission-add ManageHost --permissions="write" --
  subtree=cn=computers,cn=accounts,dc=testrelm,dc=com --attr=nshostlocation --
  memberof=admins
  ```

- **--targetgroup** は、グループが存在することを確認した後に、ターゲットを指定されたユーザーグループに設定します。

Web UI で利用可能な **Target DN** 設定はコマンドラインで利用できません。



注記

パーミッションの変更および削除の詳細は、**ipa permission-mod --help** コマンドおよび **ipa permission-del --help** コマンドを実行します。

10.4.2.3. デフォルトの管理パーミッション

管理 パーミッションは、Identity Management で事前にインストールしたパーミッションです。このパーミッションはユーザーが作成した他のパーミッションと同様に機能しますが、以下の相違点があります。

- 名前、場所、およびターゲット属性を変更することはできません。
- これらは削除できません。
- このパーミッションには 3 つの属性セットがあります。
  - *デフォルト* 属性 (IdM により管理され、ユーザーが変更できない)

- *含まれる* 属性 (ユーザーが追加する属性): 管理パーミッションに含まれる属性を追加するには、**ipa permission-mod** コマンドで **--includedattrs** オプションを指定して属性を指定します。
- *除外する* 属性 (ユーザーが削除する属性): 管理パーミッションに除外する属性を追加するには、**ipa permission-mod** コマンドで **--excludedattrs** オプションを指定して属性を指定します。

管理パーミッションは、デフォルトおよび包含属性セットに表示されている属性すべてに適用されますが、除外セットに表示されている属性には適用されません。

管理パーミッションの変更時に **--attrs** オプションを使用する場合は、含まれる属性および除外する属性セットが自動的に調整され、**--attrs** で指定された属性のみが有効になります。

注記

管理されているパーミッションを削除することはできませんが、そのバインドタイプを **permission** に設定し、すべての権限から管理パーミッションを削除すると、そのパーミッションを効果的に無効にできます。

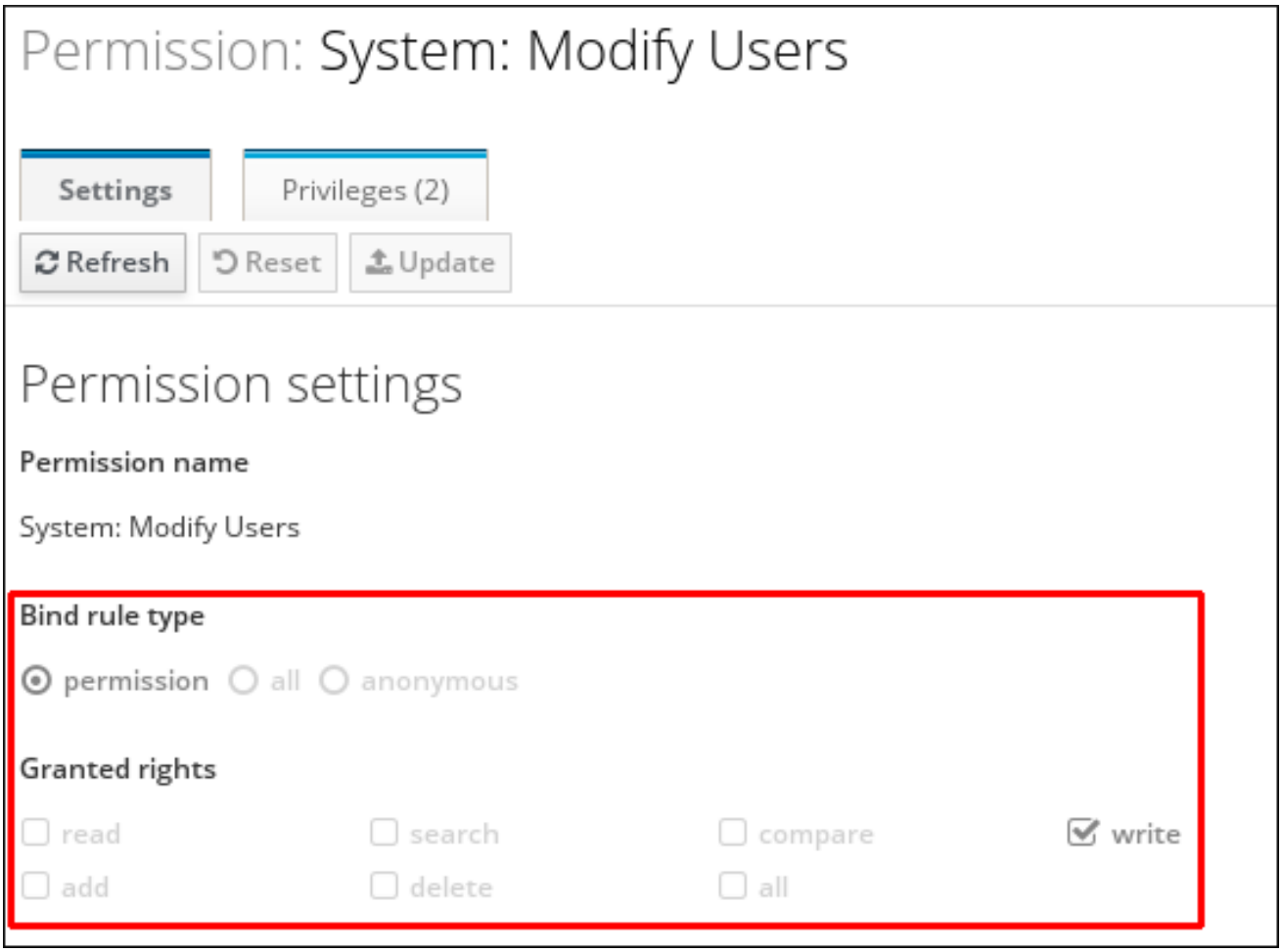
管理されたすべてのパーミッションの名前は **System:** から始まります (例: *System: Add Sudo rule* または *System: Modify Services*)。

以前のバージョンの IdM 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いました。たとえば、ユーザーがデフォルトのパーミッションを変更するのを禁止し、ユーザーはパーミションを権限に割り当てることしかできませんでした。これらのデフォルトパーミッションのほとんどは、管理パーミッションに切り替わっていますが、以下のパーミッションは引き続き以前のスキームを使用します。

- Automember Rebuild メンバーシップタスクの追加
- レプリカ合意の追加
- 証明書削除保留
- CA から証明書のステータス取得
- DNA 範囲の変更
- レプリカ合意の修正
- レプリカ合意の削除
- 証明書の要求
- 別のホストからの証明書の要求
- CA からの証明書の取得
- 証明書の取り消し
- IPA 設定の書き込み

Web UI から管理パーミッションを変更しようとすると、変更できない属性が無効になります。

図10.15 無効にされた属性



コマンドラインから管理パーミッションを変更しようとした場合、システムは変更できない属性を変更するのを許可しません。たとえば、デフォルトの**System: Modify Users**パーミッションを変更してグループに適用しようとしても失敗します。

```
$ ipa permission-mod 'System: Modify Users' --type=group
ipa: ERROR: invalid 'ipapermlocation': not modifiable on managed permissions
```

ただし、**System: Modify Users** パーミッションが **GECOS** 属性に適用されないようにすることはできます。

```
$ ipa permission-mod 'System: Modify Users' --excludedattrs=gecos
-----------------------------------------
Modified permission "System: Modify Users"
```

10.4.2.4. 以前のバージョンの Identity Management におけるパーミッション

Identity Management の以前のバージョンでは、パーミッションの処理方法が異なりました。以下に例を示します。

- グローバル IdM ACI は、匿名のユーザー (つまり認証されないユーザー) であっても、サーバーのすべてのユーザーに読み取りアクセスを付与しました。
- 書き込み、追加、および削除のパーミッションタイプのみが利用可能でした。読み取りパーミッションも利用できましたが、認証されていないユーザーを含むすべてのユーザーにはデフォルトで読み取りアクセスがあるため、実用的値はほとんどありませんでした。

Identity Management の現在のバージョンには、パーミッションを設定するオプションが含まれており、これはより粒度の細かいものになります。

- グローバル IdM ACI は、認証されていないユーザーに読み取りアクセスを付与しません。
- たとえば、フィルターとサブツリーの両方を同じパーミッションに追加できるようになりました。
- 検索および比較権限を追加できます。

パーミッションを処理する新しい方法では、以前のバージョンとの後方互換性を維持しながら、ユーザーまたはグループのアクセス制御に関して IdM 機能が大幅に改善されました。以前のバージョンの IdM からアップグレードすると、すべてのサーバー上のグローバル IdM ACI が削除され、*管理パーミッション* に置き換えられます。

以前の方法で作成されたパーミッションは、変更する際に、現在のスタイルに自動的に変換されます。それらを変更しようとしないと、以前のタイプのパーミッションは変換されません。パーミッションが現在のスタイルを使用したら、以前のスタイルにダウングレードすることはできません。

注記

以前のバージョンの IdM を実行しているサーバーで、引き続きパーミッションを権限に割り当てることはできます。

ipa permission-show コマンドおよび **ipa permission-find** コマンドは、現在のパーミッションと以前のスタイルのパーミッションの両方を認識します。これらの両方のコマンドからの出力は、パーミッションを現在のスタイルで表示しますが、パーミッション自体は変更されません。LDAP に変更をコミットせずに、コマンドはメモリー内にのみデータを出力する前にパーミッションエントリーをアップグレードします。

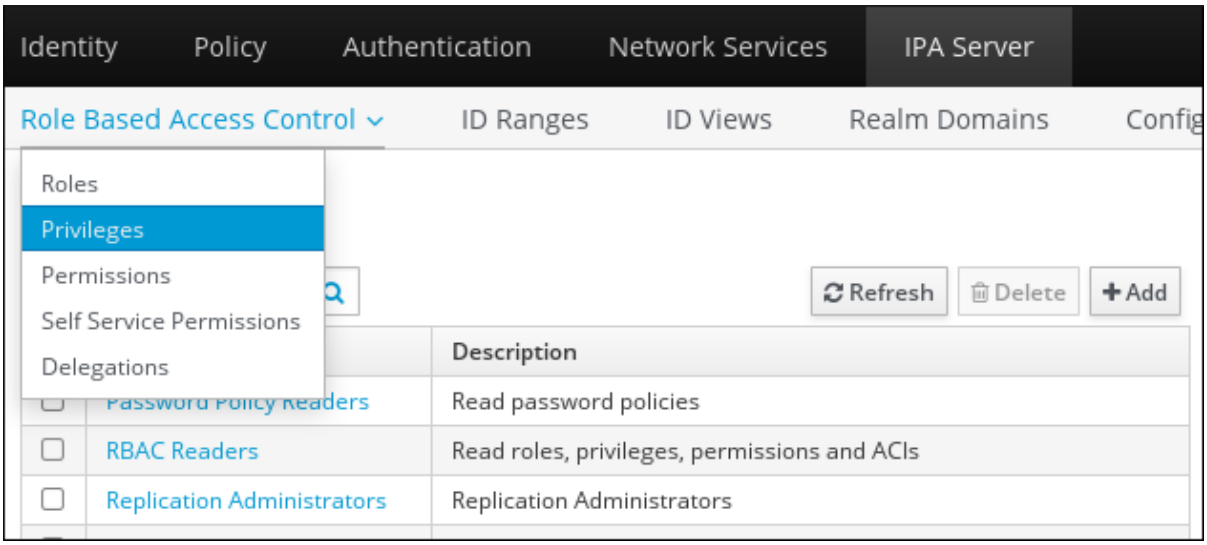
以前の特性を持つパーミッションと現在の特性を持つパーミッションは、どちらもすべてのサーバー (以前のバージョンの IdM を実行するものと、現在の IdM バージョンを実行するもの) に影響します。ただし、以前のバージョンの IdM を実行しているサーバーで、現在のパーミッションでパーミッションを作成または変更することはできません。

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 特権タスク

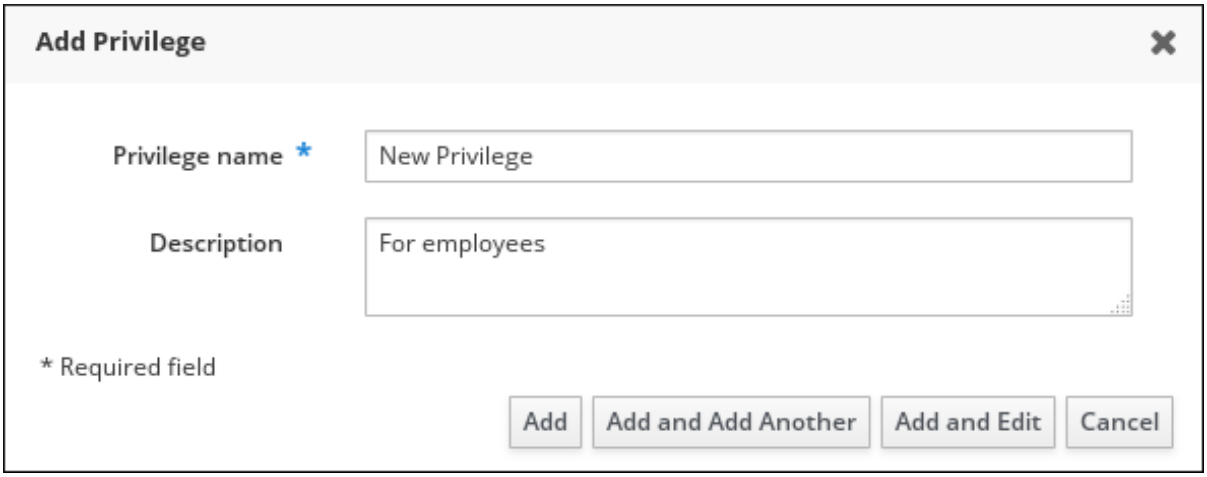


3. 特権リストの上部にある **Add** リンクをクリックします。

図10.17 新しい権限の追加

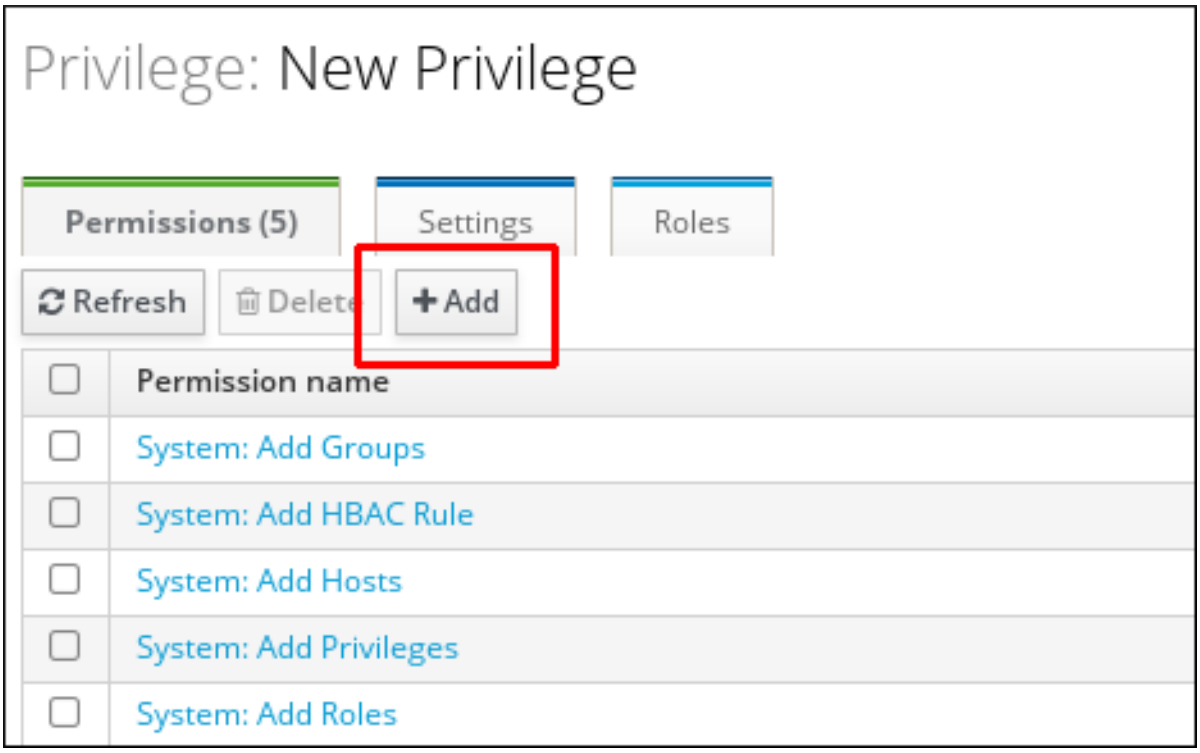
4. 権限の名前と説明を入力します。

図10.18 権限を追加するためのフォーム



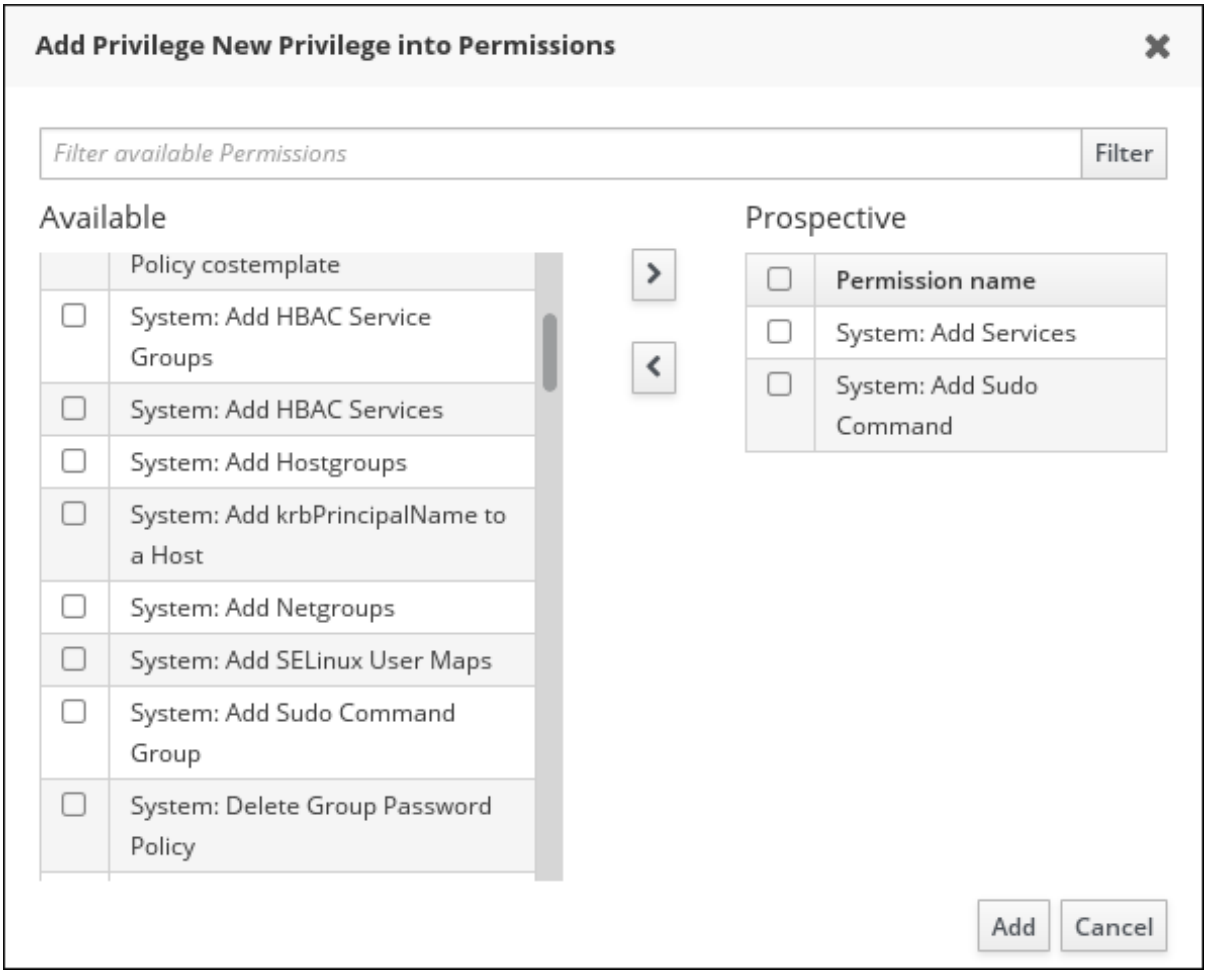
5. **Add and Edit** をクリックして、権限設定ページに移動し、パーミッションを追加します。
6. **Permissions** タブを選択します。
7. パーミッションリストの上部にある **Add** をクリックして、パーミッションを権限に追加します。

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クリックし、> ボタンを使用してパーミッションを **Prospective** 列に移動します。

図10.20 パーミッションの選択



9. **Add** ボタン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

権限エントリーは、**privilege-add** コマンドを使用して作成され、**privilege-add-permission** コマンドを使用して権限グループに追加されます。

1. 権限エントリーを作成します。

```
[jsmith@server ~]$ ipa privilege-add "managing filesystems" --desc="for filesystems"
```

2. 必要なパーミッションを割り当てます。以下に例を示します。

```
[jsmith@server ~]$ ipa privilege-add-permission "managing filesystems" --
permissions="managing automount" --permissions="managing ftp services"
```

パート IV. 管理: アイデンティティーの管理

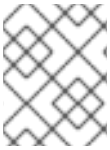
このパートでは、ユーザーアカウント、ホスト、ユーザーグループ、およびホストグループを管理する方法を説明します。さらに、一意の UID 番号および GID 番号の割り当てと表示方法、ユーザーおよびグループスキーマの機能も詳細に説明します。本章では、サービスの管理と、ホストおよびサービスへのアクセスの委譲を説明します。最後の章では、**ID 管理** ユーザー向けに **アクセス制御** を定義する方法、**Kerberos**フラグおよびプリンシパルエイリアスを管理する方法、**NIS** ドメインおよび **ネットグループ** との統合方法を説明します。

第11章 ユーザーアカウントの管理

本章では、ユーザーアカウントの一般的な管理および設定を説明します。

11.1. ユーザーホームディレクトリーの設定

すべてのユーザーにホームディレクトリーが設定されていることが推奨されます。ユーザーのホームディレクトリー用のデフォルトの場所は **/home/** ディレクトリーにあります。たとえば、IdM は、*user_login* ログインを持つユーザーに **/home/*user_login*** にホームディレクトリーが設定されていることを想定しています。



注記

ipa config-mod コマンドを使用すると、ユーザーのホームディレクトリーのデフォルトの想定される場所を変更できます。

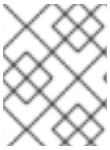
IdM は、ユーザーにホームディレクトリーを自動的に作成しません。ただし、ユーザーのログイン時に自動的にホームディレクトリーを作成する PAM ホームディレクトリーモジュールを設定できます。または、NFS 共有および **automount** ユーティリティーを使用して、ホームディレクトリーを手動で追加できます。

11.1.1. PAM ホームディレクトリーモジュールを使用したホームディレクトリーの自動マウント

サポートされる PAM ホームディレクトリーモジュール
PAM ホームディレクトリーモジュールを設定して、IdM ドメインにログインする際に、ユーザーのホームディレクトリーを自動的に作成するには、以下の PAM モジュールのいずれかを使用します。

- **pam_oddjob_mkhomedir**
- **pam_mkhomedir**

IdM は最初に **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** の使用を試行します。



注記

新規ユーザー用のホームディレクトリーの NFS 共有での自動作成はサポートされていません。

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モジュールを有効にすると、ローカルに影響します。そのため、必要な各クライアントおよびサーバーでモジュールを個別に有効に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モジュールを設定するには、マシンのインストール時に **ipa-server-install** または **ipa-client-install** ユーティリティーで **--mkhomedir** オプションを使用します。

すでにインストールされているサーバーまたはクライアントに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 を使用してホームディレクトリーを作成する方法は、System-Level Authentication Guideを参照してください。

11.1.2. ホームディレクトリーの手動マウント

NFS ファイルサーバーを使用して、IdM ドメインのすべてのマシンで利用可能な **/home/** ディレクトリーを提供してから、**automount** ユーティリティーを使用して IdM マシンにディレクトリーをマウントすることができます。

NFS 使用時の潜在的な問題
NFS を使用すると、パフォーマンスやセキュリティーに悪影響を与える可能性があります。たとえば、NFS を使用することで、NFS ユーザーへの root アクセス付与によるセキュリティーの問題、**/home** ツリー全体を読み込む際のパフォーマンスの問題、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のネットワークパフォーマンスの問題などが発生する可能性があります。

これらの問題の影響を減らすには、以下のガイドラインに従うことを推奨します。

- **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。**/home/** ツリー全体を読み込む時に使用しないでください。
- 限定的なパーミッションを割り当てた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、そのユーザーとして 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ため、**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、NFS サーバーにホームディレクトリーを作成できます。

NFS およびautomountを使用したホームディレクトリーの設定
NFS 共有と **automount** を使用して、ホームディレクトリーを別の場所から IdM サーバーに手動で追加するには、以下の手順を実施します。

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. 新しい場所の **auto.direct** ファイルに直接マッピングを追加します。**auto.direct** ファイルは、**ipa-server-install** ユーティリティーが自動作成する **自動マウント** のマッピングです。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 **自動マウント** を使用する方法は、*34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 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。

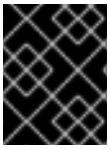
- **ステージ** ユーザーは認証できません。これは初期状態です。アクティブユーザーに必要なユーザーアカウントプロパティーの一部が設定されていない可能性があります。

- **アクティブ** ユーザーは認証が可能です。必要なユーザーアカウントプロパティーはすべて、この状態で設定する必要があります。
- **保存済み**ユーザーは、以前は **アクティブ**なユーザーでした。非アクティブであると見なされ、IdM に対して認証できません。保存済みユーザーには、アクティブユーザーのときに有効になっていたアカウントプロパティーの大部分が保持されていますが、ユーザーグループからは除外されています。

注記

保存済み 状態のユーザーのリストは、以前のユーザーアカウントの履歴を提供します。

ユーザーエントリーは、IdM データベースから完全に削除することもできます。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。ユーザーの外部設定 (システムアカウントやホームディレクトリーなど) は削除されませんが、IdM からは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を復元することはできません。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する情報がすべて永続的に失われます。

新規管理ユーザーは、デフォルトの**admin**ユーザーなど、他の管理者のみが作成できます。すべての管理者アカウントを誤って削除した場合は、Directory Manager が、Directory Server に新しい管理者を手動で作成する必要があります。

警告

admin ユーザーを削除しないでください。**admin** は IdM で必要な事前定義ユーザーであるため、この操作では特定のコマンドで問題が生じます。代替の **admin** ユーザーを定義して使用する場合は、管理者パーミッションを少なくとも 1 人のユーザーに付与した後に **ipa user-disable admin** で事前定義された admin ユーザーを無効にします。

ユーザーライフサイクル管理操作

ユーザーのプロビジョニングを管理するために、管理者はユーザーアカウントをある状態から別の状態に移行することができます。新規ユーザーアカウントは、**active** または **stage** のいずれかとして追加できますが、**preserved** としては追加できません。

IdM は、ユーザーのライフサイクル管理に対する以下の操作をサポートします。

stage → active

stage 状態のアカウントを適切にアクティブ化する準備ができると、管理者は **active** 状態に移行します。

active → preserved

ユーザーが会社を離れると、管理者はアカウントを **preserved** 状態に移行します。

preserved → active

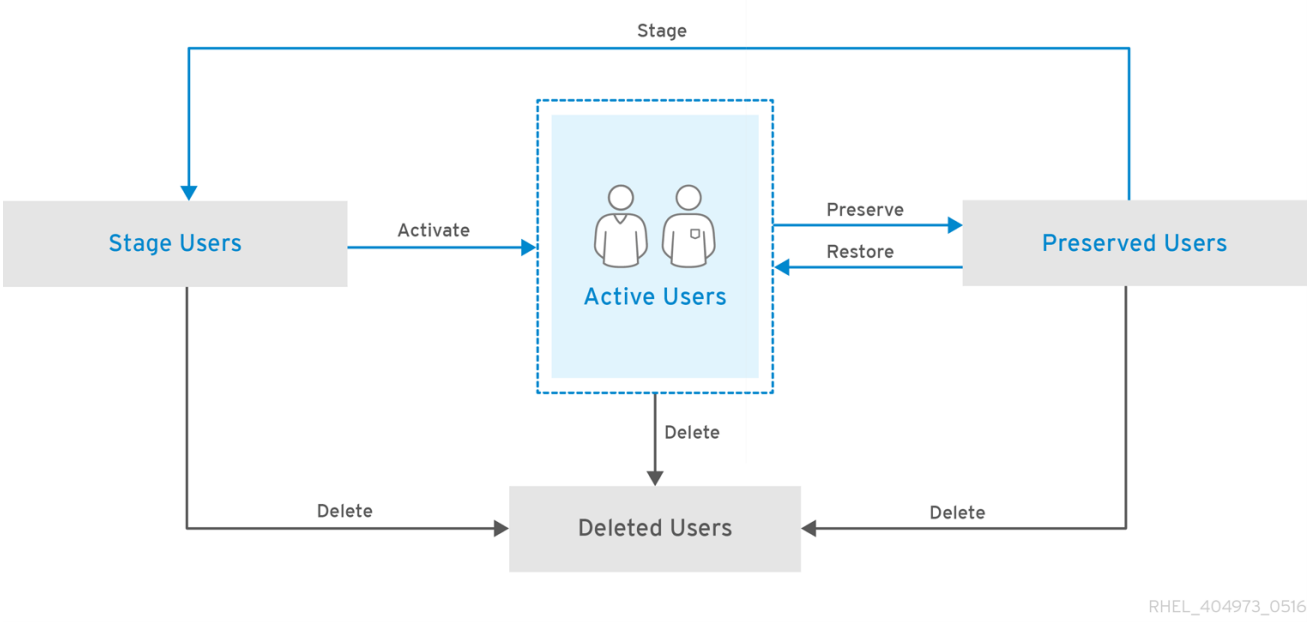
先ほどのユーザーが会社に再度参加します。管理者は、ユーザーアカウントを**preserved** 状態から **active** 状態に戻して、ユーザーアカウントを復元します。

preserved → stage

先ほどのユーザーは、会社に再び参加する計画です。管理者は、アカウントを **preserved** 状態から **stage** 状態に移動して、今後の再アクティブ化に向けてアカウントを準備します。

IdM からアクティブユーザー、ステージユーザー、および保存済みユーザーを完全に削除することもできます。ステージユーザーを **preserved** 状態に移動することはできず、永続的に削除できるだけです。

図11.1 ユーザーのライフサイクルの操作

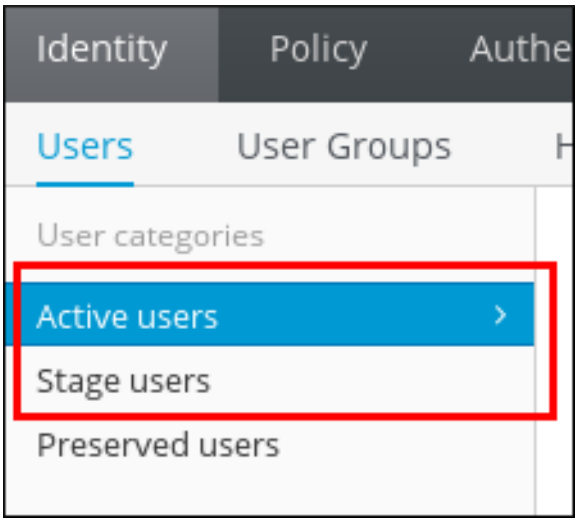


11.2.1. stage または Active ユーザーの追加

Web UI でユーザーの追加

1. **Identity → Users** タブを選択します。
2. **active** または **stage** 状態のユーザーを追加するかどうかに応じて、**Active users** または **Stage users** カテゴリーを選択します。

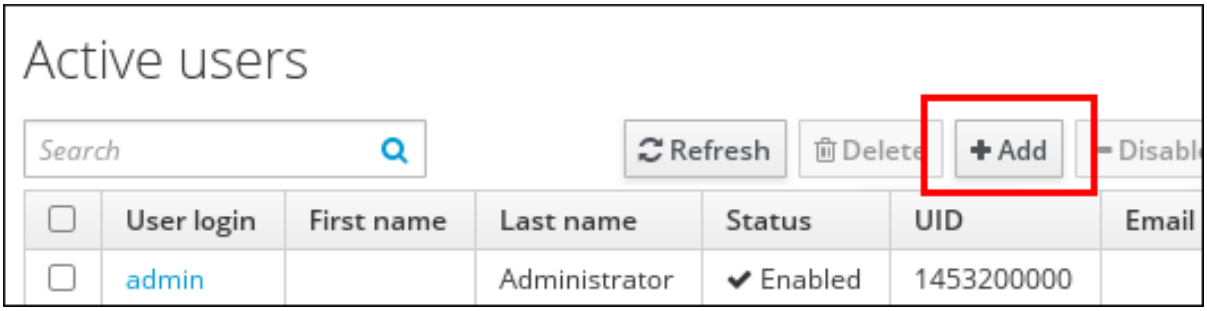
図11.2 ユーザーカテゴリーの選択



アクティブ または **ステージユーザーの** ライフサイクルの状態についての詳細は、「ユーザーのライフサイクル」 を参照してください。

3. ユーザーリストの上部にある **追加** をクリックします。

図11.3 ユーザーの追加



4. **Add User** フォームを入力します。

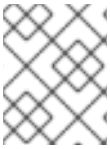
   ユーザーログインを手動で設定しないと、IdM は指定された名および姓に基づいてログインを自動的に生成することに注意してください。

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の追加を開始するか、**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 状態で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** 状態で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

アクティブ または **ステージユーザーの** ライフサイクルの状態についての詳細は、「ユーザーのライフサイクル」 を参照してください。

オプションなしで実行する場合は、**ipa user-add** および **ipa stageuser-add** により、必要最小限のユーザー属性の入力が求められ、その他の属性にはデフォルト値が使用されます。または、コマンドに直接、さまざまな属性を指定するオプションを追加することもできます。

インタラクティブセッションでは、オプションを指定せずにコマンドを実行すると、IdM は指定の名および姓に基づいて自動生成されたユーザーログインを提案し、大かっこ ([ ]) に表示します。デフォルトのログインを受け入れるには、**Enter** を押して確認します。カスタムログインを指定するには、デフォルトのログインを確認せず、代わりにカスタムログインを指定してください。

```
$ 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値を定義できます。つまり、対話型セッションよりも多くの情報を指定できます。たとえば、ステージ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
```

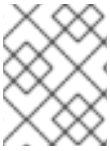
ipa user-add および **ipa stageuser-add** で使用できるオプションの完全リストは、**--help** オプションを追加してコマンドを実行します。

11.2.1.1. ユーザー名の要件

IdM は、以下の正規表現で説明できるユーザー名をサポートします。

```
'(?!^[0-9]+$)^[a-zA-Z0-9_.][a-zA-Z0-9_.-]*[a-zA-Z0-9_.$-]?$'
```

ユーザー名には、文字、数字、_、-、.、$ のみを含めることができ、少なくとも 1 文字が含まれている必要があります。



注記

ユーザー名の末尾がドル記号 ($) で終わる場合は、Samba 3.x マシンでのサポートが有効になります。

大文字を含むユーザー名を追加すると、IdM が名前を保存する際に自動的に小文字に変換されます。したがって、IdM にログインする場合、ユーザーは常にユーザー名をすべて小文字で入力する必要があります。また、**user** と **User** など、大文字と小文字のみが異なるユーザー名を追加することはできません。

ユーザー名のデフォルトの長さは、最大 32 文字です。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 文字にするには、次のコマンドを実行します。

```
$ ipa config-mod --maxusername=64
  Maximum username length: 64
  ...
```

11.2.1.2. カスタム UID または GID 番号の定義

カスタムの UID または GID 番号を指定せずに新しいユーザーエントリーを追加すると、IdM は ID 範囲で次に利用可能な ID 番号を自動的に割り当てます。これは、ユーザーの ID 番号が常に一意であることを意味します。ID 範囲の詳細は、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。

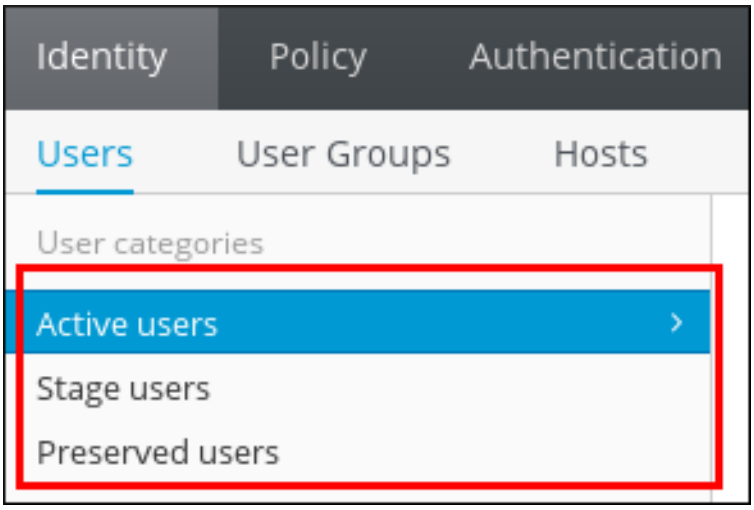
カスタム ID 番号を指定する場合、サーバーはカスタム ID 番号が一意であるかどうかを検証しません。このため、複数のユーザーエントリーに同じ ID 番号が割り当てられる可能性があります。Red Hat は、複数のエントリーに同じ ID 番号を割り当てることがないようにすることを推奨します。

11.2.2. ユーザーのリスト表示およびユーザーの検索

Web UI でのユーザーのリスト表示

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。

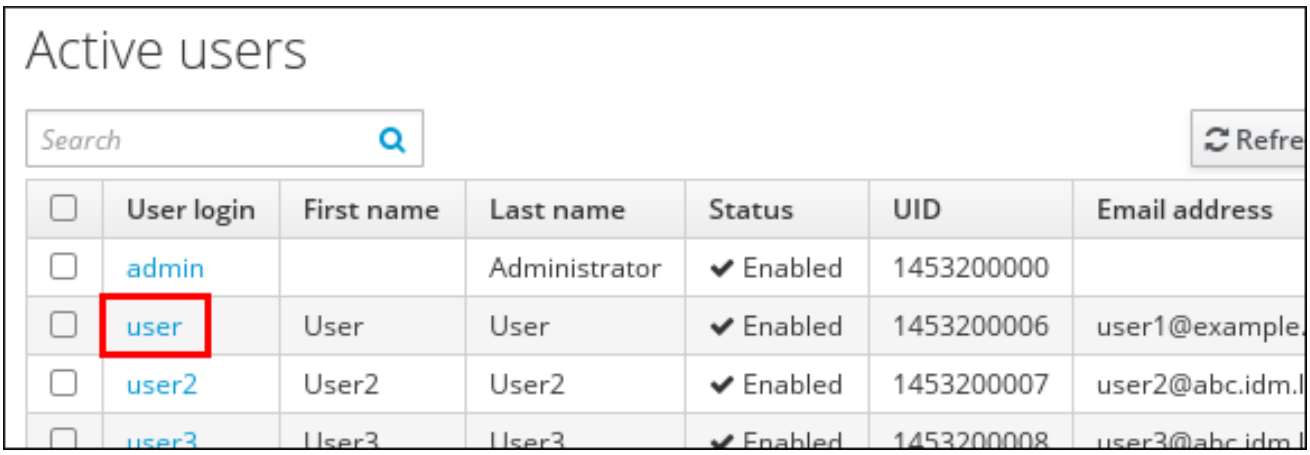
図11.4 ユーザーのリスト表示



Web UI でのユーザーに関する情報の表示

ユーザーに関する詳細情報を表示するには、ユーザーリストでユーザーの名前をクリックします。

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リスト表示

アクティブなユーザーをリスト表示するには、**ipa user-find** コマンドを実行します。すべてのステージユーザーをリスト表示するには、**ipa stageuser-find** コマンドを使用します。保存済みユーザーのリストを表示するには、**ipa user-find --preserved=true** コマンドを実行します。

以下に例を示します。

```
$ 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** にオプションと引数を追加すると、検索条件を定義し、検索結果をフィルタリングできます。たとえば、特定のタイトルが定義されているアクティブユーザーをすべて表示するには、次のコマンドを実行します。

```
$ 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** が含まれる全ステージユーザーを表示するには、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** で使用できるオプションの完全リストは、**--help** オプションを追加してコマンドを実行します。

コマンドラインからのユーザーに関する情報の表示

アクティブユーザーまたは保存済みユーザーの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
...
```

ステージ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. ユーザーのアクティベート、保存、削除、および復元

本セクションでは、ユーザーライフサイクルの異なる状態間でユーザーアカウントを移動する方法を説明します。IdM のライフサイクルの状態の詳細は、「ユーザーのライフサイクル」 を参照してください。

Web UI でのユーザーのライフサイクルの管理

ステージユーザーをアクティベートするには、以下を実行します。

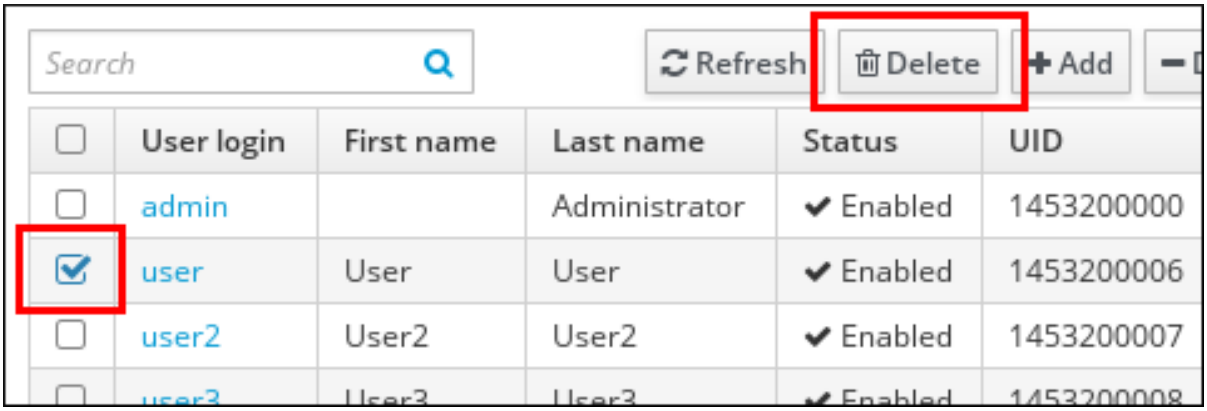
- **Stage users** リストで、アクティブにするユーザーを選択し、**Activate** をクリックします。

図11.6 ユーザーのアクティブ化

ユーザーを保存するか、削除するには、以下を実行します。

1. **アクティブユーザー** または **ステージユーザー** のリストで、ユーザーを選択します。**Delete** をクリックします。

図11.7 ユーザーの削除



2. アクティブなユーザーを選択した場合は、**delete** または **preserve** を選択します。ステージユーザーを選択している場合は、ユーザーを削除することしかできません。デフォルトの UI オプションは **delete** です。

   たとえば、アクティブユーザーを保存するには、次のコマンドを実行します。

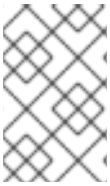
図11.8 Web UI での削除モードの選択

確認するには、**Delete** ボタンをクリックします。

保存済みユーザーを復元するには、以下を実行します。

- **Preserved users** リストで、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

ユーザーアカウントを復元しても、そのアカウントの以前の属性がすべて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ず、再度定義する必要があります。

Web UI では、ユーザーを **preserved** 状態から **stage** 状態に移行することができないことに注意してください。

コマンドラインからのユーザーのライフサイクルの管理

ステージ から **アクティブ** に移行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** コマンドまたは **ipa stageuser-del** コマンドを使用します。

- IdM データベースからアクティブなユーザーを永続的に削除するには、オプションを指定せずに **ipa user-del** を実行します。

```
$ ipa user-del user_login
--------------------
Deleted user "user3"
--------------------
```

- アクティブなユーザーアカウントを保持するには、**--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 データベースからステージユーザーを永続的に削除するには、**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ときは、**--continue** オプションを使用して、エラーに関係なくコマンドを続行します。成功および失敗した操作の概要は、コマンドが完了したときに標準出力ストリーム (**stdout**) に出力されます。

```
$ ipa user-del --continue user1 user2 user3
```

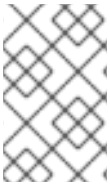
--continue を使用しないと、コマンドはエラーが発生するまでユーザーの削除を続行し、停止と終了を行います。

保存済みユーザーアカウントを**preserved** から **active** に移行して保存済み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"
------------------------------
```

保存済みユーザーアカウントを**preserved** から **stage** に移行して保存済み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、そのアカウントの以前の属性がすべて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ず、再度定義する必要があります。

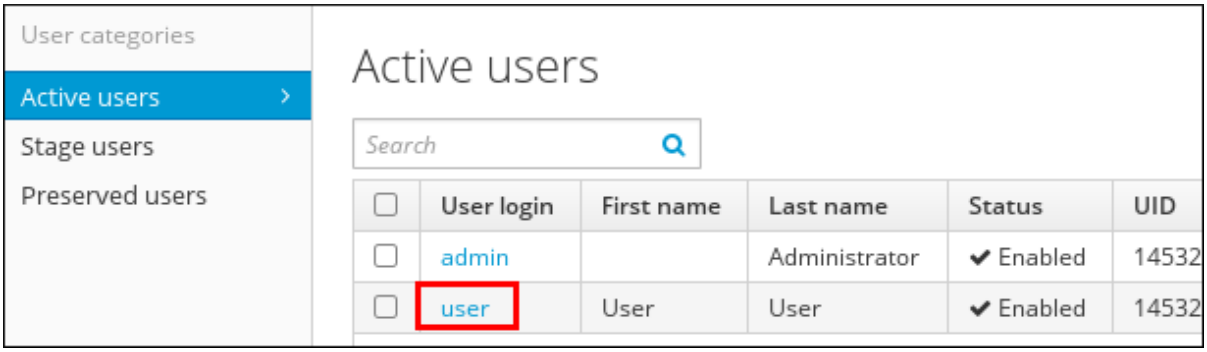
これらのコマンドとそれらが受け入れるオプションの詳細は、**-help**オプションを追加して実行してください。

11.3. ユーザーの編集

Web UI でのユーザーの編集

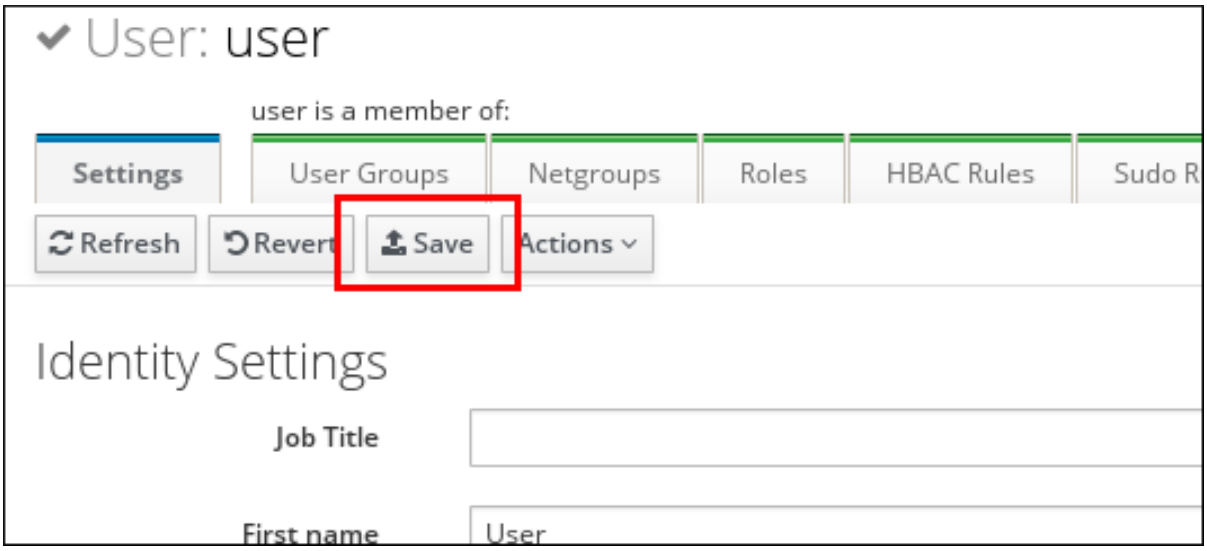
1. **Identity → Users** タブを選択します。
2. **Active users**、**Stage users**、または **Preserved users** カテゴリーを検索し、編集するユーザーを検索します。
3. 編集するユーザー名をクリックします。

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。

図11.11 変更後のユーザー属性の保存



Web UI でユーザー詳細を更新しても、新しい値は即座に同期されません。新しい値がクライアントシステムで反映されるまで、最長 5 分の時間が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** 状態のユーザーを修正するには、**ipa user-mod** コマンドを使用します。**stage** 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 コマンドおよび **ipa stageuser-mod** コマンドは、以下のオプションを受け入れます。

- 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
- 新しい属性値を指定するオプション

コマンドラインから変更できるユーザーエントリー属性の完全リストは、**ipa user-mod** および **ipa stageuser-mod** で使用できるオプションのリストを参照してください。オプションのリストを表示するには、**--help** オプションを追加してコマンドを実行します。

ipa user-mod または **ipa stageuser-mod** に属性オプションを追加すると、現在の属性値が上書きされます。たとえば、以下は、ユーザーのタイトルを変更するか、タイトルが指定されていない場合には新しいタイトルを追加します。

```
$ ipa user-mod user_login --title=new_title
```

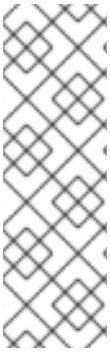
複数の値を使用できる LDAP 属性の場合、IdM でも複数の値を使用できます。たとえば、ユーザーは 2 つのメールアドレスをユーザーアカウントに保存できます。既存の値を上書きせずに別の属性値を追加するには、新しい属性値を指定するオプションと共に**--addattr** オプションを使用します。たとえば、メールアドレスがすでに指定されているユーザーアカウントに新しいメールアドレスを追加するには、次のコマンドを実行します。

```
$ 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 回使用します。

```
$ ipa user-mod user --addattr=mobile=mobile_number_1 --addattr=mobile=mobile_number_2
```

ipa user-mod コマンドでは、属性値を設定する **--setattr** オプションと、属性値を削除する **--delattr** オプションも使用できます。これらのオプションは、**--addattr** の使用と同様の方法で使用されます。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。ただし、メールアドレスを追加するには、**--addattr** オプションと共に **mail** 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
```

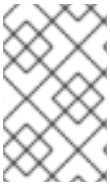
11.4. ユーザーアカウントの有効化、無効化

管理者は、アクティブユーザーのアカウントを無効および有効にすることができます。ユーザーアカウントを無効にすると、アカウントが非アクティブになります。無効にしたユーザーアカウントを使用して認証することはできません。アカウントが無効になったユーザーは IdM にログインできず、Kerberos などの IdM サービスを使用したり、タスクを実行したりすることができません。

無効にしたユーザーアカウントはそのまま IdM に残り、関連する情報は何も変更しません。保存済みユーザーのアカウントとは異なり、無効にしたユーザーアカウントは **active** 状態のままになります。したがって、**ipa user-find** コマンドの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
...
```

無効にしたユーザーアカウントは、すべて再度有効にできます。



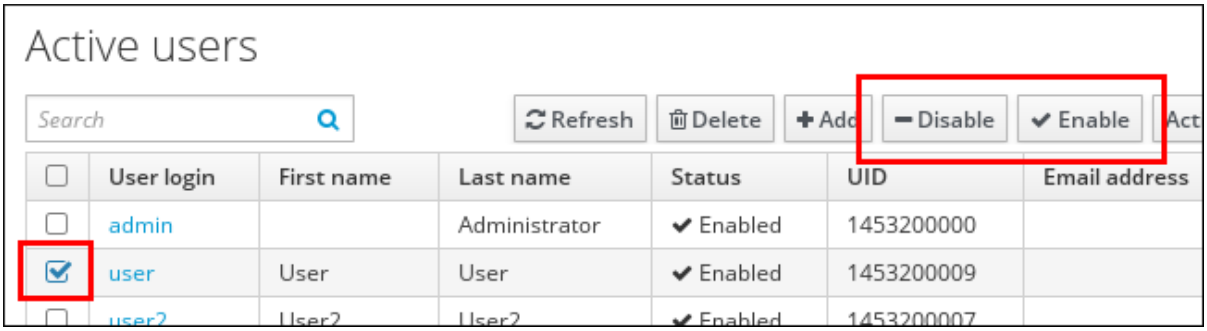
注記

ユーザーアカウントを無効にした後、既存の接続はユーザーの Kerberos TGT や他のチケットの有効期限が切れるまで有効です。チケットの期限が切れると、ユーザーが更新できなくなります。

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** リストから必要なユーザーを選択し、**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にするには、**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にするには、**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管理の許可

デフォルトでは、**admin**ユーザーしかユーザーのライフサイクルを管理したり、ユーザーアカウントを無効化または有効化したりすることができません。別の非管理者ユーザーがこれを実行できるようにするには、新規ロールを作成し、このロールに関連するパーミッションを追加し、管理者以外のユーザーをロールに割り当てます。

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。

ユーザーの変更およびパスワードのリセット

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。

ユーザー管理者

この権限には、アクティブなユーザーの追加、アクティブではないユーザーのアクティブ化、ユーザーの削除、ユーザー属性の変更を行うためのパーミッション、およびその他のパーミッションが含まれます。

ステージユーザーのプロビジョニング

この権限には、ステージユーザーを追加するパーミッションが含まれます。

ステージユーザー管理者

この権限には、ステージユーザーの追加や、ライフサイクル状態間でのユーザーの移動など、多くのライフサイクル操作を実行するパーミッションが含まれます。ただし、ユーザーを active 状態に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、および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。

異なるユーザーが異なるユーザー管理操作を実行することの許可

ユーザーアカウントの管理に関連する異なる権限を異なるユーザーに追加できます。たとえば、従業員のアカウントのエントリーとアクティベーションの権限を分けることができます。

- あるユーザーを、今後の従業員をステージユーザーとして IdM に追加できるがアクティブ化できない*ステージユーザー管理者*として設定します。
- 別のユーザーを、入社初日に従業員認証情報が検証された後にステージユーザーをアクティブ化できる *セキュリティー管理者* として設定します。

ユーザーが特定のユーザー管理操作を実行できるようにするには、必要な権限で新規ロールを作成し、ユーザーをそのロールに割り当てます。

例11.1 管理者以外のユーザーによるステージユーザー追加の許可

この例は、新規ステージユーザーの追加のみが許可され、他のステージ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示しています。

1. **admin** ユーザーまたはロールベースのアクセス制御を管理できる他のユーザーとしてログインします。

```
$ kinit admin
```

2. ステージユーザーの追加を管理する新しいカスタムロールを作成します。

   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** の権限をロールに追加します。この権限により、stage ユーザーを追加することができます。

```
$ 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付与します。

   a. 管理者以外のユーザーが存在しない場合は、新規ユーザーを作成します。この例では、user の名前は **stage_user_admin** 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** ユーザーを **システムプロビジョニングロール** に割り当てます。

```
$ 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。前の手順の 1 つで新しいユーザーとして **stage_user_admin** を作成した場合は、IdM は **admin** が設定した初期パスワードを変更するよう要求します。

```
$ 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ようにするには、**klist** ユーティリティーを使用できます。

```
$ 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. 新規ステージ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

ステージユーザーの追加後に IdM が報告するエラーは想定されたものです。**stage_user_admin** はステージユーザーの追加のみ許可され、それらのユーザーについての情報を表示できません。したがって、新たに追加した **stage_user** 設定の概要を表示する代わりに、IdM によりエラーが表示されます。

stage_user_admin ユーザーは、ステージユーザーについての情報を表示できません。したがって、**stage_user_admin** としてログインしている状態で新規 **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** としてログインできます。

```
$ 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 は、環境の設定をサポートしているため、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 をプロビジョニングできます。本セクションでは、このような設定の例を説明します。この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が使用するユーザーアカウントの設定」
- 「ステージユーザーアカウントを自動的にアクティベートするための IdM の設定」
- 「IdM アイデンティティーを管理するための外部プロビジョニングシステムの LDAP プロバイダーの設定」

11.6.1. 外部プロビジョニングシステムが使用するユーザーアカウントの設定

この手順では、外部プロビジョニングシステムが使用する 2 つの IdM ユーザーアカウントを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 ユーザーを追加する権限を持つ **provisionator** という名前のユーザーを作成します。ユーザーアカウントは、新規ステージユーザーを追加するために外部プロビジョニングシステムによって使用されます。

   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 の権限をロールに追加します。この特権により、ステージユーザーを追加できます。

```
$ 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を管理する権限を持つ **activator** ユーザーを作成します。ユーザーアカウントは、外部プロビジョニングシステムによって追加されるステージユーザーを自動的にアクティベートするために使用されます。

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** ユーザーに必要な特権を付与します。

ユーザーをデフォルトの **User Administrator** ロールに追加します。

```
$ 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有効期限やロックアウトを防ぎますが、複雑なパスワードを必要とす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および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
```

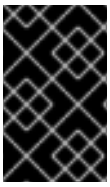
新しい IdM ユーザーのパスワードはすぐに失効するため、パスワードの変更が必要になります。

関連情報:

- 新規ユーザーの追加の詳細は、「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特権をユーザーに付与する方法は、「管理者以外のユーザーによるユーザーエントリー管理の許可」 を参照してください。
- IdM パスワードポリシーの管理の詳細は、28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. ステージユーザーアカウントを自動的にアクティベートするための IdM の設定

この手順では、ステージ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、指定した間隔でスクリプトを自動的に実行します。これにより、新規ユーザーアカウントが自動的にアクティベートされ、作成直後に使用できます。



重要

この手順では、スクリプトが IdM に追加する前に、新しいユーザーアカウントを検証する必要がないことを前提としています。たとえば、ユーザーが外部プロビジョニングシステムの所有者によってすでに検証されている場合は、検証は必要ありません。

IdM サーバーの 1 つでのみアクティベーションプロセスを有効にするだけで十分です。

1. アカウントのアクティベーション用に 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場合は、1 つのサーバーだけで keytab ファイルを生成します。次に、keytab ファイルを他のサーバーにコピーします。

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および所有権を編集して、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** ユニットファイル **/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 **/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** を有効にします。

```
# systemctl enable ipa-activate-all.timer
```

関連情報:

- **systemd** ユニットファイルの詳細は、『システム管理者のガイド』のsystemd ユニットファイルを使用したサービスの管理の章を参照してください。

11.6.3. IdM アイデンティティーを管理するための外部プロビジョニングシステムの LDAP プロバイダーの設定

本セクションでは、さまざまなユーザーおよびグループ管理操作のテンプレートを説明します。テンプレートを使用して、プロビジョニングシステムの LDAP プロバイダーを設定して、IdM ユーザーアカウントを管理できます。たとえば、従業員が退職した後にユーザーアカウントを非アクティブにするようにシステムを設定できます。

LDAP を使用したユーザーアカウントの管理

基盤の Directory Server データベースを編集して、新しいユーザーエントリーの追加、既存のエントリーの変更、ライフサイクルの異なる状態間でのユーザーの移動、またはユーザーの削除を行うことができます。データベースを編集するには、**ldapmodify** ユーティリティーを使用します。

以下の LDIF 形式のテンプレートは、**ldapmodify** を使用して変更する属性に関する情報を提供します。詳細な手順例は、例11.2「**ldapmodify** を使用したステージユーザーの追加」 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。

新規ステージユーザーの追加

UID と GID が自動的に割り当てられるユーザーの追加

```
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 と GID が静的に割り当てられるユーザーの追加

```
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
```

ステージ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ります。

既存ユーザーの変更

ユーザーを変更する前に、ユーザーのログインを検索してユーザーの識別名 (DN) を取得します。以下の例では、*user_allowed_to_read*ユーザーは、ユーザーやグループ情報の読み取りが許可されるユーザーで、*password* はこのユーザーのパスワードになります。

```
# 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には、以下を実行します。

```
dn: distinguished_name
changetype: modify
replace: attribute_to_modify
attribute_to_modify: new_value
```

ユーザーを無効にするには、以下を実行します。

```
dn: distinguished_name
changetype: modify
replace: nsAccountLock
nsAccountLock: TRUE
```

ユーザーを有効にするには、以下を実行します。

```
dn: distinguished_name
changetype: modify
```

```
replace: nsAccountLock
nsAccountLock: FALSE
```

ユーザーを保存するには、以下を実行します。

```
dn: distinguished_name
changetype: modrdn
newrdn: uid=user_login
deleteoldrdn: 0
newsuperior: cn=deleted users,cn=accounts,cn=provisioning,dc=example
```

nssAccountLock 属性の更新は、ステージユーザーおよび保存済みユーザーには影響を与えません。更新操作が正常に完了しても、属性値は **nssAccountLock: TRUE** のままになります。

新規グループの作成

新規グループを作成するには、以下を実行します。

```
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、グループ名を使用して検索してグループの識別名 (DN) を取得します。

```
# ldapsearch -YGSSAPI  -H ldap://server.example.com -b
"cn=groups,cn=accounts,dc=example,dc=com" "cn=group_name"
```

既存グループを削除するには、以下を実行します。

```
dn: group_distinguished_name
changetype: delete
```

グループにメンバーを追加するには、以下を実行します。

```
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には、以下を実行します。

```
dn: distinguished_name
changetype: modify
delete: member
member: uid=user_login,cn=users,cn=accounts,dc=example,dc=com
```

ステージまたは保存済みユーザーをグループに追加しないでください。更新操作が正常に完了しても、ユーザーはグループのメンバーとしては更新されません。アクティブなユーザーのみがグループに所属できます。

例11.2 ldapmodify を使用したステージユーザーの追加

標準の **interorgperson** オブジェクトクラスを使用して新規 **stageuser** ユーザーを追加するには、以下を実行します。

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. stage エントリーの内容を検証して、プロビジョニングシステムが必要なすべての POSIX 属性を追加し、stage エントリーをアクティブ化する準備ができていることを確認することを検討してください。新規 stage ユーザーの LDAP 属性を表示するには、**ipa stageuser-show --all --raw** コマンドを実行します。ユーザーは、**nsaccountlock** 属性により明示的に無効になっ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 の **modrdn** 操作を使用して **ユーザー** を保存するには、以下を実行します。

1. **ldapmodify** ユーティリティー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. 必要に応じて、保存済みユーザーをリスト表示して、ユーザーが保持されていることを確認します。

```
$ 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 はいずれも、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ロールは、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 (1章*Red Hat Identity Management の概要* を参照)。この関係では、ドメイン内ホストの認可および制御の **委譲** が不可欠な要素です。

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) を保存できるサービスは、IdM サービスとして設定できます。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 証明書やキータブ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。)

IdM ドメインは、共通の ID 情報、共通ポリシー、および共有サービスを使用して、マシン間で共通性を確立します。ドメインのクライアントとしてのドメイン機能に属するマシンです。これは、ドメインが提供するサービスを使用することを意味します。IdM ドメインは、マシン専用の 3 つの主なサービスを提供します。

- DNS
- Kerberos
- 証明書管理

ユーザーと同様に、マシンは IdM によって管理されるアイデンティティーです。クライアントマシンは DNS を使用して IdM サーバー、サービス、およびドメインメンバーを特定します。これは、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 または証明書を使用して、ドメインに対して認証できます。

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。IdM サーバーがマシンを信頼し、そのマシン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 接続を受け入れるには、マシン認証が必要です。クライアントを認証すると、IdM サーバーはそのリクエストに応答できます。IdM は、マシン認証において 3 つのアプローチをサポートします。

- SSH 鍵。ホストの 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、SSSD (System Security Services Daemon) は Identity Management を ID プロバイダーとして使用し、OpenSSH およびその他のサービスと一緒に機能して、IdM の中央にある公開鍵を参照できます。この操作は、「ホストの公開 SSH 鍵の管理」 に説明があります。
- キーテーブル (または *キータブ*。ユーザーパスワードに多少類似する対称キー) 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 チケットの付与、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 29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、マシンは IdM サーバーの認証局により発行され、IdM の Directory Server に保存されている SSL 証明書を使用します。次に、証明書はマシンに送信され、サーバーに対する認証時に提示されます。クライアントでは、証明書は **certmonger** というサービスにより管理されます。

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、ホストに関する情報 (物理的な場所、MAC アドレス、鍵、証明書など、システム設定を除く) 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に、この情報のほとんど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	説明
説明	**--desc**=description	ホストの説明。
局所性	**--locality**=locality	ホストの地理的な場所。
場所	**--location**=location	データセンターのラックなど、ホストの物理的な場所。
プラットフォーム	**--platform**=string	ホストのハードウェアまたはアーキテクチャー。
オペレーティングシステム	**--os**=string	ホストのオペレーティングシステムとバージョン。

UI フィールド	コマンドラインオプション	説明
MAC アドレス	**--macaddress**=address	ホストの MAC アドレス。これは多値属性です。MAC アドレスは、NIS プラグインにより、ホスト用の NIS の **ethers** マップを作成するために使用されます。
SSH 公開鍵	**--sshpubkey**=string	ホストの完全 SSH 公開鍵。これは複数値の属性であるため、複数の鍵を設定できます。
プリンシパル名 (編集不可)	**--principalname**=principal	ホストの Kerberos プリンシパル名。**-p** に別のプリンシパルを明示的に設定しない限り、クライアントのインストール時にホスト名がデフォルトになります。これはコマンドラインツールを使用して変更できますが、UI で変更することはできません。
ワンタイムパスワードの設定	**--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**	一括登録で使用されるランダムなパスワードを生成します。
-	**--certificate**=string	ホストの証明書ブロブ。
-	**--updatedns**	これにより、IP アドレスが変更した場合に、ホストが DNS エントリーを動的に更新できるかどうかが設定されます。

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

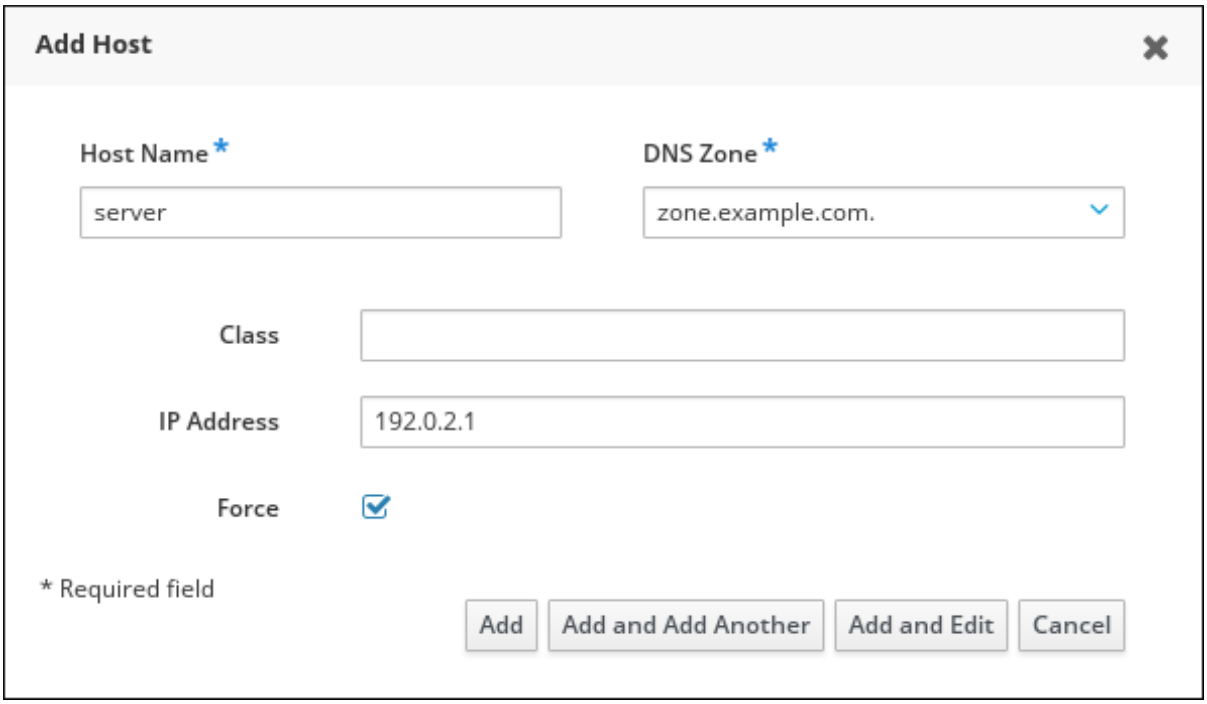
1. **Identity** タブを開き、サブタブの **ホスト** を選択します。
2. ホストリストの上部にある **追加** をクリックします。

図12.1 ホストエントリーの追加

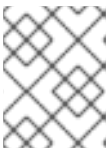
3. マシン名を入力し、ドロップダウンリストの設定済みゾーンからドメインを選択します。ホストに静的 IP アドレスが割り当てられている場合は、ホストエントリーにそのアドレスを追加して、DNS エントリーが完全に作成されるようにします。

   必要に応じて、一部のユースケースでホストに値を追加するには、**Class** フィールドを使用します。この属性に配置されるセマンティクスは、ローカル解釈用です。

図12.2 ホストウィザードの追加



「マスター DNS ゾーンの追加および削除」で説明されているように、DNS ゾーンは IdM で作成可能です。IdM サーバーが DNS サーバーを管理しない場合は、通常のテキストフィールドなど、メニューエリアでゾーンを手動で入力できます。

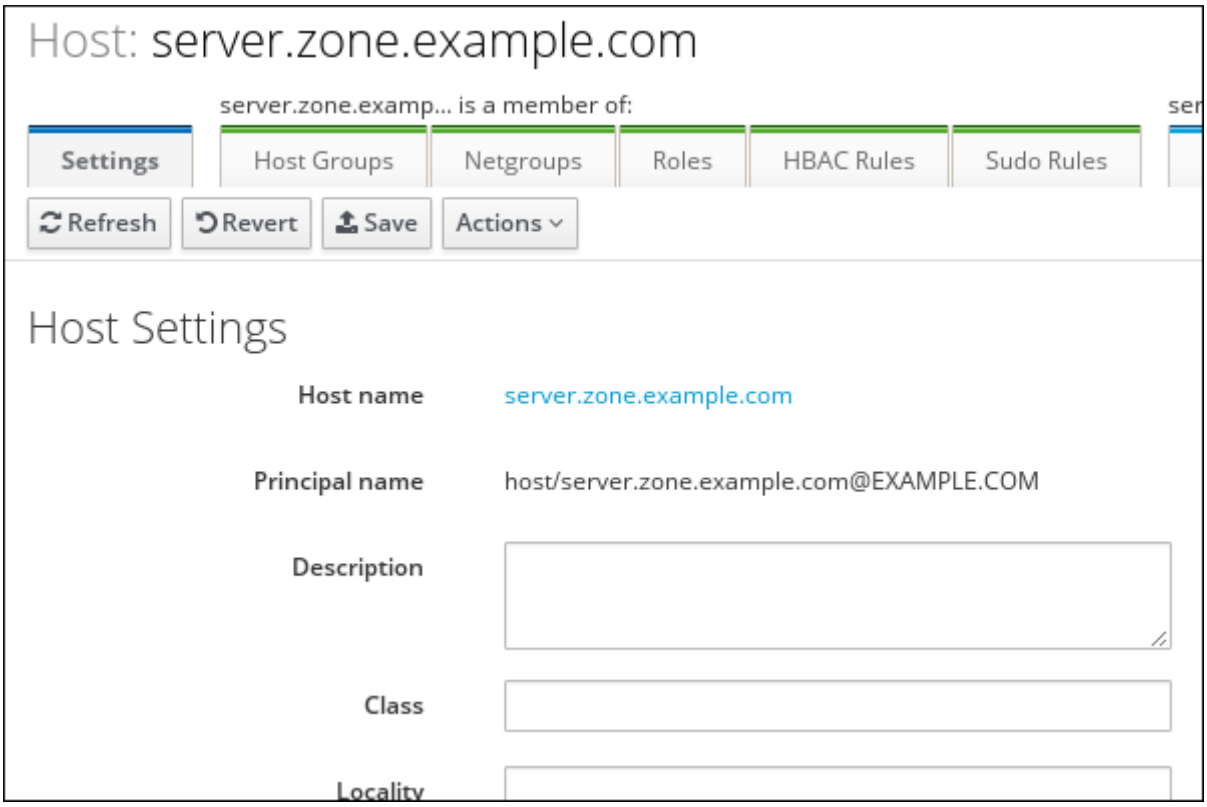


注記

ホストが DNS 経由で解決できるかどうかの確認を行わないようにするには、**強制** チェックボックスを選択します。

4. **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。ホストのハードウェアと物理的な場所に関する情報は、ホストエントリーに追加できます。

図12.3 拡張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** コマンドを使用して作成されます。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** の全オプションリストは、**ipa host** の 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 レルムに追加し、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 を管理するように設定されている場合には、**--ip-address** および **--force** オプションを使用して、DNS リソースレコードにホストも追加できます。

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 アドレスがないこと、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はよくあります。たとえば、ラップトップが Identity Management クライアントとして事前設定されている場合がありますが、設定時には IP アドレスがありません。DHCP を使用するホストは、**--force** を使用して DNS エントリーで設定可能です。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。

例12.2 DHCP でのホストエントリーの作成

```
$ ipa host-add --force client1.example.com
```

ホストレコードは、**host-del** コマンドを使用して削除されます。IdM ドメインが DNS を使用する場合には、**--updatedns** オプションを使用すると、ホストに関連のあるレコードはすべて DNS から削除されます。

```
$ ipa host-del --updatedns client1.example.com
```

12.4. ホストエントリーの無効化と再有効化

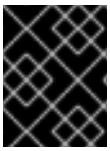
アクティブなホストは、ドメイン内の他のサービスやホスト、ユーザーからアクセス可能です。アクティビティーからホストを削除する必要がある場合もあります。ただし、ホストを削除するとエントリーや関連する設定もすべて完全に削除されてしまいます。

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。これには、**host-disable** コマンドを使用します。

以下に例を示します。

```
[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、そのホストが無効になるだけではありません。そのホストで設定されているすべてのサービスも無効にします。

12.4.2. ホストの再有効化

このセクションでは、無効な IdM ホストを再度有効にする方法を説明します。

ホストを無効にすると、アクティブなキータブが強制的に削除され、設定エントリーを変更せずにホストが IdM ドメインから削除されます。

ホストを再度有効にするには、以下を追加して、**ipa-getkeytab** コマンドを使用します。

- キータブを要求する IdM サーバーを指定する **-s** オプション
- プリンシパル名を指定する **-p** オプション
- キータブを保存するファイルを指定する **-k** オプション

たとえば、**client.example.com** の **server.example.com** から新規ホストキータブを要求し、キータブを **/etc/krb5.keytab** ファイルに保存するには、次のコマンドを実行します。

```
$ ipa-getkeytab -s server.example.com -p host/client.example.com -k /etc/krb5.keytab -D
"cn=directory manager" -w password
```

注記

管理者の認証情報を使用して、**-D "uid=admin,cn=users,cn=accounts,dc=example,dc=com"** を指定することもできます。認証情報は、ホストのキータブの作成を許可されたユーザーに対応することが重要です。

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、ユーザーが、**kinit admin** などを使用して TGT を取得した場合に、LDAP 認証情報 (**-D** および **-w**) を使用せずに実行できます。無効化されたホストでコマンドを直接実行するには、LDAP 認証情報を提供して IdM サーバーに認証します。

12.5. ホストの公開 SSH 鍵の管理

OpenSSH は、**公開鍵**を使用し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アクセスを試みてキーのペアを提示します。ホストの初回認証時には、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。次に、マシンはホストの公開鍵を **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、およびキーの 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 鍵は、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、セキュリティー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) がホストの SSH 鍵をキャッシュして取得するように設定し、アプリケーションやサービスがホストキーを 1 カ所で検索できるようにします。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ーとすることができます。このため管理者は、ホスト SSH 鍵の配布や更新、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場合には、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して保存される前に、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ブから、RSA または DSA キーといったキーのタイプを識別できます。ただし、**~/.ssh/known_hosts** などのキーファイルでは、サーバーのホスト名および IP アドレス、キーのタイプ、キー自体で、キーのエントリーが識別されます。以下に例を示します。

```
host.example.com,1.2.3.4 ssh-rsa AAA...ZZZ==
```

これは、要素の順序が **type key== comment** のユーザーの公開鍵エントリーとは多少異なります。

```
"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て表示できます。このような場合には、~/**.ssh**/**known_hosts** ファイルからのホスト公開鍵エントリーが、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。

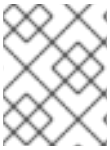
```
ssh-rsa AAA...ZZZ== host.example.com,1.2.3.4
```

キータイプは公開鍵のコンテンツから自動的に判断されます。個別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、公開鍵ブロブ自体のみとなります。

12.5.2. **ipa-client-install** および OpenSSH

デフォルトでは、**ipa-client-install** スクリプトは、IdM クライアントマシンで OpenSSH サーバーおよび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 SSSD、OpenSSH、および Identity Management を使用してキーキャッシングおよび取得に必要な全設定が実行されます。

クライアントインストール時に SSH サービスが有効な場合 (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と、クライアントには RSA と DSS の 2 つの SSH 鍵を作成されます。

他にも **--ssh-trust-dns** というクライアント設定オプションがあり、**ipa-client-install** コマンドに指定して実行でき、キーのフィンガープリントを格納する IdM DNS レコードを OpenSSH が信頼するように自動設定します。

別の方法として、クライアントのインストール時に **--no-sshd** オプションを使用して OpenSSH を無効にできます。この設定により、インストールスクリプトで OpenSSH サーバーを設定できなくなります。

別の **--no-dns-sshfp** というオプションを使用すると、ホストが独自の DNS エントリーで DNS SSHFP レコードを作成できなくなります。このオプションは、**--no-sshd** オプションと合わせて使用することも、なしでも使用できます。

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。以下に例を示します。

```
server.example.com,1.2.3.4 ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAQEApvjBvSFSkTU0WQW4eOweeo0DZZ08F9Ud21xlLy6F
OhzwpXFGIyxvXZ52+siHBHbbqGL5+14N7UvElruyslIHx9LYUR/pPKSMXCGyboLy5aTNl5OQ5
EHwrhVnFDIKXkvp45945R7SKYCUtRumm0Iw6wq0XD4o+ILeVbV3wmcB1bXs36ZvC/M6riefn
9PcJmh6vNCvIsbMY6S+FhkWUTTiOXJjUDYRLIwM273FfWhzHK+SSQXeBp/zIn1gFvJhSZMR
i9HZpDoqxLbBB9QIdIw6U4MIjNmKsSI/ASpkFm2GuQ7ZK9KuMItY2AoCuIRmRAdF8iYNHBT
XNfFurGogXwRDjQ==
```

必要に応じて、ホストキーを生成します。OpenSSH ツールを使用する場合は、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:GAUIDVVEgly7rs1ITWP6oguHz8BKvyZkpqCqVSsmi7c server.example.com
The key's randomart image is:
+--[ RSA 2048]----+
|             .. |
|              .+|
|         o   .* |
|        o . .. *|
|       S + .  o+|
|        E . .. .|
|       . = .  o |
|        o .  ..o|
|           .....|
+-----------------+
```

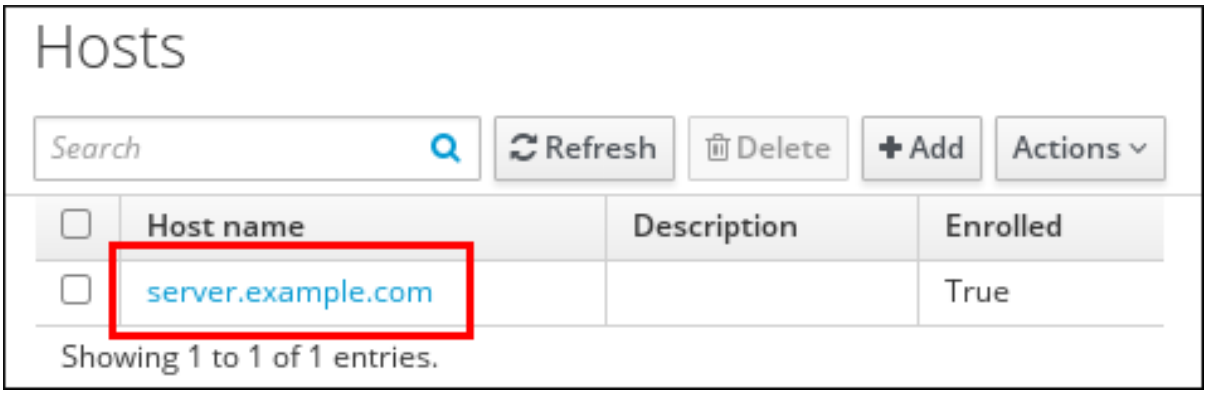
2. 公開鍵をキーファイルからコピーします。完全なキーエントリーは、**hostname,IP type key==** の形式です。**key==** は必須ですが、エントリー全体を保存できます。エントリーの全要素を使用するには、エントリーを再編成して、順番が **type key== [host name,IP]** になるように設定します。

```
[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、サブタブの **ホスト** を選択します。

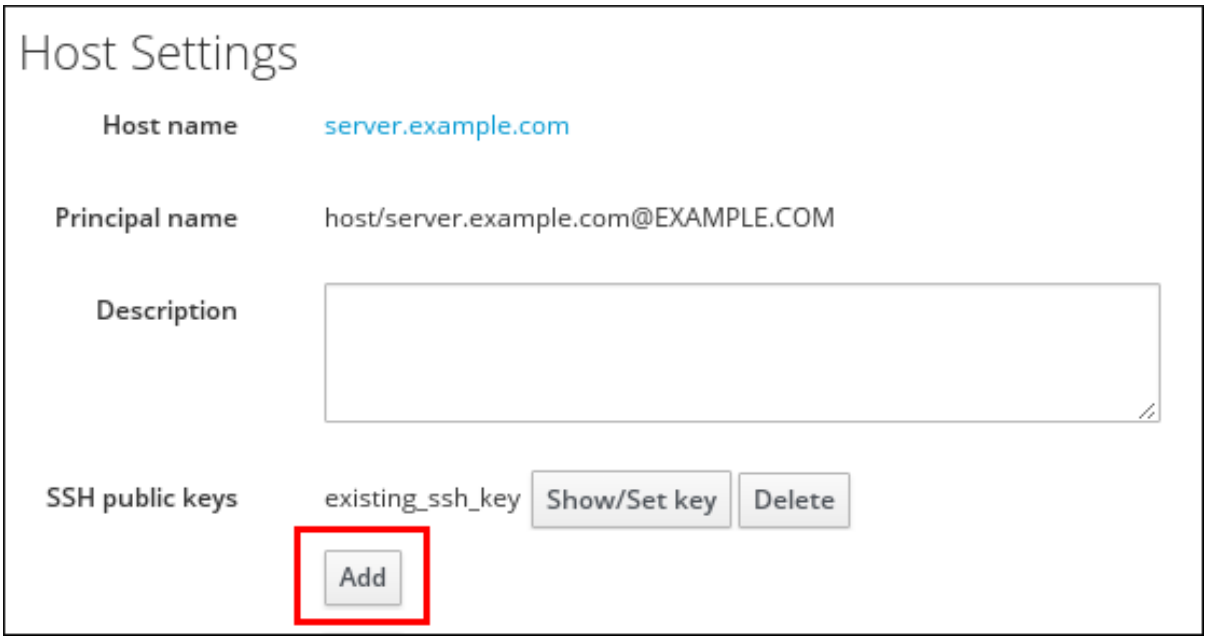
4. 編集するホスト名をクリックします。

図12.4 ホストのリスト



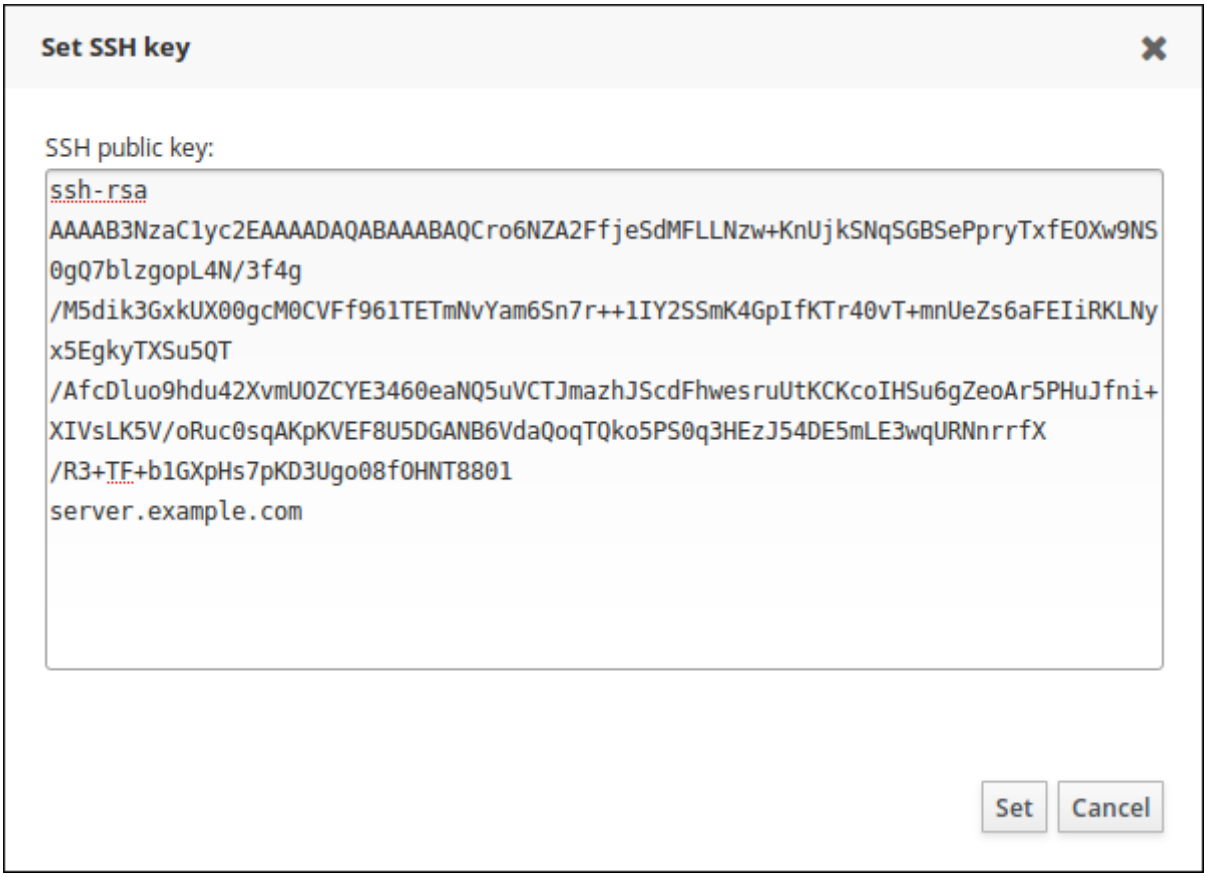
5. **Settings**タブの **Host Settings** エリアで、**SSH public keys** の横にある **Add** をクリックします。

図12.5 SSH キーの追加



6. ホストの公開鍵に貼り付けて、**Set** をクリックします。

図12.6 SSH キーの設定



SSH 公開鍵 エリアに、新しい鍵が表示されるようになりました。**Show/Set key** をクリックすると、送信したキーが開きます。

7. 複数のキーをアップロードするには、公開鍵リストの下にある **Add** をクリックして、他のキーをアップロードします。

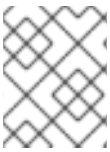
8. すべてのキーが送信されたら、ホストページ上部の **Save** をクリックして変更を保存します。

公開鍵を保存すると、エントリーは鍵フィンガープリント、コメント (存在する場合)、および鍵の種類として表示されます。[2].

ホストキーをアップロードしたら、「OpenSSH サービスのキャッシュを提供するように SSSD を設定」 で説明するように、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。

12.5.4. コマンドラインからのホストキーの追加

ホスト SSH 鍵は、**host-add** を使用してホストを作成する時か、エントリーを後で修正する時に、 IdM のホストエントリーに追加されます。



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。

1. **--sshpubkey** オプションを指定して **host-mod** コマンドを実行し、base64 にエンコードされた公開鍵をホスト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** オプションも使用してホストの DNS エントリーも更新します。

   以下に例を示します。

   ```
   [jsmith@server ~]$ ipa host-mod --sshpubkey="ssh-rsa RjlzYQo==" --updatedns
   host1.example.com
   ```

   実際のキーは通常、等号 (=) で終わりますが、より長いです。

   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

   ホストには複数の公開鍵を指定できます。

2. ホストキーをアップロードしたら、「OpenSSH サービスのキャッシュを提供するように SSSD を設定」 で説明するように、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。

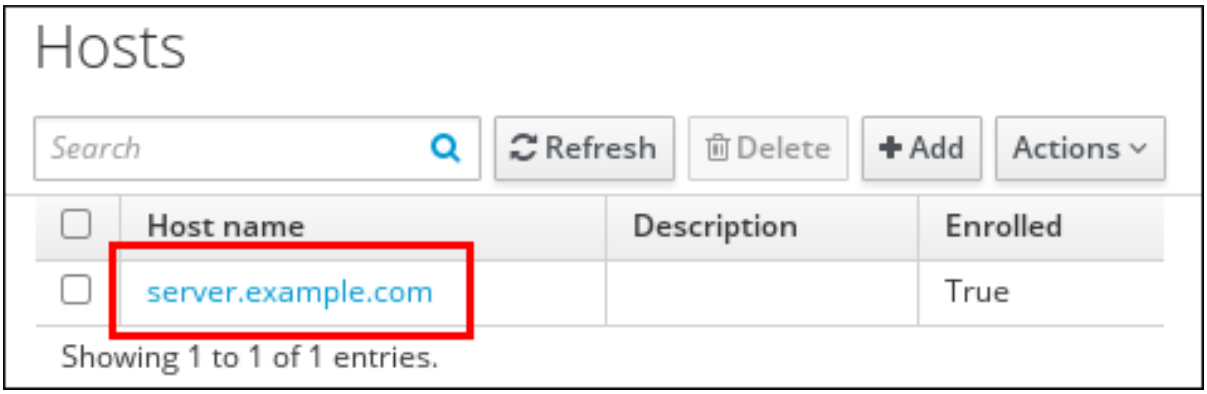
12.5.5. ホストキーの削除

ホストキーは、期限が切れるか有効でなくなると、削除できます。

Web UI を使用して個別のホストキーを削除するのが最も簡単な方法です。

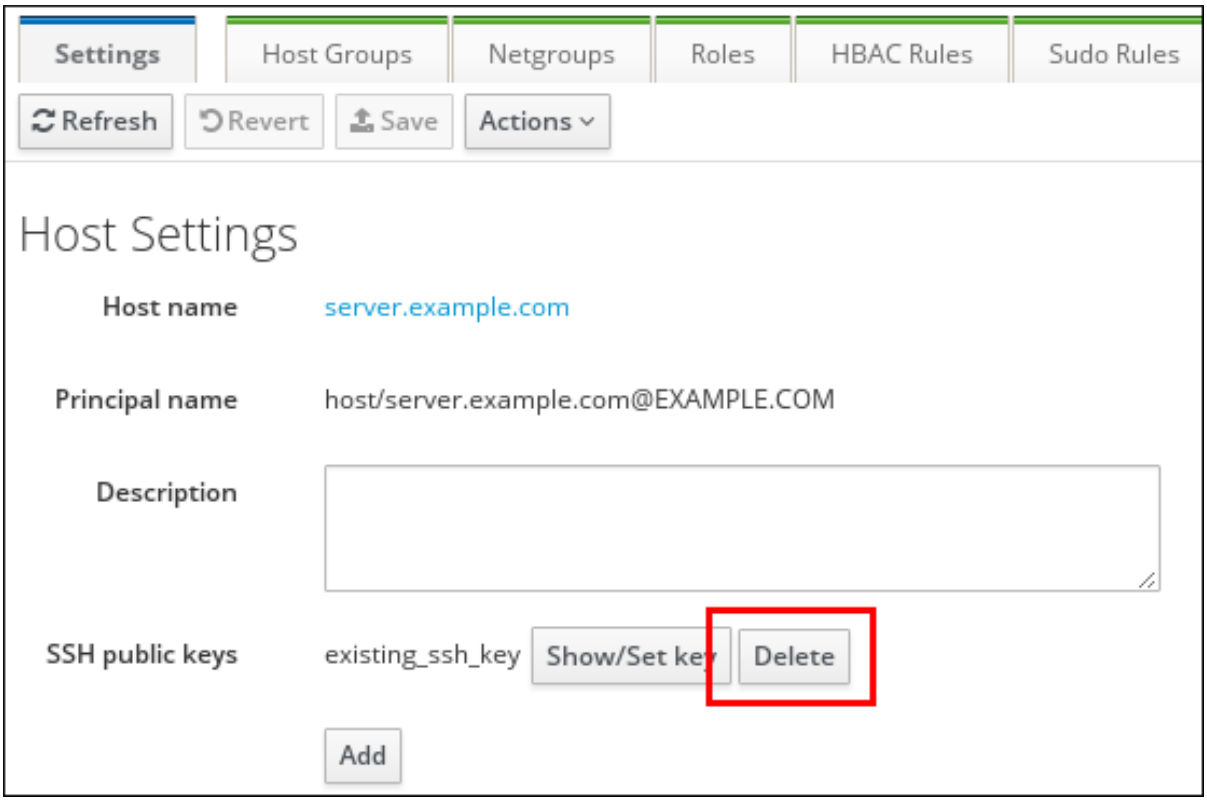
1. **Identity** タブを開き、サブタブの **ホスト** を選択します。
2. 編集するホスト名をクリックします。

図12.7 ホストのリスト



3. **SSH public keys** エリアで、削除するキーのフィンガープリントのそばにある **Delete** をクリックします。

図12.8 公開鍵の削除



4. ホスト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で、すべてのキーを削除することもできます。方法は、**--sshpubkey=** を空の値に指定して **ipa host-mod** を実行します。これで、対象ホストの公開鍵が **すべて** 削除されます。また、ホストの DNS エントリーの更新には、**--updatedns** オプションを使用します。以下に例を示します。

```
[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** テーブルをホストできます。このテーブルを使うと、システムのプラットホ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。

IdM では、**ou=ethers** サブツリーのディレクトリーに、該当の **ethers** エントリーが含まれた状態で、各システムが作成されます。

```
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、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。以下に例を示します。

```
[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] キータイプがアップロードされたキーに含まれていない場合には、キー自体をもとに自動的に決定されます。

第13章 ユーザーおよびホストグループの管理

13.1. IDM でのユーザーおよびグループの仕組み

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、共通の特権、パスワードポリシーなどの特性が指定された一連のユーザーです。

ホストグループは、共通のアクセス制御ルールやその他の特性を持つ IdM ホストセットです。

たとえば、企業の部門、物理的な場所、またはアクセス制御要件に関してグループを定義できます。

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
- 他の IdM ユーザーグループ
- 外部ユーザー (IdM の外部に存在するユーザー)

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

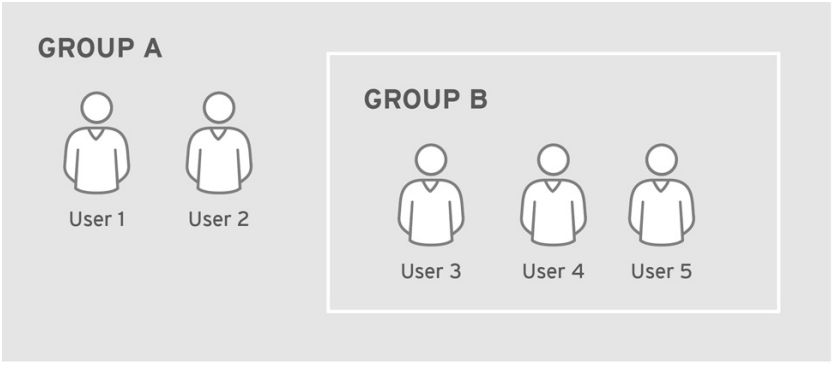
13.1.3. 直接および間接のグループメンバー

IdM のユーザーおよびホストグループ属性は、直接メンバーと間接メンバーの両方に適用されます。グループ B がグループ A のメンバーである場合、グループ B のすべての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「直接および間接グループメンバーシップ」 では以下のようになります。

- ユーザー 1 とユーザー 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。

図13.1 直接および間接グループメンバーシップ



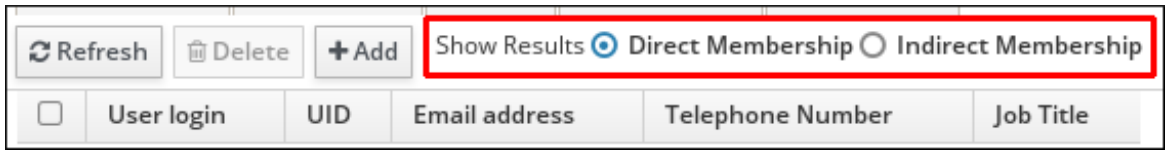
ユーザーグループ A にパスワードポリシーを設定すると、そのポリシーはユーザーグループ B のすべてのユーザーにも適用されます。

例13.1 直接および間接のグループメンバーの表示

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と削除」を参照してください。
2. 以下を追加します。
   - **group_A** のメンバーとしての 1 ユーザー
   - **group_B** のメンバーとしての別のユーザー
   - **group_A** のメンバーとする **group_B**

   「ユーザーまたはホストグループメンバーの追加と削除」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記載されている個別の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。**直接メンバーシップ** と **間接メンバーシップ** 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: **ipa group-show** コマンドを使用します。

```
$ ipa group-show group_A
  ...
  Member users: user_1
  Member groups: group_B
  Indirect Member users: user_2
```

間接メンバーのリストには、信頼された Active Directory ドメインの外部ユーザーが含まれません。Active Directory 信頼ユーザーオブジェクトは、IdM 内に LDAP オブジェクトとして存在しないため、IdM インターフェイスには表示されません。

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、メンバーの POSIX 属性に対応します。Active Directory と対話するグループは POSIX 属性を使用できないことに注意してください。

非 POSIX グループ

このタイプのグループのすべてのグループメンバーは、IdM ドメインに属している必要があります。

外部グループ

外部グループを使用して、IdM ドメイン外の ID 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、またはディレクトリーサービスです。

非 POSIX および外部グループは、POSIX 属性に対応していません。たとえば、これらのグループには GID が定義されていません。

例13.2 各種ユーザーグループの検索

1. **ipa group-find** コマンドを実行して、すべてのユーザーグループを表示します。
2. **ipa group-find --posix** コマンドを実行して、すべての POSIX グループを表示します。
3. **ipa group-find --nonposix** コマンドを実行して、すべての非 POSIX グループを表示します。
4. **ipa group-find --external** コマンドを実行して、すべての外部グループを表示します。

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	すべての IdM ユーザー
admins	ユーザーグループ	管理権限を持つユーザー (初期のデフォルトの **admin** ユーザー)
editors	ユーザーグループ	ユーザーは、管理ユーザーの権限がすべてなくても、Web UI で他の IdM ユーザーを編集できます。
trust admins	ユーザーグループ	Active Directory 信頼を管理する権限を持つユーザー
ipaservers	ホストグループ	すべての 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権限およびポリシーが適用されます。たとえば、ユーザーを **admins** グループに追加すると、ユーザーに管理者権限が付与されます。

警告

admins グループを削除しないでください。**admins** は IdM で必要な事前定義グループであるため、この操作により特定のコマンドで問題が生じます。

警告

ホストを **ipaservers** ホストグループに追加する場合は注意してください。**ipaservers** のすべてのホストは、IdM サーバーにプロモートできます。

さらに、IdM で新しいユーザーが作成されるたびに、IdM は、デフォルトで *ユーザーのプライベートグループ*を作成します。

- 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になります。
- ユーザーは、ユーザープライベートグループの唯一のメンバーです。
- プライベートグループの GID は、ユーザーの UID と一致します。

例13.3 ユーザープライベートグループの表示

ipa group-find --private コマンドを実行して、すべてのユーザープライベートグループを表示します。

```
$ 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
----------------------------
```

状況によっては、NIS グループまたは別のシステムグループが、ユーザープライベートグループに割り当てられる GID をすでに使用している場合など、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方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と削除

グループを追加するには、以下のツールを使用できます。

- Web UI (「Web UI: ユーザーまたはグループの追加」を参照)
- コマンドライン (「コマンドライン: ユーザーまたはグループの追加」を参照)

IdM では、ユーザーグループの作成時にカスタム GID を指定できます。これを行う場合は、ID の競合を避けるように注意してください。「ID 値が一意であることの確認」を参照してください。カスタム GID を指定しない場合、IdM は使用可能な ID 範囲から GID を自動的に割り当てます。

グループを削除するには、以下のツールを使用できます。

- Web UI (「Web UI: ユーザーまたはグループの削除」を参照)
- コマンドラインは、「コマンドライン: ユーザーまたはグループの削除」を参照してください。

グループを削除しても、IdM からグループメンバーは削除されません。

Web UI: ユーザーまたはグループの追加

1. **Identity → Groups** をクリックし、左側のサイドバーで **User Groups** または **Host Groups** を選択します。
2. **Add** をクリックして、グループを追加します。
3. グループの情報を入力します。

   ユーザーグループのタイプの詳細は、「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** コマンドを使用します。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。別の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。

- **--nonposix** は、非 POSIX グループを作成します。
- **--external** は、外部グループを作成します。

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: ユーザーまたはグループの削除

1. **Identity → Groups** をクリックし、左側のサイドバーで **User Groups** または **Host Groups** を選択します。
2. 削除するグループを選択し、**Delete** をクリックします。

コマンドライン: ユーザーまたはグループの削除

1. 管理者としてログインします。

```
$ kinit admin
```

2. **ipa group-del *group_name*** コマンドを使用してユーザーグループを削除します。**ipa hostgroup-del *group_name*** コマンドを使用してホストグループを削除します。

```
$ 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と削除

ユーザーグループにメンバーを追加するには、以下のツールを使用できます。

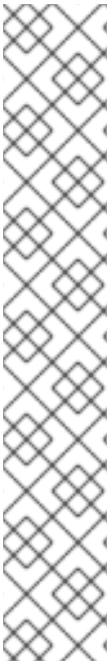
- IdM Web UI(「Web UI: ユーザーまたはグループへのメンバーの追加」を参照)
- コマンドライン (「コマンドライン: ユーザーグループへのメンバーの追加」を参照)

重要

別のユーザーグループをメンバーとして追加する場合は、再帰グループを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 のメンバーである場合は、グループ B をグループ A のメンバーとして追加しないでください。再帰的なグループにより予期しない動作が発生する可能性があります。

ユーザーグループからメンバーを削除するには、以下のツールを使用できます。

- IdM Web UI(「Web UI: ユーザーグループからのメンバーの削除」を参照)
- 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)



注記

ユーザーグループまたはホストグループにメンバーを追加した後、更新が Identity Management 環境のすべてのクライアントに広がるまでに時間がかかる場合があります。これは、特定のホストがユーザー、グループ、またはネットグループを解決するときに、**System Security Services Daemon** (SSSD) が最初にキャッシュを調べて、サーバーで不足または期限切れのレコードのみを検索するためです。

ホストグループに適用された変更を直ちに表示するには、キャッシュパージユーティリティー **sss_cache** を使用して、ホストの **SSSD** キャッシュを更新します。**sss_cache** を使用して、ホストグループの **SSSD** キャッシュ内の現在のレコードを無効化すると、**SSSD** キャッシュがアイデンティティープロバイダーから更新されたレコードを取得するように強制するため、変更はすぐに実現できます。

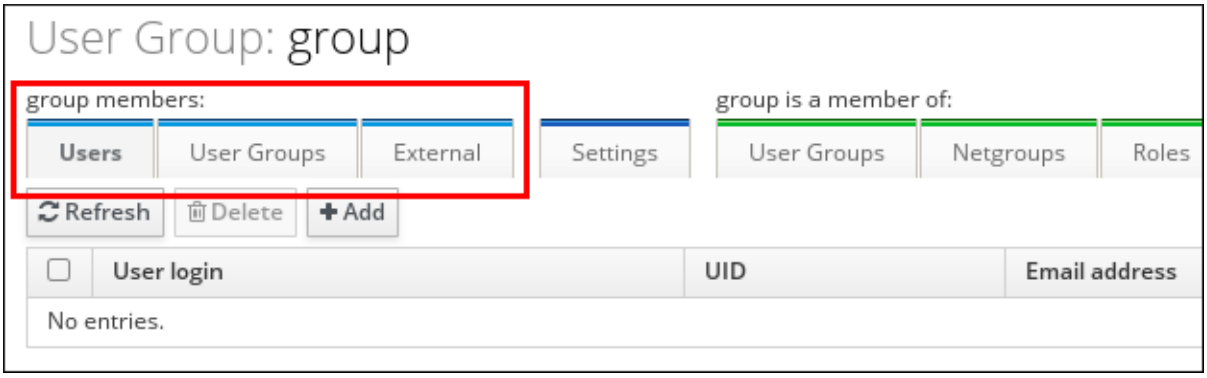
SSSD キャッシュでホストグループエントリーを消去するには、次のコマンドを実行します。

```
# sss_cache -n host_group_name
```

Web UI: ユーザーまたはグループへのメンバーの追加

1. **Identity → Groups** をクリックし、左側のサイドバーで **User Groups** または **Host Groups** を選択します。
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。たとえば、ユーザーグループの場合は、**ユーザー**、**ユーザーグループ**、**外部** などです。

   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: ユーザーグループへのメンバーの追加

1. **オプション:ipa group-find** コマンドまたは **ipa hostgroup-find** コマンドを使用して、グループを検索します。
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** コマンドを使用します。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際は、以下のオプションを使用してメンバーを指定します。

- **--users** は、IdM ユーザーを追加します
- **--external** は、**DOMAIN\user_name** 形式または **user_name@domain** 形式で、IdM ドメイン外に存在するユーザーを追加します
- **--groups** は、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際は、以下のオプションを使用してメンバーを指定します。

- **--hosts** は、IdM ホストを追加します
- **--groups** は、IdM ホストグループを追加します。

例13.4 ユーザーグループにメンバーを追加するコマンドの例

user1、*user2*、および *group1* を *group_name* という名前のグループに追加するには、次のコマンドを実行します。

```
$ ipa group-add-member group_name --users=user1 --users=user2 --groups=group1
```

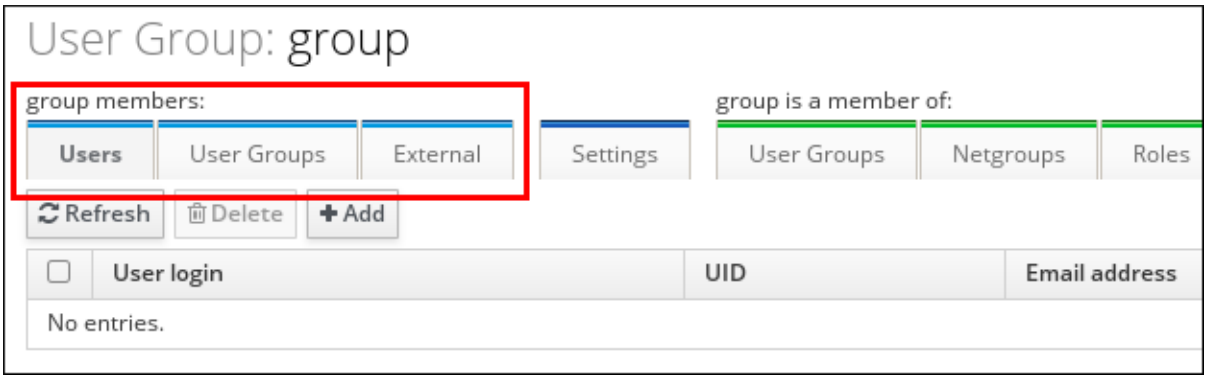
ad_domain という名前のドメインから *group_name* という名前のグループに *ad_user* を追加するには、外部ユーザーの指定方法を選択できます。以下に例を示します。

```
$ ipa group-add-member group_name --external='AD_DOMAIN\ad_user'
$ ipa group-add-member group_name --external='ad_user@AD_DOMAIN'
$ ipa group-add-member group_name --
external='ad_user@AD_DOMAIN.EXAMPLE.COM'
```

Web UI: ユーザーグループからのメンバーの削除

1. **Identity → Groups** をクリックし、左側のサイドバーで **User Groups** または **Host Groups** を選択します。
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。たとえば、ユーザーグループの場合は、**ユーザー**、**ユーザーグループ**、**外部** などです。

図13.4 ユーザーグループメンバーの削除



4. 必要な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、削除するメンバーがグループに含まれていることを確認します。

2. ユーザーグループメンバーを削除するには、**ipa group-remove-member** コマンドを使用します。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ときは、以下のオプションを使用してメンバーを指定します。

   - **--users** は、IdM ユーザーを削除します
   - **--external** は、**DOMAIN\user_name** または **user_name@domain** の形式で、IdM ドメイン外に存在するユーザーを削除します
   - **--groups** は、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ときは、以下のオプションを使用してメンバーを指定します。

   - **--hosts** は、IdM ホストを削除します
   - **--groups** は、IdM ホストグループを削除します。

   たとえば、*group_name* という名前のグループから、*user1*、*user2*、および *group1* を削除するには、次のコマンドを実行します。

   ```
   $ 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 が新規ユーザー用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のないユーザーの作成」
- 「すべてのユーザーに対してユーザープライベートグループをグローバルに無効にする」

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した後でも、IdM には新しいユーザーの追加時に GID が必要になります。ユーザーの追加が正常に行われたことを確認するには、「ユーザープライベートグループが無効であるユーザーの追加」 を参照してください。

注記

GID の競合が原因で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は、デフォルトの UID および GID の割り当て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のないユーザーの作成

ipa user-add コマンドに **--noprivate** オプションを追加します。コマンドを成功させるには、カスタムプライベートグループを指定する必要があることに注意してください。「ユーザープライベートグループが無効であるユーザーの追加」を参照してください。

13.4.2. すべてのユーザーに対してユーザープライベートグループをグローバルに無効にする

1. 管理者としてログインします。

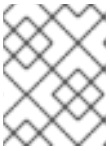
```
$ kinit admin
```

2. IdM は、Directory Server Managed Entries プラグイン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。プラグインのインスタンスをリスト表示します。

```
$ ipa-managed-entries --list
```

3. IdM が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、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

後で **UPG 定義** インスタンスを再有効にするには、**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が無効であるユーザーの追加

デフォルトのユーザープライベートグループの作成が無効になっているときに新しいユーザーの追加が成功することを確認するには、次のいずれかを選択します。

- 新しいユーザーを追加するときにカスタムの GID を指定します。GID は、既存のユーザーグループに対応す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、**--gid** オプションを **ipa user-add** コマンドに追加します。

- automember ルールを使用して、GID のある既存のグループにユーザーを追加します。「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指定のキーワードのエントリーを検索すると、IdM は特定の属性のみを検索します。以下に例を示します。

- ユーザー検索: 名、姓、ユーザー名 (ログイン ID)、役職、組織単位、電話番号、UID、メールアドレス。

- グループ検索: グループ名、説明。

以下の手順は、他の属性も検索するように IdM を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ことに注意してください。たとえば、ユーザー検索属性のリストから役職属性を削除しても、IdM は引き続きユーザータイトルを検索します。

前提条件
新しい属性を追加する前に、対応するインデックスがこの属性の LDAP ディレクトリーに存在することを確認します。ほとんどの標準的な LDAP 属性には LDAP にインデックスがありますが、カスタム属性を追加する場合は、インデックスを手動で作成する必要があります。『Red Hat Directory Server 10 管理ガイド』 の 標準インデックスの作成 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**にユーザー検索属性を設定します。
3. **Group Options** エリアで、**Group search fields**にグループ検索属性を設定します。
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
以下のオプションを指定して **ipa config-mod** コマンドを使用します。

- **--usersearch** は、ユーザーの検索属性の新しいリストを定義します。
- **--groupsearch** は、グループの検索属性の新しい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 で自動グループメンバーシップが機能する仕組み

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すると、属性に基づいてユーザーとホストをグループに自動的に割り当てることができます。たとえば、以下を行うことができます。

- 従業員のマネージャー、ロケーション、またはその他の属性に基づいて従業員のユーザーエントリーをグループに分類する。
- クラス、ロケーション、またはその他の属性に基づいてホストを分類する。
- 全ユーザーまたは全ホストを 1 つのグローバルグループに追加する。

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を手動で管理す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なくなりました。

ユーザーおよびホスト管理における一貫性の向上

自動グループメンバーシップでは、ユーザーとホストは、厳密に定義され自動評価された基準をもとにグループに割り当てられます。

グループベースの設定の容易な管理

さまざまな設定がグループに定義され、**sudo** ルール、**automount**、またはアクセス制御などの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シップを使用すると、ユーザーとホストは指定されたグループに自動的に追加されます。これにより、グループベースの設定の管理が容易になります。

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場合、管理者は *automember ルール* を定義します。automember ルールは、特定のユーザーまたはホストグループに適用されます。これには、グループに含める/除外するためにユーザーまたはホストが満たす必要がある *条件* が含まれます。

包含条件

ユーザーまたはホストのエントリーが包含条件を満たす場合は、グループに含まれます。

除外条件

ユーザーまたはホストのエントリーが除外された状態を満たす場合は、グループには **含まれません**。

この条件は、Perl 互換正規表現 (PCRE) 形式の正規表現として指定します。PCRE の詳細は、pcresyntax(3) の man ページを参照してください。

IdM は、包含条件よりも除外条件を先に評価します。競合が発生した場合は、包含条件よりも除外条件が優先されます。

13.6.2. automember ルールの追加

automember ルールを追加するには、以下のツールを使用します。

- IdM Web UI(「Web UI: automember ルールの追加」を参照)
- コマンドライン (「コマンドライン: automember ルールの追加」を参照)

automember ルールを追加すると、以下のようになります。

- 今後作成されるすべてのエントリーは、指定されたグループのメンバーになります。エントリーが複数の automember ルールで指定された条件を満たすと、対応するグループすべてに追加されます。
- 既存のエントリーは、指定されたグループのメンバーには**なりません**。詳細は、「既存のユーザーおよびホストへの automember ルールの適用」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. 1 つ以上の包括的および排他的条件を定義します。詳細は「automember ルール」を参照してください。

   a. **Inclusive** セクションまたは **Exclusive** セクションで、**Add** をクリックします。

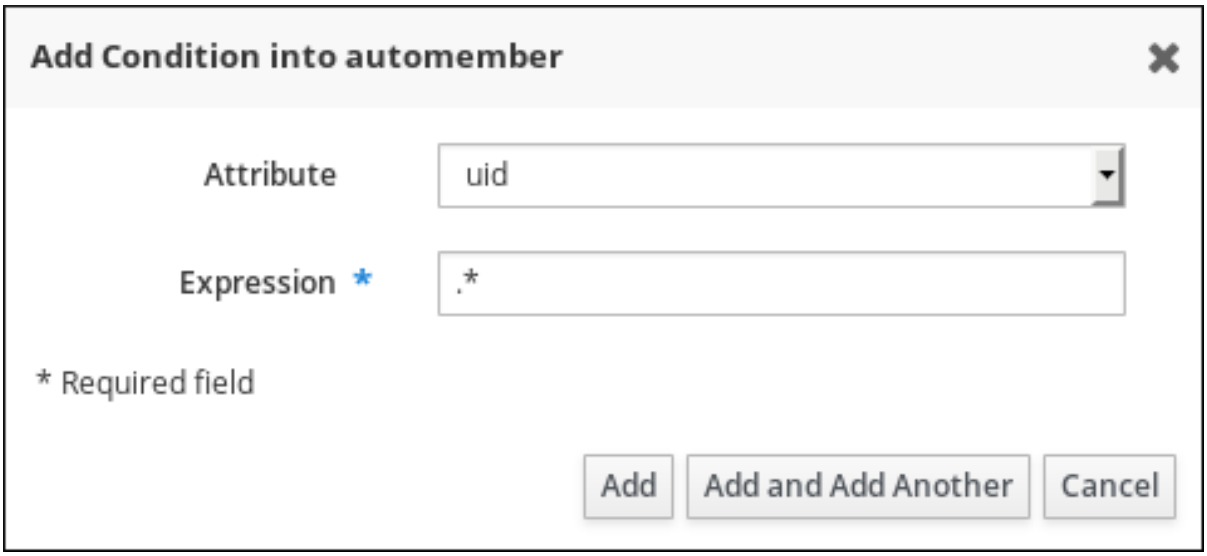
   b. **Attribute** フィールドで、必要な属性を選択します。

   c. **Expression** フィールドに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、ユーザーログイン属性 (**uid**) のすべての値 (**.***) を対象にしています。

   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。

   - **Automember rule**: 対象のグループ名と一致します。
   - **Grouping Type**: ルールのターゲットがユーザーグループか、ホストグループであるかを指定します。ユーザーグループをターゲットにするには、**group** を入力します。ホストグループをターゲットに設定するには、**ホストグループ** を入力します。

   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. 1 つ以上の包括的および排他的条件を定義します。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。

   - **Automember rule**: 対象のグループ名と一致します。
   - **属性キー**: フィルターが適用されるエントリー属性を指定します。たとえば、ユーザーの **manager** です。
   - **Grouping Type**: ルールのターゲットがユーザーグループか、ホストグループであるかを指定します。ユーザーグループをターゲットにするには、**group** を入力します。ホストグループをターゲットに設定するには、**ホストグループ** を入力します。
   - **包含正規表現** と **除外正規表現**:1 つ以上の条件を正規表現として指定します。ある条件のみを指定する場合は、他の条件が求められたら **Enter** を押します。

   たとえば、以下の条件は、ユーザーログイン属性 (**uid**) のすべての値 (**.***) を対象にしています。

```
$ 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13.5 コマンドライン: すべてのエントリーを 1 つのグループに追加する automember ルールの作成

cn または **fqdn** など、すべてのユーザーまたはホストエントリーに含まれる属性に包含条件を作成すると、今後作成されるすべてのユーザーまたはホストが 1 つのグループに追加されるようになります。

1. **all_hosts** という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と削除」を参照してください。
2. 新規ホストグループの automember ルールを追加します。以下に例を示します。

```
$ ipa automember-add
Automember Rule: all_hosts
Grouping Type: hostgroup
-----------------------------------
Added automember rule "all_hosts"
-----------------------------------
  Automember Rule: all_hosts
```

3. すべてのホストを対象とした包含条件を追加します。以下の例では、包含条件は**fqdn** 属性に任意の値 (.*) を持つホストを対象にします。

```
$ 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
---------------------------
```

今後追加されるすべてのホストは、自動的に **all_hosts** グループのメンバーになります。

例13.6 コマンドライン: 同期 AD ユーザーの automember ルールの作成

Active Directory(AD) から同期した Windows ユーザーは、**ntUser** オブジェクトクラスを共有します。**objectclass** 属性で **ntUser** が指定されたすべてのユーザーを対象とした automember 条件を作成すると、今後作成されたすべての同期 AD ユーザーが AD ユーザーの共通グループに含ま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と削除」を参照してください。

2. 新規ユーザーグループの automember ルールを追加します。以下に例を示します。

```
$ ipa automember-add
Automember Rule: ad_users
Grouping Type: group
-------------------------------------
Added automember rule "ad_users"
-------------------------------------
  Automember Rule: ad_users
```

3. AD ユーザーをフィルターする包含条件を追加します。以下の例では、包含条件は、**objectclass**属性に **ntUser** の値を持つすべての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
----------------------------
```

今後追加されるすべての AD ユーザーは、自動的に **ad_users** ユーザーグループのメンバーになります。

13.6.3. 既存のユーザーおよびホストへの automember ルールの適用

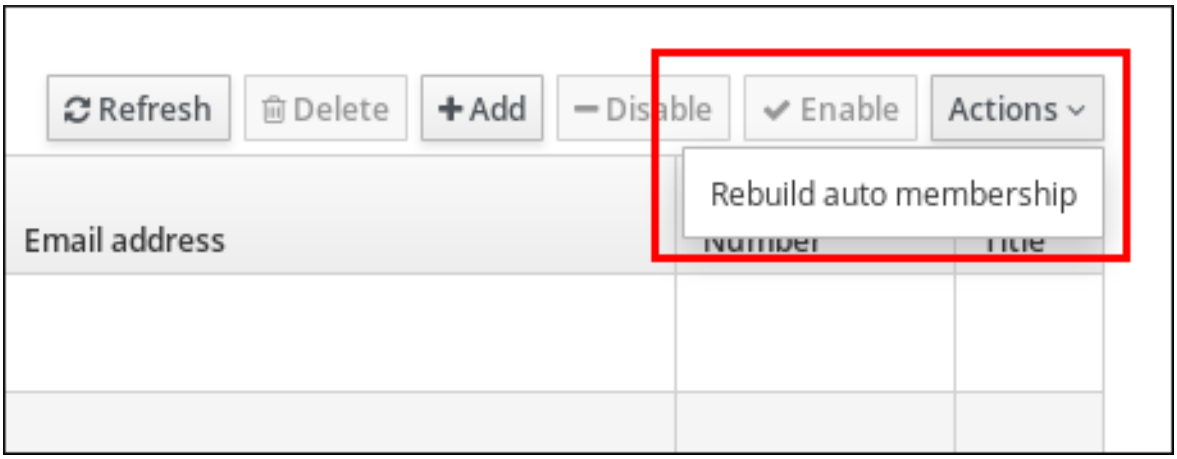
Automember ルールは、ルールの追加後に作成されたユーザーおよびホストエントリーに自動的に適用されます。ルールの追加前に存在するエントリーには、遡及的に適用されません。

ルールを追加する前に存在したエントリーに automember 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ます。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、すべてのエントリーまたは特定のエントリーに適用されます。

Web UI: 既存のエントリーの自動メンバーシップの再構築
全ユーザーまたは全ホストに対して自動メンバーシップを再構築するには、以下の手順を実施します。

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。

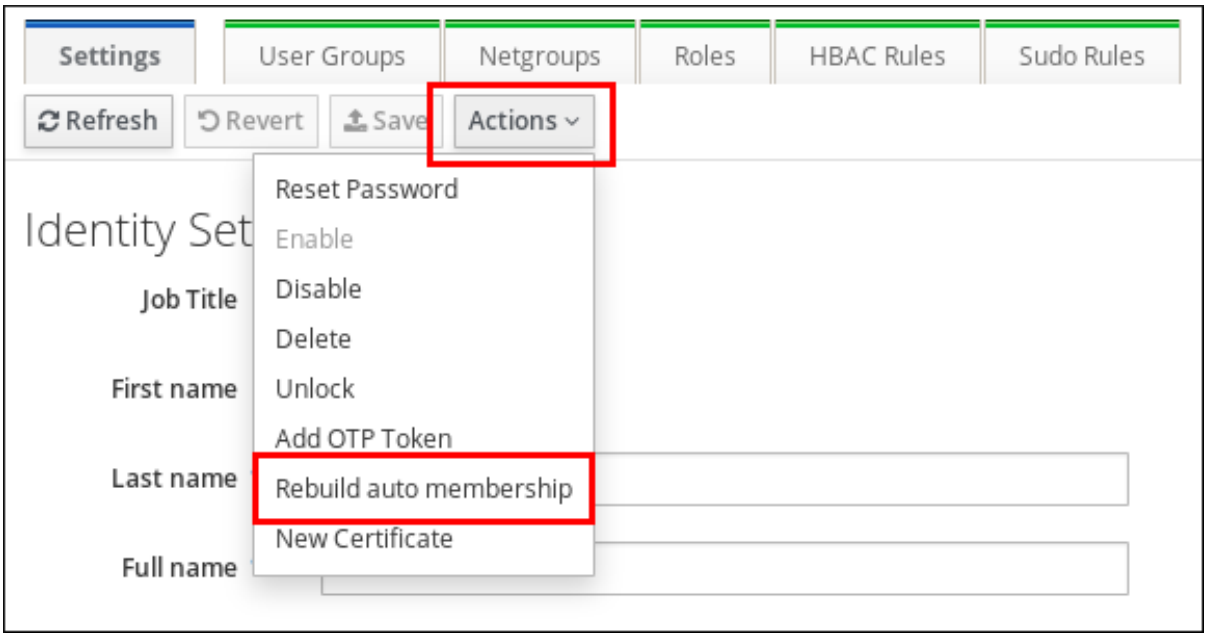
図13.6 全ユーザーまたは全ホストに対して自動メンバーシップの再構築



1 つのユーザーまたはホストに対して自動メンバーシップを再構築するには、以下を行います。

1. **Identity → Users** または **Hosts** を選択し、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 1つのユーザーまたはホストに対して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。

指定した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.
---------------------------------------------------------
```

指定した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 グループの設定

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ーは自動的にデフォルト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。
   - ターゲットグループ名を指定する **Default (fallback) Group**。

- **グルーピングタイプ**。ターゲットがユーザーグループか、ホストグループであるかを指定します。ユーザーグループをターゲットにするには、**group** を入力します。ホストグループをターゲットに設定するには、**ホストグループ** を入力します。

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。コマンドは、現在のデフォルトの automember グループを表示します。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) およびグループ 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 つの組織で異なる複数のドメインが使用される場合は、IdM ドメイン全体で UID および GID が一意になる必要があります。

14.1. ID 範囲

UID および GID 番号は *ID 範囲* に分けられます。個別のサーバーとレプリカに別々の数値範囲を指定することで、エントリーに対して発行された ID 値が別のサーバーまたはレプリカの別のエントリーですでに使用されている可能性が最小限に抑えられます。

ドメインのバックエンド 389 Directory Server インスタンスの一部として、分散型数値割り当て (DNA) プラグインにより、範囲が更新されサーバーとレプリカ間で更新されるようになります。プラグインはすべてのマスターとレプリカで ID 範囲を管理します。すべてのサーバーまたはレプリカには、現在の ID 範囲、および、現在の範囲を使い果たした後にサーバーまたはレプリカが使用する追加の **next** ID 範囲があります。DNA Directory Server プラグインの詳細は、Red Hat Directory Server デプロイメントガイド 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時に、**ipa-server-install** コマンドは、デフォルトでランダムな現在の ID 範囲をインストールされるサーバーに自動的に割り当てます。セットアップスクリプトは、合計 10,000 の可能な範囲から 200,000 の ID の範囲をランダムに選択します。このようにランダムな範囲を選択すると、今後別の 2 つの IdM ドメインを統合する場合に、ID の競合が発生する可能性を大幅に削減できます。

ただし、**ipa-server-install** で以下の 2 つのオプションを使用することで、サーバーのインストール時に現在の ID 範囲を手動で定義できます。

- **--idstart** は、UID および GID 番号の開始値を提供します。デフォルトでは、この値はランダムに選択されます。
- **--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。

IdM サーバーを 1 つインストールした場合、新しいユーザーまたはグループエントリーは、範囲全体からランダムな ID を受け取ります。新規レプリカをインストールし、レプリカが独自の ID 範囲を要求すると、サーバーの初期 ID 範囲が分割され、サーバーとレプリカの間で分散されます。レプリカは、初期マスターで使用可能な ID 範囲の残りの半分を受け取ります。次に、サーバーとレプリカは、新規エントリーに元の ID 範囲の対応する部分を使用します。また、レプリカに割り当てられた ID 範囲の残りの ID が 100 未満になると (つまり、レプリカが割り当てられた ID 範囲を使い果たした状態に近づくと)、レプリカは使用可能な他のサーバーに問い合わせて新しい ID 範囲を要求します。

サーバーは、DNA プラグインの初回使用時の ID 範囲を受け取ります。それまでは、サーバーには ID 範囲が定義されていません。たとえば、マスターサーバーからレプリカを作成すると、レプリカは ID 範囲をすぐに受信しません。レプリカは、最初の ID がレプリカ上で割り当てられる場合にのみ、最初のマスターから ID 範囲を要求します。

注記

レプリカが ID 範囲を要求する前に最初のマスターが機能しなくなると、レプリカはマスターに問い合わせて ID 範囲を要求することができません。レプリカに新しいユーザーを追加しようとすると失敗します。このような場合は、無効になったマスターに割り当てられている ID 範囲を確認し、ID 範囲を手動でレプリカに割り当てます (「手動 ID 範囲の拡張および新規 ID 範囲の割り当て」 を参照)。

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 (サーバーを指定した場合は指定されたサーバーでのみ) に設定されている現在の ID 範囲を表示します。以下に例を示します。

```
# 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、全サーバーに現在設定されている次の ID 範囲を表示します。サーバーを指定した場合は、指定されたサーバー上でのみ表示されます。以下に例を示します。

```
# 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
```

この 2 つのコマンドの詳細は、ipa-replica-manage(1) の man ページを参照してください。

14.4. レプリカの削除後の自動 ID 範囲拡張

機能しているレプリカを削除すると、**ipa-replica-manage del** コマンドは、そのレプリカに割り当てられた ID 範囲を取得して、次の範囲として利用可能な別の IdM レプリカに追加します。これにより、ID 範囲は他のレプリカで引き続き使用できる状態のままになります。

レプリカを削除したら、「現在割り当てられている ID 範囲の表示」 で説明されている **ipa-replica-manage dnarange-show** および **ipa-replica-manage dnanextrange-show** コマンドを使用して、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。

14.5. 手動 ID 範囲の拡張および新規 ID 範囲の割り当て

特定の状況では、ID 範囲を手動で調整する必要があります。

割り当てられた ID 範囲を使い果たした。

レプリカに割り当てられた ID 範囲を使い果たし、他のレプリカの IdM 範囲で使用可能な空き ID が

なくなったため、追加の ID の要求に失敗しました。レプリカに割り当てられた ID 範囲を拡張します。これには、既存の ID 範囲の分割や、サーバー用に最初に設定された ID 範囲の超過が伴う場合があります。または、新しい ID 範囲を割り当てることもできます。

注記

新しい ID 範囲を割り当てると、サーバーまたはレプリカ上の既存のエントリーの UID は同じになります。現在の ID 範囲を変更しても、IdM は過去に割り当てられた範囲の記録を保持するため、これにより問題が発生することはありません。

レプリカが機能しなくなる

レプリカが停止してしまい、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ないので、以前に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。ID 範囲を復元し、他のレプリカで使用できるようにします。

機能しなくなったサーバーに属する ID 範囲を回復し、別のサーバーに割り当てる場合は、最初に「現在割り当てられている ID 範囲の表示」に記載されている **ipa-replica-manage dnarange-show** コマンドを使用して ID 範囲の値を確認し、その ID 範囲を手動でサーバーに割り当てます。また、UID や GID が重複しないように、回復した範囲からの ID の値がユーザーまたはグループに割り当てられていないことを確認します。これは、既存のユーザーおよびグループの UID と GID を調べて実行できます。

ID 範囲を手動で定義するには、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、指定したサーバーの現在の 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、指定したサーバーの次の ID 範囲を定義できます。

```
# ipa-replica-manage dnanextrange-set masterB.example.com 1001-5000
```

これらのコマンドの詳細は、ipa-replica-manage(1) の 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た ID 範囲のいずれかが重複すると、この 2 つのサーバーにより、異なるエントリーに同じ ID 値を割り当てる可能性があります。

UID の値が 1000 以下の ID 範囲は設定しないでください。1000 以下の値はシステム使用向けに予約されています。また、**0** 値が含まれる ID 範囲は設定しないでください。SSSD サービスは ID の値 **0** を処理しません。

ID 範囲を手動で拡張する場合は、新たに拡張した範囲が 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** コマンドを使用して確認できます。**ipa idrange-find -h** コマンドを実行して、**ipa idrange-find** を使用する方法のヘルプを表示します。

14.6. ID 値が一意であることの確認

UID または GID が競合しないようにすることが推奨されます。UID および GID は、常に一意である必要があります。2 人のユーザーに同じ UID を割り当てることはできず、2 つのグループに同じ GID を割り当てることはできません。

自動 ID 割り当て

対話的に、あるいは ID 番号を手動で指定せずにユーザーまたはグループが作成される場合、サーバーは ID 範囲から次に利用可能な ID 番号をユーザーアカウントに割り当てます。これにより、UID または GID が常に一意になります。

手動 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と、サーバーは指定の UID または GID が一意であることを検証しません。別のエントリーですでに使用されている値を選択した場合は、競合を警告しません。

「変更された UID および GID 番号の修復」 で説明されているように、SSSD サービスは、同一の ID を持つエントリーを処理しません。2 つのエントリーが同じ ID 番号を共有している場合は、この ID の検索は最初のエントリーのみを返します。ただし、別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 はいずれも同じ ID 範囲から選択されます。ユーザーおよびグループには同じ ID を指定できます。UID と GID は **uidNumber** と **gidNumber** の 2 つの異なる属性に設定されているため、このような状況では競合は発生しません。

注記

ユーザーおよびグループの両方に同じ ID を設定すると、ユーザープライベートグループを設定できます。このようにユーザー用に一意のシステムグループを作成するには、ユーザーとグループに同じ ID 値を設定します。グループの唯一のメンバーはこのユーザーです。

14.7. 変更された UID および GID 番号の修復

ユーザーが IdM システムにログインすると、そのシステムの SSSD は、ユーザー名とユーザーの UID および GID をキャッシュします。SSSD は、次に UID をユーザーの識別キーとして使用します。同じユーザー名で、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と、SSSD は 2 つの異なる UID を登録し、ユーザー名が競合している 2 つの異なるユーザーがいると仮定します。これにより、ユーザーの UID が変更された場合に問題が発生する可能性があります。このような場合、SSSD は、異なる UID を持つ同じユーザーとして認識するのではなく、変更した UID を持つユーザーを新規ユーザーと誤って解釈します。既存のユーザーの UID が変更される場合は、SSSD と関連するサービスおよびドメインにログインできません。これは、ID の情報に SSSD を使用するクライアントアプリケーションにも影響します。

この問題を回避するには、UID または GID が変更された場合に SSSD キャッシュを削除します。これにより、ユーザーは再度ログインできるようになります。たとえば、指定したユーザーの SSSD キャッシュを削除するには、以下のように **sss_cache** ユーティリティーを使用します。

```
[root@server ~]# sss_cache -u user
```

第15章 ユーザーおよびグループスキーマ

ユーザーエントリー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 属性を使用して、情報がディレクトリーに保存されます。(この詳細は、『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	説明
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 エントリー (テンプレート) のオブジェクトクラス

ユーザーエントリー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使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、任意またはデフォルト[a]
User login	**username**	必須
名	**--first**	必須
姓	**--last**	必須
名前	**--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、任意またはデフォルト[a]
Initials	**--initials**	デフォルト
Home directory	**--homedir**	デフォルト
GECOS field	**--gecos**	デフォルト
シェル	**--shell**	デフォルト
Kerberos プリンシパル	**--principal**	デフォルト
メールアドレス	**--email**	オプション
Password	**--password** [b]	任意
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 番号	**--fax**	オプション
組織単位	**--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、任意またはデフォルト[a]
SSH キー	**--sshpubkey**	オプション
Additional attributes	**--addattr**	オプション
部門番号	**--departmentnumber**	オプション
従業員番号	**--employeenumber**	オプション
従業員のタイプ	**--employeetype**	オプション
希望の言語	**--preferredlanguage**	オプション

[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。

[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。

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 エントリーで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リスト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リスト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

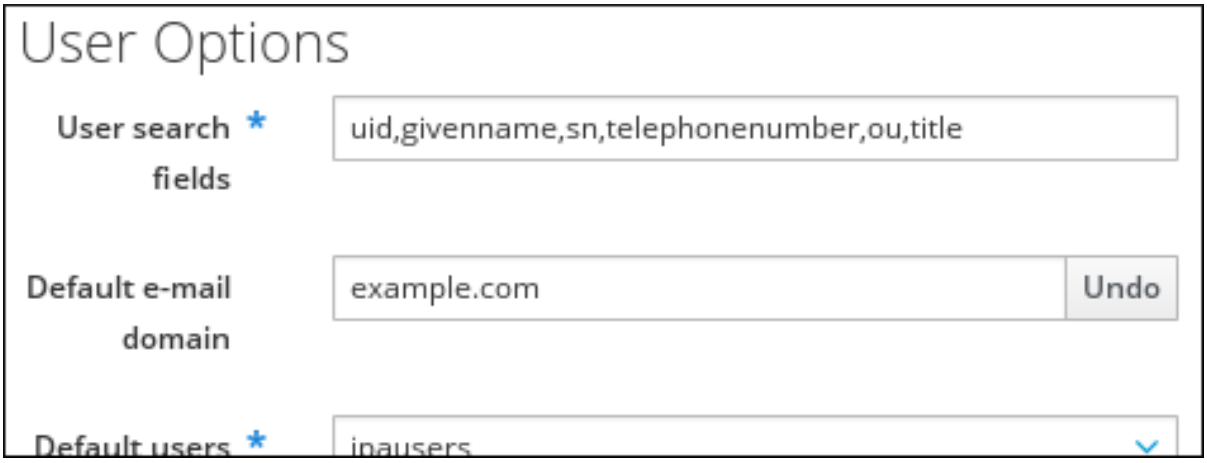
ユーザーおよびグループアカウントは、エントリーに適用する定義済みの LDAP オブジェクトクラスとともに作成されます。オブジェクトクラスに属する属性は、ユーザーエントリーに追加することができます。

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デプロイメントのシナリオに対応していますが、管理者はカスタマイズ属性を指定した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のリストを変更した後に、新しい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、古いエントリーは自動的に変更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. カスタムスキーマ要素をすべて、Identity Management が使用する 389 Directory Server インスタンスに追加します。スキーマ要素の追加については、Directory Server 管理者ガイドのスキーマの章 で説明します。
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。

   図15.1 サーバー設定のユーザーオプション

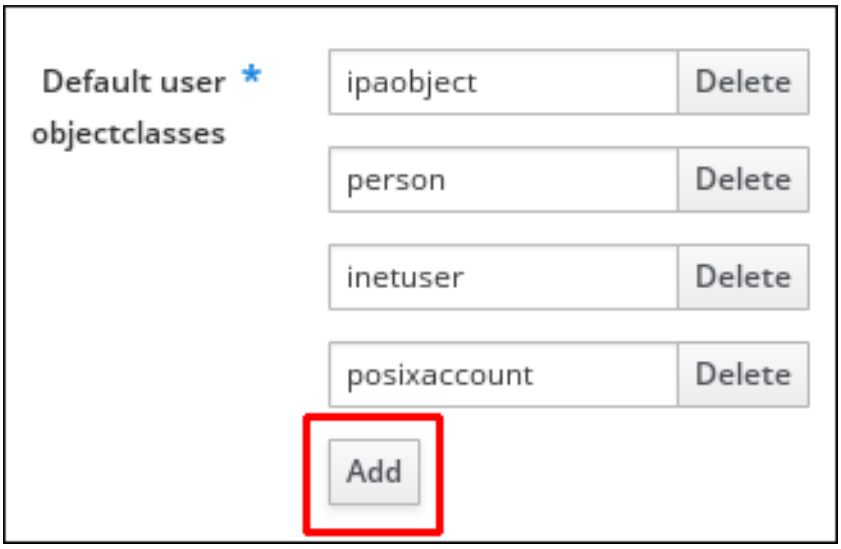
   

5. ユーザーエリア下部で、**Add** をクリックして、別のオブジェクトクラスの新規フィールドを追加します。

重要

設定の更新時は、常に**既存**のデフォルトオブジェクトクラスを追加してください。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とオブジェクトクラス違反で失敗することになります。

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

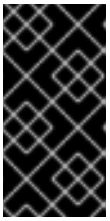


6. 変更が完了したら、**Configuration** ページ上部の **Save** をクリックします。

15.2.2. コマンドラインでの操作

1. カスタムスキーマ要素をすべて、Identity Management が使用する 389 Directory Server インスタンスに追加します。スキーマ要素の追加については、Directory Server 管理者ガイドのスキーマの章 で説明します。

2. エントリーに追加するオブジェクトクラスリスト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

設定の更新時は、常に**既存**のデフォルトオブジェクトクラスを追加してください。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とオブジェクトクラス違反で失敗することになります。

すべてのオブジェクトクラスは、オブジェクトクラスのリストに含める必要があります。**config-mod** コマンドで渡される情報は、以前の値を上書きします。これは、**--userobjectclasses** 引数を使用して各オブジェクトクラスを指定するか、**{attr1,attr2,attr3}** のように、中かっこ内にすべてのオブジェクトクラスのコンマ区切りリスト (スペースなし) を指定することで実行できます。特に、長いリストで、複数のオプションよりも中括弧を簡単に使用できます。以下に例を示します。

```
[bjensen@server ~]$ ipa config-mod --
userobjectclasses={top,person,organizationalperson,inetorgperson,inetuser,posixaccount,krbpr
incipalaux,krbticketpolicyaux,ipaobject,ipasshuser,employeeinfo}
```

注記

中かっこオプションを使用するには、**ブレース拡張機能** をオンにする必要があります。この機能を有効にするには、**set** コマンドを使用します。

```
# set -o braceexpand
```

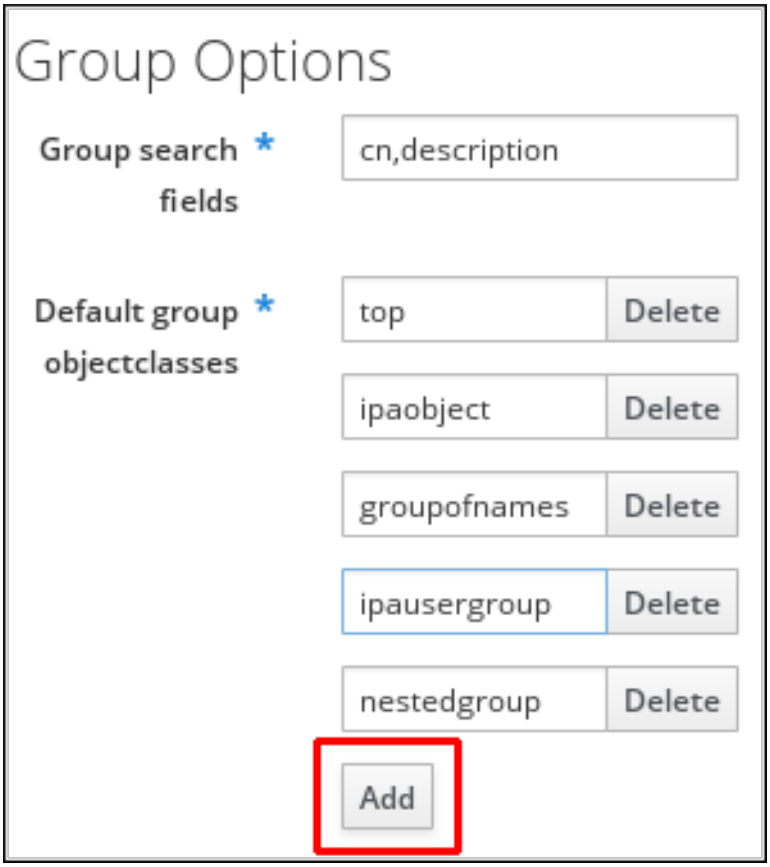
15.3. カスタムのオブジェクトクラスを新規グループエントリーに適用する

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属性を指定したカスタムのオブジェクトクラスを作成できます。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のリストを変更した後に、新しい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、古いエントリーは自動的に変更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. カスタムスキーマ要素をすべて、Identity Management が使用する 389 Directory Server インスタンスに追加します。スキーマ要素の追加については、Directory Server 管理者ガイドのスキーマの章 で説明します。
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に新規フィールドを追加します。

重要

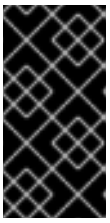
設定の更新時は、常に**既存**のデフォルトオブジェクトクラスを追加してください。これらを含めないと、現行設定は上書きされます。Identity Management で必須のオブジェクトクラスが含まれないと、これ以降にエントリーの追加を試みるとオブジェクトクラス違反で失敗することになります。

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. カスタムスキーマ要素をすべて、Identity Management が使用する 389 Directory Server インスタンスに追加します。スキーマ要素の追加については、Directory Server 管理者ガイドのスキーマの章 で説明します。

2. エントリーに追加するオブジェクトクラスリスト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

設定の更新時は、常に**既存**のデフォルトオブジェクトクラスを追加してください。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とオブジェクトクラス違反で失敗することになります。

すべてのオブジェクトクラスは、オブジェクトクラスのリストに含める必要があります。**config-mod** コマンドで渡される情報は、以前の値を上書きします。これは、**--groupobjectclasses** 引数を使用して各オブジェクトクラスを指定するか、**{attr1,attr2,attr3}** のように、中かっこ内にすべてのオブジェクトクラスのコンマ区切りリスト (スペースなし) を指定することで実行できます。特に、長いリストで、複数のオプションよりも中括弧を簡単に使用できます。以下に例を示します。

```
[bjensen@server ~]$ ipa config-mod --
groupobjectclasses={top,groupofnames,nestedgroup,ipausergroup,ipaobject,ipasshuser,empl
oyeegroup}
```

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時にテンプレートを使用します。

ユーザーの場合は、テンプレートは非常に特有です。Identity Management は、IdM ユーザーアカウントの複数のコア属性に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** コマンドを使用して変更できます。

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
ユーザー名の最大長	**--maxusername**	ユーザー名の最大長を設定します。デフォルト値は 32 です。

フィールド	コマンドラインオプション	説明
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。
検索時間の制限	**--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。
検索サイズ制限	**--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。
Certificate subject base		クライアント証明書用に発行先 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。

フィールド	コマンドラインオプション	説明
Default user object classes	**--userobjectclasses**	IdM ユーザーアカウントの作成に使用されるオブジェクトクラスを定義します。これは複数回呼び出すことができます。オブジェクトクラスのリストは、コマンドの実行時に上書きされるため、指定する必要があります。
Default group object classes	**--groupobjectclasses**	IdM グループアカウントの作成に使用されるオブジェクトクラスを定義します。これは複数回呼び出すことができます。オブジェクトクラスのリストは、コマンドの実行時に上書きされるため、指定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。

図15.4 検索での制限設定

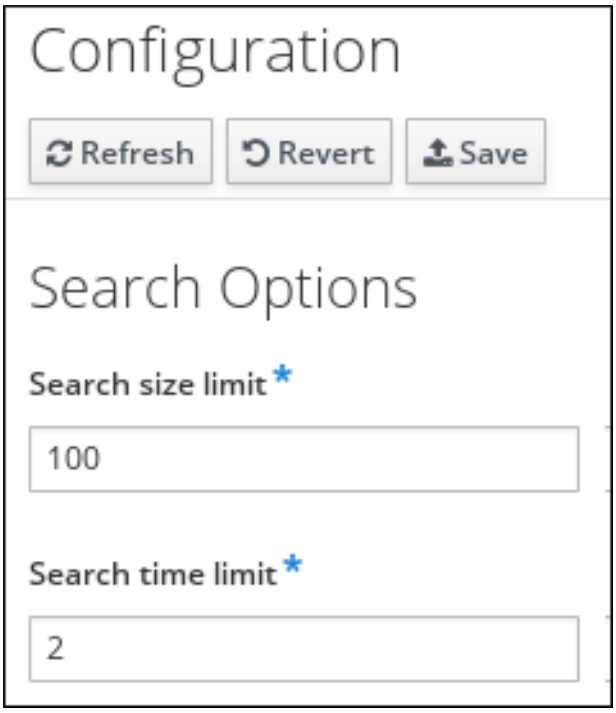


図15.5 user 属性

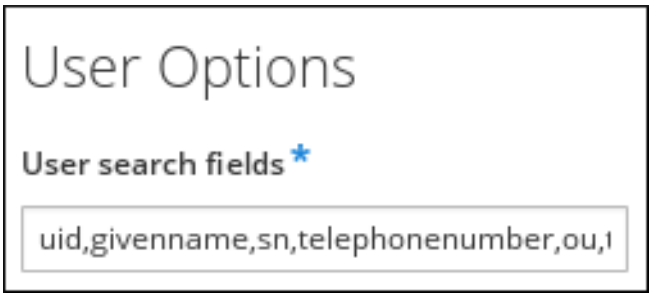
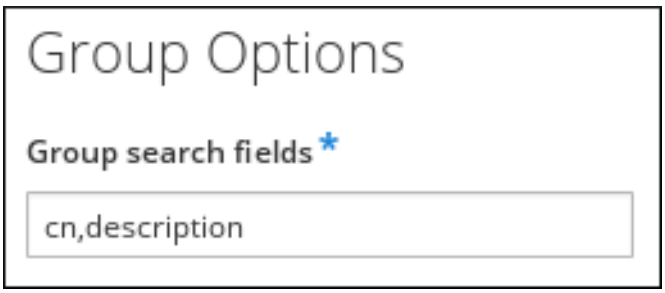


図15.6 グループ属性



15.4.2. コマンドラインでの属性表示

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、**--all** オプションを使用すると設定すべてが表示されます。

```
[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) を保存できるサービスは、IdM サービスとして設定できます。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 証明書やキータブ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。)

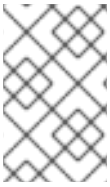
IdM ドメインは、共通の ID 情報、共通ポリシー、および共有サービスを使用して、マシン間で共通性を確立します。ドメインのクライアントとしてのドメイン機能に属するマシンです。これは、ドメインが提供するサービスを使用することを意味します。(1章*Red Hat Identity Management の概要* で説明されているように) IdM ドメインは、マシン専用の 3 つの主要サービスを提供します。

- DNS
- Kerberos
- 証明書管理

16.1. サービスエントリーおよびキータブの追加と編集

ホストエントリーの場合と同様に、ホストのサービスエントリー (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ーを作成し、次にそのサービスがドメインへのアクセスに使用するキータブを作成します。

デフォルトでは、Identity Management は **/etc/httpd/conf/ipa.keytab** に HTTP キータブに保存します。



注記

このキータブは、Web UI に使用します。キーが **ipa.keytab** に保存され、そのキータブファイルが削除された場合には、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 に対応させる必要のある各サービスで、同様の場所を指定できます。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キータブを含めるべきではありません。各サービスはキータブを特定の場所に保存し、そのサービスのみがキータブにアクセスできるようにアクセス権限 (場合によっては追加で SELinux ルール) を設定します。

16.1.1. Web UI でのサービスとキータブの追加

1. **Identity** タブを開き、**Services** サブタブを選択します。
2. サービスリストの上部にある **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設定します。
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、サービスプリンシパルの新規キータブを生成して割り当てます。

```
[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。別のレルムは指定できません。
- 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。**--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** 引数を指定すると、暗号化タイプのリストをキータブに追加できます。これは、デフォルトの暗号化タイプより優先されます。エントリーのリストは、同じコマンド呼び出しでオプションを複数回使用するか、**--option={val1,val2,val3}**のように、中括弧内のコンマ区切りリストにオプションをリストすることで設定できます。

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シークレットがリセットされます。つまり、そのプリンシパルの他のキータブ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キータブの追加

1. サービスプリンシパルを作成します。サービスは、**service/FQDN** などの名前で認識されます。

```
# ipa service-add serviceName/hostname
```

以下に例を示します。

```
$ 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** コマンドを使用して、サービスキータブ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できます。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。)

このコマンドには、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**) が必要です。キータブ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。

以下に例を示します。

```
# 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。別のレルムは指定できません。
- 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。**--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** 引数を指定すると、コンマ区切りの暗号化タイプのリストをキータブに追加できます。これは、デフォルトの暗号化タイプより優先されます。エントリーのリストは、同じコマンド呼び出しでオプションを複数回使用するか、**--option={val1,val2,val3}**のように、中括弧内のコンマ区切りリストにオプションをリストすることで設定できます。

警告

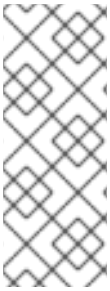
ipa-getkeytab コマンドは、指定されたプリンシパルのシークレットをリセットします。つまり、そのプリンシパルの他のキータブすべてが無効になります。

16.2. クラスターサービスの設定

IdM サーバーは、*クラスターに対応*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、クラスターサービスを IdM の一部として設定きます。

1. クラスター内の全ホストを 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、必要なキータブを生成します。
3. **/etc/krb5.keytab** にあるホストキータブなど、ホスト上のサービスに設定された全キータブを収集します。
4. **ktutil** コマンドを使用して、全キータブファイルのコンテンツを含む単一のキータブファイルを作成します。
   a. 各ファイルで **rkt** コマンドを使用してそのファイルからキーを読み取ります。
   b. 新規キータブ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. 各ホスト上のキータブファイルを新たに作成した結合キータブ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. サービスによっては、追加の設定を行い、障害のあるサービスから引き継いだ時にリセットされないクラスターのメンバーに対応する必要がある場合があります。
   - **sshd** の場合は、**/etc/ssh/sshd_config** に ***GSSAPIStrictAcceptorCheck no*** を設定します。

- **mod_auth_kerb** の場合には、**/etc/httpd/conf.d/auth_kerb.conf** に ***KrbServiceName Any*** を設定します。



注記

SSL サーバーの場合には、クライアントがクラスター化したホストに接続する時に、サーバー証明書の発行先名または代わりの発行先名が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発行先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** コマンドを使用して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. 複数サーバーの既存キータブの取得

クラスター環境など、一部のシナリオでは、異なるマシンによって 1 つの共通ホスト名で表されるサービスに同じキータブファイルが必要になります。IdM コマンドを使用して、各ホストで同じキータブを取得できます。

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します。

```
[root@ipaserver ~]# kinit admin
```

2. このホスト名を共有するすべての IP アドレスの共通の正引き 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 名用に新規ホストの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 からキータブを取得できる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. 1 つのホストに新規キータブを作成するパーミッションを付与します。

```
[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、以下の手順に従います。

1. ホストの Kerberos キータブで認証を行います。

```
# kinit -kt /etc/krb5.keytab
```

2. 1. 該当のパーミッションを付与したクライアントで、新規キータブを生成し、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. 他のすべてのクライアントで、**-r** オプションをコマンドに追加して、IdM サーバーから既存のキータブを取得します。

```
[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、新しいキータブが生成されることに注意してください。これは、このサービスプリンシパル用に以前に取得したキータブをすべて無効にします。

16.5. サービスエントリーの無効化および再有効化

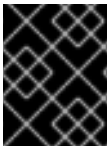
アクティブなサービスは、ドメイン内の他のサービスやホスト、ユーザーからアクセス可能です。アクティビティーからホストやサービスを削除する必要がある場合もあります。ただし、サービスやホストを削除するとエントリーやすべての関連する設定も永続的に削除されてしまいます。

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、ドメインユーザーは、サービスをドメインから完全に削除されない場合にはサービスにアクセスできなくなります。これは、**service-disable** コマンドを使用して実行できます。

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。以下に例を示します。

```
[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、そのホストが無効になるだけではありません。そのホストで設定されているすべてのサービスも無効にします。

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キータブを強制終了することになります。キータブを削除すると、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、キータブを要求する IdM サーバーを、**-p** はプリンシパル名を、**-k** はキータブを保存するファイルを指定します。

新規の HTTP キータブを要求する場合は、以下のようになります。

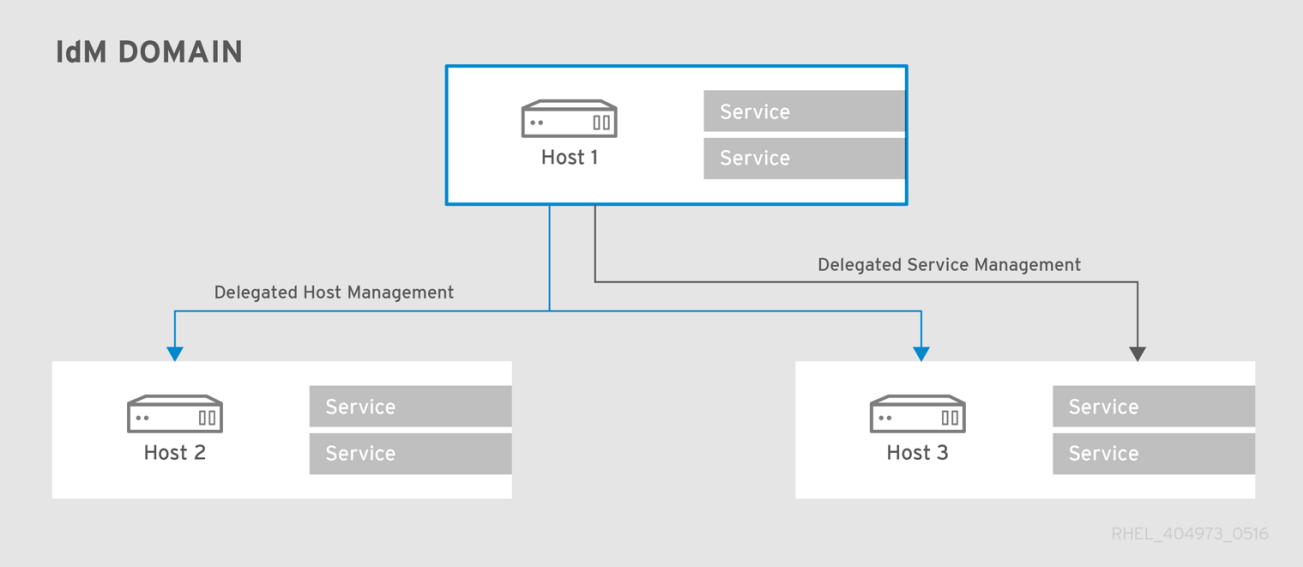
```
[root@ipaserver ~]# ipa-getkeytab -s ipaserver.example.com -p HTTP/server.example.com -k
/etc/httpd/conf/krb5.keytab -e aes256-cts
```

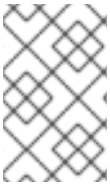
第17章 ホストおよびサービスへのアクセス委譲

本章のコンテキストで **管理する**とは、別のホストまたはサービスのキータブと証明書を取得できることを意味します。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、適切な委譲更新や、適切な ***managedby*** エントリーを指定して、ホストが他のホストや他のホスト上のサービスを管理できるようにすることも可能です。

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**委譲** 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譲できます。

図17.1 ホストおよびサービスの委譲





注記

managedBy エントリーで別のホストに権限が委譲されている場合に、そのホスト上の全サービスの管理を委譲されたわけではありません。委譲は個別に行われる必要があります。

17.1. サービス管理の委譲

service-add-host ユーティリティーを使用してサービスの制御をホストに委譲します。

```
# ipa service-add-host principal --hosts=hostname
```

サービスを委譲するには、2 つの部分があります。

- *principal* 引数を使用したプリンシパルの指定
- **--hosts**オプションを使用して、制御でホストを特定します。

以下に例を示します。

```
[root@server ~]# ipa service-add HTTP/web.example.com
[root@server ~]# ipa service-add-host HTTP/web.example.com --hosts=client1.example.com
```

ホストに権限が委譲されると、ホストプリンシパルを使用してサービスを管理できます。

```
[root@client1 ~]# kinit -kt /etc/krb5.keytab host/client1.example.com
[root@client1 ~]# ipa-getkeytab -s server.example.com -k /tmp/test.keytab -p
HTTP/web.example.com
Keytab successfully retrieved and stored in: /tmp/test.keytab
```

このサービスのチケットを作成するには、委譲された認証局を使用してホストで証明書要求を作成します。

```
[root@client1]# kinit -kt /etc/krb5.keytab host/client1.example.com
[root@client1]# openssl req -newkey rsa:2048 -subj '/CN=web.example.com/O=EXAMPLE.COM' -
keyout /etc/pki/tls/web.key -out /tmp/web.csr -nodes
Generating a 2048 bit RSA private key
.....+++
.....+++
Writing new private key to '/etc/pki/tls/private/web.key'
```

cert-request ユーティリティーを使用してサービスエントリーを作成し、認定情報を読み込みます。

```
[root@client1]# ipa cert-request --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
```

証明書要求の作成と **ipa cert-request** の使用方法は、「ユーザー、ホスト、またはサービスの新規証明書の要求」 を参照してください。

17.2. ホスト管理の委譲

他のホストへの権限の委譲は **host-add-managedby** ユーティリティーを使用します。これにより、**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、ホストが権限を委譲したホストのキータブを取得できるようになります。

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、以下は権限を *client2* **から** *client1* **に** 委譲します。

   ```
   [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ます。

   ```
   [root@client1 ~]# kinit -kt /etc/krb5.keytab host/client1.example.com
   ```

4. **client2** の keytab を取得します。

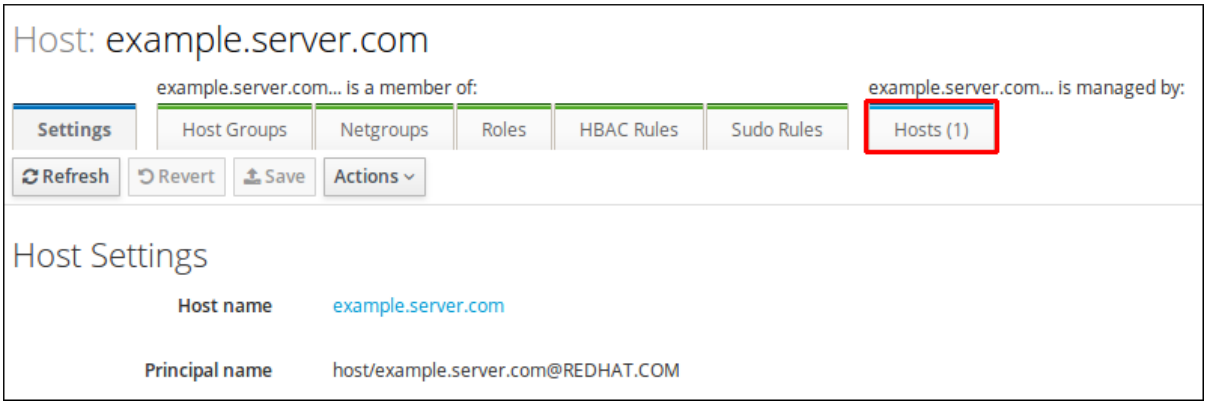
```
[root@client1 ~]# ipa-getkeytab -s server.example.com -k /tmp/client2.keytab -p
host/client2.example.com
Keytab successfully retrieved and stored in: /tmp/client2.keytab
```

17.3. WEB UI を使用したホストまたはサービス管理の委譲

IdM Web UI の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譲されているかを示す設定タブがあります。

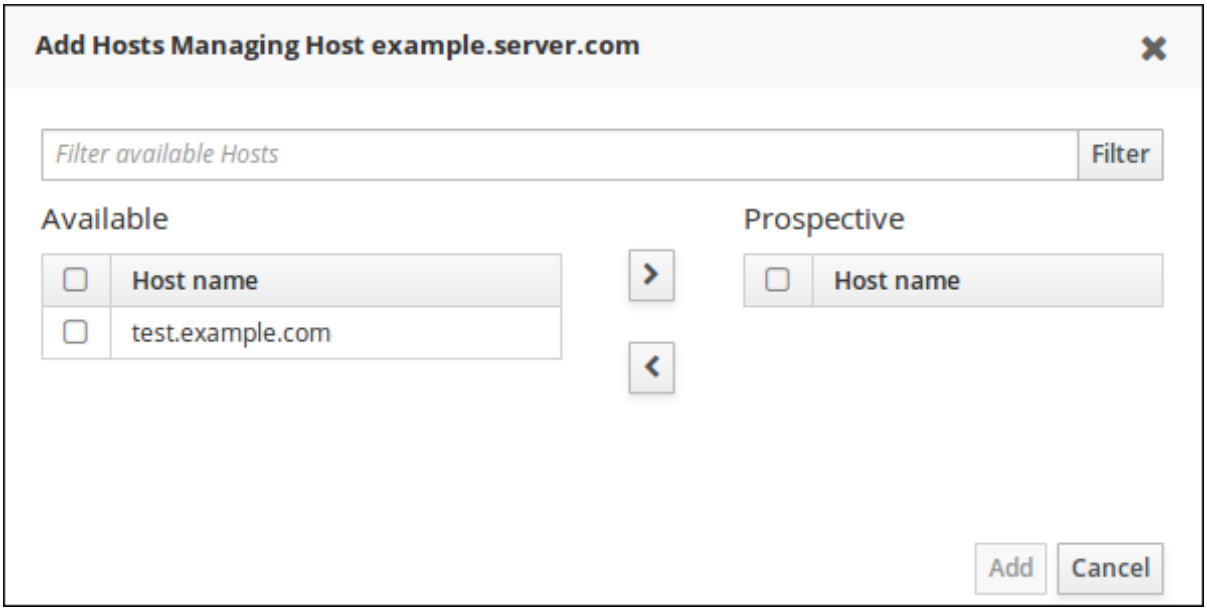
1. **Identity** タブを開き、**Hosts** または **Services** サブタブを選択します。
2. **委譲管理の付与先となる** ホストもしくはサービス名をクリックします。
3. ホストまたはサービスエントリーの右端にある **Hosts** サブタブをクリックします。このタブでは、選択したホストまたはサービスを **管理できる** ホストが表示されます。

図17.2 Host サブタブ



4. リスト上部にある **Add** をクリックします。
5. ホストまはサービスの管理を委譲する先のホスト名の横にあるチェックボックスをクリックします。右矢印 (>) をクリックし、ホストを選択ボックスに移動します。

図17.3 ホスト/サービス委譲の管理



6. **Add** をクリックして選択ボックスを閉じて、委譲設定を保存します。

17.4. 委譲サービスへのアクセス

サービスおよびホストの両方でクライアントに権限が委譲されている場合には、ローカルマシンでそのプリンシパルのキータブを取得できます。サービスの場合には、**service/hostname@REALM** という形式になります。ホストの場合には、**サービス** は **host** となります。

kinit では、**-k** オプションを指定してキータブを読み込み、**-t** オプションでキータブを指定します。以下に例を示します。

ホストにアクセスするには、以下を実行します。

```
[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 ユーザーまたはグループ属性に新しい値を指定でき、新しい値が適用されるクライアントホストを 1 つまたは複数定義できます。

たとえば、ID ビューを使用して以下を行うことができます。

- 環境ごとに異なる属性値を定義してください。「異なるホストのユーザーアカウントに対する異なる属性値の定義」を参照してください。
- 以前生成された属性の値を別の値に置き換えます。

重要

ID ビューは、IdM サーバーではなく、IdM クライアントにのみ適用できます。

SSSD パフォーマンスに対する潜在的な悪影響
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 がサーバー上でグループを検索するプロセスの最適化を防ぎます。

- 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だけを収集できます。

この負の影響は、多くの場合 SSSD のキャッシュが空の場合やキャッシュを消去した後に現れ、全エントリーが無効になります。

関連情報
ID ビューには、Active Directory に関連する環境でのいくつかのユースケースもあります。詳細は、『Windows Integration Guide』のMigrate from Synchronization to Trust Manually Using ID Views の章を参照してください。

18.1. ID ビューが上書きできる属性

ID ビューは、ユーザーおよびグループ ID のオーバーライドで構成されます。オーバーライドは、新しい属性値を定義します。

ユーザー ID およびグループ ID の上書きは、以下の属性の新しい値を定義できます。

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 公開鍵 (**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 VIEW コマンドのヘルプの取得

ID ビューおよびオーバーライドの管理に使用するコマンドをすべて表示するには、次のコマンドを実行します。

```
$ ipa help idviews
```

特定のコマンドの詳細なヘルプを表示するには、コマンドに **--help** オプションを追加します。

```
$ ipa idview-add --help
```

18.3. 異なるホストのユーザーアカウントに対する異なる属性値の定義

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 ビューを別のクライアントホストに適用することができます。例: サービスアカウントは、異なるホストで認証を行う際に異なる SSH 公開鍵を使用するように設定されます。

本セクションには、以下の手順が含まれています。

- 「Web UI: 特定ホストの属性値の上書き」
- 「コマンドライン: 特定ホストの属性値の上書き」

この手順では、**host1.example.com** という名前のクライアントホストの ID ビューを作成する方法を説明します。他のホストの属性値も上書きするには、手順を使用して、ホストごとに 1 つずつ、複数の ID ビューを作成します。

手順では、以下を前提としています。

- **user** は、属性を上書きする必要があるユーザーアカウントです
- **host1.example.com** は、ID ビューが適用されるホストです。

重要

新しい ID ビューを作成したら、ID ビューが適用されるすべてのクライアントで SSSD を再起動します。

新しい ID ビューが UID または GID を変更する場合は、これらのクライアントで SSSD キャッシュも消去します。

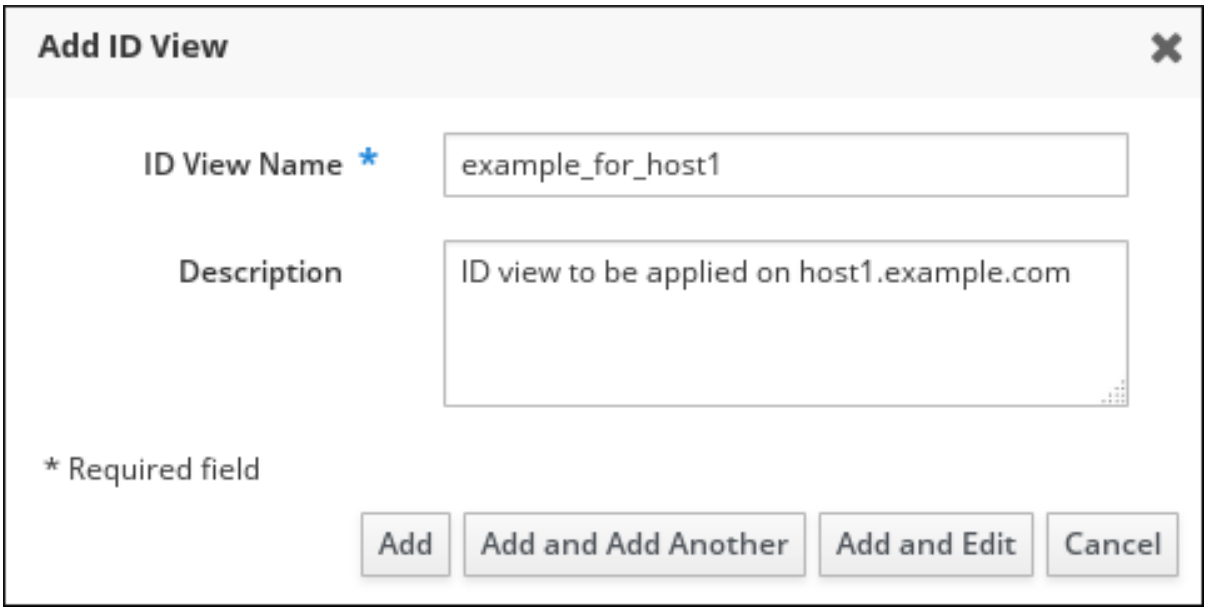
18.3.1. Web UI: 特定ホストの属性値の上書き

ID ビューを管理するには、最初に IdM 管理者として 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 ビューの名前を指定します。

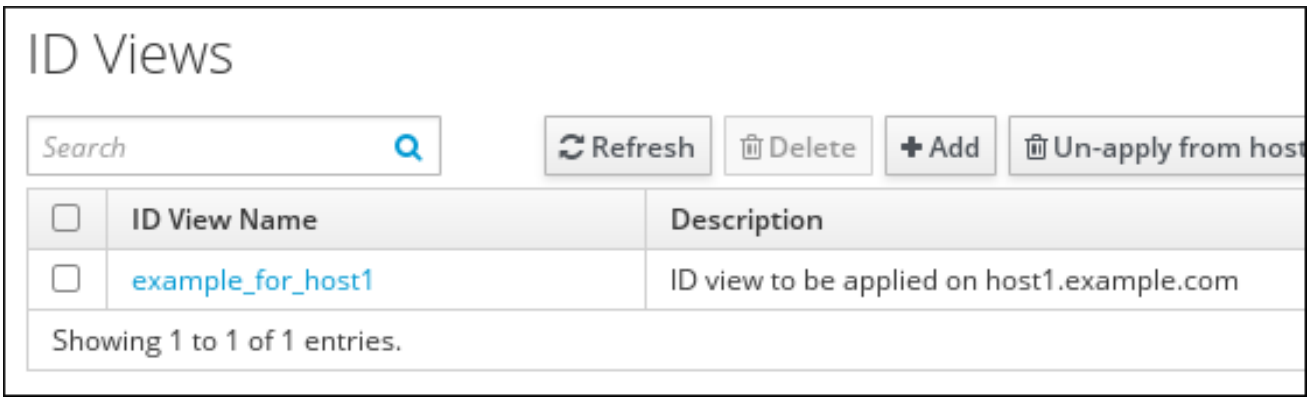
図18.1 ID ビューの追加



3. **Add** をクリックして確定します。

新しい ID ビューが ID ビューのリストに表示されます。

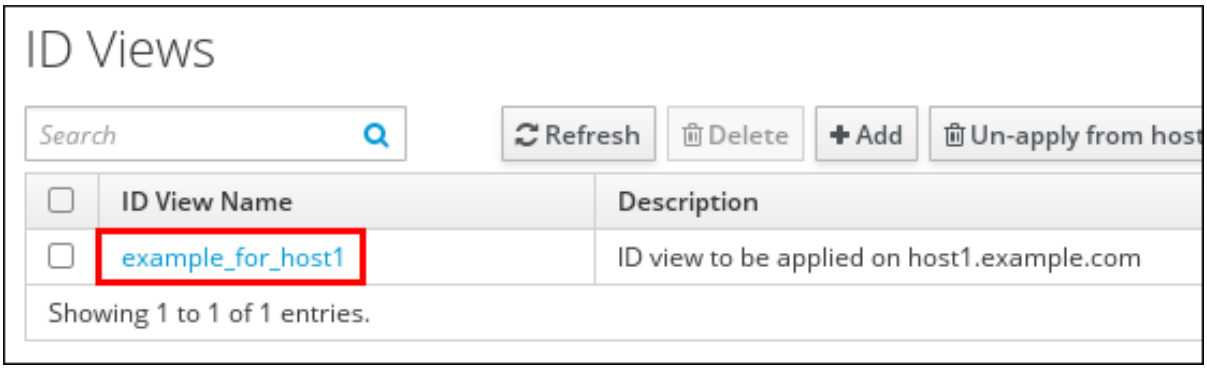
図18.2 ID ビューのリスト



ID ビューへのユーザーオーバーライドの追加

1. ID ビューのリストで、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

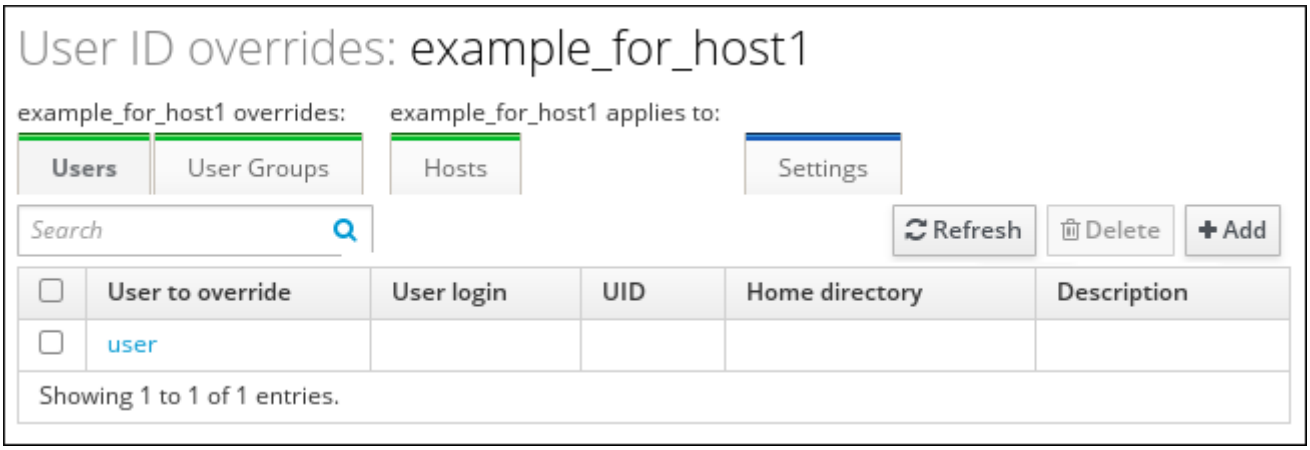


2. **Users** タブで **Add** をクリックして、ユーザーの上書きを追加します。

3. 上書きする属性値を持つユーザーアカウントを選択し、**Add** をクリックします。

ユーザーオーバーライドが **example_for_host1** ID ビューページに表示されるようになります。

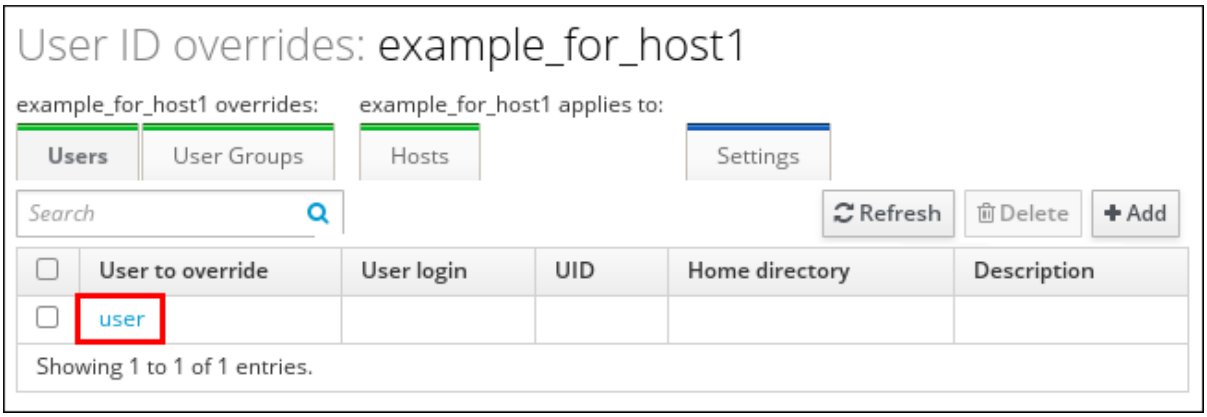
図18.4 オーバーライドのリスト



上書きする属性の指定

1. 属性値を変更するために使用するオーバーライドをクリックします。

図18.5 オーバーライドの編集

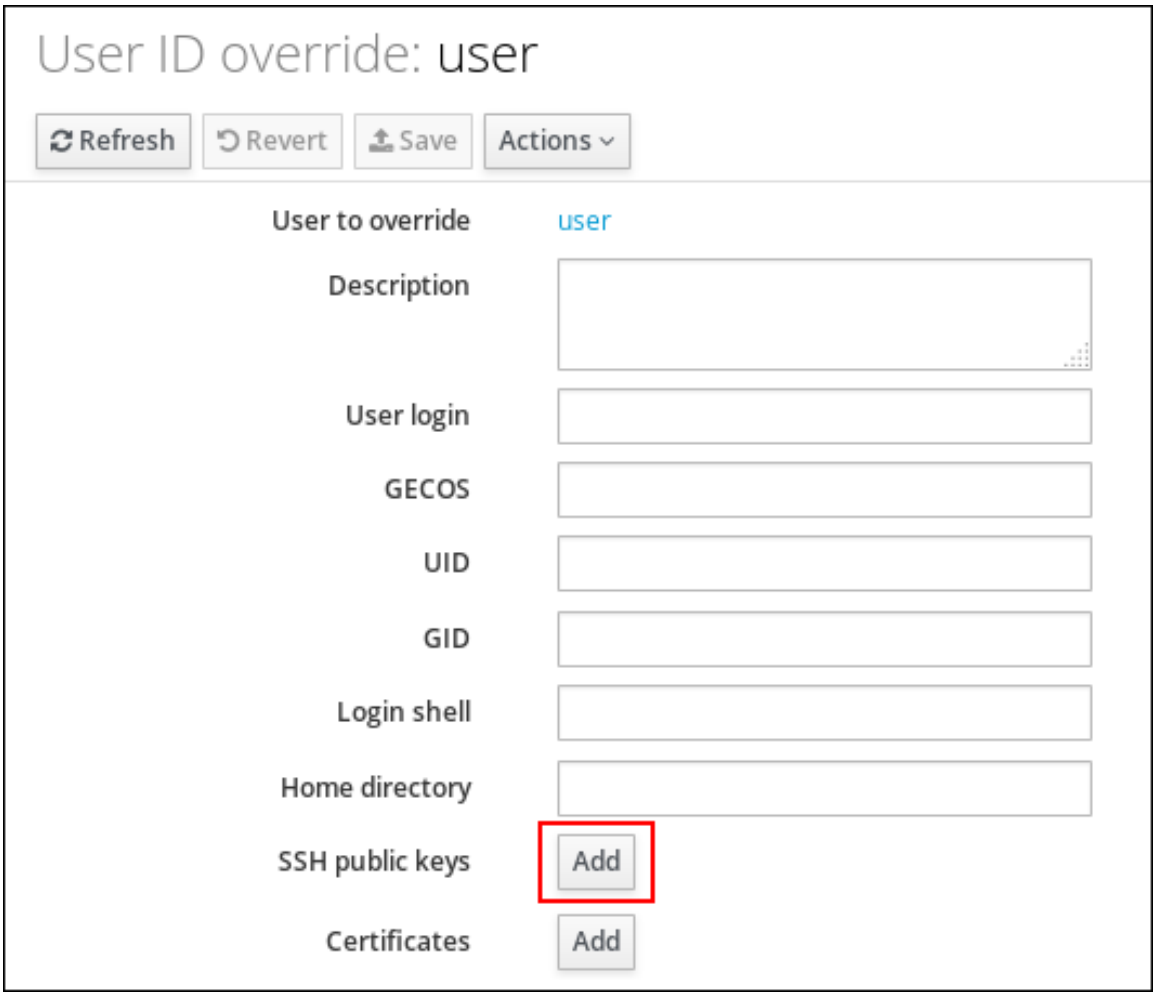


2. 属性の新しい値を定義します。

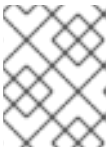
   たとえば、ユーザーアカウントが使用する SSH 公開鍵を上書きするには、以下の手順を実施します。

a. **SSH public keys: Add** をクリックします。

図18.6 SSH 公開鍵の追加



b. 公開鍵に貼り付けます。



注記

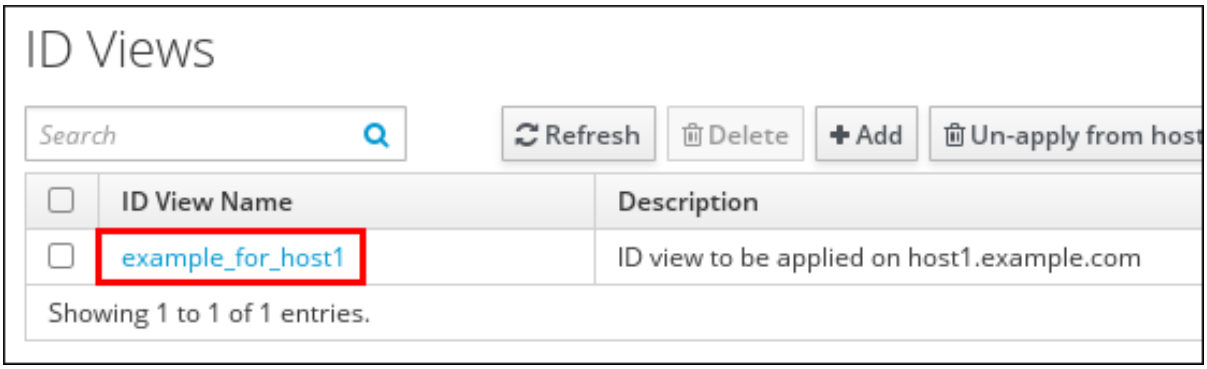
IdM に SSH キーを追加する方法は、「ユーザーの公開 SSH 鍵の管理」 を参照してください。

3. **Save** をクリックして上書きを更新します。

特定のホストへの ID ビューの適用

1. ID ビューのリストで、ID ビューの名前をクリックします。

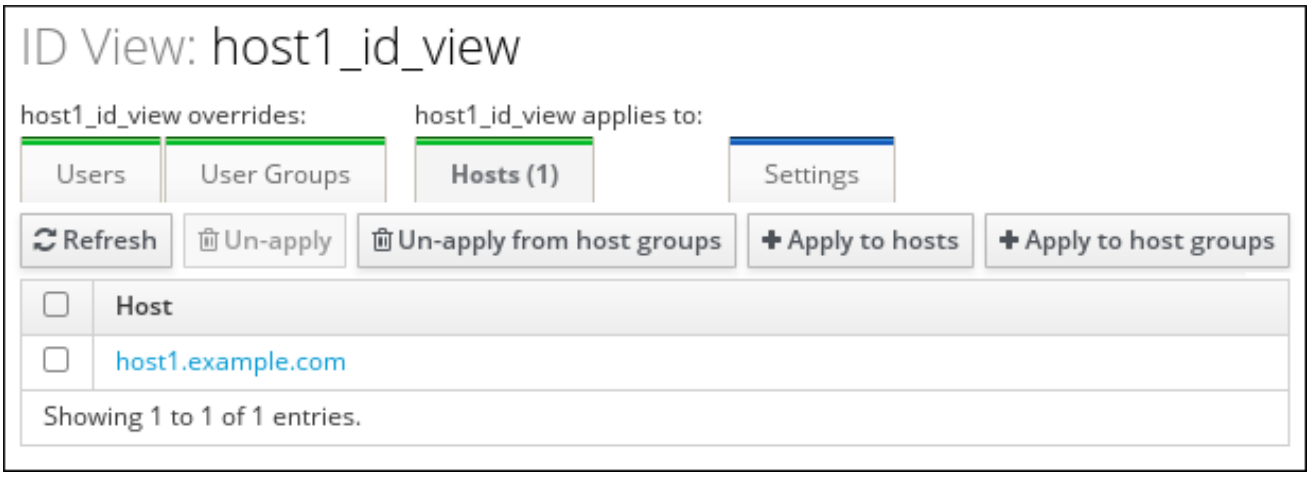
図18.7 ID ビューの編集



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、**Prospective** 列に移動します。
4. **Apply** をクリックします。

これで、ID ビューが適用されるホストリストにホストが表示されます。

図18.8 ID ビューが適用されるホストのリスト表示



18.3.2. コマンドライン: 特定ホストの属性値の上書き

ID ビューを管理する前に、IdM 管理者としてチケットを要求します。以下に例を示します。

```
$ kinit admin
```

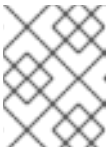
1. 新しい ID ビューを作成します。たとえば、**example_for_host1** という名前の ID ビューを作成するには、次のコマンドを実行します。

```
$ ipa idview-add example_for_host1
----------------------------
Added ID View "example_for_host1"
----------------------------
  ID View Name: example_for_host1
```

2. ユーザーオーバーライドを **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です。

- 新しい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のリストは、**ipa idoverrideuser-add --help** を実行します。たとえば、**--sshpubkey** オプションを使用して、SSH 公開鍵の値を上書きします。

```
$ 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
                  AAAB3NzaC1yrRqFE...gWRL71/miPIZ
    user@example.com
```



注記

IdM に SSH キーを追加する方法は、「ユーザーの公開 SSH 鍵の管理」 を参照してください。

- **ipa idoverrideuser-add --certificate** コマンドは、指定された ID ビューのアカウントの既存証明書をすべて置き換えます。追加の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、既存のユーザーのオーバーライドに新しい属性値を指定することもできます。

- **ipa idoverrideuser-del** コマンドを使用して、ユーザーの上書きを削除します。

注記

このコマンドを使用して SSH キーのオーバーライドを削除しても、キャッシュから SSH キーはすぐに削除されません。デフォルトのキャッシュタイムアウト値 (***entry_cache_timeout = 5400***) では、キーが 1 時間半の間キャッシュに残ります。

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 コマンドでは、**--hostgroups** オプションも使用できます。このオプションは、ID ビューを、指定のホストグループに所属するホストに適用しますが、ホストグループ自体との関連付けは行いません。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て、**--hosts** オプションを個別に適用します。

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、マシン、サービス、エントリーなどの特定のリソースにアクセスできるユーザーや、実行可能な操作の種類を定義するセキュリティー機能のセットです。Identity Management は複数のアクセス制御機能を提供し、付与されているアクセスの種類と、誰に付与されているかが明らかになります。この一環として、Identity Management は、ドメイン内のリソースへのアクセス制御と、IdM 設定自体へのアクセス制御を区別します。

IdM サーバーや IdM ユーザーに対する、IdM 内のユーザーが利用可能なさまざまな内部アクセス制御の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 フラグおよび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

さまざまな Kerberos フラグを使用して、Kerberos チケットの動作に関する特定の側面を定義できます。これらのフラグは、サービスとホストの Kerberos プリンシパルに追加できます。

Identity Management(IdM) のプリンシパルは、以下の Kerberos フラグを受け入れます。

OK_AS_DELEGATE

このフラグを使用して、委譲用に信頼される Kerberos チケットを指定します。

Active Directory(AD) クライアントは、Kerberos チケットで **OK_AS_DELEGATE** フラグをチェックして、ユーザーの認証情報を特定サーバーに転送または委譲できるかどうかを判断します。AD は、TGT(Ticket-granting Ticket) を **OK_AS_DELEGATE** が設定されたサービスまたはホストにのみ転送します。このフラグを使用すると、SSSD(System Security Services デーモン) は、IdM クライアントマシンのデフォルトの Kerberos 認証情報キャッシュに AD ユーザー TGT を追加できます。

REQUIRES_PRE_AUTH

このフラグを使用して、事前認証チケットのみがプリンシパルに対して認証できることを指定します。

REQUIRES_PRE_AUTH フラグを設定すると、キー配布センター (KDC) は追加の認証を要求します。KDC は、TGT が事前認証されている場合に限り、**REQUIRES_PRE_AUTH** が設定されたプリンシパルに TGT を発行します。

REQUIRES_PRE_AUTH を削除して、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。これにより、KDC への負荷が軽減されますが、以降の長期的なキーでブルートフォース攻撃の可能性が若干増大します。

OK_TO_AUTH_AS_DELEGATE

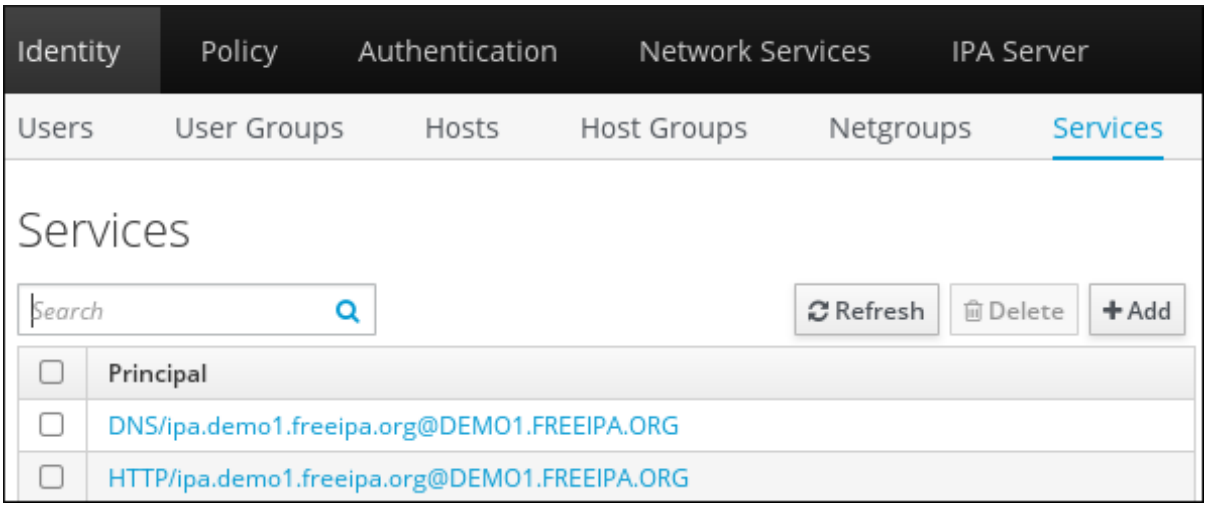
OK_TO_AUTH_AS_DELEGATE フラグを使用して、サービスがユーザーの代わりに kerberos チケットを取得できることを指定します。これは、プロトコルの移行を実行するのに十分ですが、ユーザーの代わりに他のチケットを取得するためには、サービスには **OK_AS_DELEGATE** フラグと、鍵配布センターで許可される対応するポリシー決定が必要です。

20.1.1. Web UI からの Kerberos フラグの設定

OK_AS_DELEGATE、**REQUIRES_PRE_AUTH**、または **OK_TO_AUTH_AS_DELEGATE** をプリンシパルに追加するには、以下を実行します。

1. **Identity** メインタブでアクセス可能な **Services** サブタブを選択します。

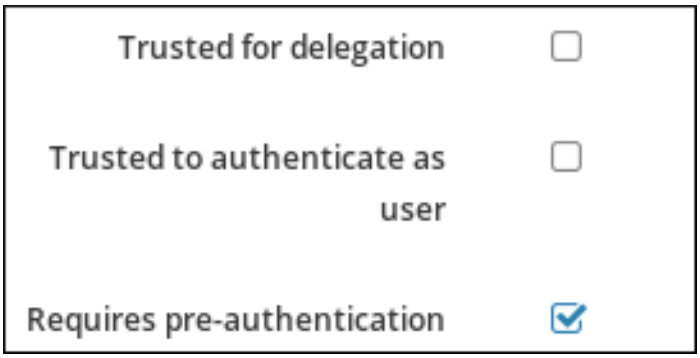
図20.1 サービスのリスト



2. フラグを追加するサービスをクリックします。

3. 設定するオプションのチェックを選択します。たとえば、**REQUIRES_PRE_AUTH** フラグを設定するには、**Requires pre-authentication** オプションのチェックを選択します。

図20.2 REQUIRES_PRE_AUTH フラグの追加



以下の表は、Kerberos フラグの名前と、Web UI での対応する名前を示しています。

表20.1 WebUI での Kerberos フラグのマッピング

Kerberos フラグ名	Web UI オプション
OK_AS_DELEGATE	**Trusted for delegation**
REQUIRES_PRE_AUTH	**認証の事前認証が必要**
OK_TO_AUTH_AS_DELEGATE	**ユーザーとして認証を信頼**

20.1.2. コマンドラインからの Kerberos フラグの設定および削除

フラグをコマンドラインからプリンシパルに追加するか、フラグを削除するには、以下のオプションのいずれかを **ipa service-mod** コマンドに追加します。

- **OK_AS_DELEGATE:--ok-as-delegate**
- **REQUIRES_PRE_AUTH:--requires-pre-auth**

- **OK_TO_AUTH_AS_DELEGATE:--ok-to-auth-as-delegate**

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 *service/ipa.example.com@EXAMPLE.COM* プリンシパルに追加するには、次のコマンドを実行します。

```
$ ipa service-mod service/ipa.example.com@EXAMPLE.COM --ok-as-delegate=1
```

フラグを削除するか、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、*test/ipa.example.com@EXAMPLE.COM* プリンシパルの **REQUIRES_PRE_AUTH** フラグを無効にするには、次のコマンドを実行します。

```
$ ipa service-mod test/ipa.example.com@EXAMPLE.COM --requires-pre-auth=0
```

20.1.3. コマンドラインからの Kerberos フラグの表示

OK_AS_DELEGATE がプリンシパルに現在設定されているかどうかを確認するには、以下を実行します。

1. **kvno** ユーティリティーを実行します。
2. **klist -f** コマンドを実行します。

OK_AS_DELEGATE は、**klist -f** 出力の **O** 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
```

表20.2 kerberos フラグの省略形

Kerberos フラグ名	省略形
OK_AS_DELEGATE	O
REQUIRES_PRE_AUTH	A
OK_TO_AUTH_AS_DELEGATE	F

プリンシパルに現在設定されているフラグを確認するには、**kadmin.local** ユーティリティーを使用します。現在のフラグは、**kadmin.local** 出力の **Attributes** 行に表示されます。以下に例を示します。

```
# kadmin.local
kadmin.local: getprinc test/ipa.example.com
Principal: test/ipa.example.com@EXAMPLE.COM
Expiration date: [never]
```

```
...
Attributes: REQUIRES_PRE_AUTH OK_AS_DELEGATE OK_TO_AUTH_AS_DELEGATE
Policy: [none]
```

20.2. ユーザー、ホスト、およびサービス用の KERBEROS プリンシパルエイリアスの管理

新しいユーザー、ホスト、またはサービスを作成すると、以下の形式で 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*

シナリオによっては、ユーザー、ホストまたはサービスがエイリアスを使用して Kerberos アプリケーションに対して認証できるようにすることは、管理者に役立ちます。

- ユーザー名の変更後に、ユーザーが以前のユーザー名と新しいユーザー名の両方でログインできるようにする。
- IdM Kerberos レルムがメールドメインと異なる場合でも、ユーザーはメールアドレスを使用してログインする必要がある。

ユーザーの名前を変更すると、オブジェクトはエイリアスと以前の正規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**ユーザー**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 **ipa host-add-principal** コマンドまたは **ipa service-add-principal** コマンドを使用します。

エイリアス名を使用して認証する場合は、**kinit** コマンドに **-C** オプションを渡します。

```
[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
アカウント **user**からエイリアス**useralias**を削除するには、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
```

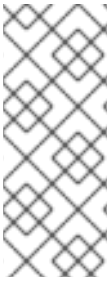
ホストまたはサービスからエイリアスを削除するには、代わりにそれぞれ **ipa host-remove-principal** または **ipa service-remove-principal** コマンドを使用します。

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 Enterprise Principal Alias

エンタープライズプリンシパルエイリアスは、ユーザープリンシパル名 (UPN) 接尾辞、NetBIOS 名、または信頼された Active Directory フォレストドメインのドメイン名以外の任意のドメイン接尾辞を使用できます。



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場合は、2 つのバックスラッシュ (\\) を使用して @ 記号をエスケープします。そうしないと、シェルは @ シンボルを Kerberos レルム名の一部として解釈し、以下のエラーが発生します。

```
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 **ipa host-add-principal** コマンドまたは **ipa service-add-principal** コマンドを使用します。

エンタープライズプリンシパル名を使用して認証する場合は、**kinit** コマンドに **-E** オプションを渡します。

```
[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、以下のコマンドを入力します。

```
[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 および IDENTITY MANAGEMENT の概要

UNIX 環境では、ネットワーク情報サービス (NIS) は ID と認証を一元管理する一般的な方法です。元々 **Yellow Pages** (YP) という名前で呼ばれていた NIS は、以下のような認証や ID 情報を一元管理します。

- ユーザーおよびパスワード
- ホスト名および IP アドレス
- POSIX グループ

今日のネットワークインフラストラクチャーでは、NIS は、ホスト認証を提供しておらず、データが暗号化せずにネットワークに送信されるため、セキュリティーが非常に低いと見なされます。この問題を回避するため、NIS はセキュリティーを強化するために他のプロトコルと統合されることが多くあります。

Identity Management(IdM) を使用する場合は、NIS サーバープラグインを使用して、IdM に完全に移行することができないクライアントに接続できます。IdM は、ネットグループおよびその他の NIS データを IdM ドメインに統合します。また、NIS ドメインから IdM にユーザーおよびホストの ID を簡単に移行することもできます。

Identity Management での NIS

NIS オブジェクトは、RFC 2307 に準拠し、Directory Server バックエンドに統合され、保存されます。IdM は、LDAP ディレクトリーに NIS オブジェクトを作成し、クライアントは例えば System Security Services Daemon (SSSD) または暗号化された LDAP 接続を使用する**nss_ldap**を通じてそのオブジェクトを取得します。

IdM は、ネットグループ、アカウント、グループ、ホスト、およびその他のデータを管理します。IdM は NIS リスナーを使用してパスワード、グループ、およびネットグループを 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 サポートのために、IdM は slapi-nis パッケージで提供される以下のプラグインを使用します。

NIS サーバープラグイン

NIS サーバープラグインにより、IdM 統合 LDAP サーバーがクライアントの NIS サーバーとして機能できるようになります。このロールでは、Directory Server は設定に応じて NIS マップを動的に生成し、更新します。プラグインを使用すると、IdM は NIS プロトコルを使用するクライアントに対して NIS サーバーとして機能します。

詳細は、「Identity Management での NIS の有効化」 を参照してください。

スキーマ互換性プラグイン

スキーマ互換性プラグインを使用すると、Directory Server バックエンドは、ディレクトリー情報ツリー (DIT) の一部に保存されたエントリーの代替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、ドロップ、名前変更、およびオプションでツリー内の複数のエントリーからの属性値の取得が含まれます。

詳細は、**/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 ネットグループ

NIS エンティティーはネットグループに保存できます。UNIX グループと比較すると、ネットグループは以下のサポートを提供します。

- ネスト化されたグループ (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

ネットグループは、ホスト、ユーザー、およびドメインなどの一連の情報を定義します。このセットは **トリプル** と呼ばれています。以下の 3 つのフィールドを含めることができます。

- 値。
- 有効な値なしを指定するダッシュ (**-**)
- 値なし。空のフィールドはワイルドカードを指定します。

> (*host.example.com*,,*nisdomain.example.com*)
> (-,*user*,*nisdomain.example.com*)

クライアントが NIS ネットグループを要求すると、IdM は以下の項目に LDAP エントリーを変換します。

- 従来の NIS マップへと変換し、これを NIS プラグインを使用して NIS プロトコル経由でクライアントに送信します。
- RFC 2307 または RFC 2307bis に準拠する LDAP 形式。

21.1.1.1. NIS ネットグループエントリーの表示

IdM は、ユーザーおよびグループを ***memberUser*** 属性に保存します。また、 ***memberHost*** にホストおよびホストグループを保存します。以下の例は、IdM の Directory Server コンポーネントのネットグループエントリーを示しています。

例21.1 Directory Server 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-*** コマンドを使用してネットグループエントリーを管理できます。たとえば、ネットグループエントリーを表示するには、次のコマンドを実行します。

例21.2 ネットグループエントリーの表示

```
[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 での NIS の有効化

Identity Management で NIS を有効にするには、以下を実行します。

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:** NIS リモート手順呼び出し (RPC) に固定ポートを設定します。

   NIS を使用する場合、クライアントは接続を確立するために使用する IdM サーバーのポートを認識する必要があります。デフォルト設定を使用すると、IdM はサーバーの起動時に未使用のランダムな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を要求するために使用するポートマッパーサービスに送信されます。

   より厳密なファイアウォール設定を行うには、固定ポートを設定できます。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

設定には、1024 未満の未使用のポート番号を設定できます。

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. ネットグループの追加

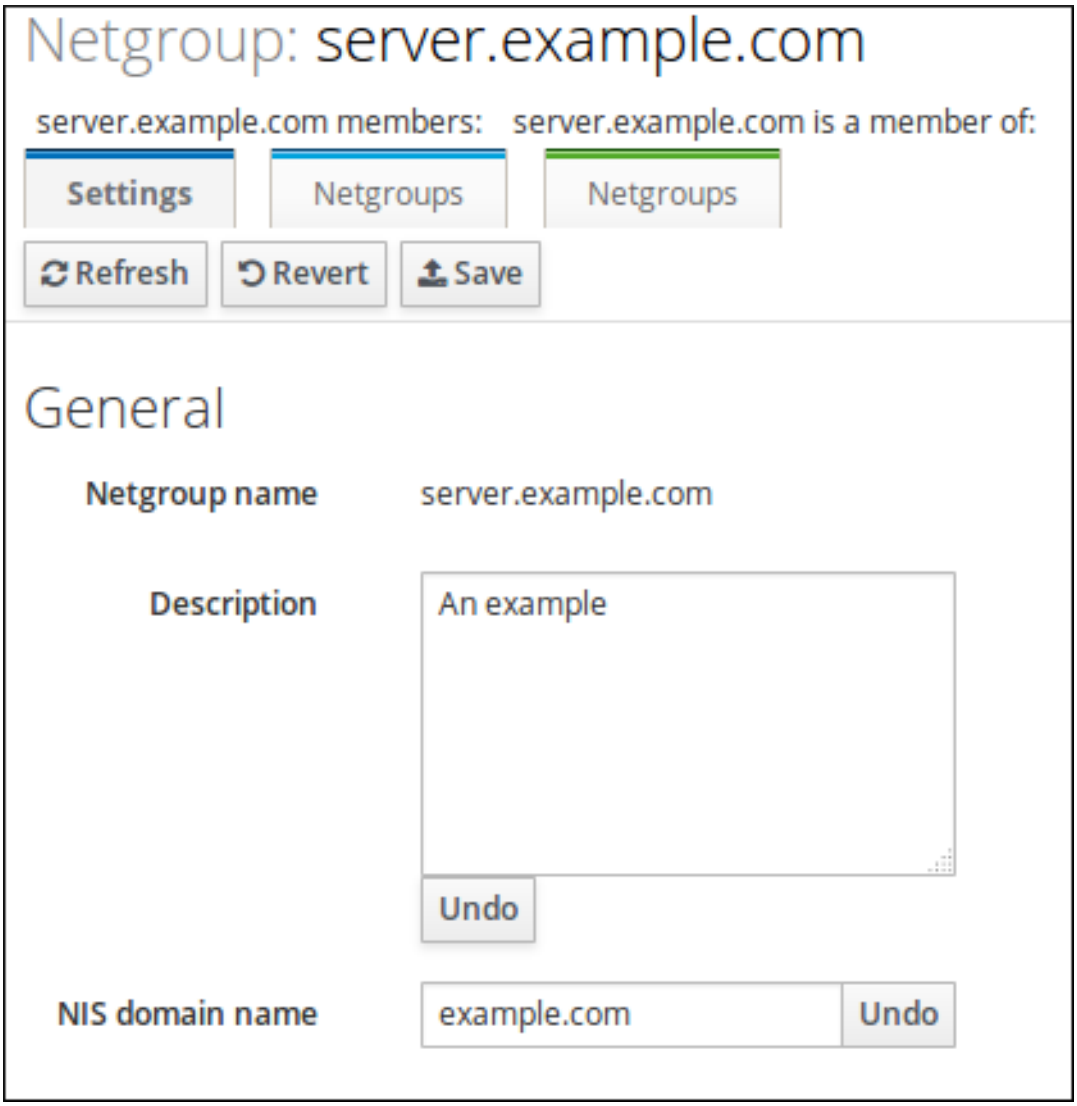
ネットグループを追加するには、以下のツールを使用できます。

- IdM Web UI(「Web UI: ネットグループの追加」を参照)
- コマンドライン (「コマンドライン: ネットグループの追加」を参照)

Web UI: ネットグループの追加

1. **Identity → Groups → Netgroups**を選択します。
2. **追加** をクリックします。
3. 一意の名前を入力し、必要に応じて説明を入力します。グループ名は、IdM ドメインの netgroup に使用する ID です。後で変更することはできません。
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 ドメインは IdM ドメイン名に設定されます。オプションで、NIS domain name フィールドに別の NIS ドメインの名前を入力できます。

   図21.1 Netgroup タブ



   NIS domain name フィールドでは、netgroup のトリプルに表示されるドメインを設定します。Identity Management NIS リスナーが応答する NIS ドメインには影響はありません。

6. 「Web UI: ネットグループへのメンバーの追加」 にあるように、メンバーを追加します。
7. **Save** をクリックします。

コマンドライン: ネットグループの追加
ipa netgroup-add コマンドを使用して、新しい netgroup を追加できます。以下の項目を指定します。

- グループ名
- 任意の説明
- 必要に応じて、IdM ドメイン名と異なる場合は NIS ドメイン名

注記

この **--nisdomain** オプションは、netgroup トリプルに表示されるドメインを設定します。Identity Management リスナーが応答する NIS ドメインには影響はありません。

以下に例を示します。

```
[root@server ~]# ipa netgroup-add --desc="Netgroup description" --nisdomain="example.com"
example-netgroup
```

ネットグループにメンバーを追加するには、「コマンドライン: ネットグループへのメンバーの追加」を参照してください。

21.3.2. ネットグループへのメンバーの追加

ユーザーとホスト以外に、ネットグループには、ユーザーグループ、ホストグループ、およびその他のネットグループ (ネストされたグループ) 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、ネスト化されたグループを作成した後に、子グループのメンバーが親グループのメンバーとして表示されるまでに、数分かかることがあります。

ネットグループにメンバーを追加するには、以下を使用できます。

- IdM Web UI は、「Web UI: ネットグループへのメンバーの追加」を参照してください。
- コマンドラインは、「コマンドライン: ネットグループへのメンバーの追加」を参照してください。

警告

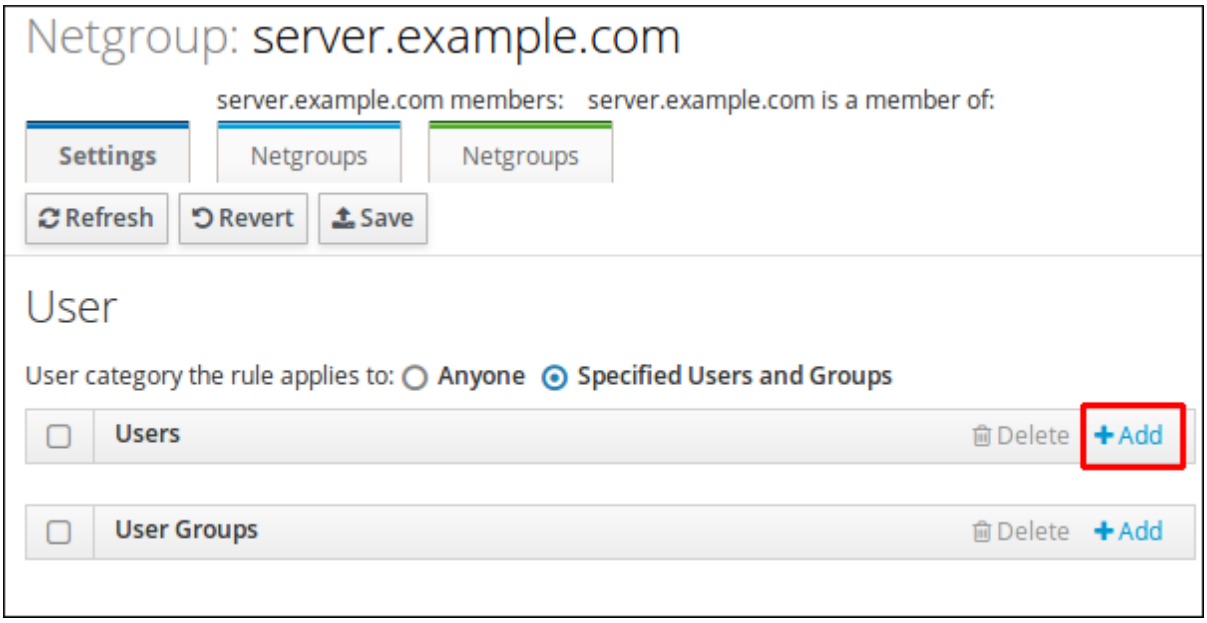
再帰的なネスト化されたグループは作成しないでください。たとえば、*GroupA* が *GroupB* のメンバーの場合には、*GroupB* を *GroupA* のメンバーとして追加しないでください。再帰グループはサポートされておらず、予測不可能な動作を引き起こす可能性があります。

Web UI: ネットグループへのメンバーの追加
Web UI を使用してネットグループにメンバーを追加するには、以下を実行します。

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。

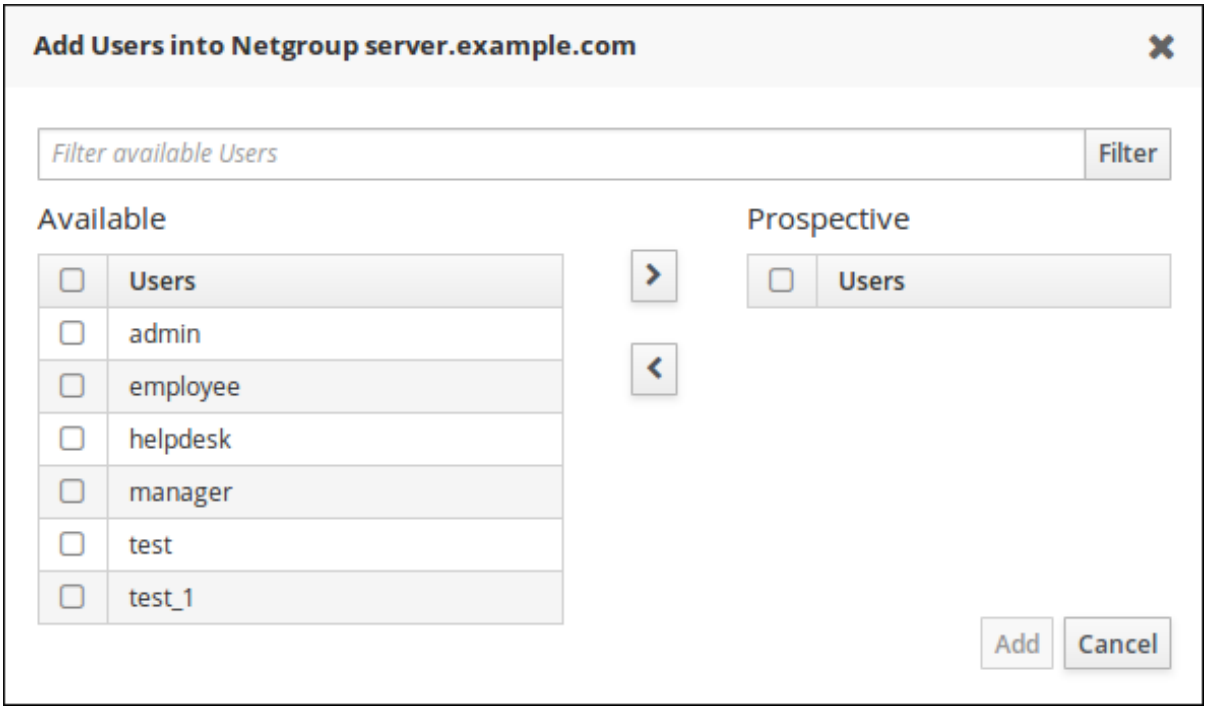
3. 必要なメンバータイプの横にある **Add** をクリックします。

図21.2 Netgroup タブで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 タブでのユーザーメニューの追加



5. **Add** をクリックします。

コマンドライン: ネットグループへのメンバーの追加

ネットグループを作成したら、**ipa netgroup-add-member** コマンドを使用してメンバーを追加できます。

```
# ipa netgroup-add-member --users=user_name --groups=group_name --hosts=host_name \
    --hostgroups=host_group_name --netgroups=netgroup_name group_nameame
```

複数のメンバーを設定するには、中かっこ内でコンマ区切りリストを使用します。以下に例を示します。

```
[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すでに定義されている場合は、IdM の NIS 設定に手動で追加する必要があります。これにより、マップが NIS クライアントに公開されます。

NIS サーバーは、IdM LDAP ディレクトリーの特別なプラグインエントリーによって管理されます。各 NIS ドメインおよび NIS サーバーによって使用されるマップは、このコンテナー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が含まれます。

- NIS ドメインの名前
- NIS マップの名前
- NIS マップの内容として使用するディレクトリーエントリーを検索する方法に関する情報
- NIS マップのキーおよび値として使用する属性に関する情報

これらの設定のほとんどは、すべてのマップで同じです。

21.4.1. 自動マウントマップの追加

IdM は、IdM ディレクトリーツリーの **cn=automount** ブランチに、自動マウントの場所別にグループ化された自動マウントマップを保存します。LDAP プロトコルを使用して NIS ドメインおよびマップを追加できます。

たとえば、**example.com** ドメインの **default**に **auto.example** という名前の自動マウントマップを追加するには、次のコマンドを実行します。

```
[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 属性を NIS ドメインの名前に設定します。

nis-base 属性に設定された値は以下に対応している必要があります。

- **ipa automountmap-*** コマンドを使用して設定され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された既存の自動マウントの場所

エントリーを設定したら、自動マウントマップを確認できます。

```
[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、以下の手順が必要です。

1. Identity Management の NIS リスナーの有効化
2. NIS から既存データをエクスポートおよびインポート

21.5.1. IdM での netgroup エントリーの準備

移行前に、現在の NIS サーバーで管理されているアイデンティティーの種類を特定します。

ユーザーエントリー

NIS によって提供されるユーザー情報を使用するアプリケーションを判定します。**sudo** などの一部のユーティリティーは NIS ネットグループを必要としますが、複数のユーティリティーが通常の UNIX グループを使用できます。

移行は、以下の手順で実行します。

1. IdM で対応するユーザーアカウントを作成します。「ユーザーエントリーの移行」を参照してください。
2. さらにネットグループが必要な場合は、以下を行います。
   a. ネットグループを追加します。「ネットグループの追加」を参照してください。
   b. ユーザーをネットグループに追加します。「ネットグループエントリーの移行」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、対応するシャドウの NIS グループが自動的に作成されます。これらのシャドウ NIS グループに **ipa netgroup-*** コマンドを使用しないでください。**ipa netgroup-*** コマンドは、**netgroup-add** コマンドで作成された **ネイティブ** の netgroups の管理にだけ使用します。

直接変換の場合

すべてのユーザーエントリーとホストエントリーが同じ名前を使用する必要がある場合は、IdM で同じ名前を使用してエントリーを作成できます。

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、ネットグループにこれらのグループを追加することもできます。

21.5.2. Identity Management での NIS リスナーの有効化

「Identity Management での NIS の有効化」 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、netgroups、および自動マウントマップに関する情報を追加できます。これらのエントリータイプを IdM に移行できます。

次のセクションでは、**ypcat** コマンドを使用して現在の NIS サーバーからデータをエクスポートし、出力を使用して、対応する **ipa *-add** コマンドを使用してエントリーを IdM にインポートします。

- 移行スクリプトで使用される **ypcat** コマンドを提供するため、**yp-tools** パッケージをインストールするようにしてください。

```
[root@nis-server ~]# yum install yp-tools -y
```

21.5.3.1. ユーザーエントリーの移行

NIS の **passwd** マップには、名前、UID、プライマリーグループ、GECOS、シェル、ホームディレクトリーなどのユーザーに関する情報が含まれます。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**: パスワード強度の弱いパスワードに対応する必要がある場合には、「NIS ユーザー認証用の脆弱なパスワードハッシュ化の有効化」 を参照してください。
2. 以下の内容で **/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 passw0rd1 | 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. スクリプトを実行します。以下に例を示します。

```
[root@nis-server ~]# sh /root/nis-users.sh nisdomain nis-master.example.com
```

注記

このスクリプトは、名、姓にハードコードされた値を使用し、パスワードを **passw0rd1** に設定します。ユーザーは、次回ログイン時に一時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 **グループ** 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. 以下の内容で **/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. スクリプトを実行します。以下に例を示します。

```
[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 **ホスト** マップには、ホスト名や IP アドレスなどのホストに関する情報が含まれます。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. 以下の内容で **/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. スクリプトを実行します。以下に例を示します。

```
[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別なホスト設定は移行されません。

21.5.3.4. ネットグループエントリーの移行

NIS **netgroup** マップには、netgroup に関する情報が含まれます。このデータを使用して、NIS ネットグループを IdM に移行します。

1. 以下の内容で **/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 $netgroup $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. スクリプトを実行します。以下に例を示します。

```
[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)、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です。NIS 自動マウントマップを IdM に移行するには、以下を実行します。

1. 以下の内容で **/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 自動マウント情報のエクスポート、自動マウントの場所と関連マップの LDAP データ交換形式 (LDIF) の生成、IdM Directory Server への LDIF ファイルのインポートが行われます。詳細は、「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. スクリプトを実行します。以下に例を示します。

```
[root@nis-server ~]# sh /root/nis-automounts.sh location nisdomain \
    nis-master.example.com map_name
```

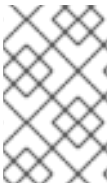
21.5.4. NIS ユーザー認証用の脆弱なパスワードハッシュ化の有効化

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でセキュア化されたハッシュアルゴリズム (SSHA) を使用してハッシュ化されます。NIS クライアントにパスワードの弱いハッシュアルゴリズムが必要な場合は、パスワードストレージスキーム設定を更新します。

弱いパスワードハッシュスキームを有効にすると、***userPassword*** 属性に保存されているパスワードのみが影響を受けます。Kerberos はこの属性を使用しないため、Kerberos 暗号化はこの設定の影響を受けません。

たとえば、**CRYPT** ハッシュ化された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

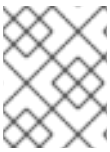
パスワードハッシュは復号できないため、Directory Server は既存のパスワードハッシュを変換しません。サーバーは、ストレージスキームの変更後に設定したパスワードにのみ新しいパスワードストレージを適用します。

パート V. 管理: 認証の管理

このパートでは、スマートカード認証の設定および管理方法を説明します。さらに、証明書の発行、証明書ベースの認証設定、**ID 管理** での証明書の有効性の制御など、証明書関連のトピックを説明します。

第22章 ユーザー認証

本章では、ユーザーのパスワード、SSH キー、および証明書を管理する方法や、ワンタイムパスワード (OTP) およびスマートカード認証を設定する方法など、ユーザー認証メカニズムの管理について説明します。



注記

Kerberos を使用して Identity Management (IdM) にログインする方法は、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、独自の個人パスワードのみを変更できます。この方法で個人パスワードが変更されました。

- IdM パスワードポリシーを満たしている必要があります。パスワードポリシーの設定に関する詳細は、28章*パスワードポリシーの定義* を参照してください。

管理者およびパスワード変更権限を持つユーザーは、新しいユーザーに初期パスワードを設定し、既存のユーザーのパスワードをリセットできます。この方法でパスワードが変更されました。

- IdM パスワードポリシーを満たす必要はありません。
- 最初のログインに成功したら失効します。このような場合、IdM はユーザーが期限切れのパスワードを直ちに変更するよう要求します。この動作を無効にするには、「Next login でパスワード変更を要求しないパスワードリセットの有効化」 を参照してください。

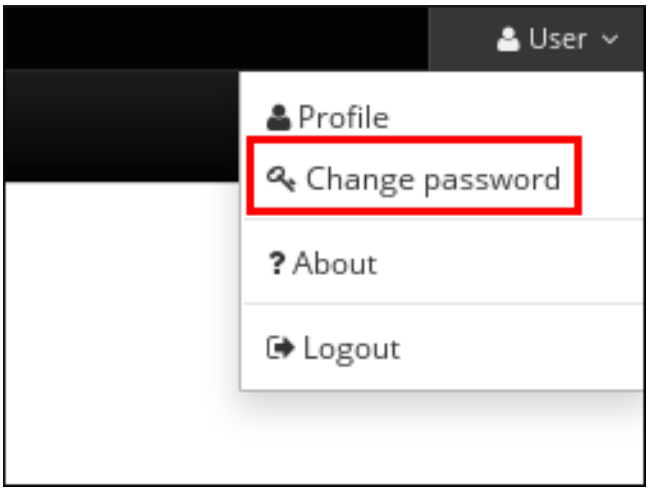
注記

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、任意の IdM パスワードポリシーを上書きできます。DM によって設定されたパスワードは最初のログイン後に有効期限が切れません。

22.1.1.1. Web UI: 独自の個人パスワードの変更

1. 右上の ***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

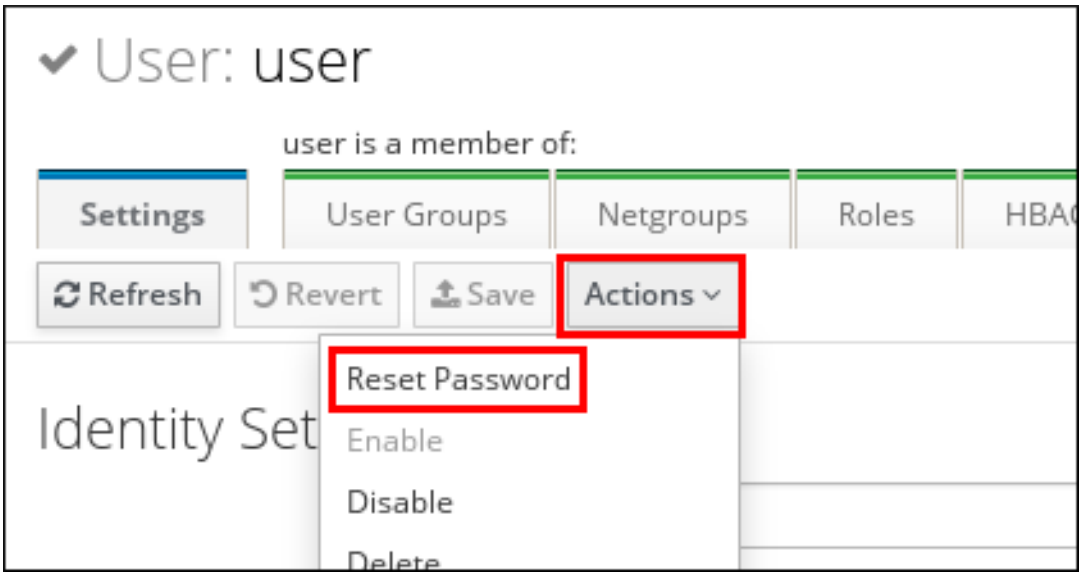


2. 新しいパスワードを入力します。

22.1.1.2. Web UI: 別のユーザーのパスワードの設定

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. 新しいパスワードを入力し、**Reset Password** をクリックします。

図22.3 新しいパスワードの確認

22.1.1.3. コマンドライン: 別のユーザーのパスワードの変更または再設定

独自の個人パスワードを変更したり、別のユーザーのパスワードを変更またはリセットするには、**--password** オプションを **ipa user-mod** コマンドに追加します。このコマンドにより、新しいパスワードの入力が求められます。

```
$ ipa user-mod user --password
Password:
Enter Password again to verify:
--------------------
Modified user "user"
--------------------
...
```

22.1.2. Next login でパスワード変更を要求しないパスワードリセットの有効化

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、初回のログインに成功したらパスワードが期限切れになります。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が期限切れにならないようにするには、ドメイン内のすべての Identity Management サーバーでこれらの変更を加えます。

- パスワード同期エントリー (**cn=ipa_pwd_extop,cn=plugins,cn=config**) を編集します。
- ***PassSyncManagersDNs*** 属性に管理ユーザーアカウントを指定します。属性は多値です。

たとえば、**ldapmodify** ユーティリティーを使用して **admin** ユーザーを指定するには、次のコマンドを実行します。

```
$ 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
```

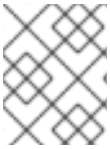
警告

これらの追加権限を必要とするユーザーのみが指定します。***PassSyncManagerDNs*** にリスト表示されているすべてのユーザーが、以下を行うことができます。

- 後続のパスワードリセットを必要とせずにパスワード変更操作を実行する
- 強度や履歴の強制が適用されないようにパスワードポリシーをバイパスします。

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

ユーザーが間違ったパスワードを使用してログインしようとすると、IdM はユーザーアカウントをロックし、ユーザーがログインできなくなります。IdM は、ユーザーアカウントがロックされていることを示す警告メッセージが表示されないことに注意してください。



注記

許容される失敗した試行の数とロックアウトの期間を正確に設定する方法は、28章 *パスワードポリシーの定義* を参照してください。

IdM は、指定した時間が経過した後にユーザーアカウントを自動的にアンロックします。また、管理者は、ユーザーアカウントを手動でアンロックできます。

ユーザーアカウントの手動ロック解除
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"
----------------------
```

その後、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の確認

ユーザーの失敗したログイン試行の数を表示するには、**ipa user-status** コマンドを使用します。表示される数が、許可されるログイン試行回数を超えると、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
----------------------------
```

デフォルトでは、Red Hat Enterprise Linux 7.4 以降の IdM は、最後に成功した Kerberos 認証のタイムスタンプを保存しません。この機能を有効にするには、「最後に成功した Kerberos 認証の追跡の有効化」 を参照してください。

22.2. 最後に成功した KERBEROS 認証の追跡の有効化

パフォーマンス上の理由から、Red Hat Enterprise Linux 7.4 以降で実行している IdM は、最後に成功した Kerberos 認証のタイムスタンプを保存しません。そのため、**ipa user-status** などの特定のコマンドはタイムスタンプを表示しません。

最後に成功した Kerberos 認証の追跡を有効にするには、以下を実行します。

1. 現在有効なパスワードプラグイン機能を表示します。

```
# ipa config-show | grep "Password plugin features"
  Password plugin features: AllowNThash, KDC:Disable Last Success
```

次の手順で **KDC:Disable Last Success** 以外の機能の名前が必要です。

2. **KDC:Disable Last Success** を除き、現在有効な **ipa config-mod** コマンドに ***--ipaconfigstring=feature*** パラメーターを渡します。

```
# ipa config-mod --ipaconfigstring='AllowNThash'
```

このコマンドは、**AllowNThash** プラグインのみを有効にします。複数の機能を有効にするには、***--ipaconfigstring=feature*** パラメーターを複数回指定します。たとえば、**AllowNThash** および **KDC:Disable Lockout** 機能を有効にするには、以下を実行します。

```
# ipa config-mod --ipaconfigstring='AllowNThash' --ipaconfigstring='KDC:Disable Lockout'
```

3. IdM を再起動します。

```
# ipactl restart
```

22.3. 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 以降を実行しているクライアントでのみサポートされます。

ワンタイムパスワード (OTP) は、1 つの認証セッションにのみ有効になり、使用後に無効になります。従来の静的パスワードとは異なり、認証トークンによって生成された OTP は変更を維持します。OTP は、2 要素認証の一部として使用されます。

1. ユーザーは従来のパスワードで認証します。
2. ユーザーは、認識された OTP トークンによって生成された OTP コードを提供します。

2 要素認証は、従来のパスワードのみを使用した認証よりも安全であると考えられます。ログイン中に OTP を傍受しても、その時点で傍受された OTP はすでに無効になり、認証の成功にのみ使用可能であるためです。

警告

現在、以下のセキュリティーと、IdM の OTP サポートに関連しています。

- 最も重要なセキュリティー制限は、システム全体で攻撃を再生する可能性のある脆弱性です。レプリケーションは非同期であるため、OTP コードはレプリケーション期間中に再利用できます。ユーザーは 2 つのサーバーに同時にログインできる場合があります。ただし、この脆弱性は通常、包括的な暗号化のために悪用するのが難しくなります。
- OTP 認証をサポートしないクライアントを使用して、TGT (Ticket-Granting Ticket) を取得することはできません。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、特定のユースケースに影響を及ぼす可能性があります。
- FIPS モードが有効な場合は、IdM ソリューションでパスワード + OTP を使用することはできません。

22.3.1. IdM での OTP 認証の仕組み

22.3.1.1. IdM でサポートされている 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、ソフトウェアトークンとハードウェアトークンの両方をサポートします。

ユーザー管理のトークンおよび管理者管理のトークン
ユーザーは独自のトークンを管理でき、管理者はそれらのトークンを管理できます。

ユーザー管理のトークン

ユーザーは、Identity Management のユーザー管理トークンを完全に制御できます。トークンの作成、編集、または削除が可能です。

管理者管理のトークン

管理者は、管理者管理のトークンをユーザーのアカウントに追加します。ユーザー自体には、このようなトークンに対する読み取り専用アクセスがあります。トークンを管理または変更するパーミッションがなく、これらをいずれの方法でも設定する必要はありません。

現在、アクティブなトークンのみであれば、ユーザーはトークンを削除または非アクティブにすることはできません。管理者は、最後にアクティブなトークンを削除または非アクティブ化することはできませんが、別のユーザーの最後のアクティブなトークンを削除または非アクティブ化することができます。

対応している OTP アルゴリズム
Identity Management は、以下にある、2 つの標準 OTP メカニズムに対応しています。

- 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に対応する HOTP の拡張機能です。

22.3.1.2. 利用可能な OTP 認証方法

OTP 認証を有効にする場合、以下の認証方法を選択できます。

2 要素認証 (パスワード + OTP)

この方法では、標準パスワードと OTP コードの両方を入力する必要があります。

Password

この方法では、標準のパスワードのみを使用して認証を行うオプションがあります。

RADIUS プロキシーサーバー認証

OTP 検証に RADIUS サーバーを設定する方法は、「プロプライエタリー 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

これらの認証方法は、グローバルまたは個々のユーザーに対して設定できます。

- デフォルトでは、ユーザー固有の認証方法設定はグローバル設定よりも優先されます。ユーザーに認証方法が設定されていない場合、グローバルに定義されているメソッドが適用されます。
- 任意のユーザーのユーザーごとの認証方法設定を無効にできます。これにより、IdM はユーザーごとの設定を無視し、常にユーザーにグローバル設定を適用するようになります。

複数の認証方法の統合

複数のメソッドを一度に設定すると、認証を成功させるには、いずれかの方法で十分です。以下に例を示します。

- 2 要素とパスワード認証の両方を設定する場合、ユーザーはパスワード (最初の係数) を指定する必要がありますが、コマンドラインを使用する場合は OTP(2 番目の係数) を提供することは任意です。

```
First Factor:
Second Factor (optional):
```

- Web UI では、ユーザーは両方の要素を指定する必要があります。

注記

個々のホストまたはサービスは、OTP などの特定の認証方法を必要とするように設定できます。最初の要素を使用してこのようなホストまたはサービスに対して認証しようとすると、アクセスは拒否されます。「ユーザーの認証方法に基づいたサービスとホストへのアクセス制限」を参照してください。

ただし、RADIUS と別の認証方法が設定されている場合には、マイナーな例外が存在します。

- Kerberos は常に RADIUS を使用しますが、LDAP は使用しません。LDAP は、パスワードと 2 要素認証メソッドのみを認識します。

- 外部の 2 要素認証プロバイダーを使用する場合は、アプリケーションから Kerberos を使用します。パスワードを使用した認証のみを許可する場合は、LDAP を使用します。アプリケーションは、Kerberos または LDAP のいずれかの設定を可能にする Apache モジュールおよび SSSD を利用することが推奨されます。

22.3.1.3. GNOME Keyring サービスのサポート

IdM は、OTP 認証と GNOME Keyring サービスを統合します。GNOME Keyring 統合では、ユーザーは最初の要素と 2 番目の要素を個別に入力する必要があります。

> First factor: *static_password*
> Second factor: *one-time_password*

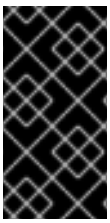
22.3.1.4. OTP を使用したオフライン認証

IdM は、オフラインの OTP 認証に対応します。ただし、オフラインでログインできるようにするには、静的パスワードと OTP を個別に入力して、システムがオンラインになると、最初に認証する必要があります。

> First factor: *static_password*
> Second factor: *one-time_password*

オンラインでログインする際に両方のパスワードが別個に入力されると、中央認証サーバーが利用できない場合でもユーザーは認証が可能になります。IdM は、ユーザーがオフライン時に従来の静的パスワードのみを要求することに注意してください。

IdM は、**First factor** プロンプトで 1 つの文字列に静的パスワードと OTP の両方を入力することもできます。ただし、これはオフラインの OTP 認証と互換性がありません。ユーザーが両方の要素を 1 つのプロンプトで入力すると、認証時に IdM が常に中央認証サーバーに接続する必要があります。これには、システムをオンラインにする必要があります。



重要

ノートパソコンなど、オフラインでも動作するデバイスで OTP 認証を使用する場合は、オフライン認証が利用可能になるように、Red Hat は、静的パスワードと OTP を個別に入力することを推奨します。そうしないと、IdM では、システムがオフラインになった後にログインできなくなります。

OTP オフライン認証を活用する場合は、静的パスワードと OTP パスワードを別々に入力する以外には、以下の条件を満たしていることを確認してください。

- **/etc/sssd/sssd.conf** ファイルの ***cache_credentials*** オプションは **True** に設定され、最初のファクターパスワードをキャッシュできるようにします。

- First-factor の静的パスワードは、**/etc/sssd/sssd.conf** で設定した ***cache_credentials_minimal_first_factor_length*** オプションに定義されたパスワードの長さの要件に対応します。デフォルトの最小長は 8 文字です。オプションの詳細は、sssd.conf(5) の man ページを参照してください。

/etc/sssd/sssd.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、SSSD は、システムがオンラインになったときに Kerberos チケット保証チケット (TGT) の更新を

試行しません。この時点で OTP が無効である可能性があるためです。この状況で TGT を取得するには、両方の要素を使用して再度認証する必要があります。

22.3.2. FIPS モードで実行している IdM サーバーで RADIUS プロキシーを設定するために必要な設定

Federal Information Processing Standard(FIPS) モードでは、OpenSSL はデフォルトで MD5 ダイジェストアルゴリズムの使用を無効にします。したがって、RADIUS プロトコルでは、RADIUS クライアントと RADIUS サーバー間のシークレットを暗号化するのに MD5 が必要になるため、FIPS モードで MD5 が利用できないと、IdM RADIUS プロキシーサーバーが失敗します。

RADIUS サーバーが IdM マスターと同じホストで実行されている場合は、以下の手順に従って問題を回避し、セキュアなメーター内で MD5 を有効にすることができます。

1. 以下の内容で **/etc/systemd/system/radiusd.service.d/ipa-otp.conf** ファイルを作成します。

```
[Service]
Environment=OPENSSL_FIPS_NON_APPROVED_MD5_ALLOW=1
```

2. **systemd** 設定をリロードします。

```
# systemctl daemon-reload
```

3. **radiusd** サービスを起動します。

```
# systemctl start radiusd
```

22.3.3. 2 つのファクター認証の有効化

OTP に関連する利用可能な認証方法の詳細は、「利用可能な OTP 認証方法」 を参照してください。

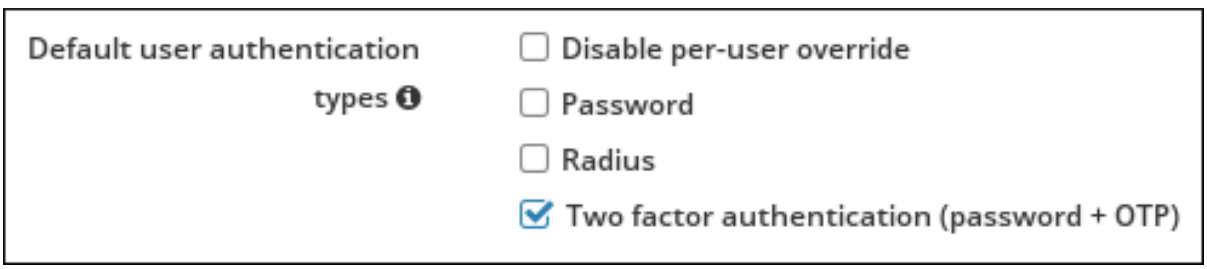
以下を使用して 2 つのファクター認証を有効にするには、以下を実行します。

- Web UI は 「Web UI: 2 つのファクター認証の有効化」 を参照してください。
- コマンドラインは、「コマンドライン: 2 二要素認証の有効化」を参照してください。

Web UI: 2 つのファクター認証の有効化
すべての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** 認証タイプを選択します。

図22.4 ユーザー認証方法

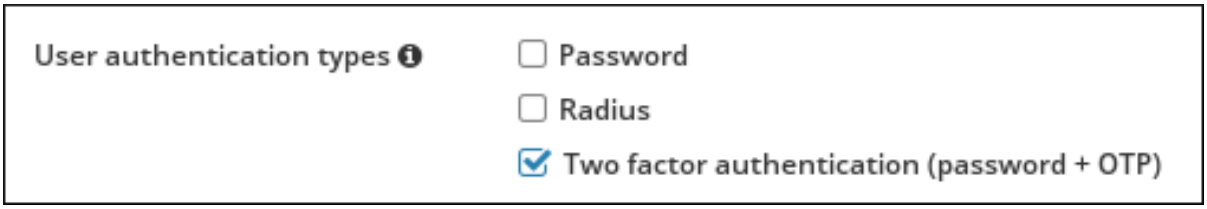


グローバル設定がユーザー別の設定で上書きされないようにするには、**Disable per-user override** を選択します。**Disable per-user override** を選択しない場合は、ユーザーごとに設定された認証方法がグローバル設定よりも優先されます。

認証方法はユーザーごとに個別に設定するには、以下を実行します。

1. **Identity → Users** を選択し、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 **ユーザー認証タイプ**を選択します。

図22.5 ユーザー認証方法



コマンドライン: 2 二要素認証の有効化
すべての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 2 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 で使用できる値のリストは、**ipa config-mod --help** コマンドを実行します。

2. ユーザーごとの上書きを無効にするには、グローバル設定がユーザーごとの設定で上書きされないようにするには **--user-auth-type=disabled** オプションを追加します。たとえば、グローバル認証方法を 2 要素認証に設定し、ユーザーごとの上書きを無効にするには、以下を実行します。

```
$ ipa config-mod --user-auth-type=otp --user-auth-type=disabled
```

--user-auth-type=disabled を設定しないと、ユーザーごとに設定された認証方法がグローバル設定よりも優先されます。

指定されたユーザーに認証方法を個別に設定するには、以下を実行します。

- **ipa user-mod --user-auth-type** コマンドを実行します。たとえば、**user** が 2 要素認証を使用するために必要な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、すべてのユーザーにパスワードと 2 要素認証を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
```

22.3.4. ユーザー管理のソフトウェアトークンの追加

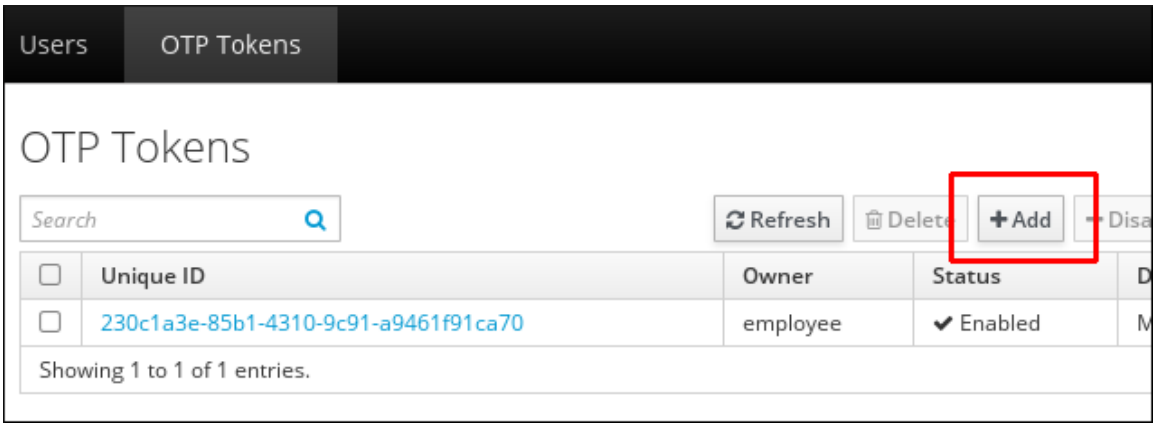
1. 標準のパスワードでログインします。

2. **FreeOTP Authenticator** アプリケーションがモバイルデバイスにインストールされていることを確認します。**FreeOTP Authenticator** をダウンロードするには、FreeOTP のソースページ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** タブの **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、**OTP Tokens** タブは **Authentication** タブから アクセスできます。

     図22.6 ユーザーの 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-cb32fecd548a
  Type: TOTP
...
```

     Ipa otptoken-add の詳細は、**--help** オプションを追加してコマンドを実行します。

4. QR コードは、Web UI またはコマンドラインに表示されます。**FreeOTP Authenticator** で QR コードをスキャンし、モバイルデバイスにトークンをプロビジョニングします。

22.3.5. ユーザー管理の YubiKey ハードウェアトークンの追加

YubiKey トークンなどの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しか追加できません。トークンを所有するユーザーとして YubiKey ハードウェアトークンを追加するには、以下を実行します。

1. 標準のパスワードでログインします。

2. YubiKey トークンを挿入します。

3. **ipa otptoken-add-yubikey** コマンドを実行します。

   - YubiKey に空のスロットが利用可能な場合は、このコマンドにより、空のスロットが自動的に選択されます。

   - 空のスロットが使用できない場合は、**--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
```

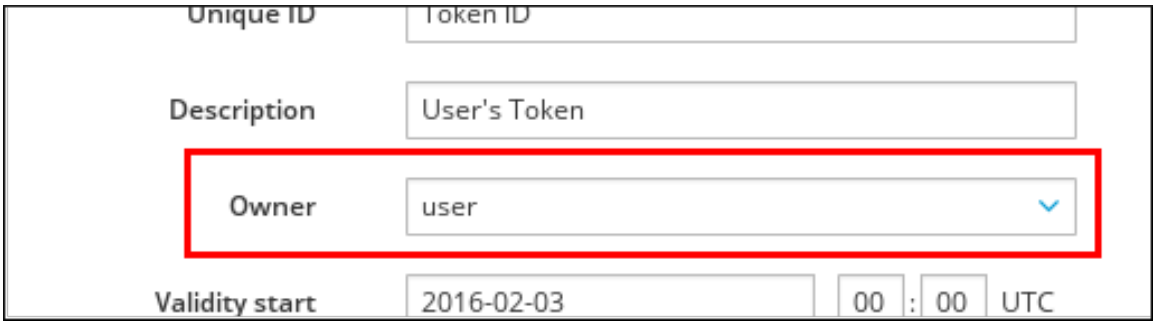
これにより、選択したスロットが上書きされる点に注意してください。

22.3.6. 管理者としてのユーザーのトークンの追加

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. **FreeOTP Authenticator** アプリケーションがモバイルデバイスにインストールされていることを確認します。**FreeOTP Authenticator** をダウンロードするには、FreeOTP のソースページ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 → OTPTokens** を選択し、OTP トークンのリストの上部にある **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 管理者管理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. QR コードは、Web UI またはコマンドラインに表示されます。**FreeOTP Authenticator** で QR コードをスキャンし、モバイルデバイスにトークンをプロビジョニングします。

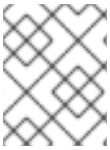
管理者として、YubiKey トークンなどのプログラム可能なハードウェアトークンを追加するには、以下を実行します。

1. 管理者としてログインしていることを確認します。
2. 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
```

22.3.7. プロプライエタリー OTP ソリューションからの移行

IdM は、プロプライエタリー OTP ソリューションから IdM ネイティブの OTP ソリューションへの大規模なデプロイメントの移行を可能にするため、IdM では、ユーザーのサブセットに対して OTP 検証をサードパーティーの RADIUS サーバーにオフロードすることができます。管理者は、各プロキシーが単一の RADIUS サーバーのみを参照できる RADIUS プロキシーのセットを作成します。複数のサーバーに対応する必要がある場合は、複数の RADIUS サーバーを参照する仮想 IP ソリューションを作成することが推奨されます。このようなソリューションは、keepalived デーモンなどを使用して、RHEL IdM の外部で構築する必要があります。次に、管理者はこれらのプロキシーセットのいずれかをユーザーに割り当てます。ユーザーが RADIUS プロキシーが設定されている限り、IdM は他のすべての認証メカニズムをバイパスします。



注記

IdM は、サードパーティーシステムのトークンに対するトークン管理または同期のサポートを提供しません。

OTP 検証用に RADIUS サーバーを設定し、ユーザーをプロキシーサーバーに追加するには、以下を実行します。

1. **radius** 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。詳細は「2 つのファクター認証の有効化」を参照してください。
2. **ipa radiusproxy-add *proxy_name* --secret *secret*** コマンドを実行して RADIUS プロキシーを追加します。このコマンドは、必要な情報を挿入するように求められます。

   RADIUS プロキシーの設定には、クライアントとサーバーとの間の共通のシークレットを使用して認証情報をラップする必要があります。***--secret*** パラメーターにこのシークレットを指定します。
3. **ipa user-mod radiususer --radius=*proxy_name*** コマンドを実行して、追加したプロキシー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*** コマンドを実行して、RADIUS に送信されるユーザー名を設定します。

これにより、ユーザー OTP 認証は RADIUS プロキシーサーバーを介して処理されます。

注記

FIPS モードが有効になっている IdM マスターで RADIUS サーバーを実行するには、「FIPS モードで実行している IdM サーバーで RADIUS プロキシーを設定するために必要な設定」 で説明されている手順を実行します。

ユーザーが IdM ネイティブ OTP システムに移行する準備ができたら、ユーザーの RADIUS プロキシー割り当てを削除するだけです。

22.3.7.1. 低速ネットワークでの RADIUS サーバーを実行する場合の KDC のタイムアウト値の変更

低速なネットワークで RADIUS プロキシーを実行する場合など、IdM KDC は、ユーザーがトークンに入るのを待たずに接続をタイムアウトしたため、RADIUS サーバーが応答する前に接続を閉じます。

KDC のタイムアウト設定を変更するには、以下を実行します。

1. **/var/kerberos/krb5kdc/kdc.conf** ファイルの **[otp]** セクションで ***timeout*** パラメーターの値を変更します。たとえば、タイムアウトを **120** 秒に設定するには、次のコマンドを実行します。

```
[otp]
DEFAULT = {
  timeout = 120
  ...
}
```

2. **krb5kdc** サービスを再起動します。

```
# systemctl restart krb5kdc
```

22.3.8. 現在の認証情報の 2 要素認証へのプロモート

パスワードと 2 要素認証の両方が設定されている場合、パスワードを使用して認証される場合は、特定サービスまたはホストへのアクセスを拒否することができます (「ユーザーの認証方法に基づいたサービスとホストへのアクセス制限」を参照)。このような場合には、再度認証することで、認証情報を 1 要素から 2 要素認証にプロモートします。

1. 画面をロックします。画面をロックするデフォルトのキーボードショートカットは **Super キー+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。

22.3.9. OTP トークンの再同期

「同期されていない OTP トークン」を参照してください。

22.3.10. ロット OTP トークンの置き換え

以下の手順では、OTP トークンを紛失したユーザーがトークンを置き換える方法を説明します。

1. 管理者として、ユーザーのパスワードと OTP 認証を有効にします。

```
[admin@server]# ipa user-mod --user-auth-type=password --user-auth-type=otp user_name
```

2. ユーザーは新しいトークンを追加できるようになりました。たとえば、説明に **New Token** が設定された新しいトークンを追加するには、以下を実行します。

```
[user@server]# ipa otptoken-add --desc="New Token"
```

詳細は、***ipa otptoken-add --help*** パラメーターを追加してコマンドを入力します。

3. ユーザーは、古いトークンを削除できるようになりました。
   a. 必要に応じて、アカウントに関連付けられたトークンをリスト表示します。

```
[user@server]# ipa otptoken-find
---------------------
```

```
2 OTP tokens matched
-------------------
  Unique ID: 4ce8ec29-0bf7-4100-ab6d-5d26697f0d8f
  Type: TOTP
  Description: New Token
  Owner: user

  Unique ID: e1e9e1ef-172c-4fa9-b637-6b017ce79315
  Type: TOTP
  Description: Old Token
  Owner: user
----------------------------
Number of entries returned 2
----------------------------
```

b. 古いトークンを削除します。たとえば、**e1e9e1ef-172c-4fa9-b637-6b017ce79315** ID でトークンを削除するには、以下を実行します。

```
[user@server]# # ipa otptoken-del e1e9e1ef-172c-4fa9-b637-6b017ce79315
----------------------------------------------------------
Deleted OTP token "e1e9e1ef-172c-4fa9-b637-6b017ce79315"
----------------------------------------------------------
```

4. 管理者として、ユーザーの OTP 認証のみを有効にします。

```
[admin@server]# ipa user-mod --user-auth-type=otp user_name
```

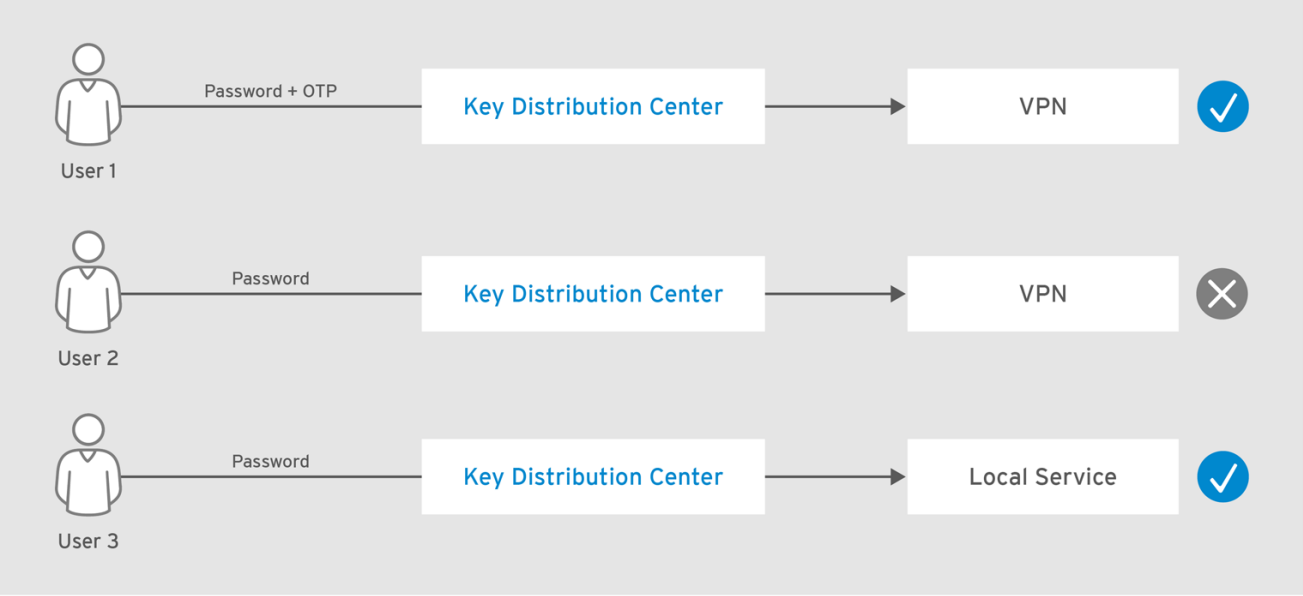
22.4. ユーザーの認証方法に基づいたサービスとホストへのアクセス制限

IdM がサポートする認証メカニズムは認証強度によって異なります。たとえば、ワンタイムパスワード (OTP) を使用した認証は、標準のパスワードを使用した認証のみであると見なされます。本セクションでは、ユーザーの認証方法に基づいてサービスとホストへ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ため、VPN などのセキュリティーに重要なサービス
- ローカルログインなどの非クリティカルなサービスは、弱い認証を可能にするが、より便利な認証方法になります。

図22.8 複数の異なるメソッドを使用した認証の例



RHEL_404973_1016

認証インジケーター
サービスおよびホストへのアクセスは、*authentication indicators* で定義されます。

- サービスまたはホストエントリーに含まれるインジケーターは、ユーザーがそのサービスまたはホストへのアクセスに使用でき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ーターは、チケットの取得に使用された認証方法を示します。

プリンシパルのインジケーターが TGT のインジケーターと一致しない場合は、ユーザーはアクセスが拒否されます。

22.4.1. 特定の認証方法を要求するホストまたはサービスを設定

以下を使用してホストまたはサービスを設定するには、以下を実行します。

- Web UI は、「Web UI: ホストまたはサービスを特定の認証方法を要求する設定」を参照してください。
- コマンドラインは、「コマンドライン: ホストまたはサービスを特定の認証方法を要求する設定」を参照してください。

Web UI: ホストまたはサービスを特定の認証方法を要求する設定

1. **Identity → Hosts** または **Identity → Services** を選択します。
2. 必要な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、必要な認証方法を選択します。
   - たとえば、**OTP** を選択すると、パスワードで有効な OTP コードを使用しているユーザーのみがホストまたはサービスにアクセスできます。
   - **OTP** と **RADIUS** の両方を選択する場合は、アクセスを許可するのに OTP または RADIUS のどちらかを選択できます。

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: ホストまたはサービスを特定の認証方法を要求する設定

1. **オプション:lpa host-find** コマンドまたは **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. **lpa 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、必要な認証インジケーターを追加します。**--auth-ind** で使用できる値のリストは、**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** は、パスワードで有効な OTP コードを使用しているユーザーのみが、ホストまたはサービスにアクセスできるようにします。

   ```
   $ 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ーを追加すると、アクセスを許可するのに十分な OTP または RADIUS で十分です。

22.4.2. Kerberos 認証の変更

デフォルトでは、Identity Management(IdM) は、**PKINIT** 事前認証プラグインを使用して Kerberos 認証用の証明書マッピングに **pkinit** インジケーターを使用します。認証プロバイダーを変更する必要がある場合は、Kerberos Distribution Center(KDC) が TGT(Ticket-Granting Ticket) に挿入します。以下のように、**PKINIT** 機能を提供するすべての IdM マスターで設定を変更します。

1. **/var/kerberos/krb5kdc/kdc.conf** ファイルで、***pkinit_indicator*** パラメーターを **[kdcdefaults]** セクションに追加します。

   ```
   # pkinit_indicator = indicator
   ```

   以下の値を設定できます。

   - **OTP** - 2 要素認証
   - **RADIUS** ベースの認証用の RADIUS
   - スマートカード認証用の **pkinit**

2. **krb5kdc** サービスを再起動します。

   ```
   # systemctl restart krb5kdc
   ```

22.5. ユーザーの公開 SSH 鍵の管理

Identity Management を使用すると、公開 SSH キーをユーザーエントリーにアップロードできます。対応する SSH 鍵にアクセスできるユーザーは、**ssh** を使用して Kerberos 認証情報を使用せずに IdM マ

シン にログインすることができます。**Pam_krb5** が正しく設定されている場合や、SSSD が IdM サーバーのアイデンティティープロバイダーとして使用されている場合は、ユーザーはログイン後に Kerberos チケット保証チケット (TGT) も受け取ります。詳細は、「Kerberos チケットの自動取得」 を参照してください。

SSH 秘密鍵ファイルが利用できない場合でも、ユーザーは Kerberos 認証情報を提供して認証できることに注意してください。

SSH キー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時に、SSSD は、ユーザーおよびホストの SSH 鍵をキャッシュし、取得するようにマシンに自動的に設定されます。これにより、IdM は SSH 鍵の汎用および集中化されたリポジトリーとして機能できます。

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は、マシンで SSSD を手動で設定できます。その方法は、「OpenSSH サービスのキャッシュを提供するように SSSD を設定」 を参照してください。SSSD による SSH 鍵のキャッシュには、ローカルマシンでの管理権限が必要です。

SSH キーの形式要件

IdM では、以下の 2 つの SSH 鍵形式を使用できます。

OpenSSH-style key

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-style key

この形式に関する詳細は、RFC 4253 を参照してください。

IdM は、IdM の LDAP サーバーに保存する前に、RFC 4253 形式の鍵を OpenSSH スタイルの鍵に自動的に変換することに注意してください。

id_rsa.pub などのキーファイルは、キータイプ、キー自体、追加のコメントまたは識別子の 3 つの部分で設定されます。以下の例では、キータイプは RSA で、コメントは鍵を **client.example.com** のホスト名に関連付けます。

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQDMM4xPu54Kf2dx7C4Ta2F7vnIzuL1i6P21TTKniSkjFuA+r

qW06588e7v14Im4VejwnNk352gp49A62qSVOzp8IKA9xdtyRmHYCTUvmkcyspZvFRI713zfRKQVFyJO
qHmW/m
dCmak7QBxYou2ELSPhH3pe8MYTQIuIKDSu5Zbsrqedg1VGkSJxf7mDnCSPNWWzAY9AFB9Lmd2m
2xZmNgVAQEQ
nZXNMaIIroLD/51rmMSkJGHGb1O68kEq9Z client.example.com
```

キーを IdM にアップロードする場合は、3 つのキーの部分すべてをアップロードするか、キー自体のみをアップロードします。キー自体をアップロードした場合、IdM は、アップロードした鍵から RSA や DSA などの鍵タイプを自動的に識別します。

22.5.1. SSH キーの生成

SSH キーは、OpenSSH の **ssh-keygen** ユーティリティーを使用して生成できます。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:GAUIDVVEgly7rs1ITWP6oguHz8BKvyZkpqCqVSsmi7c user@example.com
The key's randomart image is:
+--[ RSA 2048]----+
|             |
|    + .      |
|   + =   .   |
|    =   +    |
|   . E S..   |
|  .    . .o  |
|   . .  . oo.|
|  . o .  +.+o|
|   o  .o..o+o|
+-----------------+
```

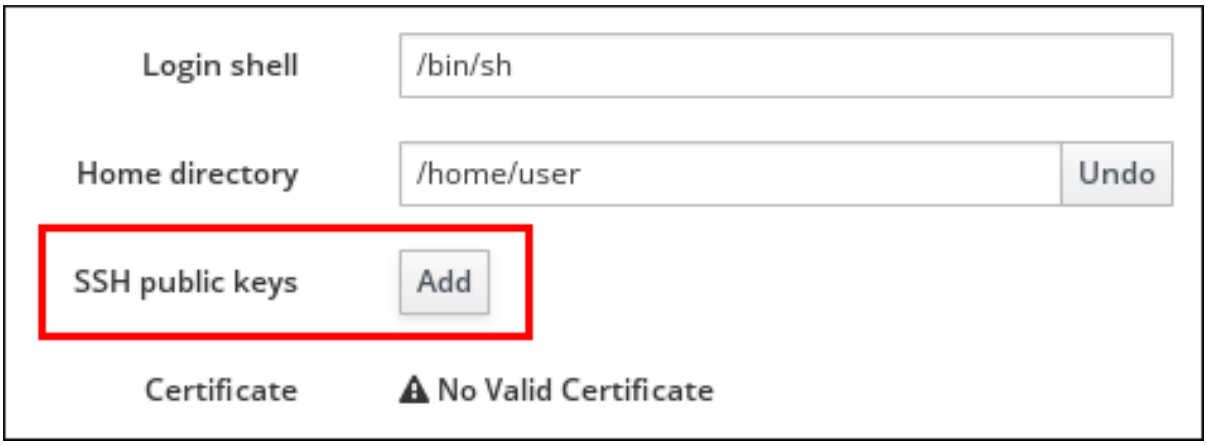
ユーザーの SSH キーをアップロードするには、表示されるファイルに保存されている公開鍵文字列を使用します。

22.5.2. ユーザーの SSH 鍵のアップロード

22.5.2.1. 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** をクリックします。

   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文字列に貼り付け、**Set** をクリックします。

図22.10 公開鍵での貼り付け

5. ページ上部にある **Save** をクリックします。

22.5.2.2. 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** オプションを使用して Base 64 でエンコードされた公開鍵文字列を渡します。

たとえば、キータイプ、キー自体、およびホスト名識別子をアップロードするには、次のコマンドを実行します。

```
$ ipa user-mod user --sshpubkey="ssh-rsa AAAAB3Nza...SNc5dv== client.example.com"
```

複数のキーをアップロードするには、**--sshpubkey** を複数回使用します。たとえば、SSH 鍵を 2 つアップロードするには、次のコマンドを実行します。

```
--sshpubkey="AAAAB3Nza...SNc5dv==" --sshpubkey="RjlzYQo...ZEt0TAo="
```

注記

キー文字列をコマンドラインに手動で貼り付ける代わりに、コマンドリダイレクトを使用して、キーを含むファイルを参照することができます。以下に例を示します。

```
$ ipa user-mod user --sshpubkey="$(cat ~/.ssh/id_rsa.pub)" --sshpubkey="$(cat
~/.ssh/id_rsa2.pub)"
```

22.5.3. ユーザーキーの削除

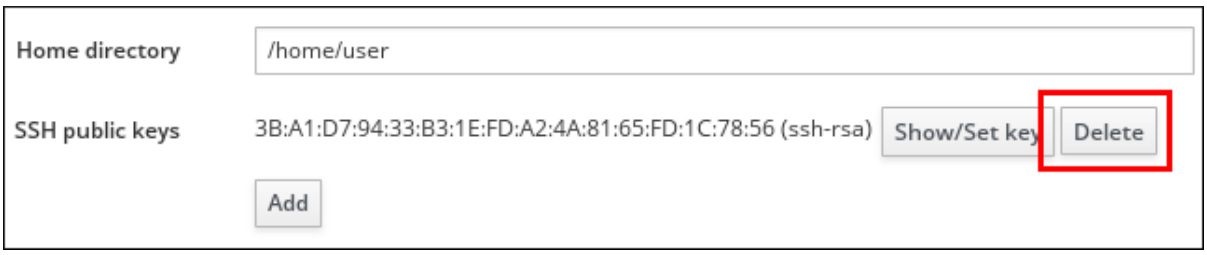
SSH キーを削除するには、以下を実行します。

- Web UI の使用については、「Web UI: ユーザーの SSH キーの削除」 を参照してください。
- コマンドラインの使用方法については、「コマンドライン: ユーザーの SSH キーの削除」 を参照してください。

22.5.3.1. Web UI: ユーザーの SSH キーの削除

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、削除するキーの横にある **Delete** をクリックします。

図22.11 ユーザー SSH 公開鍵の削除



4. ページ上部にある **Save** をクリックします。

22.5.3.2. コマンドライン: ユーザーの SSH キーの削除

ユーザーアカウントに割り当てられたすべての SSH キーを削除するには、キーを指定せずに **--sshpubkey** オプションを **ipa user-mod** コマンドに追加します。

```
$ ipa user-mod user --sshpubkey=
```

特定の SSH キーまたはキーのみを削除する場合は、**--sshpubkey** オプションを使用して、保持するキーまたはキーを指定します。

注記

このコマンドは、キャッシュから SSH 鍵をすぐに削除しません。デフォルトのキャッシュタイムアウト値 (***entry_cache_timeout = 5400***) では、キーが 1 時間半の間キャッシュに残ります。

22.6. OPENSSH サービスのキャッシュを提供するように SSSD を設定

SSSD(System Security Services Daemon) は、OpenSSH を含む複数のシステムサービスにインターフェイスを提供します。

本セクションでは、マシンおよびユーザーの SSH 鍵をキャッシュするように SSSD を設定する方法を説明します。

22.6.1. OpenSSH での SSSD の仕組み

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識別する *public-private key pairs* に基づいて、2 つのシステム間で暗号化された接続を作成します。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、マシンおよびユーザーの 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、以下が行われます。

1. OpenSSH は、SSSD を参照して、キャッシュされた鍵を確認するように設定されています。

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 の鍵キャッシュとして SSSD を使用できます。Windows マシンなどの他のマシンはできません。

SSSD によるホストキーの管理方法
ホストキーを管理するには、SSSD は以下を実行します。

1. ホストシステムからパブリックホストキーを取得します。
2. ホストキーを **/var/lib/sss/pubconf/known_hosts** ファイルに保存します。
3. ホストマシンで接続を確立します。

必要な設定手順については、「ホストキーに SSSD を使用するように OpenSSH の設定」 を参照してください。

SSSD によるユーザーキーの管理方法
ユーザーキーを管理するには、SSSD は以下を実行します。

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. ユーザーキーを標準の認証鍵形式で **.ssh/sss_authorized_keys** ファイルに保存します。

必要な設定手順については、「ユーザーキーに SSSD を使用するように OpenSSH の設定」 を参照してください。

22.6.2. ホストキーに SSSD を使用するように OpenSSH の設定

ユーザーごとまたはシステム全体で設定を変更できます。

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、SSH クライアント (必要な引数とホスト名の **sss_ssh_knownhostsproxy** ユーティリティー) への接続に使用されるコマンドを指定します。

   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 ホストファイルの場所を指定します (**/var/lib/sss/pubconf/known_hosts**)。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されます。

以下の例では、SSH が SSSD ドメインで公開鍵を検索し、提供された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 の設定および設定ファイルの詳細は、ssh_config(5) の man ページを参照してください。

22.6.3. ユーザーキーに SSSD を使用するように OpenSSH の設定

システム全体の設定を変更できます。

1. **/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、ユーザーキー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、コマンドが実行されるアカウントのユーザーを指定します。

以下の例では、**ユーザー**アカウントで **sss_ssh_authorizedkeys** ユーティリティーを実行するように SSH を設定します 。

```
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 の設定および設定ファイルの詳細は、ssh_config(5) の man ページを参照してください。

22.7. IDENTITY MANAGEMENT でのスマートカード認証

Identity Management でのスマートカード認証の詳細は、23章*Identity Management でのスマートカード認証* を参照してください。

22.8. ユーザー証明書

ユーザー証明書の詳細は、24章 *ユーザー、ホスト、およびサービスの証明書の管理* 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

スマートカードに基づいた認証は、パスワードの代替手段です。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、スマートカードをリーダーに配置し、スマートカードの PIN コードを提供します。

本章では、Identity Management でスマートカード認証を管理者が設定する方法と、ユーザーがスマートカードを使用して Identity Management に認証する方法を説明します。

23.1. 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には、以下のようにします。

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証明書をファイルにデプロイメントします。たとえば、Base64 形式の証明書を **user.crt** という名前のファイルに抽出するには、次のコマンドを実行します。

```
$ 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。

23.2. IDENTITY MANAGEMENT に証明書マッピングルールを設定

23.2.1. スマートカードにおける認証を設定するための証明書マッピングルール

証明書マッピングルールは、Identity Management (IdM) 管理者が特定のユーザーの証明書にアクセスしない場合に、シナリオで証明書を使用して認証できるため便利な方法です。通常、このようなアクセスがない理由は、証明書が外部認証局によって発行されたためです。特別なユースケースは、IdM ドメインが信頼関係にある Active Directory (AD) の証明書システムが発行した証明書によって表されます。

証明書マッピングルールは、スマートカードを使用するユーザーが多く、IdM 環境が大きい場合にも便利です。このような場合、完全な証明書を追加すると複雑になります。ほとんどの場合、発行先と発行者は予測可能であるため、完全な証明書よりも簡単に追加できます。システム管理者は、証明書マッピングルールを作成し、特定のユーザーに証明書を発行する前に、ユーザーエントリーに証明書マッピングデータを追加できます。証明書が発行されると、完全な証明書が自分のエントリーにアップロードされていなくても、ユーザーは証明書を使用してログインできるようになります。

さらに、証明書は一定間隔で更新する必要があるため、証明書マッピングルールは管理のオーバーヘッドを軽減します。ユーザーの証明書を更新する際に、管理者がユーザーエントリーを更新する必要がありません。たとえば、マッピングが ***Subject*** と ***Issuer*** の値に基づいている場合、および新しい証明書

の Subject と Issuer が以前と同じ場合は、マッピングは引き続き適用されます。一方で、完全な証明書を使用した場合、管理者は古い証明書に置き換わる新しい証明書をユーザーエントリーにアップロードする必要があります。

証明書マッピングを設定するには、以下を実行します。

1. 管理者は、証明書マッピングデータ (通常は発行者と題名)、または完全な証明書をユーザーアカウントに読み込む必要があります。

2. ユーザーが IdM へのログインを問題なく行えるようにするために、管理者が証明書マッピングルールを作成する必要があります。

   - アカウントに、証明書マッピングデータエントリーが含まれる
   - 証明書マッピングデータエントリーが、証明書の情報と一致する

   マッピングルールを設定する個々のコンポーネントの詳細と、そのコンポーネントの取得方法および使用方法は、IdM での ID マッピングルールのコンポーネントおよび一致するルールで使用する証明書の発行者の取得を参照してください。

23.2.1.1. Active Directory ドメインとの信頼に対する証明書マッピングルール

本セクションでは、IdM デプロイメントが Active Directory (AD) ドメインと信頼関係にある場合に可能な、別の証明書マッピングのユースケースを簡単に説明します。

証明書マッピングルールは、信頼された AD 証明書システムが発行したスマートカード証明書を持つユーザーに対して、IdM リソースにアクセスするのに便利な方法です。AD 設定によっては、以下の状況が考えられます。

- 証明書が AD で発行され、ユーザーと証明書が IdM に保存されている場合、マッピングと、認証リクエストの全処理は IdM 側で行われます。このシナリオの設定に関する詳細は、「IdM に保存されたユーザーの証明書マッピングの設定」 を参照してください。
- ユーザーが AD に保存されている場合は、認証要求の処理が AD で実行されます。サブケースは 3 つあります。
  - AD ユーザーエントリーに、証明書全体が含まれる場合。このシナリオで IdM を設定する方法は、「AD ユーザーエントリーに証明書全体が含まれるユーザーに証明書マッピングを設定」 を参照してください。
  - AD が、ユーザー証明書をユーザーアカウントにマップするように設定されている場合。この場合、AD ユーザーエントリーには証明書全体が含まれず、代わりに ***altSecurityIdentities*** と呼ばれる属性が含まれます。このシナリオで IdM を設定する方法は、「ユーザー証明書をユーザーアカウントにマッピングするように AD が設定されている場合に、証明書マッピングの設定」 を参照してください。
  - AD ユーザーエントリーに、証明書全体またはマッピングデータが含まれない場合。この場合の解決策として、**ipa idoverrideuser-add** コマンドを使用して、IdM で AD ユーザーの ID オーバーライドに証明書全体を追加します。詳細は、「AD ユーザーエントリーに証明書やマッピングデータが含まれていない場合に、証明書マッピングの設定」を参照してください。

23.2.1.2. IdM における ID マッピングルールのコンポーネント

本セクションでは、IdM の ID マッピングルール のコンポーネントと、その設定方法を説明します。各コンポーネントには、オーバーライドできるデフォルト値があります。コンポーネントは、Web UI ま

たはコマンドラインで定義できます。コマンドラインで、**ipa certmaprule-add** コマンドを使用して、ID マッピングルールが作成されます。

マッピングルール

マッピングルールコンポーネントでは、証明書を 1 人または複数のユーザーアカウントに関連付けます (またはマップします)。ルールは、証明書を目的のユーザーアカウントに関連付ける LDAP 検索フィルターを定義します。

さまざまな認証局 (CA) が発行する証明書にはさまざまなプロパティーがあり、さまざまなドメインで使用される可能性があります。そのため、IdM はマッピングルールを無条件に適用せず、適切な証明書にのみ適用されます。適切な証明書は、マッチングルールを使用して定義されます。

マッピングルールのオプションを空のままにすると、証明書は、DER でエンコードされたバイナリーファイルとして、***userCertificate*** 属性で検索されることに注意してください。

--maprule オプションを使用して、コマンドラインでマッピングルールを定義します。

マッチングルール

ドメインリストは、ID マッピングルールの処理時に IdM がユーザーを検索する ID ドメインを指定します。このオプションを指定しないと、IdM は、IdM クライアントが所属しているローカルドメイン内でのみユーザーを検索します。

--domain オプションを使用して、コマンドラインでドメインを定義します。

優先度

複数のルールが証明書に適用される場合は、最も優先度が高いルールが優先されます。その他のルールはすべて無視されます。

- 数値が低いほど、ID マッピングルールの優先度が高くなります。たとえば、優先度 1 のルールは、優先度 2 のルールよりも高く設定されています。
- ルールに優先度の値が定義されていないと、優先度が最も低くなります。

--priority オプションを使用して、コマンドラインでマッピングルールの優先度を定義します。

例23.1 証明書マッピングルールの例 1

コマンドラインを使用して、その証明書の ***Subject*** が IdM のユーザーアカウントの ***certmapdata*** エントリーと一致している場合に限り、**EXAMPLE.ORG** 組織の スマートカード CA が発行する証明書の認証を可能にする証明書マッピングルール **simple_rule** を定義するには、次のコマンドを実行します。

```
# ipa certmaprule-add simple_rule --matchrule '<ISSUER>CN=Smart Card
CA,O=EXAMPLE.ORG' --maprule '(ipacertmapdata=X509:<I>{issuer_dn!nss_x500}<S>
{subject_dn!nss_x500})'
```

23.2.1.3. マッチングルールで使用する証明書から発行者の取得

この手順では、証明書から発行者情報を取得して、証明書マッピングルールのマッチングルールにコピーする方法を説明します。マッチングルールで必要な発行者の形式を取得するには、**openssl x509** コマンドを使用します。

前提条件

- **.pem** 形式または **.crt** 形式のユーザー証明書がある。

手順

1. 証明書からユーザー情報を取得します。以下のように、**openssl** 証明書の表示および署名ユーティリティーを使用します。

   - リクエストのエンコードされたバージョンの出力を防ぐ **-noout** オプション
   - 発行者名を出力する **-issuer** オプション
   - 証明書を読み込む入力ファイル名を指定する **-in** オプション
   - **RFC2253** 値と共に **-nameopt** オプションを指定して、最初に最も具体的な相対識別名 (RDN) で出力を表示します。

   入力ファイルに Identity Management 証明書が含まれる場合は、コマンドの出力で、***Organization*** 情報を使用して 発行者が定義されていることを示しています。

   ```
   # openssl x509 -noout -issuer -in idm_user.crt -nameopt RFC2253
   issuer=CN=Certificate Authority,O=REALM.EXAMPLE.COM
   ```

   入力ファイルに Active Directory 証明書が含まれる場合は、コマンドの出力で、***ドメインコンポーネント*** の情報を使用して発行者が定義されていることを示しています。

   ```
   # # openssl x509 -noout -issuer -in ad_user.crt -nameopt RFC2253
   issuer=CN=AD-WIN2012R2-CA,DC=AD,DC=EXAMPLE,DC=COM
   ```

2. 必要に応じて、証明書発行者が、**ad.example.com** ドメインからデプロイメントした **AD-WIN2012R2-CA** であることを指定する マッチングルールに基づいて、コマンドラインで新しいマッピングルールを作成する場合は、証明書の発行先が、IdM のユーザーアカウントにある ***certmapdata*** エントリーと一致する必要があります。

   ```
   # ipa certmaprule-add simple_rule --matchrule '<ISSUER>CN=AD-WIN2012R2-
   CA,DC=AD,DC=EXAMPLE,DC=COM' --maprule '(ipacertmapdata=X509:<I>
   {issuer_dn!nss_x500}<S>{subject_dn!nss_x500})'
   ```

追加情報

certmap コマンドの詳細は、man ページの sss-certmap(5) を参照してください。ここには、マッチングルールおよびマッピングルールで対応している形式の紹介と、優先順位およびドメインフィールドの説明もあります。

23.2.2. IdM に保存されたユーザーの証明書マッピングの設定

このセクションでは、証明書認証が設定されているユーザーが IdM に保存されている場合に、システム管理者が IdM での証明書マッピングを有効にする必要がある手順を説明します。

前提条件

- IdM にユーザーがアカウントがある。
- 管理者が、ユーザーエントリーに追加する証明書全体または証明書マッピングデータのいずれかを所有している。

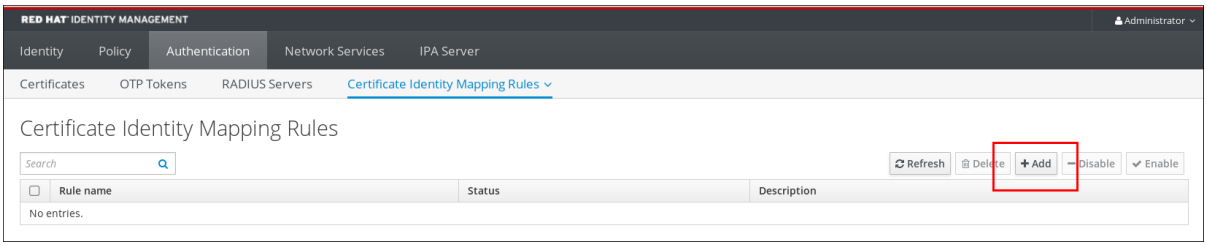
23.2.2.1. IdM での証明書マッピングルールの追加

このセクションでは、マッピングルールおよび証明書マッピングデータエントリーで指定された条件に一致する証明書を持つ IdM ユーザーが IdM に対して認証できるように、証明書マッピングルールを設定する方法を説明します。

23.2.2.1.1. IdM Web UI で証明書マッピングルールの追加

1. 管理者として IdM Web UI にログインします。
2. **Authentication → Certificate Identity Mapping Rules→ Certificate Identity Mapping Rules**に移動します。
3. **Add** をクリックします。

図23.1 IdM Web UI で新しい証明書マッピングルールの追加



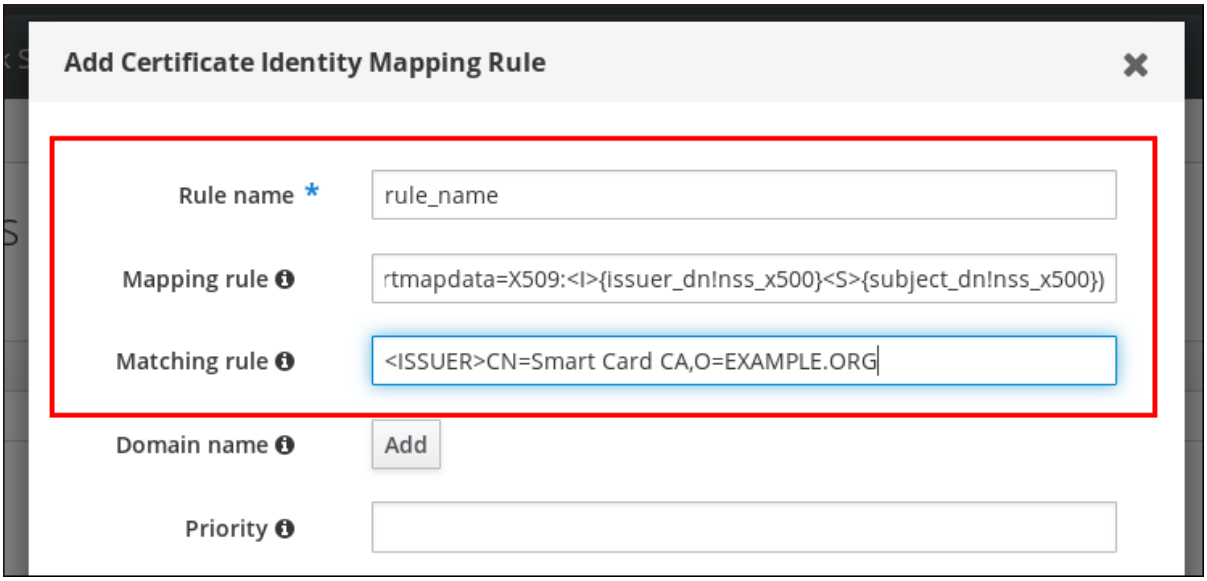
4. ルール名を入力します。
5. マッピングルールを入力します。たとえば、IdM に提示された証明書の ***Issuer*** エントリーおよび ***Subject*** エントリーを IdM で検索し、提示された証明書に含まれるこの 2 つのエントリーで見つかった情報に基づいて認証するかどうかを決定するには、次のコマンドを実行します。

```
(ipacertmapdata=X509:<I>{issuer_dn!nss_x500}<S>{subject_dn!nss_x500})
```

6. マッチングルールを入力します。たとえば、**EXAMPLE.ORG** 組織の **スマートカード CA** が発行する証明書のみが IdM に対して認証できるようにするには、次のコマンドを実行します。

```
<ISSUER>CN=Smart Card CA,O=EXAMPLE.ORG
```

図23.2 IdM Web UI への証明書マッピングルールの詳細の入力



7. ダイアログボックスの下部にある **Add** をクリックして、ルールを追加し、ダイアログボックスを閉じます。

8. System Security Services Daemon (SSSD) は、証明書マッピングルールを定期的に再読み込みします。新たに作成したルールがすぐに読み込まれるようにする場合は、次のコマンドを実行して SSSD を再起動します。

```
# systemctl restart sssd
```

これで、証明書マッピングルールセットが設定され、スマートカードの証明書で検出されたマッピングルールで指定されたデータの種類と、IdM ユーザーエントリーの証明書マッピングデータを比較します。一致するファイルが見つかると、一致するユーザーが認証されます。

23.2.2.1.2. コマンドラインを使用した証明書マッピングルールの追加

1. 管理者の認証情報を取得します。

```
# kinit admin
```

2. マッピングルールを入力し、マッピングルールの基となっているマッチングルールを入力します。たとえば、提示する証明書の ***Issuer*** エントリーおよび ***Subject*** エントリーを IdM で検索し、提示された証明書に含まれるこの 2 つのエントリーで見つかった情報に基づいて認証するかどうかを決定するには、**EXAMPLE.ORG** 組織の **Smart Card CA** が発行する証明書のみ を認識するには、次のコマンドを実行します。

```
# ipa certmaprule-add rule_name --matchrule '<ISSUER>CN=Smart Card
CA,O=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,O=EXAMPLE.ORG
  Enabled: TRUE
```

3. System Security Services Daemon (SSSD) は、証明書マッピングルールを定期的に再読み込みします。新たに作成したルールがすぐに読み込まれるようにする場合は、次のコマンドを実行して SSSD を再起動します。

```
# systemctl restart sssd
```

これで、証明書マッピングルールセットが設定され、スマートカードの証明書で検出されたマッピングルールで指定されたデータの種類と、IdM ユーザーエントリーの証明書マッピングデータを比較します。一致するファイルが見つかると、一致するユーザーが認証されます。

23.2.2.2. IdM のユーザーエントリーへの証明書マッピングデータの追加

本セクションでは、証明書マッピングデータエントリーで指定された値が含まれている場合に限り、ユーザーが複数の証明書を使用して認証できるように、IdM ユーザーエントリーへの証明書マッピングデータを入力する方法を説明します。

23.2.2.2.1. IdM Web UI のユーザーエントリーへの証明書マッピングデータの追加

1. 管理者として IdM Web UI にログインします。
2. **Users → Active users** に移動し、ユーザーエントリーをクリックします。
3. **Certificate mapping data** オプションを見つけ、**Add** をクリックします。
4. 利用できるユーザーの証明書がある場合は、次のコマンドを実行します。

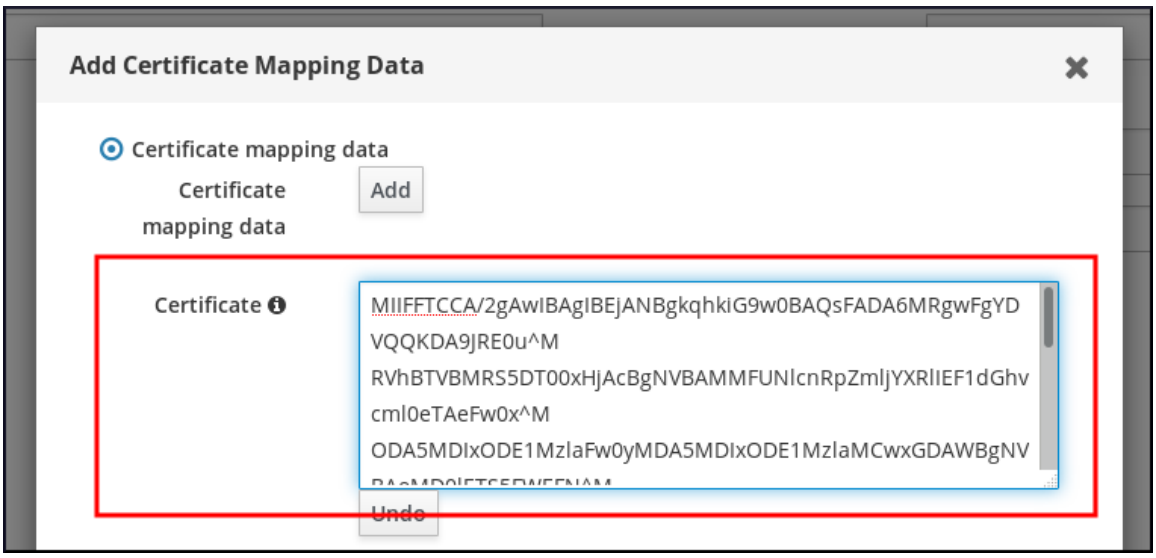
   a. コマンドラインインターフェイスで、**cat** ユーティリティーまたはテキストエディターで証明書を表示します。

   ```
   # [root@server ~]# cat idm_user_certificate.pem
   -----BEGIN CERTIFICATE-----
   MIIFFTCCA/2gAwIBAgIBEjANBgkqhkiG9w0BAQsFADA6MRgwFgYDVQQKDA9JRE0u
   RVhBTVBMRS5DT00xHjAcBgNVBAMMFUNlcnRpZmljYXRlIEF1dGhvcml0eTAeFw0x
   ODA5MDIxODE1MzlaFw0yMDA5MDIxODE1MzlaMCwxGDAWBgNVBAoMD0lETS5FWE
   FN
   [...output truncated...]
   ```

   b. 証明書をコピーします。

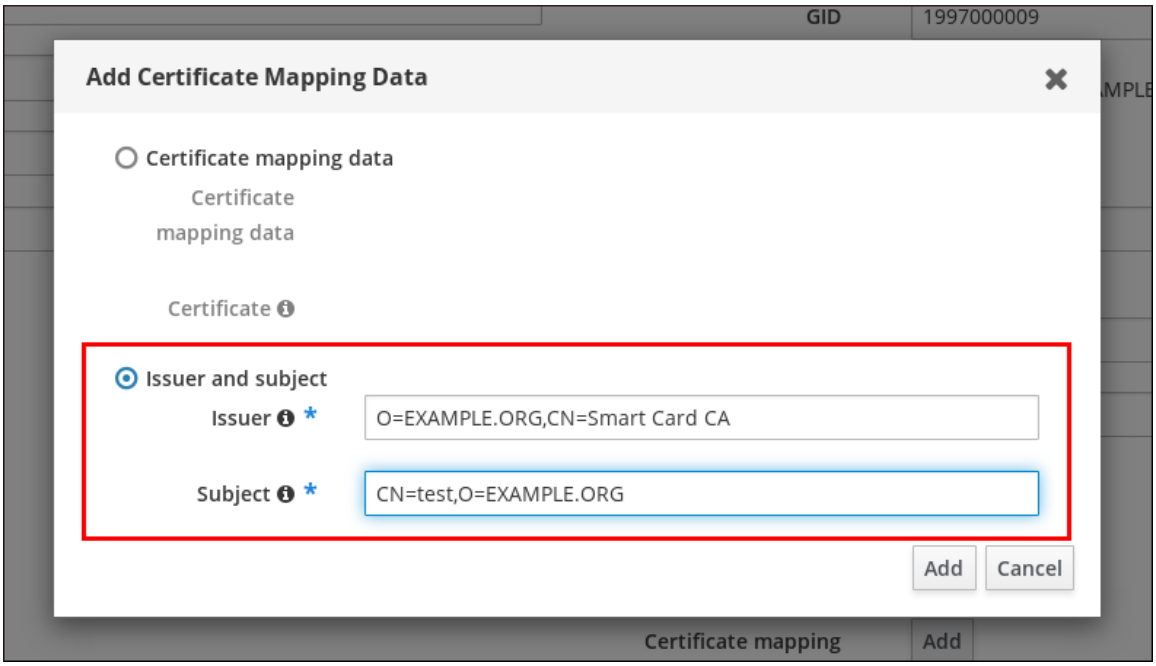
   c. IdM Web UI で、**Certificate** の横にある **Add** をクリックして、開いたウィンドウに証明書を貼り付けます。

   図23.3 ユーザーの証明書マッピングデータの追加 - 証明書

   または、利用できるユーザーの証明書がなくても、証明書の ***Issuer*** および ***Subject*** を知っている場合は、**Issuer and subject** のラジオボタンをオンにして、該当するボックスに値を入力します。

図23.4 ユーザーの証明書マッピングデータの追加 - 発行者および発行先



5. **Add** をクリックします。

6. 必要に応じて、**.pem** 形式の証明書全体へのアクセスがある場合は、ユーザーと証明書がリンクされていることを確認します。

   a. **sss_cache** ユーティリティーを使用して、SSSD キャッシュでユーザーのレコードを無効にし、ユーザーの情報を再読み込みします。

   ```
   # sss_cache -u user_name
   ```

   b. **ipa certmap-match** コマンドに、IdM ユーザーの証明書が含まれるファイルの名前を付けて実行します。

   ```
   # ipa certmap-match idm_user_cert.pem
   --------------
   1 user matched
   --------------
    Domain: IDM.EXAMPLE.COM
    User logins: idm_user
   ----------------------------
   Number of entries returned 1
   ----------------------------
   ```

   この出力では、証明書マッピングデータがユーザーに追加され、「IdM での証明書マッピングルールの追加」で定義された対応するマッピングルールが存在することを確認します。これは、定義した証明書マッピングデータに一致する証明書を使用して、ユーザーとして認証できることを意味します。

23.2.2.2.2. コマンドラインを使用したユーザーエントリーへの証明書マッピングデータの追加

1. 管理者の認証情報を取得します。

   ```
   # kinit admin
   ```

2. 利用できるユーザーの証明書がある場合は、**ipa user-add-cert** コマンドを使用して、証明書をユーザーアカウントに追加します。

```
# CERT=`cat idm_user_cert.pem | tail -n +2 | head -n -1 | tr -d '\r\n'\`
# ipa user-add-certmapdata idm_user --certificate $CERT
```

または、利用できるユーザーの証明書がなくても、ユーザーの証明書の ***Issuer*** および ***Subject*** を知っている場合は、次のコマンドを実行します。

```
# 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
```

3. 必要に応じて、**.pem** 形式の証明書全体へのアクセスがある場合は、ユーザーと証明書がリンクされていることを確認します。

   a. **sss_cache** ユーティリティーを使用して、SSSD キャッシュでユーザーのレコードを無効にし、ユーザーの情報を再読み込みします。

```
# sss_cache -u user_name
```

   b. **ipa certmap-match** コマンドに、IdM ユーザーの証明書が含まれるファイルの名前を付けて実行します。

```
# ipa certmap-match idm_user_cert.pem
--------------
1 user matched
--------------
 Domain: IDM.EXAMPLE.COM
 User logins: idm_user
----------------------------
Number of entries returned 1
----------------------------
```

23.2.3. AD ユーザーエントリーに証明書全体が含まれるユーザーに証明書マッピングを設定

このセクションでは、IdM デプロイメントが Active Directory (AD) を信頼し、そのユーザーが AD に保存され、AD のユーザーエントリーに証明書全体が含まれる場合に、IdM で証明書マッピングを有効にするのに必要な手順を説明します。

前提条件

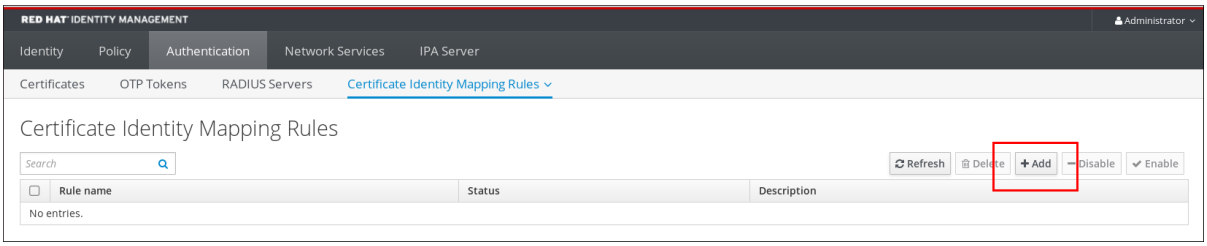
- IdM にユーザーアカウントがない。
- ユーザーに、証明書を含む AD のアカウントがある。
- IdM 管理者が、IdM 証明書マッピングルールが基になっているデータにアクセスできる。

23.2.3.1. IdM Web UI を使用した Whole 証明書が含まれるユーザーの証明書マッピングルールの追加

IdM Web UI で証明書マッピングルールを追加するには、以下のようにします。

1. 管理者として IdM Web UI にログインします。
2. **Authentication → Certificate Identity Mapping Rules→ Certificate Identity Mapping Rules**に移動します。
3. **Add** をクリックします。

図23.5 IdM Web UI で新しい証明書マッピングルールの追加



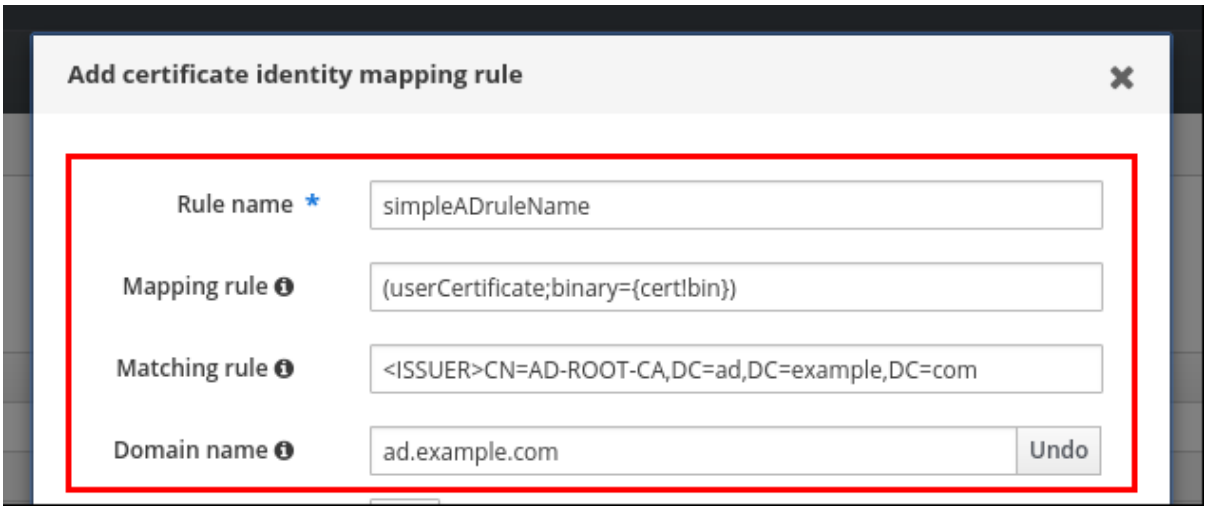
4. ルール名を入力します。
5. マッピングルールを入力します。認証のために IdM に提示された証明書全体を、AD で利用可能な証明書全体と比較するには、次のコマンドを実行します。

```
(userCertificate;binary={cert!bin})
```

6. マッチングルールを入力します。たとえば、**AD.EXAMPLE.COM** ドメインの **AD-ROOT-CA** が発行する証明書のみを認証できるようにするには、次のコマンドを実行します。

```
<ISSUER>CN=AD-ROOT-CA,DC=ad,DC=example,DC=com
```

図23.6 AD に保存されている証明書があるユーザーの証明書マッピングルール



7. **Add** をクリックします。
8. System Security Services Daemon (SSSD) は、証明書マッピングルールを定期的に再読み込みします。新たに作成したルールがすぐに読み込まれるようにする場合は、次のコマンドを実行して SSSD を再起動します。

```
# systemctl restart sssd
```

23.2.3.2. ユーザー選択の AD ユーザーエントリーに、コマンドラインを使用した Whole 証明書が含まれるに対する証明書マッピングルールの追加

コマンドラインで証明書マッピングルールを追加するには、以下を行います。

1. 管理者の認証情報を取得します。

```
# kinit admin
```

2. マッピングルールを入力し、マッピングルールの基となっているマッチングルールを入力します。AD で利用可能な証明書と比較する、認証用に提示される証明書全体を取得して、**AD.EXAMPLE.COM** ドメインの **AD-ROOT-CA** により発行された証明書のみの認証を許可するには、次のコマンドを実行します。

```
# ipa certmaprule-add simpleADrule --matchrule '<ISSUER>CN=AD-ROOT-
CA,DC=ad,DC=example,DC=com' --maprule '(userCertificate;binary={cert!bin})' --domain
ad.example.com
----------------------------------------------------
Added Certificate Identity Mapping Rule "simpleADrule"
----------------------------------------------------
  Rule name: simpleADrule
  Mapping rule: (userCertificate;binary={cert!bin})
  Matching rule: <ISSUER>CN=AD-ROOT-CA,DC=ad,DC=example,DC=com
  Domain name: ad.example.com
  Enabled: TRUE
```

3. System Security Services Daemon (SSSD) は、証明書マッピングルールを定期的に再読み込みします。新たに作成したルールがすぐに読み込まれるようにする場合は、次のコマンドを実行して SSSD を再起動します。

```
# systemctl restart sssd
```

23.2.4. ユーザー証明書をユーザーアカウントにマッピングするように AD が設定されている場合に、証明書マッピングの設定

このセクションでは、IdM デプロイメントが Active Directory (AD) を信頼し、そのユーザーが AD に保存され、AD のユーザーエントリーに証明書マッピングデータが含まれる場合に、IdM で証明書マッピングを有効にするのに必要な手順を説明します。

前提条件

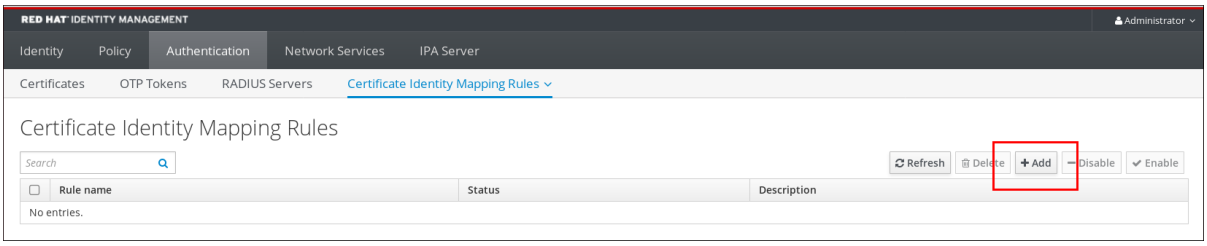
- IdM にユーザーアカウントがない。
- このユーザーに、***altSecurityIdentities*** 属性を含む AD にアカウントがある。AD は、IdM の ***certmapdata*** 属性に相当します。
- IdM 管理者が、IdM 証明書マッピングルールが基になっているデータにアクセスできる。

23.2.4.1. 信頼された AD ドメインがユーザー証明書をマッピングするように設定されている場合に、Web UI を使用した証明書マッピングルールの追加

信頼された AD ドメインがユーザー証明書をマッピングするように設定されている場合に、証明書マッピングルールを追加するには、以下を実行します。

1. 管理者として IdM Web UI にログインします。
2. **Authentication → Certificate Identity Mapping Rules→ Certificate Identity Mapping Rules**に移動します。
3. **Add** をクリックします。

図23.7 IdM Web UI で新しい証明書マッピングルールの追加



4. ルール名を入力します。
5. マッピングルールを入力します。たとえば、提示された証明書で ***Issuer*** エントリーおよび ***Subject*** エントリーを AD DC で検索し、提示された証明書に含まれるこの 2 つのエントリーで見つかった情報に基づいて認証するかどうかを決定するには、次のコマンドを実行します。

```
(altSecurityIdentities=X509:<I>{issuer_dn!ad_x500}<S>{subject_dn!ad_x500})
```

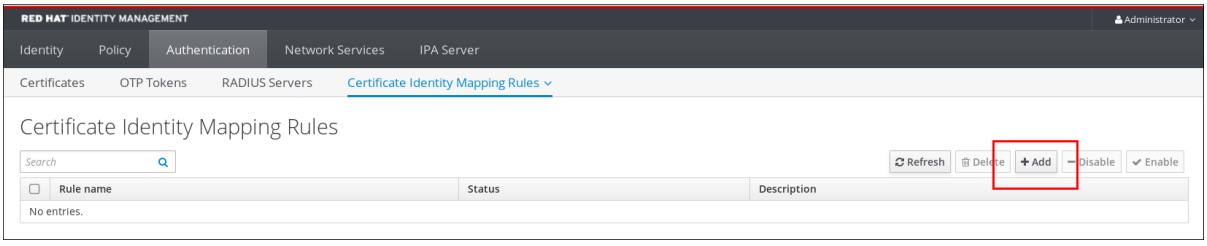
6. マッチングルールを入力します。たとえば、**AD.EXAMPLE.COM** ドメインの **AD-ROOT-CA** が発行する証明書のみを許可し、IdM に対してユーザーを認証するには、次のコマンドを実行します。

```
<ISSUER>CN=AD-ROOT-CA,DC=ad,DC=example,DC=com
```

7. ドメインを入力します。

```
ad.example.com
```

図23.8 AD がマッピング用に設定されている場合の証明書マッピングルール



8. **Add** をクリックします。
9. System Security Services Daemon (SSSD) は、証明書マッピングルールを定期的に再読み込みします。新たに作成したルールがすぐに読み込まれるようにする場合は、次のコマンドを実行して SSSD を再起動します。

```
# systemctl restart sssd
```

23.2.4.2. 信頼された AD ドメインがユーザー証明書をマッピングするように設定されている場合にコマンドラインを使用した証明書マッピングルールの追加

コマンドラインで証明書マッピングルールを追加するには、以下を行います。

1. 管理者の認証情報を取得します。

```
# kinit admin
```

2. マッピングルールを入力し、マッピングルールの基となっているマッチングルールを入力します。たとえば、提示する証明書の ***Issuer*** エントリーおよび ***Subject*** エントリーを AD で検索し、**AD.EXAMPLE.COM** ドメインの **AD-ROOT-CA** により発行された証明書のみを許可するには、次のコマンドを実行します。

```
# ipa certmaprule-add ad_configured_for_mapping_rule --matchrule '<ISSUER>CN=AD-
ROOT-CA,DC=ad,DC=example,DC=com' --maprule '(altSecurityIdentities=X509:<I>
{issuer_dn!ad_x500}<S>{subject_dn!ad_x500})' --domain=ad.example.com
----------------------------------------------------
Added Certificate Identity Mapping Rule "ad_configured_for_mapping_rule"
----------------------------------------------------
  Rule name: ad_configured_for_mapping_rule
  Mapping rule: (altSecurityIdentities=X509:<I>{issuer_dn!ad_x500}<S>
{subject_dn!ad_x500})
  Matching rule: <ISSUER>CN=AD-ROOT-CA,DC=ad,DC=example,DC=com
  Domain name: ad.example.com
  Enabled: TRUE
```

3. System Security Services Daemon (SSSD) は、証明書マッピングルールを定期的に再読み込みします。新たに作成したルールがすぐに読み込まれるようにする場合は、次のコマンドを実行して SSSD を再起動します。

```
# systemctl restart sssd
```

23.2.4.3. AD で証明書マッピングデータの確認

altSecurityIdentities 属性は、IdM の ***certmapdata*** ユーザー属性と同等の Active Directory (AD) です。信頼されている AD ドメインが、ユーザーアカウントにユーザー証明書をマッピングするように設定されている時に IdM で証明書マッピングを設定する場合は、IdM システム管理者が、AD のユーザーエントリーに ***altSecurityIdentities*** 属性が正しく設定されていることを確認する必要があります。

AD に保存されているユーザーに対して、AD が正しい情報が含まれていることを確認する場合は、**ldapsearch** コマンドを使用します。

たとえば、**ad_user** のユーザーエントリーに ***altSecurityIdentities*** 属性が設定されており、**ad_user** が AD の認証に使用する証明書が、**ad.example.com** ドメインの **AD-ROOT-CA** により発行され、発行先が **<S<>DC=com,DC=example,DC=ad,CN=Users,CN=ad_user** であることを matchrule が規定していることを、**adserver.ad.example.com** サーバーに確認する場合は、次のコマンド実行します。

```
$ ldapsearch -o ldif-wrap=no -LLL -h adserver.ad.example.com \
-p 389 -D cn=Administrator,cn=users,dc=ad,dc=example,dc=com \
-W -b cn=users,dc=ad,dc=example,dc=com "(cn=ad_user)" \
altSecurityIdentities
Enter LDAP Password:
```

```
dn: CN=ad_user,CN=Users,DC=ad,DC=example,DC=com
altSecurityIdentities: X509:<I>DC=com,DC=example,DC=ad,CN=AD-ROOT-
CA<S>DC=com,DC=example,DC=ad,CN=Users,CN=ad_user
```

23.2.5. AD ユーザーエントリーに証明書やマッピングデータが含まれていない場合に、証明書マッピングの設定

このセクションでは、IdM デプロイメントが Active Directory (AD) を信頼し、そのユーザーが AD に保存され、AD のユーザーエントリーに証明書全体または証明書マッピングデータが含まれる場合に、IdM で証明書マッピングを有効にするのに必要な手順を説明します。

前提条件

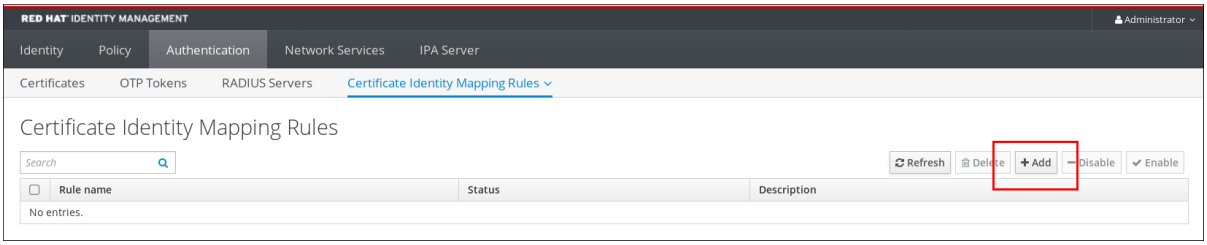
- IdM にユーザーアカウントがない。
- ユーザーのアカウントがある AD に、証明書全体、または ***altSecurityIdentities*** 属性、IdM ***certmapdata*** 属性で AD に相当するものがない。
- IdM 管理者が、IdM に、AD ユーザーの ***ユーザー ID オーバーライド*** に追加する AD ユーザー証明書全体を所有している。

23.2.5.1. AD ユーザーエントリーに証明書やマッピングデータが含まれていない場合に、Web UI を使用した証明書マッピングルールの追加

AD ユーザーエントリーに、証明書やマッピングデータが含まれていない場合に、Web UI を使用して証明書マッピングルールを追加するには、以下を実行します。

1. 管理者として IdM Web UI にログインします。
2. **Authentication → Certificate Identity Mapping Rules→ Certificate Identity Mapping Rules**に移動します。
3. **Add** をクリックします。

図23.9 IdM Web UI で新しい証明書マッピングルールの追加



4. ルール名を入力します。
5. マッピングルールを入力します。認証するために IdM に提示された証明書全体を、IdM の AD ユーザーエントリーのユーザー ID オーバーライドエントリーに保存されている証明書と比較できるようにするには、次のコマンドを実行します。

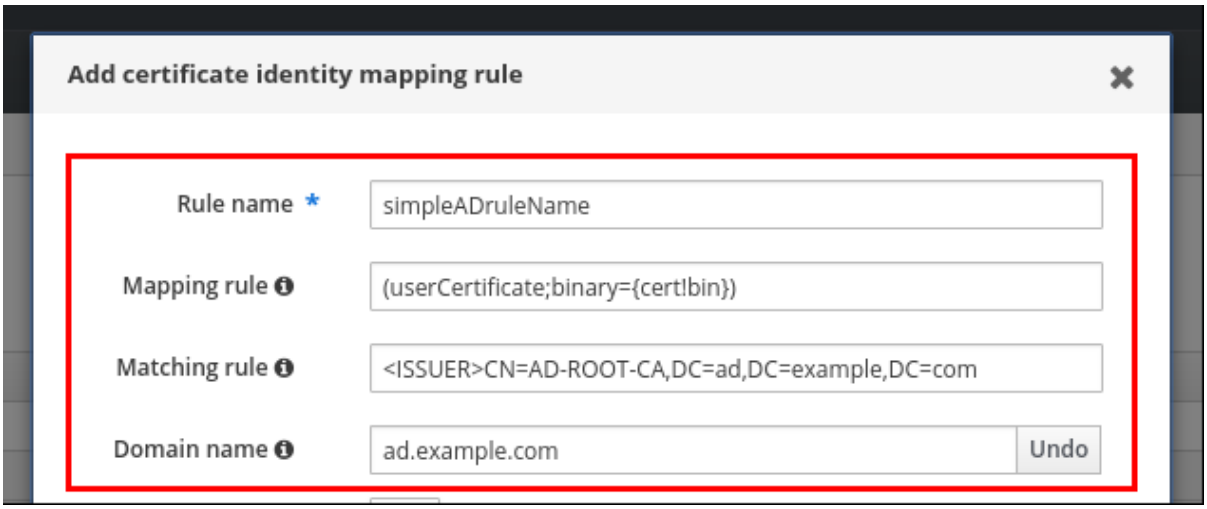
```
(userCertificate;binary={cert!bin})
```

6. マッチングルールを入力します。たとえば、**AD.EXAMPLE.COM** ドメインの **AD-ROOT-CA** が発行する証明書のみを認証できるようにするには、次のコマンドを実行します。

```
<ISSUER>CN=AD-ROOT-CA,DC=ad,DC=example,DC=com
```

7. ドメイン名を入力します。たとえば、**ad.example.com** ドメインでユーザーを検索するには、以下を実行します。

図23.10 AD に証明書やマッピングデータが保存されていないユーザーに対する証明書マッピングルール



8. **Add** をクリックします。

9. System Security Services Daemon (SSSD) は、証明書マッピングルールを定期的に再読み込みします。新たに作成したルールがすぐに読み込まれるようにする場合は、次のコマンドを実行して SSSD を再起動します。

```
# systemctl restart sssd
```

23.2.5.2. AD ユーザーエントリーに証明書やマッピングデータが含まれていない場合に、コマンドラインを使用して証明書マッピングルールを追加

AD ユーザーエントリーに、証明書やマッピングデータが含まれていない場合に、コマンドラインを使用して証明書マッピングルールを追加するには、以下を実行します。

1. 管理者の認証情報を取得します。

```
# kinit admin
```

2. マッピングルールを入力し、マッピングルールの基となっているマッチングルールを入力します。IdM の AD ユーザーエントリーのユーザー ID オーバーライドエントリーに保存されている証明書と比較する、認証用に提示される証明書全体を取得して、**AD.EXAMPLE.COM** ドメインの **AD-ROOT-CA** により発行された証明書のみを認証できるようにするには、以下のコマンドを実行します。

```
# ipa certmaprule-add simpleADrule --matchrule '<ISSUER>CN=AD-ROOT-
CA,DC=ad,DC=example,DC=com' --maprule '(userCertificate;binary={cert!bin})' --domain
ad.example.com
----------------------------------------------------
Added Certificate Identity Mapping Rule "simpleADrule"
----------------------------------------------------
  Rule name: simpleADrule
  Mapping rule: (userCertificate;binary={cert!bin})
```

```
Matching rule: <ISSUER>CN=AD-ROOT-CA,DC=ad,DC=example,DC=com
Domain name: ad.example.com
Enabled: TRUE
```

3. System Security Services Daemon (SSSD) は、証明書マッピングルールを定期的に再読み込みします。新たに作成したルールがすぐに読み込まれるようにする場合は、次のコマンドを実行して SSSD を再起動します。

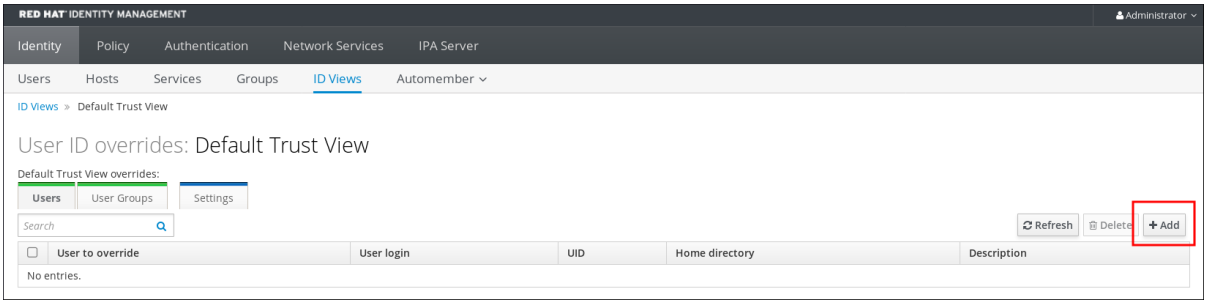
```
# systemctl restart sssd
```

23.2.5.3. Web UI を使用した AD ユーザーの ID オーバーライドへの証明書の追加

AD のユーザーエントリーに、証明書やマッピングデータが含まれていない場合に、Web UI を使用して AD ユーザーの ID オーバーライドに証明書を追加するには、次のコマンドを実行します。

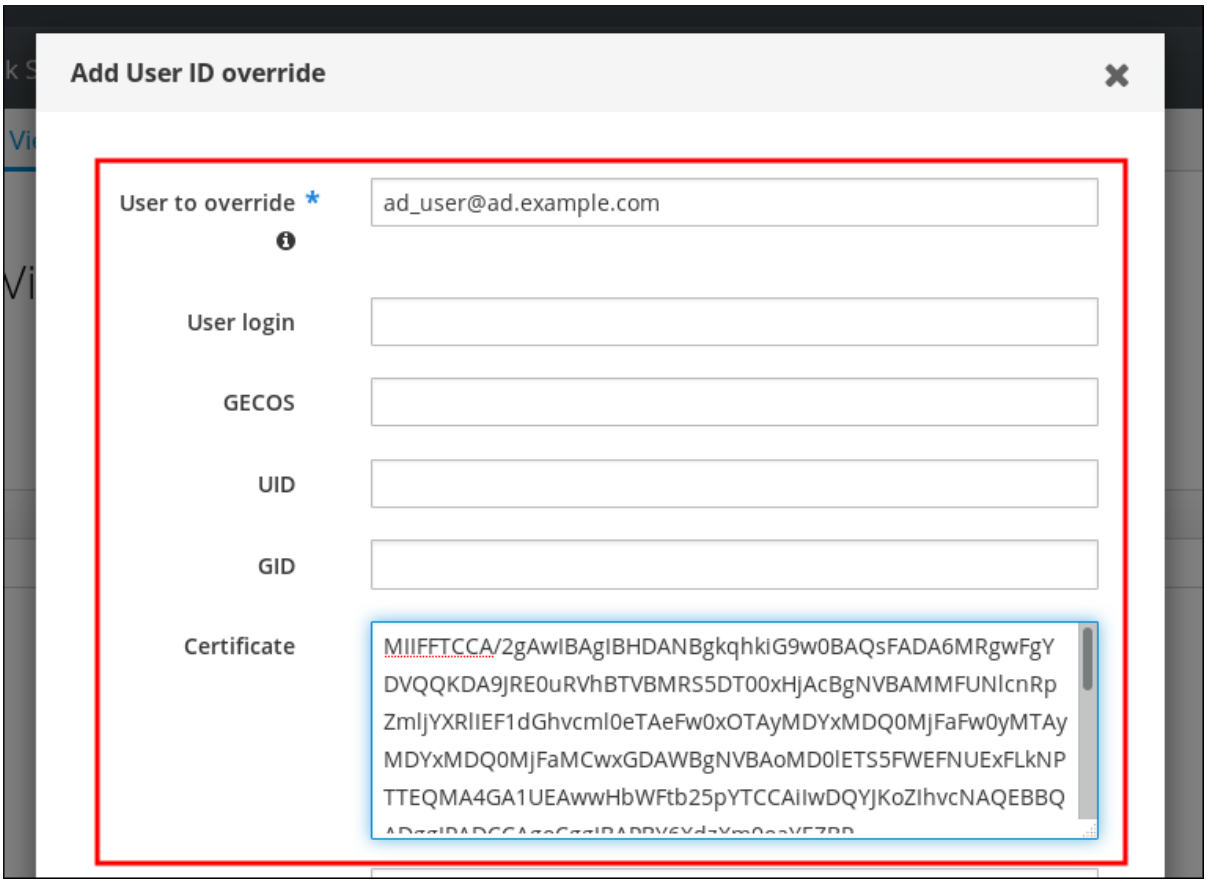
1. 管理者として IdM Web UI にログインします。
2. **Identity → ID Views → Default Trust View** に移動します。
3. **Add** をクリックします。

図23.11 IdM Web UI で新規ユーザー ID オーバーライドの追加



4. **User to override** フィールドに、***user_name@domain_name***の形式でユーザー名を入力します。
5. このユーザーの証明書を、**Certificate** フィールドにコピーアンドペーストします。

図23.12 AD ユーザーにユーザー ID オーバーライドの設定



6. 必要に応じて、ユーザーと証明書がリンクされていることを確認します。

   a. **sss_cache** ユーティリティーを使用して、SSSD キャッシュでユーザーのレコードを無効にし、ユーザーの情報を再読み込みします。

   ```
   # sss_cache -u ad_user@ad.example.com
   ```

   b. AD ユーザーの証明書が含まれるファイルの名前で、**ipa certmap-match** コマンドを実行します。

   ```
   # ipa certmap-match ad_user_cert.pem
   --------------
   1 user matched
   --------------
    Domain: AD.EXAMPLE.COM
    User logins: ad_user@ad.example.com
   ----------------------------
   Number of entries returned 1
   ----------------------------
   ```

   この出力では、証明書マッピングデータが **ad_user@ad.example.com** に追加され、対応するマッピングルールが存在することを確認します。これは、定義した証明書マッピングデータに一致する証明書を使用して、**ad_user@ad.example.com** として認証できることを意味します。

23.2.5.4. コマンドラインを使用した AD ユーザーの ID オーバーライドへの証明書の追加

AD のユーザーエントリーに、証明書やマッピングデータが含まれていない場合に、コマンドラインを使用して AD ユーザーの ID オーバーライドに証明書を追加するには、次のコマンドを実行します。

1. 管理者の認証情報を取得します。

```
# kinit admin
```

2. **ipa idoverrideuser-add-cert** コマンドを使用して、ユーザーの証明書をユーザーアカウントに追加します。

```
# CERT=`cat ad_user_cert.pem | tail -n +2 | head -n -1 | tr -d '\r\n'\`
# ipa idoverrideuser-add-cert ad_user@ad.example.com --certificate $CERT
```

3. 必要に応じて、ユーザーと証明書がリンクされていることを確認します。

   a. **sss_cache** ユーティリティーを使用して、SSSD キャッシュでユーザーのレコードを無効にし、ユーザーの情報を再読み込みします。

```
# sss_cache -u ad_user@ad.example.com
```

   b. AD ユーザーの証明書が含まれるファイルの名前で、**ipa certmap-match** コマンドを実行します。

```
# ipa certmap-match ad_user_cert.pem
--------------
1 user matched
--------------
 Domain: AD.EXAMPLE.COM
 User logins: ad_user@ad.example.com
----------------------------
Number of entries returned 1
----------------------------
```

   この出力では、証明書マッピングデータが **ad_user@ad.example.com** に追加され、対応するマッピングルールが存在することを確認します。これは、定義した証明書マッピングデータに一致する証明書を使用して、**ad_user@ad.example.com** として認証できることを意味します。

23.2.6. 複数のアイデンティティーマッピングルールを 1 つに結合

複数の ID マッピングルールを 1 つのルールに結合するには、個々のマッピングルールの前に | (or) 文字を追加し、括弧 **()** で区切ります。以下に例を示します。

```
$ 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-ROOT-
CA,DC=ad,DC=example,DC=com' \ --domain=ad.example.com
```

上記の例では、**--maprule** オプションのフィルター定義には、以下の基準が含まれます。

- ***ipacertmapdata=X509:<I>{issuer_dn!nss_x500}<S>{subject_dn!nss_x500}*** は、「IdM での証明書マッピングルールの追加」 の説明のとおりに、IdM ユーザーアカウントの ipacertmapdata 属性の値に、スマートカードの発行先および発行者をリンクさせるフィルターです。

- ***altSecurityIdentities=X509:<I>{issuer_dn!ad_x500}<S>{subject_dn!ad_x500}*** は、スマートカード証明書から発行先および発行者を、AD ユーザーアカウントの altSecurityIdentities の値にリンクするフィルターです。これは、「ユーザー証明書をユーザーアカウントにマッピングするように AD が設定されている場合に、証明書マッピングの設定」 で説明されています。

- **--domain=ad.example.com** オプションを追加すると、指定した証明書にマッピングされたユーザーが、ローカルの **idm.example.com** ドメインだけでなく、ad.example.com**ad.example.com** ドメイン内でも検索されます。

--maprule オプションのフィルターの定義では、論理演算子 | (or) が使用できるため、複数の基準を指定できます。この場合、ルールは、1 つ以上の基準を満たすユーザーアカウントをすべてマップします。

```
$ 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,O=REALM.EXAMPLE.COM' \
  --domain=idm.example.com --domain=ad.example.com
```

上記の例では、**--maprule** オプションのフィルター定義には、以下の基準が含まれます。

- ***userCertificate;binary={cert!bin}*** は、証明書全体を含むユーザーエントリーを返すフィルターです。AD ユーザーについては、このタイプのフィルターの作成については、「AD ユーザーエントリーに証明書やマッピングデータが含まれていない場合に、証明書マッピングの設定」 で詳細に説明されています。

- ***ipacertmapdata=X509:<I>{issuer_dn!nss_x500}<S>{subject_dn!nss_x500}*** は、「IdM での証明書マッピングルールの追加」 の説明のとおりに、IdM ユーザーアカウントの ***ipacertmapdata*** 属性の値に、スマートカードの発行先および発行者をリンクさせるフィルターです。

- ***altSecurityIdentities=X509:<I>{issuer_dn!ad_x500}<S>{subject_dn!ad_x500}*** は、スマートカード証明書から発行先および発行者を、AD ユーザーアカウントの ***altSecurityIdentities*** の値にリンクするフィルターです。これは、「ユーザー証明書をユーザーアカウントにマッピングするように AD が設定されている場合に、証明書マッピングの設定」 で説明されています。

--maprule オプションのフィルターの定義では、論理演算子 | (or) が使用できるため、複数の基準を指定できます。この場合、ルールは、1 つ以上の基準を満たすユーザーアカウントをすべてマップします。

23.3. スマートカードを使用した IDENTITY MANAGEMENT クライアントに対する認証

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、Identity Management ドメインに参加しているデスクトップクライアントシステムに、スマートカードで認証できます。これにより、選択したロールとしてクライアントシステムを使用できます。

対応しているオプションの基本概要は、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ベースの認証オプション」

認証を有効にするための環境の設定については、以下を参照してください。

- 「スマートカード認証用の Identity Management クライアントの準備」

認証方法の詳細は、以下を参照してください。

- 「コンソールログインを使用したスマートカードによる Identity Management クライアントでの認証」

23.3.1. Identity Management クライアントでサポートされるスマートカードベースの認証オプション

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とき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**、**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 等の他のサービスを使用したスマートカード認証はサポートされていません。

23.3.2. スマートカード認証用の Identity Management クライアントの準備

Identity Management 管理者は、以下の手順を実行します。

1. サーバーで、クライアントを設定するためのシェルスクリプトを作成します。
   a. **ipa-advise config-client-for-smart-card-auth** コマンドを使用し、その出力をファイルに保存します。

```
# ipa-advise 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、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
```

また、外部認証局 (CA) がスマートカードの証明書に署名した場合は、スマートカード CA を信頼できる 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。

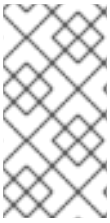
```
# ipa-cacert-manage -n "SmartCard CA" -t CT,C,C install ca.pem
# ipa-certupdate
```

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 を使用すると、管理者は、証明書に定義された Online Certificate Status Protocol (OCSP) サーバーがクライアントから到達できない場合など、***certificate_verification*** パラメーターを使用して証明書の検証プロセスをチューニングできます。詳細は、sssd.conf(5) の man ページを参照してください。

23.3.3. コンソールログインを使用したスマートカードによる Identity Management クライアントでの認証

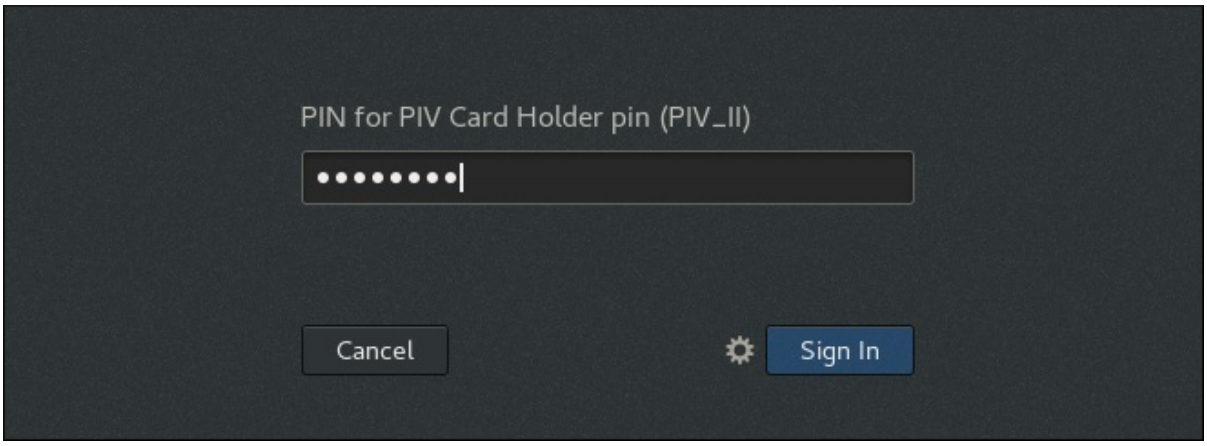
Identity Management ユーザーとして認証するには、ユーザー名と PIN を入力します。

- 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場合、必要なユーザーを選択した後に GDM によりスマートカード PIN の入力が求められます。

図23.13 Gnome Desktop Manager へのスマートカード PIN の入力



Active Directory ユーザーとして認証するには、NetBIOS ドメイン名 (**AD.EXAMPLE.COM\ad_user** または **ad_user@AD.EXAMPLE.COM**) を使用する形式でユーザー名を入力します。

認証に失敗した場合は、「スマートカードの認証エラーの調査」 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、**-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、目的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
```

認証に失敗した場合は、「スマートカードの認証エラーの調査」 を参照してください。

23.3.5. 関連情報

- スマートカードで **ssh** を使用した認証は、リモートシステムで TGT(Ticket-granting Ticket) を取得しません。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で Kerberos を設定し、Kerberos 委譲を有効にする必要があります。必要な設定の例は、この Kerberos ナレッジベースの記事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、『セキュリティーガイド』のOpenSSH へのスマートカードを使用した OpenSSH への認証情報の指定を参照してください。

23.4. スマートカード認証用のユーザー名 HINT ポリシーの設定

Identity Management 管理者は、複数のアカウントにリンクしたスマートカードの *ユーザー名ヒント* ポリシーを設定できます。

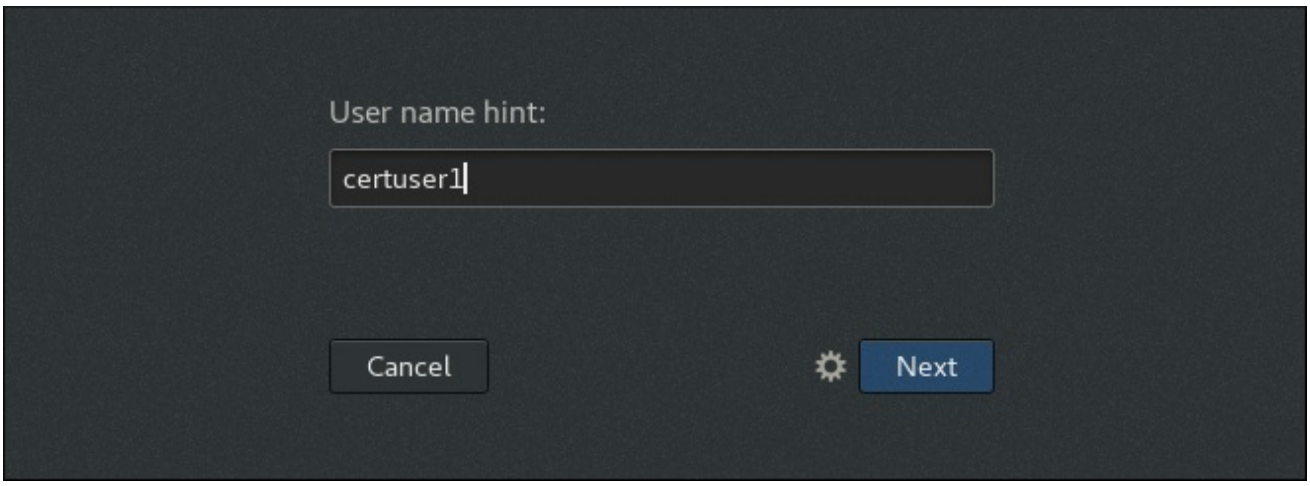
23.4.1. Identity Management のユーザー名ヒント

ユーザー名ヒントポリシーは、スマートカードユーザーにユーザー名を要求するように 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一致するスマートカード証明書で認証を試みると、以下のいずれかが発生します。

- ユーザー名のヒントポリシーを有効にすると、ユーザーにはユーザー名の入力が求められ、認証に進むことができます。
- ユーザー名のヒントポリシーが無効になっていると、認証情報を要求せずに認証に失敗します。

Identity Management は、ユーザー名を求めずに、デフォルトでスマートカード PIN を要求す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) ログインのみが対象になります。

図23.14 Gnome Desktop Manager のユーザー名ヒント



Identity Management は、デフォルトでユーザー名を尋ねるアプリケーションにユーザー名ヒントを追加しません。以下に例を示します。

- GUI は常に **Username** フィールドを表示するため、Identity Management の Web UI 認証
- **ssh** は、**-l** オプションまたは **username@host** 形式で提供された現在のユーザーのログイン名または名前を使用するため、**ssh** 認証
- コンソール認証。ログイン名は常に指定されます。

このような状況では、複数のユーザーに一致する証明書を使用した認証が常に許可されます。

23.4.2. Identity Management での User Name Hints の有効化

Identity Management 管理者は、ユーザー名ヒントポリシーを一元的に設定します。このポリシーは、Identity Management ドメインに登録されている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 User Name Hints の有効化

1. Identity Management 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。

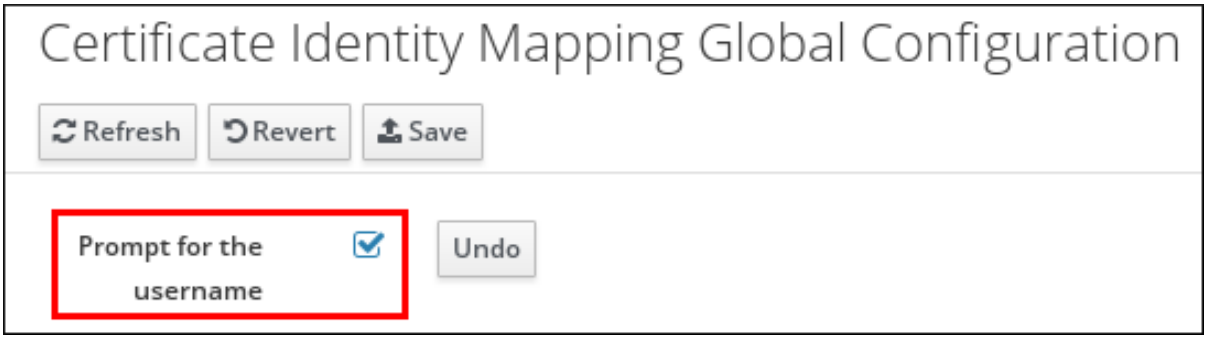
```
$ ipa certmapconfig-mod --promptusername=TRUE
Prompt for the username: TRUE
```

ユーザー名の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 User Name Hints 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23.15 Web UI でユーザー名ヒントの有効化



関連情報

- **ipa certmapconfig-mod** コマンドの詳細は、**--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、Identity Management に参加しているデスクトップクライアントシステムにスマートカードを使用して認証し、Kerberos チケット保証チケット (TGT) を自動的に取得できます。ユーザーは、チケットを使用してクライアントから追加の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 認証のための Identity Management クライアントの準備

Identity Management 管理者は、ユーザーが認証するクライアントで以下の手順を実行します。

1. サーバーで、クライアントを設定するためのシェルスクリプトを作成します。

   a. **ipa-advise config-client-for-smart-card-auth** コマンドを使用し、その出力をファイルに保存します。

```
# ipa-advise 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、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。

また、外部認証局 (CA) がスマートカードの証明書に署名した場合は、スマートカード CA を信頼できる 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。

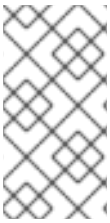
```
# ipa-cacert-manage -n "SmartCard CA" -t CT,C,C install ca.pem
# ipa-certupdate
```

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 を使用すると、管理者は、証明書に定義された Online Certificate Status Protocol (OCSP) サーバーがクライアントから到達できない場合など、***certificate_verification*** パラメーターを使用して証明書の検証プロセスをチューニングできます。詳細は、sssd.conf(5) の man ページを参照してください。

23.5.2. Identity Management ユーザーとして - 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** ユーティリティーを使用して認証します。

```
$ kinit -X X509_user_identity='PKCS11:opensc-pkcs11.so' idm_user
```

-X オプションは **opensc-pkcs11.so** モジュールを pre-authentication 属性として指定します。詳細は kinit(1) の man ページを参照してください。

23.5.3. Active Directory ユーザーとして - Identity Management クライアントでの PKINIT を使用した認証

前提条件
管理者は、Active Directory ユーザーの PKINIT 認証を使用するように環境を設定します。

- スマートカード証明書を発行した認証局 (CA) を信頼するように Active Directory サーバーを設定します。NTAuth ストアに CA をインポートして (Microsoft サポートを参照)、CA を信頼できる CA として追加します。詳細は、Active Directory の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した 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) ディストリビューションポイント拡張が含まれていない場合は、Active Directory を設定して失効エラーを無視します。

  1. 以下の REG 形式のコンテンツをプレーンテキストファイルに保存し、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** ユーティリティーを使用して認証します。Active Directory ユーザーに、ユーザー名とドメイン名を指定します。

```
$ kinit -X X509_user_identity='PKCS11:opensc-pkcs11.so' ad_user@AD.DOMAIN.COM
```

-X オプションは **opensc-pkcs11.so** モジュールを pre-authentication 属性として指定します。詳細は 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た IDENTITY MANAGEMENT WEB UI への認証

Identity Management サーバーに複数のロールを持つ Identity Management ユーザーは、選択したロールとして、スマートカードを使用して Identity Management Web UI に対して認証できます。これにより、選択したロールとして Web UI を使用できます。

注記

Identity Management ユーザーのみが、スマートカードを使用して Web UI にログインできます。Active Directory ユーザーは、ユーザー名とパスワードを使用してログインできます。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。

認証を有効にするための環境の設定については、以下を参照してください。

- 「Web UI でのスマートカード認証用の Identity Management サーバーの準備」
- 「スマートカード認証用にブラウザーを用意」

認証方法の詳細は、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た Identity Management Web UI への認証」

23.6.1. Web UI でのスマートカード認証用の Identity Management サーバーの準備

Identity Management 管理者が、以下を行います。

1. Identity Management サーバーで、サーバーを設定するシェルスクリプトを作成します。
   a. **ipa-advise config-server-for-smart-card-auth** コマンドを使用して、その出力をファイルに保存します。

   ```
   # ipa-advise 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 ドメイン内の全サーバー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。

また、外部認証局 (CA) がスマートカードで証明書に署名した場合:

1. Identity Management サーバーで、HTTP サーバーが使用する NSS データベースに CA 証明書を追加します。

   ```
   # 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 **ipa-certupdate** を繰り返します。

2. HTTP サーバーおよび Kerberos サーバーを再起動します。

   ```
   # systemctl restart httpd
   # systemctl restart krb5kdc
   ```

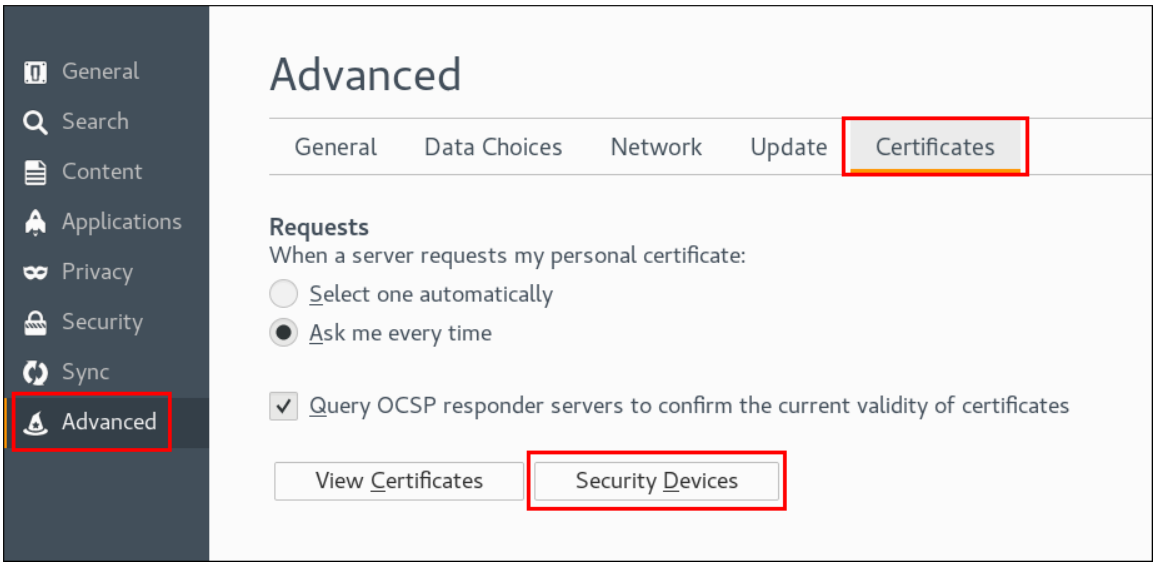
   すべてのレプリカでコマンドを繰り返し実行します。

23.6.2. スマートカード認証用にブラウザーを用意

スマートカード認証にブラウザーを設定するには、ユーザーが Web ブラウザーを起動して Web UI にアクセスするクライアントで以下の手順を実行します。ブラウザーが実行しているシステムは、Identity Management ドメインの一部にする必要はありません。この手順では、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を読み込むように Firefox を設定します。
   a. **Edit → Preferences → Advanced → Certificates → Security Devices**を選択します。

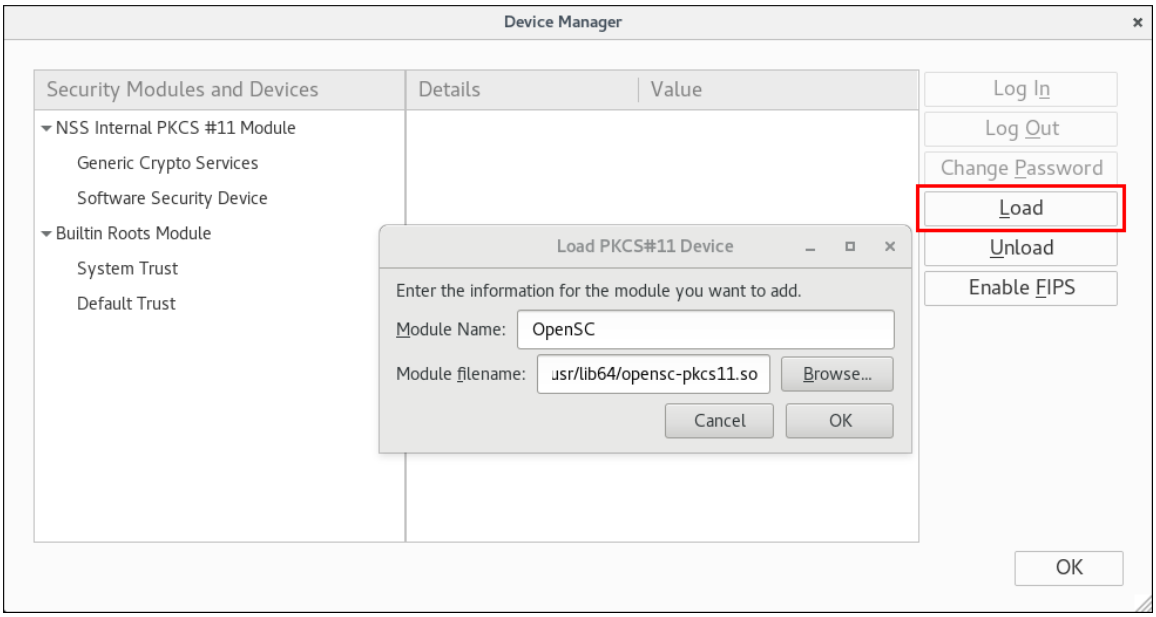
図23.16 Firefox でのセキュリティーデバイスの設定



   b. **Load** をクリックします。**PKCS#11 デバイスの読み込み** 画面に、次の情報を入力します。

- **Module Name**: **OpenSC**
- **Module filename**: **/usr/lib64/opensc-pkcs11.so**

図23.17 Firefox のデバイスマネージャー



c. **OK** をクリックして確定します。次に、**OK** をクリックしてデバイスマネージャーを閉じます。

Firefox が、認証にスマートカード証明書を使用できるようになりました。

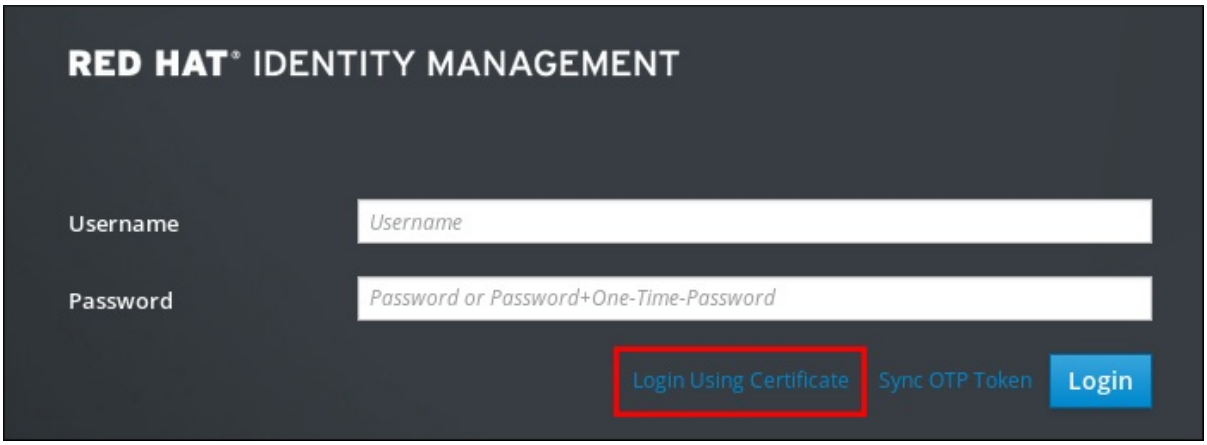
23.6.3. Identity Management ユーザーとしてスマートカードを使用した Identity Management Web UI への認証

認証するには、以下を実行します。

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は、**Username** フィールドに入力しないでください。

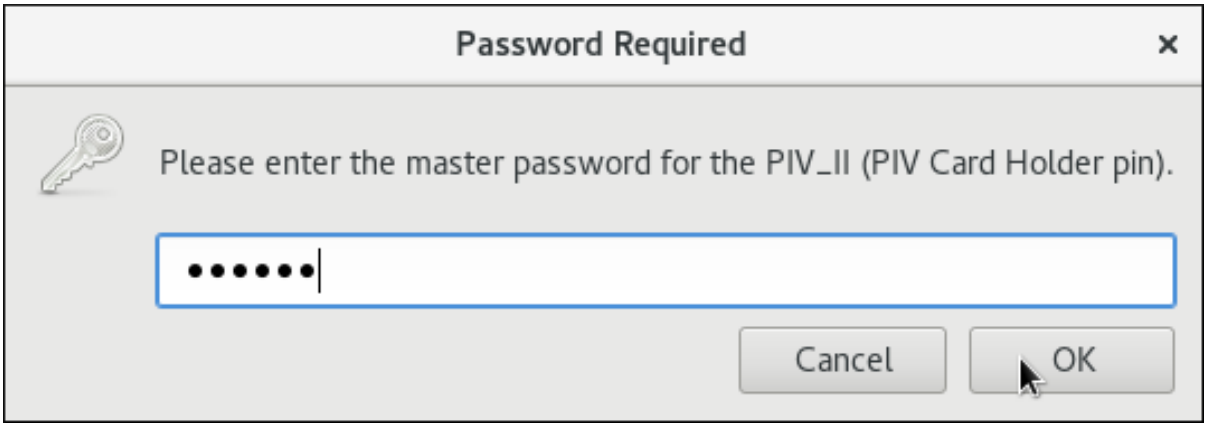
   スマートカード証明書が複数のユーザーアカウントにリンクされている場合は、**Username** フィールドに入力して必要なアカウントを指定します。
4. **Login Using Certificate** をクリックします。

   図23.18 Identity Management の Web UI でLogin Using Certificate



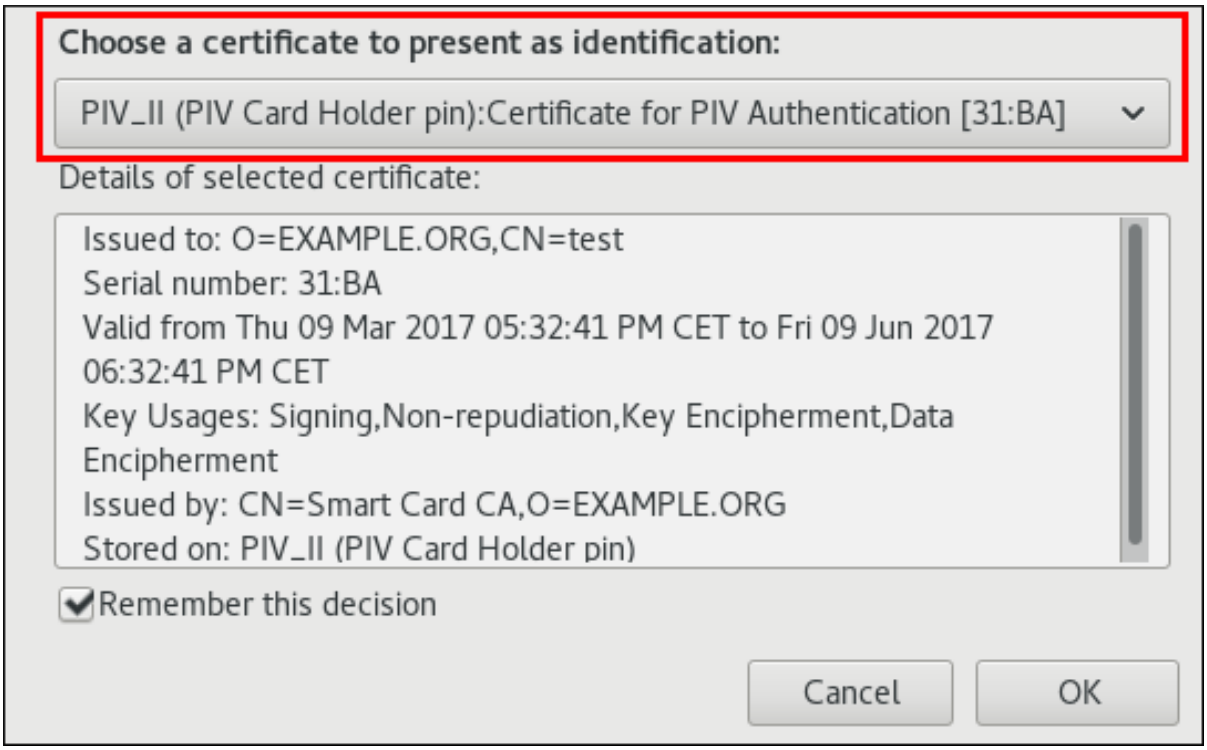
5. プロンプトが表示されたら、スマートカードの PIN を入力します。

   図23.19 スマートカードの PIN の入力



6. 新しいウィンドウが開きます。使用する証明書を提案します。スマートカード証明書を選択します。

図23.20 スマートカード証明書の選択



これで、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。

注記

管理者がユーザーのパスワードをリセットすると、IdM Web UI は、kinit ユーティリティーなどを使用して、ユーザーが新しいパスワードを設定するまでアクセスを拒否します。

関連情報

- 認証に失敗した場合は、「スマートカードの認証エラーの調査」 を参照してください。

23.6.4. 関連情報

- Identity Management の Web UI の詳細は、「IdM Web UI」 を参照してください。

23.7. IDENTITY MANAGEMENT のスマートカード認証と WEB アプリケーションの統合

Identity Management の Web インフラストラクチャー Apache モジュールを介して、アプリケーションが Identity Management サーバーを認証バックエンドとして使用する開発者は、複数のロールアカウントをスマートカードにリンクしたユーザーの認証を有効にするようにアプリケーションを設定できます。これにより、これらのユーザーは、許可されたロールアカウントでアプリケーションを使用できます。

23.7.1. スマートカードを使用した Web アプリケーション認証の前提条件

Apache Web アプリケーションが実行しているサーバーで、以下を行います。

- Identity Management ドメインで、サーバーをクライアントとして登録します。

- sssd-dbus パッケージおよび mod_lookup_identity パッケージをインストールします。
- **mod_nss** モジュールを使用して、Apache に作業用の HTTPS 接続が設定されていることを確認してください。

23.7.2. Web アプリケーションの 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 再ネゴシエーションを有効にします。

```
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に対して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、ログインページと保護された領域で設定される、ほぼ最小限の例を示します。
   - **/login** エンドポイントでは、ユーザーはユーザー名のみを指定し、アプリケーションの保護された部分にユーザーを送信することができます。
   - **/app** のエンドポイントでは、**REMOTE_USER** 環境変数を確認します。ログインに成功すると、変数にはログイン中のユーザーの ID が含まれます。それ以外の場合は、変数は設定されません。
4. ディレクトリーを作成し、そのグループを **apache** に、モードを少なくとも **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、そのグループを **apache** に設定し、モードを少なくとも **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、そのグループを **apache** に設定し、モードを少なくとも **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. アプリケーションの設定ファイルを作成します。この手順では、以下の内容を含む **/etc/httpd/conf.d/app.conf** という名前のファイルを使用しています。

```
<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>
```

   このファイルで以下を行います。

   - 最初にロードされていない場合は、**mod_lookup_identity** を読み込みます。
   - 次の部分では、**/login** と **/app** はそれぞれの Web Server Gateway Interface(WSGI) スクリプトを参照します。
   - 最後の部分は、TLS ハンドシェイク中にクライアント証明書を必要とし、それを使用するように **/app** エンドポイントに **mod_nss** を設定します。さらに、ユーザーのアイデンティティーを検索するようにオプションの要求パラメーター ***username*** を設定します。

23.8. KDC からチケットを取得する際の特定の認証の強制

特定の認証インジケーターを有効にするには、以下を実行します。

- ホストオブジェクト:

```
# ipa host-mod host_name --auth-ind=indicator
```

- Kerberos サービス。以下を実行します。

```
# ipa service-mod service/host_name --auth-ind=indicator
```

複数の認証インジケーターを設定するには、***--auth-ind*** パラメーターを複数回指定します。

警告

HTTP/*IdM_master* サービスに認証インジケーターを設定すると、IdM マスターが失敗します。さらに、IdM が提供するユーティリティーでは、マスターを復元できません。

例23.2 特定のホストでの pkinit インドシュレーターの有効化

以下のコマンドは、スマートカードで認証したユーザーのみが **host.idm.example.com** ホストのサービスチケットを取得できるように設定します。

```
# ipa host-mod host.idm.example.com --auth-ind=pkinit
```

上記の設定により、Kerberos チケットを要求するユーザーの TGT(Ticket-Granting Ticket) に **pkinit** 認証インジケーターが含まれているようになります。

第24章 ユーザー、ホスト、およびサービスの証明書の管理

Identity Management(IdM) は、2 種類の認証局 (CA) をサポートします。

統合 IdM CA

統合 CA は、ユーザー、ホスト、およびサービスの証明書の作成、取り消し、および発行が可能です。詳細は、「統合 IdM CA を使用した証明書の管理」 を参照してください。

IdM は、軽量のサブ CA の作成に対応します。詳細は、「軽量サブ 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 ツールを使用して、これらの CA が発行する証明書をユーザー、サービス、またはホストに対して追加し、削除します。詳細は、「外部 CA が発行する証明書の管理」 を参照してください。

各ユーザー、ホスト、または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 設定の詳細は、「使用する CA 設定の決定」 を参照してください。

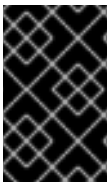
24.1. 統合 IDM CA を使用した証明書の管理

24.1.1. ユーザー、ホスト、またはサービスの新規証明書の要求

以下を使用して証明書を要求するには、以下を実行します。

- IdM Web UI は、「Web UI: 新しい証明書の要求」を参照してください。
- コマンドラインは、「コマンドライン: 新しい証明書の要求」を参照してください。

サードパーティーツールで証明書要求自体を生成する必要があることに注意してください。以下の手順では、**certutil** ユーティリティーおよび **openSSL** ユーティリティーを使用します。



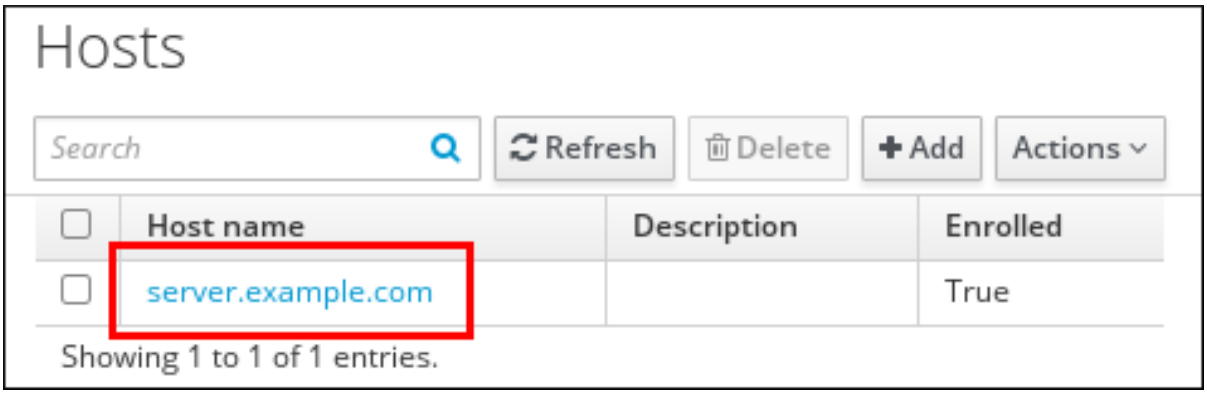
重要

サービスは通常、秘密鍵の保存先となる専用のサービスノードで実行されます。サービスの秘密鍵を IdM サーバーにコピーすることは、安全ではないとみなされます。したがって、サービスの証明書を要求する場合には、サービスノードで CSR を作成します。

Web UI: 新しい証明書の要求

1. **Identity** タブで、 **Users**、**Hosts**、または **Services** のサブタブを選択します。
2. ユーザー、ホスト、またはサービス名をクリックして、設定ページを開きます。

図24.1 ホストのリスト



3. **Actions → New Certificate** をクリックします。
4. 必要に応じて、発行元の CA およびプロファイル ID を選択します。
5. **certutil** の使用に関する画面の手順に従います。
6. **Issue** をクリックします。

コマンドライン: 新しい証明書の要求

標準の状況で **certutil** を使用して新しい証明書を要求します。「Certutil を使用した新規証明書の要求」 を参照してください。Open **SSL** を使用して新しい証明書を要求し、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します。「OpenSSL を使用した複数の SAN フィールドを使用した証明書要求の準備」 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要求

1. 証明書データベースの一時ディレクトリーを作成します。

```
# mkdir ~/certdb/
```

2. 以下のように、新しい一時証明書データベースを作成します。

```
# certutil -N -d ~/certdb/
```

3. 証明書署名要求 (CSR) を作成し、その出力をファイルにリダイレクトします。たとえば、4096 ビット証明書の CSR を作成し、発行先を *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
```

4. 証明書要求を CA に送信します。詳細は、「IdM CA への証明書要求の送信」 を参照してください。

24.1.1.2. OpenSSL を使用した複数の SAN フィールドを使用した証明書要求の準備

1. Kerberos プリンシパル *test/server.example.com* に対して、1 つ以上のエイリアス (例: *test1/server.example.com*、*test2/server.example.com*) を作成します。詳細は、「Kerberos プリンシパルエイリアス」 を参照してください。

2. CSR で dnsName (*server.example.com*) と otherName (*test2/server.example.com*) の subjectAltName を追加します。これには、UPN otherName および subjectAltName を指定する以下の行が含まれるように、**openssl.conf** ファイルを設定します。

```
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** を使用して証明書要求を作成します。

```
openssl req -new -newkey rsa:2048 -keyout test2service.key -sha256 -nodes -out
certificate_request.csr -config openssl.conf
```

4. 証明書要求を CA に送信します。詳細は、「IdM CA への証明書要求の送信」を参照してください。

24.1.1.3. Certmonger を使用した新規証明書の要求

Certmonger サービスを使用して、 IdM CA から証明書を要求できます。詳細は、『システムレベルの認証ガイド』の『SCEP 経由での CA 署名証明書の要求』 を参照してください。

24.1.1.4. IdM CA への証明書要求の送信

IdM サーバーで実行している CA に証明書要求ファイルを送信します。新しく発行した証明書に関連付ける Kerberos プリンシパルを指定します。

```
# ipa cert-request certificate_request.csr --principal=host/server.example.com
```

IdM の **ipa cert-request** コマンドは、次のデフォルトを使用します。

- 証明書プロファイル: **caIPAserviceCert**

  カスタムプロファイルを選択するには、**ipa cert-request** コマンドで **--profile-id** オプションを使用します。

  カスタム証明書プロファイルの作成方法は、「証明書プロファイルの作成」 を参照してください。

- 統合 CA: **ipa**(IdM ルート CA)

  サブ CA を選択するには、**ipa cert-request** コマンドで **--ca** オプションを使用します。

詳細は、**ipa cert-request --help** コマンドの出力を参照してください。

24.1.2. 統合 IdM CA を使用した証明書の失効

有効期限が切れる前に証明書を無効にする必要がある場合は、取り消すことができます。以下を使用して証明書を取り消すには、次のコマンドを実行します。

- IdM Web UI は、「Web UI: 証明書の失効」を参照してください。
- コマンドラインは、「コマンドライン: 証明書の失効」を参照してください。

失効した証明書は無効であり、認証に使用できません。理由 6 の証明書の保留を除き、すべての失効は永続的です。

表24.1 証明書の失効理由

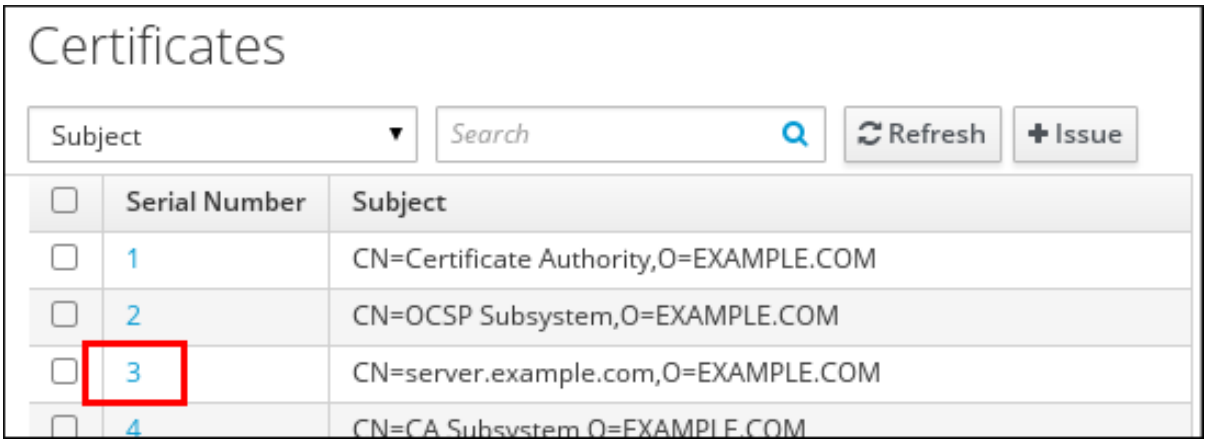
ID	理由	説明
0	指定なし	
1	鍵が侵害された	証明書を発行した鍵が信頼されなくなった。 考えられる原因 - トークンの消失、ファイルへの不適切なアクセス。
2	CA が侵害された	証明書を発行した CA は信頼されなくなった。
3	所属が変更した	考えられる原因: ● * 人が退職したか、別の部門に移動した。 ● * ホストまたはサービスが廃止された。
4	置き換え	現在の証明書から新しい証明書に置き換えられた。
5	運用停止	ホストまたはサービスの使用を停止している。
6	証明書が保留になっている	証明書は一時的に取り消されている。証明書は後で復元できます。
8	CRL から削除された	証明書は、証明書失効リスト (CRL) に含まれていない。
9	特権が撤回された	ユーザー、ホスト、またはサービスは、証明書の使用を許可されなくなった。
10	侵害された属性機関 (Attribute Authority)	属性機関証明書は信頼されなくなった。

Web UI: 証明書の失効

証明書を取り消すには、以下を実行します。

1. **Authentication** タブを開き、**Certificates** サブタブを選択します。
2. 証明書のシリアル番号をクリックして、証明書情報ページを開きます。

図24.2 証明書のリスト



3. **Actions → Revoke Certificate** をクリックします。

4. 取り消しの理由を選択し、**Revoke** をクリックします。詳細は、表24.1「証明書の失効理由」を参照してください。

コマンドライン: 証明書の失効

ipa cert-revoke コマンドを使用して、次を指定します。

- 証明書のシリアル番号
- 失効の理由を示す数字。詳細は 表24.1「証明書の失効理由」 を参照してください。

たとえば、理由 1 (侵害された鍵) のためにシリアル番号 **1032** の証明書を失効させるには、次のコマンドを実行します。

```
$ ipa cert-revoke 1032 --revocation-reason=1
```

24.1.3. 統合 IdM CA を使用した証明書の復元

理由 6 (証明書の保留) のために証明書を失効させている場合は、再度復元できます。以下を使用して証明書を復元するには、以下を実行します。

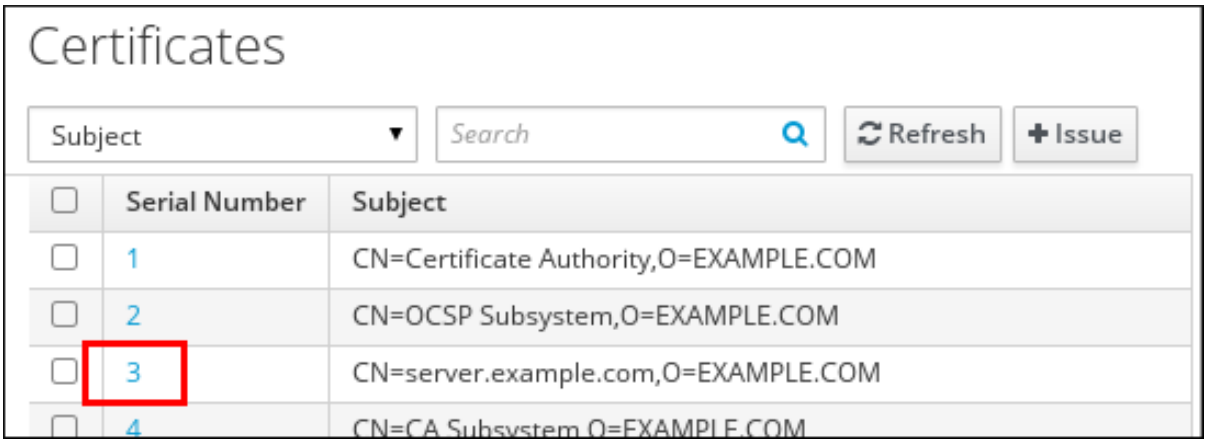
- IdM Web UI は、「Web UI: 証明書の復元」を参照してください。
- コマンドラインは、「コマンドライン: 証明書の復元」を参照してください。

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. 証明書のシリアル番号をクリックして、証明書情報ページを開きます。

図24.3 証明書のリスト



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
ipa cert-remove-hold コマンドを使用して、証明書のシリアル番号を指定します。以下に例を示します。

```
$ ipa cert-remove-hold 1032
```

24.2. 外部 CA が発行する証明書の管理

24.2.1. コマンドライン: 外部 CA が発行する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 が発行する証明書は、IdM から削除しても取り消されません。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 側からのみ手動で取り消すことができます。

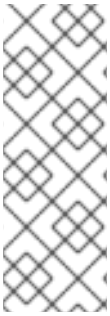
コマンドには、以下の情報を指定する必要があります。

- 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 DER 証明書

コマンドを対話的に実行するには、オプションを追加せずに実行します。

コマンドを使用して必要な情報を直接指定するには、コマンドライン引数およびオプションを使用します。

```
$ ipa user-add-cert user --certificate=MIQTPrajQAwg...
```



注記

証明書の内容をコマンドラインにコピーして貼り付ける代わりに、証明書を DER 形式に変換し、これを base64 に再プロビジョニングできます。たとえば、**user_cert.pem**証明書を**ユーザー**に追加するには、次のコマンドを実行します。

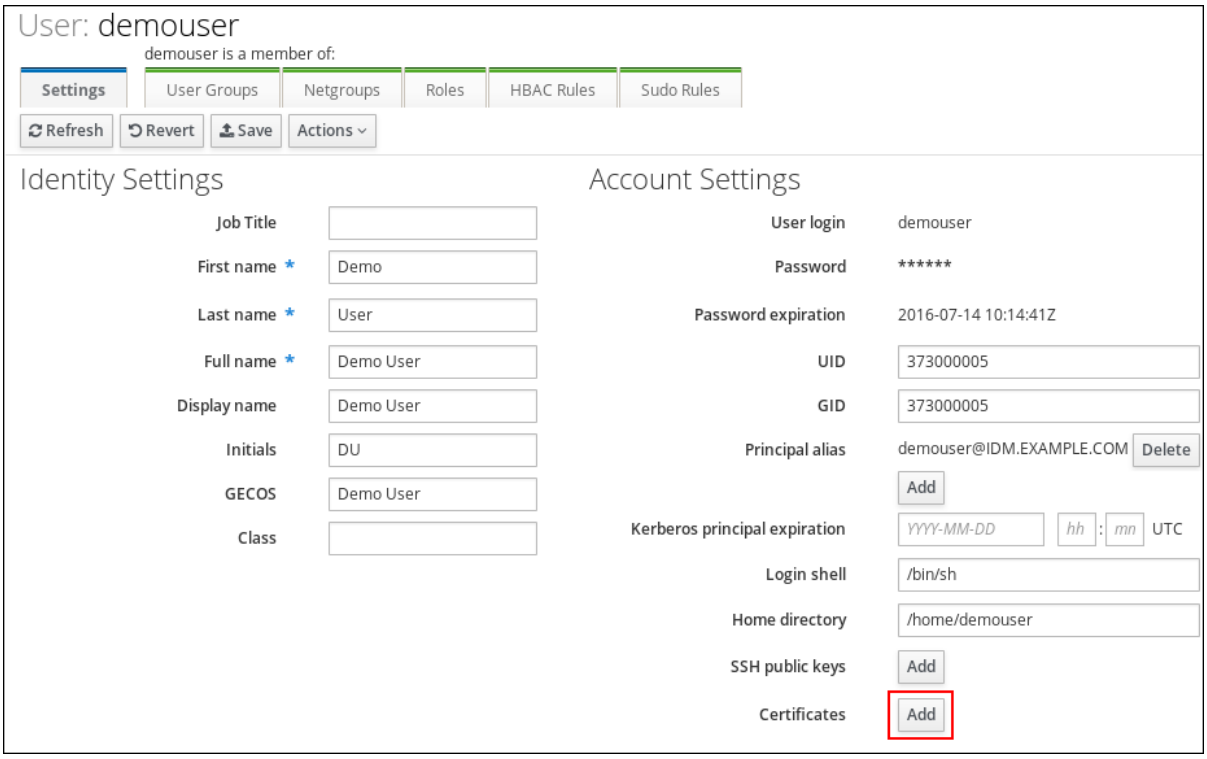
```
$ ipa user-add-cert user --certificate="$(openssl x509 -outform der -in user_cert.pem | base64 -w 0)"
```

24.2.2. Web UI: 外部 CA が発行する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** サブタブを選択します。
2. ユーザー、ホスト、またはサービス名をクリックして、設定ページを開きます。
3. **Certificates** エントリーの横にある **Add** をクリックします。

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 でエンコードされた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** 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** サブタブを選択します。
2. ユーザー、ホスト、またはサービス名をクリックして、設定ページを開きます。
3. 削除する証明書の横にある **Actions** をクリックし、**Delete** を選択します。
4. **Save** をクリックして変更を保存します。

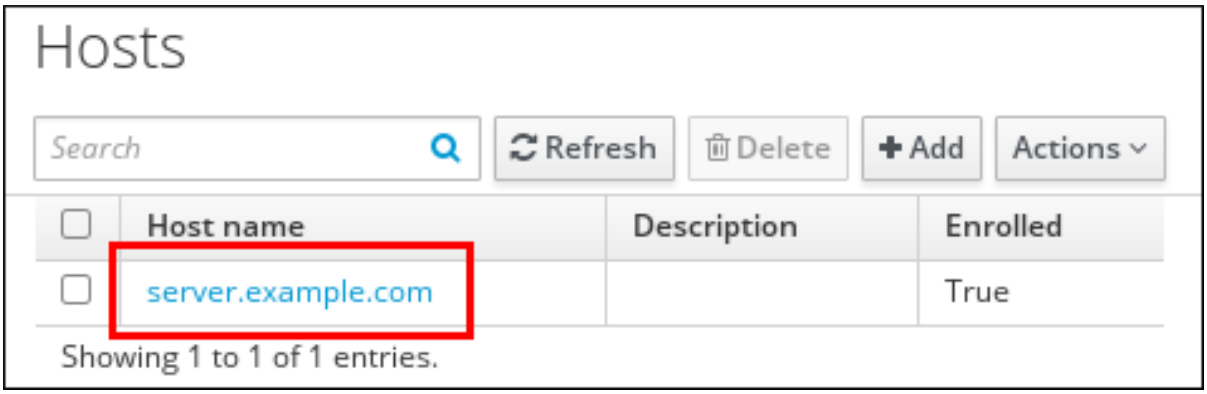
24.3. 証明書のリスト表示および表示

Web UI での証明書のリスト表示および表示

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リスト表示するには、以下を実行します。

1. **Identity** タブを開き、**Users**、**Hosts**、または **Services** サブタブを選択します。
2. ユーザー、ホスト、またはサービス名をクリックして、設定ページを開きます。

図24.5 ホストのリスト

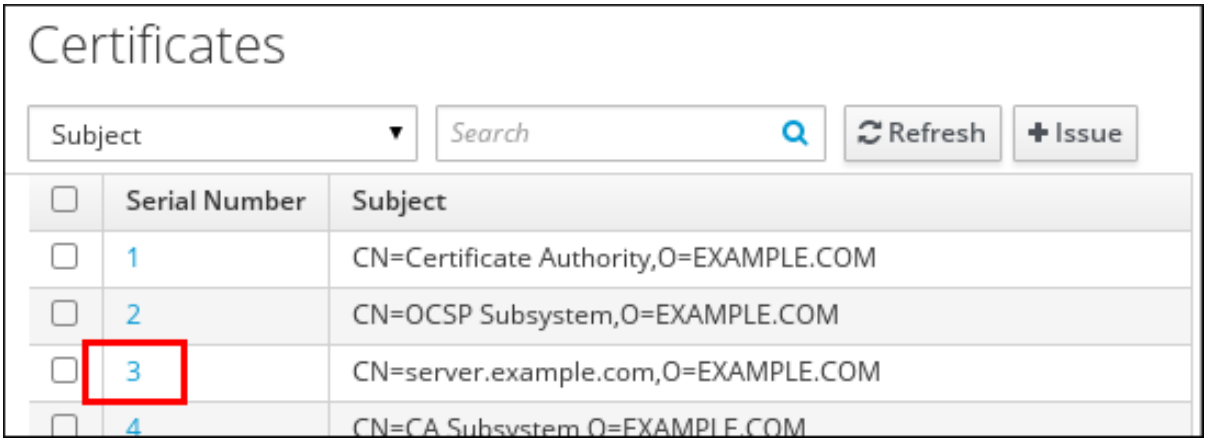


3. 設定ページには、エントリーに割り当てられたすべての証明書がリスト表示されます。また、**Show** をクリックすると、特定の証明書が表示されます。

IdM サーバーに登録されている証明書のリストを表示するには、次のコマンドを実行します。

1. **Authentication** タブを開き、**Certificates** サブタブを選択します。
2. すべての証明書のリストは **Certificates** セクションに表示されます。特定の証明書を表示するには、シリアル番号をクリックします。

図24.6 証明書のリスト



コマンドラインでの証明書のリスト表示

IdM データベース内のすべての証明書をリスト表示するには、**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
----------------------------
```

問題の日付や有効日など、特定の証明書プロパティーを指定して検索結果をフィルタリングできます。たとえば、課題の間隔で検索するには、--issued **on-from** オプションまたは **--issuedon-to** オプションを使用して、開始時間と終了点または期間を指定します。

```
ipa cert-find --issuedon-from=2020-01-07 --issuedon-to=2020-02-07
```

証明書の検索のフィルターに使用されるオプションの完全なリストは、**--help** オプションを追加して **ipa cert-find** を実行します。

コマンドラインでの証明書の表示

証明書を表示するには、**ipa cert-show** コマンドを使用してシリアル番号を指定します。

```
$ ipa cert-show 132
Serial number: 132
  Certificate:
MIIDtzCCAp+gAwIBAgIBATANBgkqhkiG9w0BAQsFADBBMR8wHQYDVQQKExZMQUlu
...
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** を使用して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、次のコマンドを実行します。

```
$ ipa user-show user
  User login: user
  ...
  Certificate: MIICfzCCAWcCAQA...
  ...
```

--out オプションを **ipa cert-show** に追加して、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、**--out** オプションはすべての証明書をエクスポートします。証明書または証明書は PEM オブジェクトとしてエクスポートされます。

24.4. 証明書プロファイル

証明書プロファイルは、特定のプロファイルに属する証明書の内容と、登録のために証明書、登録方法、入力フォームの発行に関する制約を定義します。1 つの証明書プロファイルが、特定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、およびホストに、さまざまな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、以下を決定します。

- CA が証明書署名要求 (CSR) を受け入れるかどうか。
- 証明書にどのような機能と拡張機能が存在するべきか

IdM には、デフォルトで、**caIPAserviceCert** および **IECUserRoles** の 2 つの証明書プロファイルが含まれています。さらに、カスタムプロファイルをインポートできます。

カスタム証明書プロファイルを使用すると、特定の非関連目的の証明書を発行できます。たとえば、特定のプロファイルの使用を 1 つのユーザーまたはグループに制限し、他のユーザーやグループがそのプロファイルを使用して認証用の証明書を発行しないようにすることができます。

サポートされる証明書プロファイル設定の詳細は、Red Hat Certificate System 『管理ガイド』のデフォルトの参照および制約の参照 を参照してください。

注記

「認証局 ACL ルール」 では、証明書プロファイルと CA ACL を組み合わせることで、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し、制御できます。プロファイルおよび CA ACL を使用してユーザー証明書を発行する方法は、「IdM CA でユーザー証明書を発行するための証明書プロファイルおよび ACL の使用」 を参照してください。

24.4.1. 証明書プロファイルの作成

証明書プロファイルの作成に関する詳細は、Red 『Hat Certificate System 9 管理ガイドの 』 以下のドキュメントを参照してください。

- 『証明書プロファイルの設定』 セクションでは、新しい証明書プロファイルの作成方法と、その構築方法を説明します。
- 証明書および 『CRL 付録のデフォルト、制約、および拡張』 には、証明書プロファイルで使用できるその他のオブジェクト識別子 (OID) がリスト表示されます。

24.4.2. コマンドラインでの証明書プロファイル管理

IdM プロファイルを管理するための **certprofile** プラグインを使用すると、特権ユーザーが IdM 証明書プロファイルのインポート、変更、または削除を行うことができます。プラグインがサポートするすべてのコマンドを表示するには、**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を持つユーザーとして操作する必要があります。IdM には、デフォルトで次の証明書プロファイル関連のパーミッションが含まれています。

システム: 証明書プロファイルの読み取り

ユーザーがすべてのプロファイル属性を読み取りできるようにします。

System: Certificate Profile のインポート

ユーザーが証明書プロファイルを IdM にインポートできるようにします。

System: 証明書プロファイルの削除

ユーザーが既存の証明書プロファイルを削除できるようにします。

システム: 証明書プロファイルの変更

ユーザーがプロファイル属性を変更し、プロファイルを無効化または有効化できるようにします。

これらのすべてのパーミッションは、デフォルトの **CA Administrator** 特権に含まれます。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関する詳細は、「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

証明書を要求する際に、**--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できます。プロファイル ID が指定されていない場合、証明書にデフォルトの **caIPAserviceCert** プロファイルが使用されます。

本セクションでは、プロファイル管理に **ipa certprofile** コマンドを使用する最も重要な側面のみを説明します。コマンドの詳細は、以下のように **--help** オプションを指定して実行します。

```
$ 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=BOOL   Whether to store certs issued using this profile
...
```

証明書プロファイルのインポート

新しい証明書プロファイルを IdM にインポートするには、**ipa certprofile-import** コマンドを使用します。オプションを指定せずにコマンドを実行すると、**certpofile-import** スクリプトにより、証明書のインポートに必要な情報の入力が求められます。

```
$ 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、複数のコマンドラインオプションを受け入れます。以下に例を示します。

--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。以下に例を示します。

```
$ 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 つの値を受け入れます。

- **True**: 発行した証明書をクライアントに配信し、ターゲット IdM プリンシパルの **userCertificate** 属性に保存します。
- **False**。発行された証明書をクライアントに配信しますが、IdM に保存しません。このオプションは、複数の短期証明書を発行する場合に最も一般的に使用されます。

ipa certprofile-import が指定されたプロファイル ID がすでに使用されている場合や、プロファイルコンテンツが正しくないと、インポートに失敗します。たとえば、必要な属性がない場合や、提供されたファイルで定義されたプロファイル ID の値が **ipa certprofile-import** で指定されたプロファイル ID と一致しない場合、インポートに失敗します。

新規プロファイルのテンプレートを取得するには、**--out** オプションを指定して **ipa certprofile-show** コマンドを実行し、指定した既存プロファイルをファイルにエクスポートします。以下に例を示します。

```
$ ipa certprofile-show caIPAserviceCert --out=file_name
```

必要に応じてエクスポートされたファイルを編集し、新しい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 に現在保存されている証明書プロファイルをすべて表示するには、**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
...
```

特定のプロファイルに関する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変更
既存の証明書プロファイルを変更するには、**ipa certprofile-mod** コマンドを使用します。**ipa certprofile-mod** で使用できるコマンドラインオプションを使用して、コマンドで必要な変更を渡します。たとえば、プロファイルの説明を変更し、IdM が発行した証明書を保存するかどうかを変更するには、次のコマンドを実行します。

```
$ 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。以下に例を示します。

```
$ 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。

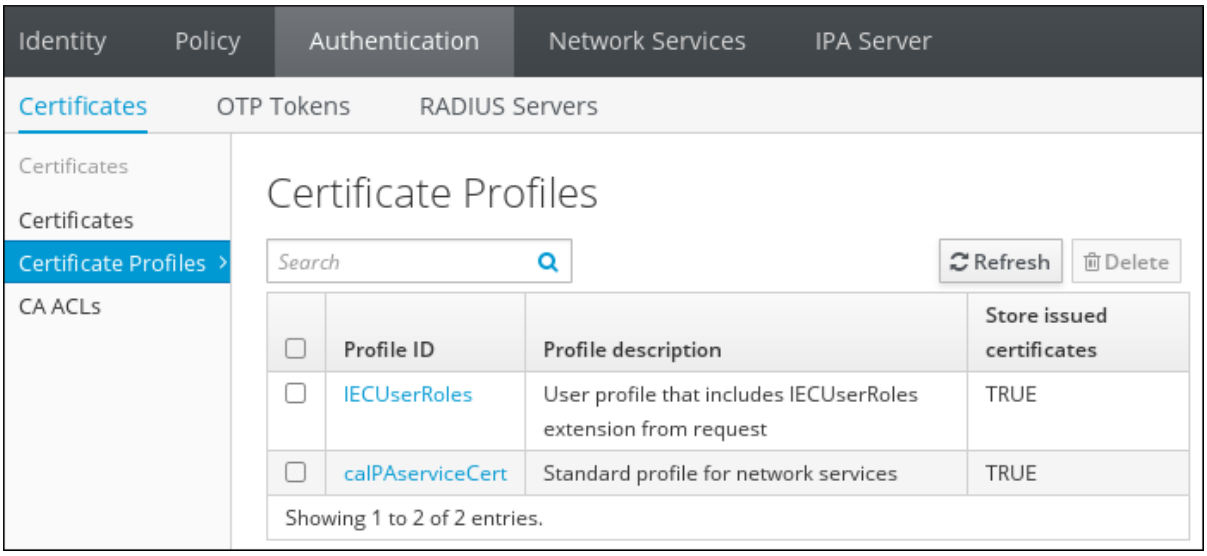
```
$ ipa certprofile-del profile_ID
----------------------
Deleted profile "profile_ID"
----------------------
```

24.4.3. 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、次のコマンドを実行します。

1. **Authentication** タブと **Certificates** サブタブを開きます。
2. **証明書プロファイル** セクションを開きます。

図24.7 Web UI の証明書プロファイル管理

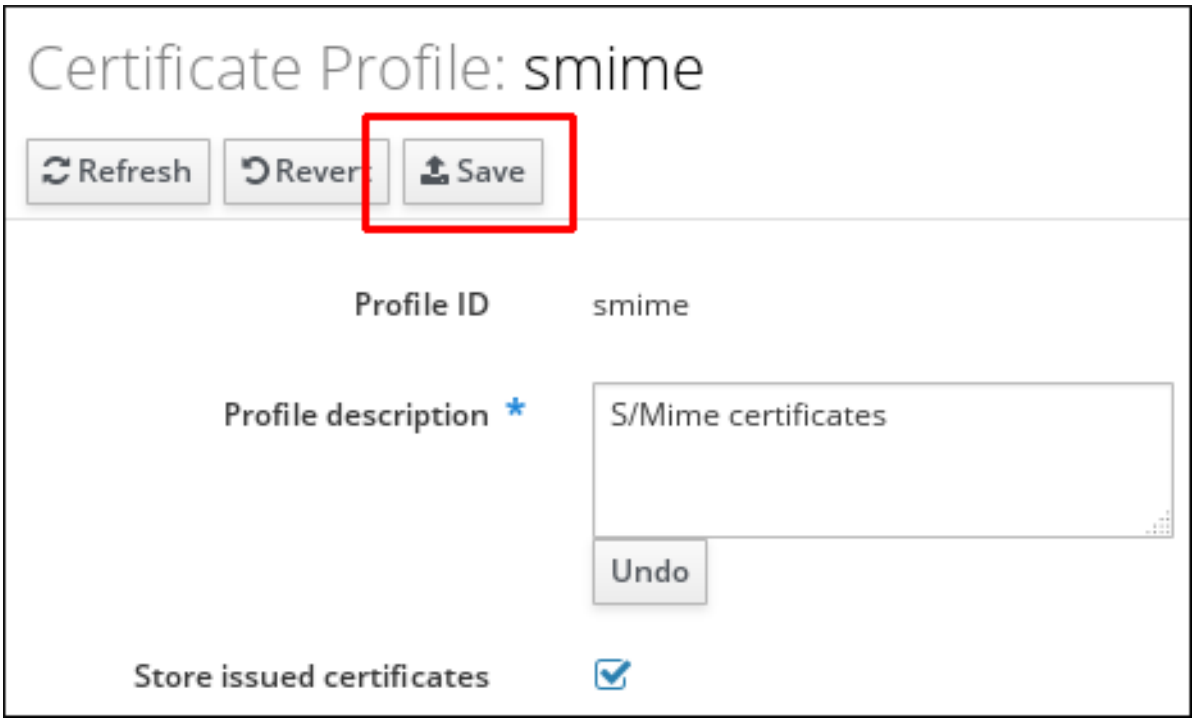


Certificate Profiles セクションで、既存のプロファイルに関する情報を表示したり、属性を変更したり、選択したプロファイルを削除したりできます。

たとえば、既存の証明書プロファイルを変更するには、次のコマンドを実行します。

1. プロファイル名をクリックして、プロファイル設定ページを開きます。
2. プロファイル設定ページで、必要な情報を入力します。
3. **Save** をクリックして、新しい設定を確定します。

図24.8 Web UI での証明書プロファイルの変更



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信され、ターゲットの IdM プリンシパルの **userCertificate** 属性に保存されます。オプションが無効になっていると、発行された証明書はクライアントに配信されますが、IdM には保存されません。複数の有効期限の短い証明書を発行する必要がある場合は、多くの場合、証明書の保存は無効になります。

Web UI では、証明書プロファイルの管理操作は現在利用できません。

- Web UI で証明書プロファイルをインポートすることはできません。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。
- 属性と値のペアの設定、追加、または削除はできません。属性と値のペアを変更するには、**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設定をインポートすることはできません。更新されたプロファイル設定を含むファイルをインポートするには、**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の管理に使用するコマンドの詳細は、「コマンドラインでの証明書プロファイル管理」 を参照してください。

24.4.4. 証明書プロファイルを使用した 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、サーバーに含まれるプロファイルはすべてインポートされ、有効になります。

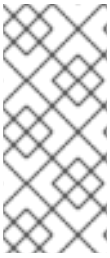
複数のサーバーレプリカをアップグレードすると、最初にアップグレードしたレプリカのプロファイルがインポートされます。他のレプリカでは、IdM は他のプロファイルの存在を検出し、インポートしないか、2 つのプロファイル間で競合を解決します。レプリカにカスタムプロファイルが定義されている場合は、アップグレード前にすべてのレプリカのプロファイルが一貫していることを確認してください。

24.5. 認証局 ACL ルール

認証局のアクセス制御リスト (CA ACL) ルールは、どのユーザー、サービス、またはホストにどのプロファイルを使用して証明書を発行するかを定義します。CA ACL は、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、特定のプロファイルを使用した証明書をプリンシパルまたはグループが要求できるようにします。

- ACL が複数のプロファイルへのアクセスを許可
- ACL には複数のユーザー、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を関連付けることができます。

たとえば、管理者は CA ACL を使用して、ロンドンのオフィスから作業す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であるユーザーに限定することができます。



注記

「証明書プロファイル」 および CA ACL で説明されている証明書プロファイルを組み合わせることにより、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し、制御できます。プロファイルおよび CA ACL を使用してユーザー証明書を発行する方法は、「IdM CA でユーザー証明書を発行するための証明書プロファイルおよび ACL の使用」 を参照してください。

24.5.1. コマンドラインでの CA ACL 管理

CA ACL ルールを管理する **caacl** プラグインを使用すると、特権ユーザーは、指定された CA ACL を追加、表示、変更、または削除できます。プラグインがサポートするすべてのコマンドを表示するには、**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を持つ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含まれています。

システム:CA ACL の読み取り

ユーザーが CA ACL のすべての属性を読み取りできるようにします。

システム:CA ACL の追加

ユーザーが新規 CA ACL を追加できるようにします。

システム:CA ACL の削除

ユーザーが既存の CA ACL を削除できるようにします。

システム:CA ACL の変更

ユーザーが CA ACL の属性を変更し、CA ACL を無効化または有効化できるようにします。

システム:CA ACL メンバーシップの管理

ユーザーが CA ACL 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できるようにします。

これらのすべてのパーミッションは、デフォルトの **CA Administrator** 特権に含まれます。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関する詳細は、「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、CA ACL 管理に **ipa caacl** コマンドを使用する最も重要な側面のみを説明します。コマンドの詳細は、以下のように **--help** オプションを指定して実行します。

```
$ 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

新しい CA ACL を作成するには、**ipa caacl-add** コマンドを使用します。オプションを指定せずにコマンドを実行すると、**ipa caacl-add** スクリプトにより、新しい CA ACL に必要な情報の入力が求められます。

```
$ ipa caacl-add
ACL name: smime_acl
------------------------
Added CA ACL "smime_acl"
------------------------
  ACL name: smime_acl
  Enabled: TRUE
```

新しい CA ACL はデフォルトで有効になっています。

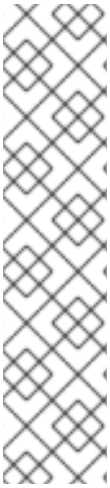
ipa caacl-add で使用できる最も注目すべきオプションは、CA ACL を CA、証明書プロファイル、ユーザー、ホスト、またはサービスカテゴリーに関連付けるオプションです。

- **--cacat**
- **--profilecat**
- **--usercat**
- **--hostcat**
- **--servicecat**

IdM は、このオプションを使用した**すべて**の値のみを受け入れ、CA ACL をすべての CA、プロファイル、ユーザー、ホスト、またはサービスに関連付けます。たとえば、CA ACL をすべてのユーザーおよびユーザーグループに関連付けるには、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、およびサービスカテゴリーは、特定のオブジェクトまたはオブジェクトのグループを CA ACL に追加する代わりに、「CA ACL へのエントリーの追加および CA ACL からエントリーの削除」 で説明されています。カテゴリーを使用したり、同じタイプのオブジェクトまたはグループを追加することはできません。たとえば、**--usercat=all** オプションを使用して、**ipa caacl-add-user --users=*user_name*** コマンドで CA ACL にユーザーを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要求しても失敗します。以下に例を示します。

```
$ 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.
```

「CA ACL へのエントリーの追加および CA ACL からエントリーの削除」 で説明されているように、ユーザーまたはグループを CA ACL に追加するか、CA ACL を **すべて**の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 は、**--cacat**、**--profilecat**、**--usercat**、**--hostcat** オプション、および **--servicecat** オプションが使用できることに注意してください。これは、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で CA ACL への検索の結果をフィルタリングするために使用できます。IdM では、このオプションの **すべて** のカテゴリーのみを許可することに注意してください。オプションの詳細は、「CA ACL の作成」 を参照してください。

特定の CA ACL に関する情報を表示するには、**ipa caacl-show** コマンドを使用します。

```
$ ipa caacl-show ca_acl_name
  ACL name: ca_acl_name
  Enabled: TRUE
  Host category: all
...
```

CA ACL の変更

既存の CA ACL を変更するには、**ipa caacl-mod** コマンドを使用します。**ipa caacl-mod** で使用できるコマンドラインオプションを使用して、必要な変更を渡します。たとえば、CA ACL の説明を変更し、CA ACL をすべての証明書プロファイルに関連付けるには、次のコマンドを実行します。

```
$ 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 が許可する最も注目すべきオプションは、**--cacat**、**--profilecat**、**--usercat**、**--hostcat**、および **--servicecat** です。これらのオプションの説明は、「CA ACL の作成」 を参照してください。

CA ACL の無効化および有効化

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"
----------------------------
```

無効にされた CA ACL は適用されず、証明書を要求するのに使用できません。CA ACL を無効にしても、IdM から削除されません。

無効にした 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 へのエントリーの追加および CA ACL からエントリーの削除

ipa caacl-add-* コマンドおよび **ipa caacl-remove-*** コマンドを使用すると、CA ACL に新しいエントリーを追加するか、既存のエントリーを削除できます。

ipa caacl-add-ca および ipa caacl-remove-ca

CA を追加または削除します。

ipa caacl-add-host および ipa caacl-remove-host

ホストまたはホストグループを追加または削除します。

ipa caacl-add-profile および 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 および ipa caacl-remove-service

サービスを追加または削除します。

ipa caacl-add-user および ipa caacl-remove-user

ユーザーまたはグループを追加または削除します。

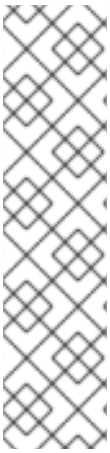
以下に例を示します。

```
$ ipa caacl-add-user ca_acl_name --groups=group_name
```

「CA ACL の作成」 で説明されているように、オブジェクトまたはオブジェクトのグループを CA ACL

に追加できず、で説明されているように、同じオブジェクトのカテゴリーを使用することができないことに注意してください。この設定は相互に排他的です。たとえば、**--user cat=all** オプションで指定した CA ACL で **ipa caacl-add-user --users=*user_name*** コマンドを実行しようとすると、コマンドは失敗します。

```
$ 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要求しても失敗します。以下に例を示します。

```
$ 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.
```

「CA ACL の作成」 で説明されているように、ユーザーまたはグループを CA ACL に追加するか、CA ACL を **すべて**のユーザーカテゴリーに関連付ける必要があります。

これらのコマンドと利用可能なオプションに必要な構文の詳細については **--help** オプションを追加してコマンドを実行します。以下に例を示します。

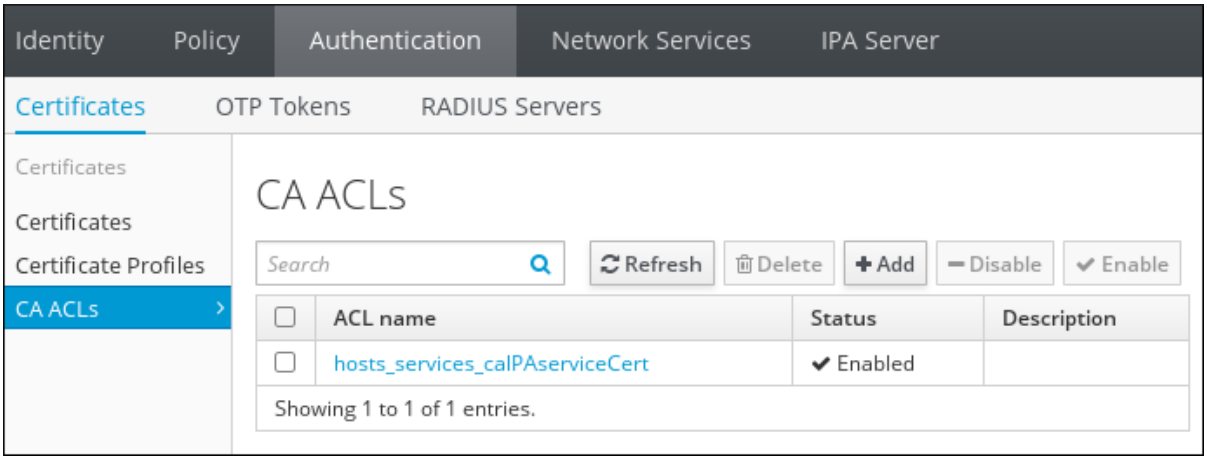
```
$ ipa caacl-add-user --help
```

24.5.2. Web UI からの CA ACL 管理

IdM Web UI で CA ACL を管理するには、以下を実行します。

1. **Authentication** タブと **Certificates** サブタブを開きます。
2. **CA ACL** 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



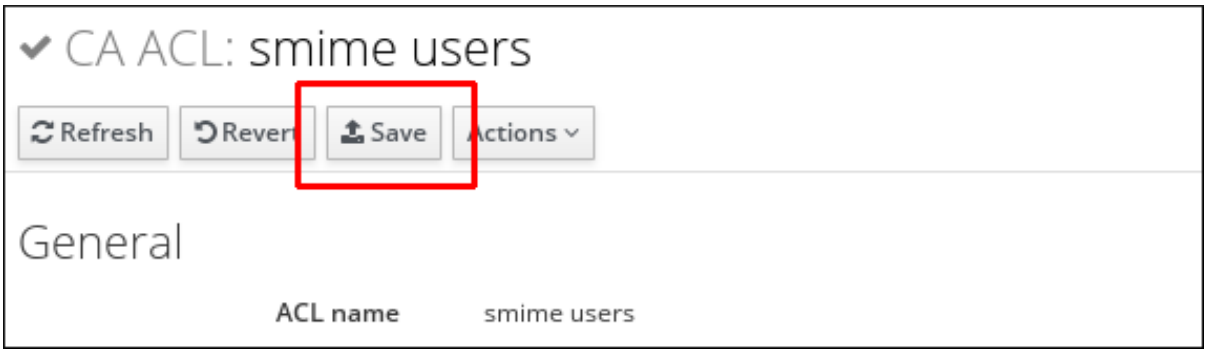
CA ACL セクションで、新しい CA ACL を追加、既存の CA ACL に関する情報の表示、属性の変更、選択した CA ACL の有効化、無効化、または削除を行うことができます。

たとえば、既存の CA ACL を変更するには、以下を実行します。

1. CA ACL の名前をクリックし、CA ACL 設定ページを開きます。
2. CA ACL 設定ページで、必要な情報を入力します。

   Profiles および **Permitted to have certificates issued** セクションを使用すると、CA ACL を証明書プロファイル、ユーザー、ユーザーグループ、ホスト、またはホストグループ、またはサービスに関連付けることができます。**Add** ボタンを使用して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 をすべてのユーザー、ホスト、またはサービスに関連付けることができます。
3. **Save** をクリックして、新しい設定を確定します。

図24.10 Web UI での CA ACL ルールの変更



24.6. IDM CA でユーザー証明書を発行するための証明書プロファイルおよび ACL の使用

ユーザーは、認証局のアクセス制御リスト (CA ACL) で許可される場合に、証明書を要求できます。以下の手順では、「証明書プロファイル」 および 「認証局 ACL ルール」 で別々に説明されている証明書プロファイルおよび CA ACL を使用します。証明書プロファイルおよび CA ACL の使用に関する詳細は、以下のセクションを参照してください。

コマンドラインでの証明書の発行

1. ユーザー証明書の要求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。以下に例を示します。

```
$ 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要求を許可するために使用される新しい認証局 (CA)ACL を追加します。以下に例を示します。

```
$ 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. ユーザーの証明書要求を生成します。例: OpenSSL の使用:

```
$ 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 にユーザーの新しい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
```

必要に応じて、**--ca *sub-CA_name*** オプションをコマンドに渡して、ルート CA **ipa** ではなくサブ CA から証明書を要求します。

新たに発行した証明書が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** コマンドを使用できます。

```
$ ipa user-show user
  User login: user
  ...
  Certificate: MIICfzCCAWcCAQA...
  ...
```

Web UI で証明書のユーザーへの発行

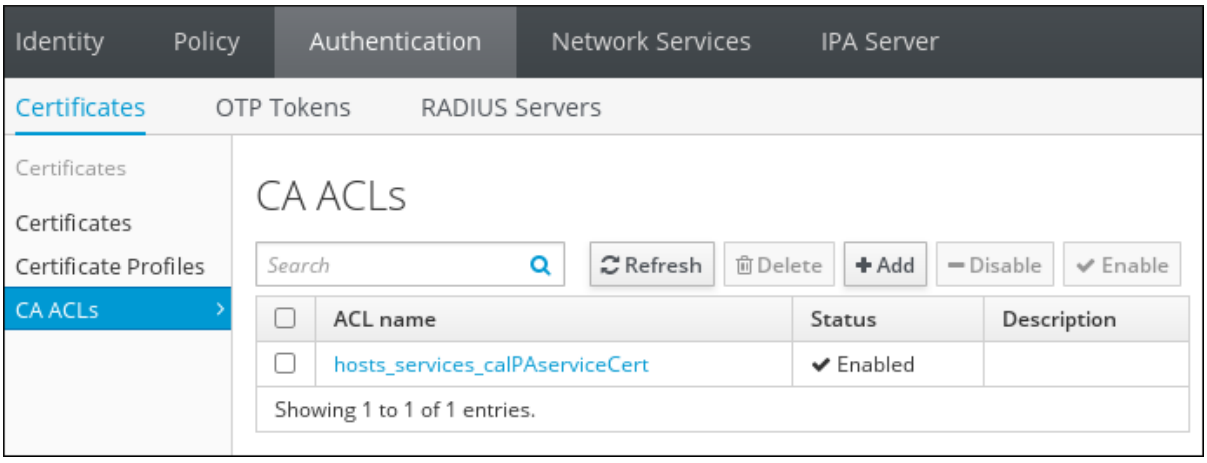
1. ユーザー証明書の要求を処理する新規のカスタム証明書プロファイルを作成またはインポートします。プロファイルのインポートはコマンドラインからのみ可能です。以下に例を示します。

```
$ ipa certprofile-import certificate_profile --file=certificate_profile.txt --store=True
```

証明書プロファイルの詳細は、「証明書プロファイル」 を参照してください。

2. Web UI の **Authentication** タブで、**CA ACL** セクションを開きます。

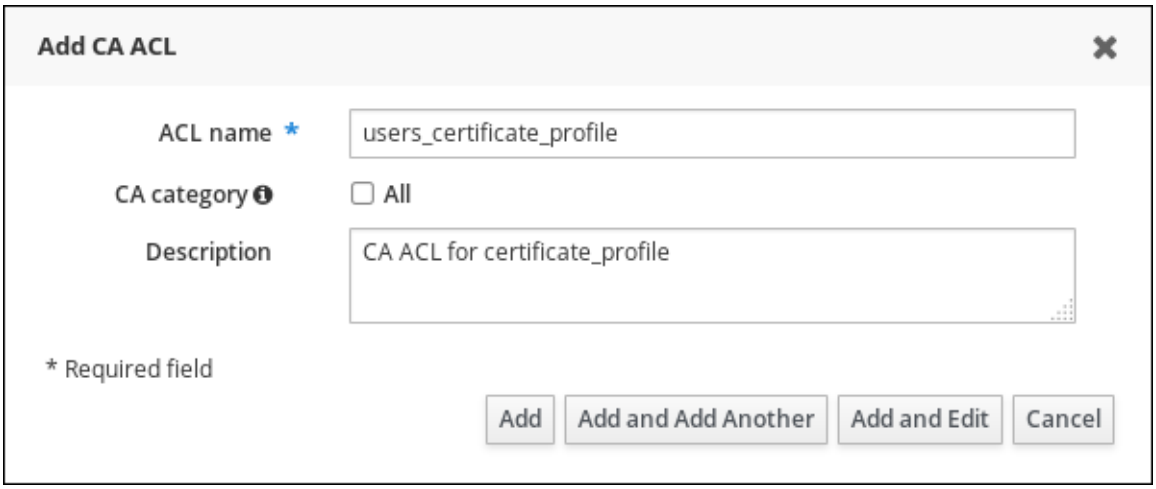
図24.11 Web UI での CA ACL ルールの管理



認証局 (CA) ACL リストの上部にある **Add** をクリックして、ユーザーエントリーの証明書を要求する新しい CA ACL を追加します。

a. 開いている **Add CA ACL** ウィンドウで、新規 CA ACL に必要な情報を入力します。

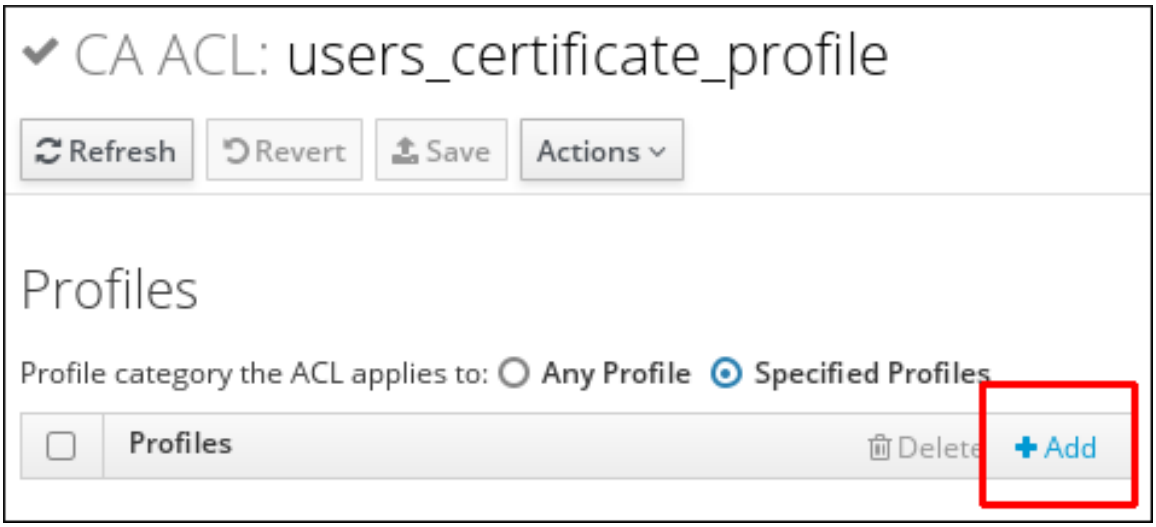
図24.12 新規 CA ACL の追加



次に、**Add and Edit** をクリックして CA ACL 設定ページに直接移動します。

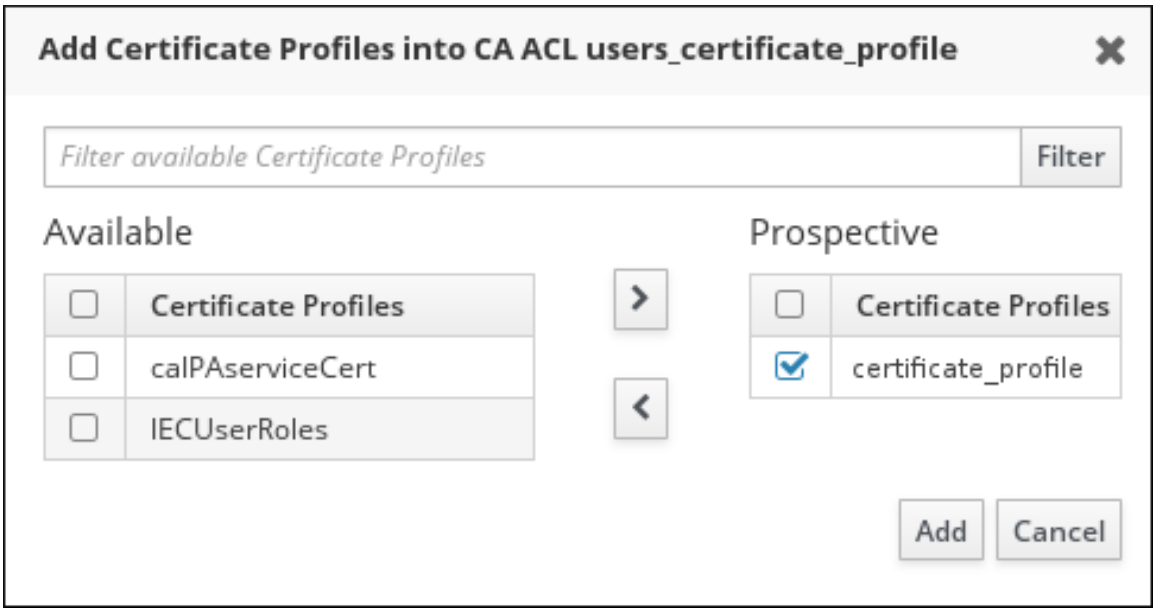
b. CA ACL 設定ページで **Profiles** セクションまでスクロールし、プロファイルリストの上部にある **Add** をクリックします。

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、**Prospective 列** に移動し、カスタム証明書プロファイルを CA ACL に追加します。

図24.14 証明書プロファイルの選択

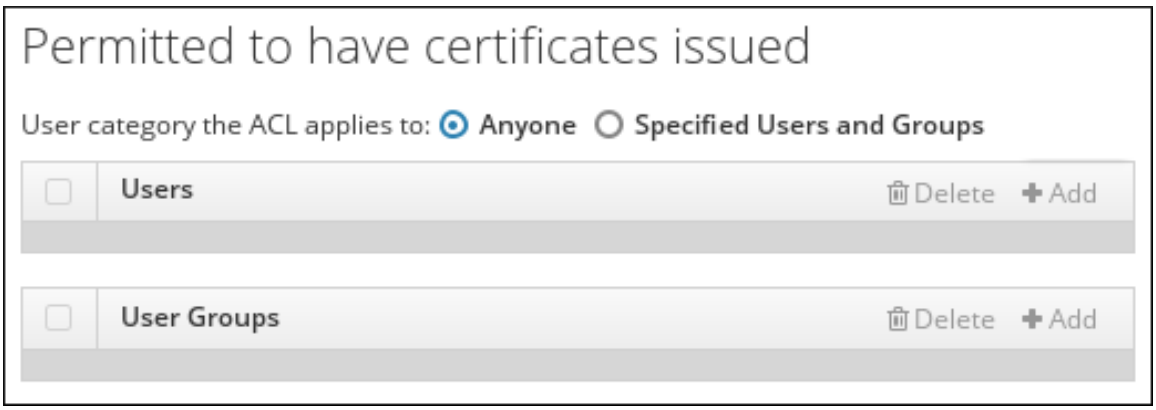


次に、**Add** をクリックします。

d. **Permitted to have certificates issued** セクションまでスクロールし、CA ACL をユーザーまたはユーザーグループに関連付けます。

Add ボタンを使用してユーザーまたはグループを追加するか、**Anyone** オプションを選択して CA ACL をすべてのユーザーに関連付けることができます。

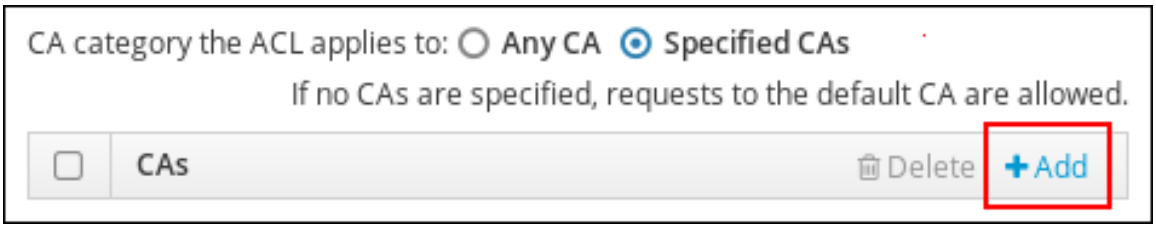
図24.15 CA ACL へのユーザーの追加



e. **Permitted to have certificates issued** セクションで、CA ACL を 1 つ以上の CA に関連付けることができます。

Add ボタンを使用して CA を追加するか、**Any CA** オプションを選択して CA ACL とすべての CA を関連付けることができます。

図24.16 CA ACL への CA の追加



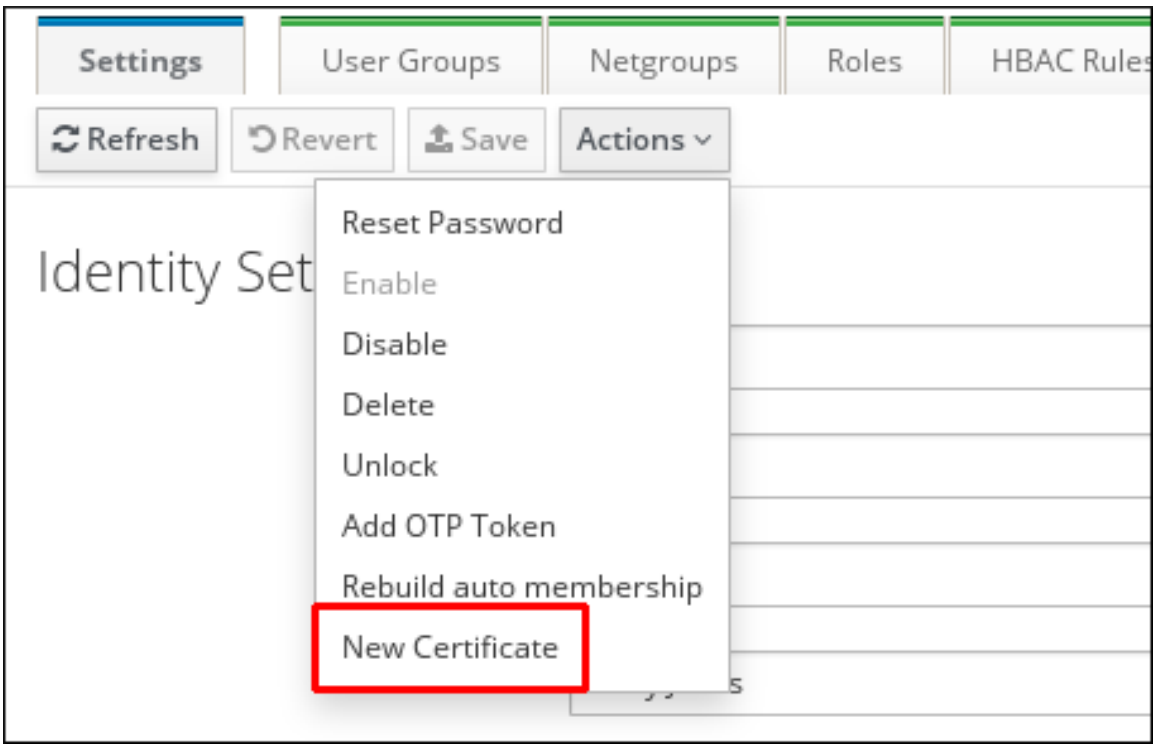
f. CA ACL 設定ページの上部にある **Save** をクリックし、CA ACL への変更を確認します。

3. ユーザーの新しい証明書を要求します。

   a. **Identity** タブおよび **Users** サブタブで、証明書が要求されるユーザーを選択します。ユーザーのユーザー名をクリックして、ユーザーエントリー設定ページを開きます。

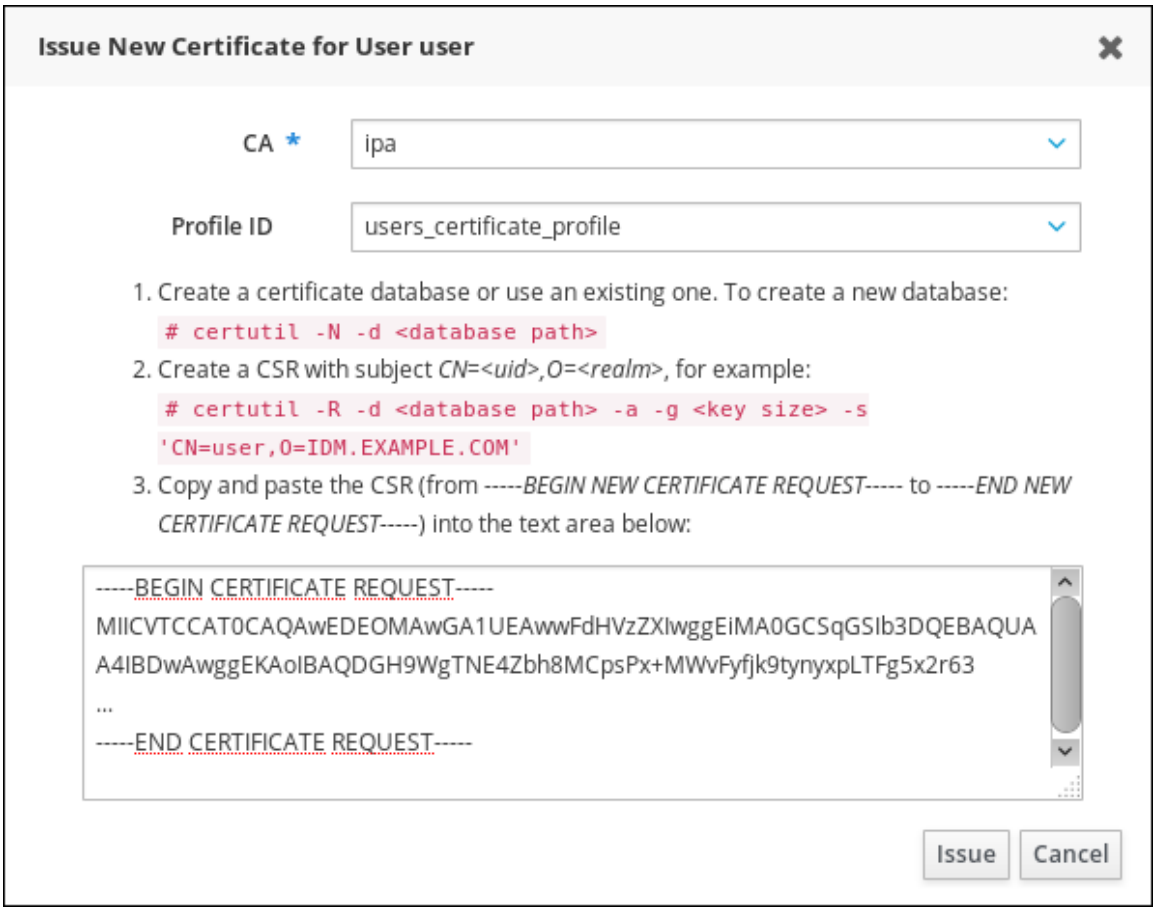
   b. ユーザー設定ページの上部にある **Actions** をクリックし、**New Certificate** をクリックします。

   図24.17 ユーザーの証明書の要求



   c. 必要な情報を入力します。

図24.18 ユーザーの証明書の発行



次に、**Issue** をクリックします。

これにより、新たに発行した証明書がユーザー設定ページに表示されます。

第25章 VAULT での認証シークレットの保存

Vault は、シークレットの保存、取得、共有、および復旧のセキュアな場所です。シークレットは、限られたユーザーまたはグループまたはエンティティーグループのみがアクセスできるようにするセキュリティーの影響を受けるデータです。たとえば、シークレットには以下が含まれます。

- パスワード
- 暗証番号
- SSH 秘密鍵

ユーザーおよびサービスは、Identity Management(IdM) ドメインに登録されているマシンから vault に保存されているシークレットにアクセスできます。

注記

Vault はコマンドラインからのみ利用でき、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が含まれます。

ユーザーの個人シークレットの保存

詳細は「ユーザーの個人シークレットの保存」を参照してください。

サービスのシークレットの保存

詳細は「Vault でのサービスシークレットの保存」を参照してください。

複数のユーザーによって使用される共通のシークレットの保存

詳細は「複数ユーザーの共通シークレットの保存」を参照してください。

Vault を使用するには、「Vault を使用するための前提条件」 に記載の条件を満たしている必要があります。

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 所有者は、vault の基本的な管理権限のあるユーザーまたはサービスです。たとえば、vault の所有者は vault のプロパティーを変更したり、新しい vault メンバーを追加したりできます。

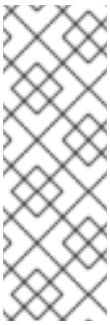
各 vault には最低でも所有者が 1 人必要です。vault には複数の所有者を指定することもできます。

Vault メンバー

vault メンバーは、別のユーザーまたは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 管理者は全 vault に制限なくアクセスでき、vault の操作をすべて実行できます。



注記

対称と非対称 vault は、パスワードまたは鍵で保護されており、特別なアクセス制御ルールが適用されます (「標準、対称および非対称 vault」 を参照)。管理者は、以下を行うためにこの特別なルールを満たす必要があります。

- 対称および非対称 vault のシークレットにアクセスする。
- vault パスワードまたはキーを変更またはリセットする。

vault 管理者は、**vault administrators** 特権を持つユーザーです。ユーザー権限の定義については、「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の特権は、vault のタイプによって異なります。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 コマンドなどの一部のコマンドの出力には、ユーザー vault 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 のあるコンテナー内のユーザーです。vault コンテナーおよびユーザー vault の詳細は、「各種 Vault コンテナー」 および 「ユーザー、サービスおよび共有 vault」 を参照してください。

25.1.2. 標準、対称および非対称 vault

以下の vault タイプは、セキュリティーおよびアクセス制御のレベルに基づいています。

標準 vault

Vault の所有者と vault メンバーは、パスワードやキーを使用せずにシークレットをアーカイブして取得できます。

対称 vault

vault のシークレットは対称キーを使用して保護されます。vault のメンバーおよび所有者は、シークレットをアーカイブして取得できますが、vault パスワードを指定する必要があります。

非対称 vault

vault のシークレットは非対称キーを使用して保護されます。ユーザーは公開鍵でシークレットをアーカイブし、秘密鍵でシークレットを取得します。vault メンバーはシークレットのアーカイブのみが可能ですが、vault 所有者はシークレットのアーカイブと取得の両方が可能です。

25.1.3. ユーザー、サービスおよび共有 vault

以下の Vault タイプは所有権に基づいています。

ユーザー vault: ユーザーのプライベート vault

所有者: 単一ユーザー。

どのユーザーも 1 人以上のユーザー vault を所有できます。

サービス vault: サービスのプライベート vault

所有者: 単一サービス

サービスは、1 つまたは複数のサービス vault を所有することができます。

共有 vault

所有者: vault を作成した vault の管理者他の vault 管理者は、vault への完全アクセスもあります。

共有 vault は、複数のユーザーまたはサービスが使用できます。

25.1.4. 各種 Vault コンテナー

vault コンテナーは vault のコレクションです。

IdM は、以下のデフォルトの vault コンテナーを提供します。

ユーザーコンテナー: ユーザーのプライベートコンテナー

このコンテナーは、特定ユーザーのユーザー vault を保存します。

サービスコンテナー: サービスのプライベートコンテナー

このコンテナーは、特定のサービスのサービス vault を格納します。

共有コンテナー

このコンテナーは、複数のユーザーまたはサービスで共有できる vault を格納します。

IdM では、ユーザーまたはサービスのプライベート vault が初めて作成されると、ユーザーまたはサービスごとにユーザーコンテナーおよびサービスコンテナーを自動的に作成します。ユーザーまたはサービスが削除されると、IdM はコンテナーとそのコンテンツを削除します。

25.2. VAULT を使用するための前提条件

Vault を有効にするには、IdM ドメインの 1 つ以上のサーバーに Key Recovery Authority(KRA)Certificate System コンポーネントをインストールします。

```
# ipa-kra-install
```

注記

vault サービスを高可用性に設定するには、IdM サーバー 2 台以上に KRA をインストールします。

25.3. VAULT コマンドのヘルプの取得

Vault および vault コンテナーの管理に使用するコマンドをすべて表示するには、次のコマンドを実行します。

```
$ ipa help vault
```

特定のコマンドの詳細なヘルプを表示するには、コマンドに **--help** オプションを追加します。

```
$ ipa vault-add --help
```

vault コマンドが失敗し、vault not found エラーが出力
コマンドによっては、以下のオプションを使用して vault の所有者またはタイプを指定する必要があります。

- **--user** または **--service** は、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 **--user** を追加せずに別のユーザーの vault を表示しようとすると、IdM により vault が見つからないことが通知されます。

```
[admin@server ~]$ ipa vault-show user_vault
ipa: ERROR: user_vault: vault not found
```

25.4. ユーザーの個人シークレットの保存

このセクションでは、1 つ以上のプライベート vault を作成して個人のシークレットをセキュアに保存する方法を説明します。その後、ユーザーは必要に応じてドメインの任意のマシンでシークレットを取得します。たとえば、ユーザーは vault に個人証明書をアーカイブできるため、証明書を一元的にセキュアに保存できます。

このセクションでは、以下の手順について説明します。

- 「ユーザーの個人シークレットのアーカイブ」
- 「ユーザーの個人シークレットの取得」

本手順での以下の用語について説明します。

- **user** は vault を作成するユーザーである。
- **my_vault** はユーザーの証明書保存に使用する vault である。
- アーカイブした証明書にアクセスするのに vault のパスワードを指定しなくてもいいように vault タイプが **standard** に設定されている。
- **secret.txt** は vault に保存する証明書が含まれるファイルです。
- **secret_exported.txt** は、ユーザーがアーカイブした証明書をエクスポートするファイルです。

25.4.1. ユーザーの個人シークレットのアーカイブ

プライベートユーザー vault を作成し、証明書を保存します。Vault タイプは standard です。これにより、証明書へのアクセス時に認証は必要ありません。

1. **user**としてログインします。

```
$ kinit user
```

2. **ipa vault-add** コマンドを使用して、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
```

重要

最初のユーザー vault の作成には、同じユーザーが使用されているようにしてください。たとえば、**admin** などの別のユーザーが **user1** の最初のユーザー vault を作成する場合には、ユーザーの vault コンテナーの所有者も **admin** になり、**user1** はユーザー vault にアクセスしたり、新しいユーザー vault を作成したりできません。「十分な追加権限がないことが原因で Vault にユーザーがアクセスできない」も参照してください。

3. **ipa vault-archive --in** コマンドを使用して、**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

1 つの vault は 1 つのシークレットのみを保存することができます。

25.4.2. ユーザーの個人シークレットの取得

プライベート標準 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** コマンドを使用して vault の内容を取得し、**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 でのサービスシークレットの保存

このセクションでは、管理者が vault を使用してサービスシークレットを一元的にセキュアに保存する方法を説明します。サービスシークレットはサービスの公開鍵で暗号化されます。その後、サービスはドメイン内のマシン上の秘密鍵を使用してシークレットを取得します。シークレットにアクセスできるのは、サービスと管理者のみです。

このセクションでは、以下の手順について説明します。

- 「サービス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プロビジョニング」
- 「サービスインスタンスのサービスパスワードの取得」
- 「サービス vault パスワードの変更」

本手順での以下の用語について説明します。

- **admin** は、サービスパスワードを管理する管理者です。
- **http_password** は、管理者が作成したプライベートユーザー vault の名前です。
- **password.txt** はサービスパスワードが含まれるファイルです。
- **password_vault** は、サービス用に作成された vault です。
- **Http/server.example.com** は、パスワードがアーカイブされるサービスです。
- **service-public.pem** 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鍵です。

25.5.1. サービス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、これを使用してサービスパスワードを保存します。Vault タイプは standard で、vault の内容にアクセスする際に管理者が認証する必要がないようにします。

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 にアーカイブしたら、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プロビジョニング

サービス用に作成された非対称 vault を使用して、サービスインスタンスにサービスパスワードをプロビジョニングします。

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鍵を取得します。たとえば、**openssl** ユーティリティーを使用する場合は以下を行います。

   a. **service-private.pem** 秘密鍵を生成します。

```
$ openssl genrsa -out service-private.pem 2048
Generating RSA private key, 2048 bit long modulus
.+++
.....................................+++
e is 65537 (0x10001)
```

   b. 秘密鍵をもとに **service-public.pem** 公開鍵を生成します。

```
$ 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 vault として非対称 vault を作成し、公開鍵を指定します。

```
$ 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 にアーカイブされたパスワードはこの鍵で保護されます。

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てから、新しいサービス 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"
----------------------------------
```

これにより、サービスインスタンスの公開鍵でパスワードを暗号化します。

警告

パスワードを vault にアーカイブしたら、システムから **password.txt** を削除します。

上記の手順を、パスワードを必要とする全サービスインスタンスで繰り返します。サービスインスタンスごとに新規の非対称 vault を作成します。

25.5.3. サービスインスタンスのサービスパスワードの取得

サービスインスタンスは、ローカルに保存されたサービスの秘密鍵を使用してサービス vault パスワードを取得できます。

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 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 パスワードの変更

サービスインスタンスが危険にさらされたら、サービス vault パスワードを変更して、新しいパスワードを侵害されていないサービスインスタンスにのみ再プロビジョニングして分離します。

1. 管理者のユーザー vault に新しい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"
----------------------------------------
```

これにより、vault に保存されている現在のパスワードが上書きされます。

2. 不正アクセスされたインスタンスを除く、各サービスインスタンスに新しいパスワードを再プロビジョニングします。

a. 管理者の vault から新しい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. 新しいパスワードをサービスインスタンス 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 にアーカイブしたら、システムから **password.txt** を削除します。

25.6. 複数ユーザーの共通シークレットの保存

このセクションでは、管理者が共有 vault を作成し、他のユーザーが vault のシークレットにアクセスできるようにする方法を説明します。管理者は共通のパスワードを vault にアーカイブし、他のユーザーはドメイン内の任意のマシンでパスワードを取得できます。

このセクションでは、以下の手順について説明します。

- 「メンバーユーザーとしての共有 Vault からのシークレットの取得」
- 「Common Secret を使用した共有 vault の作成」

本手順での以下の用語について説明します。

- **shared_vault** とは、共通パスワードを保存する vault です。

- **admin** は、共有 vault を作成する管理者です。
- アーカイブしたパスワードにアクセスするのに vault のパスワードを指定しなくてもいいように vault タイプが **standard** に設定されている。
- **secret.txt** は共通のシークレットが含まれるファイルです。
- **user1** および **user2** は vault へのアクセスが許可されるユーザーです。

25.6.1. Common Secret を使用した共有 vault の作成

共有 vault を作成し、これを使用して共通のシークレットを保存します。シークレット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 standard で、シークレットにアクセスするユーザーが認証する必要がないようにします。

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. シークレットを 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ことを指定します。

```
$ ipa vault-archive shared_vault --shared --in secret.txt
-----------------------------------
Archived data into vault "shared_vault"
-----------------------------------
```

注記

1 つの vault は 1 つのシークレットのみを保存することができます。

4. **user1** および **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の共有 Vault からのシークレットの取得

Vault のメンバーユーザーとしてログインし、vault からシークレットのあるファイルを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 からシークレットを取得します。

```
$ ipa vault-retrieve shared_vault --shared --out secret_exported.txt
----------------------------------------
Retrieved data from vault "shared_vault"
----------------------------------------
```

25.7. VAULT のパスワードまたは公開鍵の変更

Vault の所有者は vault のパスワードを変更できます。Vault が対称または非対称であるかに応じて、コマンドは以下のようになります。

- 対称 vault のパスワードを変更するには、次のコマンドを実行します。

```
# ipa vault-mod --change-password
Vault name: example_symmetric_vault
Password: old_password
New password: new_password
Enter New password again to verify: new_password
----------------------
Modified vault "example_symmetric_vault"
----------------------
  Vault name: example_symmetric_vault
  Type: symmetric
  Salt: dT+M+4ik/ltgnpstmCG1sw==
  Owner users: admin
  Vault user: admin
```

- 非対称 vault の公開鍵を変更するには、次のコマンドを実行します。

```
# ipa vault-mod example_asymmetric_vault --private-key-file=old_private_key.pem --public-
key-file=new_public_key.pem
------------------------------
Modified vault "example_assymmetric_vault"
------------------------------
  Vault name: example_assymmetric_vault
  Typ: asymmetric
  Public key: ...
  Owner users: admin
  Vault user: admin
```

第26章 証明書と認証局の管理

26.1. 軽量サブ CA

IdM インストールが統合証明書システム (CS) 認証局 (CA) で設定されている場合は、軽量のサブ CA を作成できます。これにより、1つのサブ CA が発行する証明書のみを受け入れるように、仮想プライベートネットワーク (VPN) ゲートウェイなどの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 が発行する証明書のみを受け入れるように他のサービスを設定できます。

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 を使用して IdM を設定する場合は、自動作成された **ipa** CA は、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 の下位局になります。

26.1.1. 軽量のサブ 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。

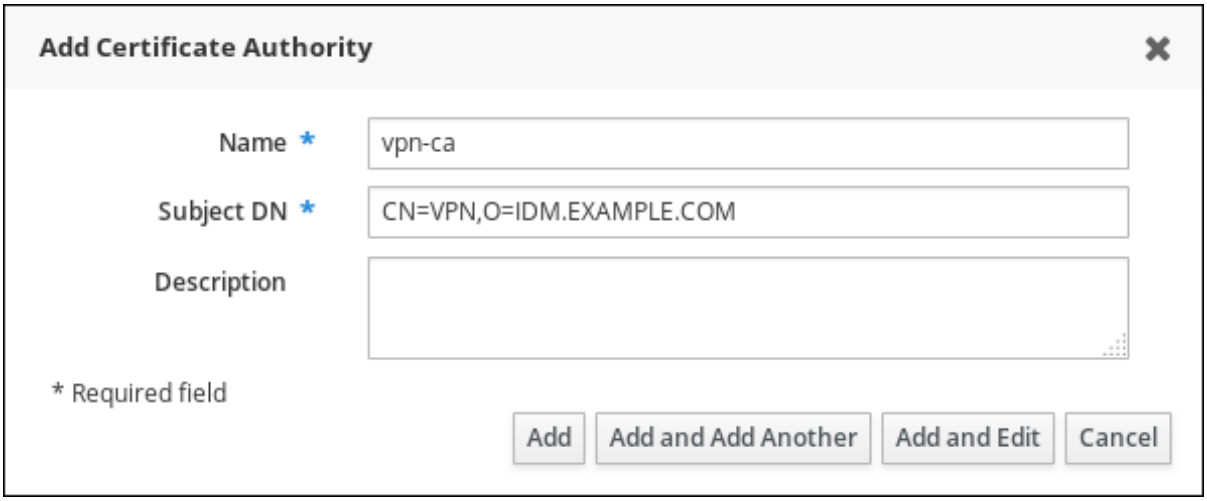
- 「Web UI でのサブ CA の作成」
- 「コマンドラインでのサブ CA の作成」

Web UI でのサブ CA の作成

vpn-ca という名前のサブ CA を新たに作成するには、以下を実行します。

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、**Add** をクリックします。
3. CA の名前およびサブジェクト 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ある必要があります。

コマンドラインでのサブ CA の作成

vpn-ca という名前のサブ CA を新たに作成するには、次のコマンドを実行します。

```
[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
```

名前

CA の名前

認証局 ID

CA 用に自動作成される個別 ID。

発行先 DN

サブジェクト識別名 (DN)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ある必要があります。

発行者 DN

サブ CA 証明書を発行した親 CA。サブ CA はすべて、IdM のルート CA の子として作成されます。

新しい CA 署名証明書が IdM データベースに正常に追加されたことを確認するには、次のコマンドを実行します。

```
[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

新しい CA 証明書は、証明書システムインスタンスがインストールされていると、すべてのレプリカに自動的に転送されます。

26.1.2. 軽量サブ CA の削除

サブ CA の削除に関する詳細は、を参照してください。

- 「Web UI からのサブ CA の削除」
- 「コマンドラインでのサブ 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、**Delete** をクリックします。
4. **Delete** をクリックして確定します。

コマンドラインでのサブ 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

詳細は、以下を参照してください。

- 証明書の更新が自動で行われています。「証明書の自動更新」を参照してください。
- 手動による証明書の更新。「CA 証明書の手動更新」 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、有効期限が切れる前に以下の証明書の 28 日を自動的に更新します。

- IdM CA がルート CA として発行する CA 証明書
- 内部 IdM サービスが使用する統合 IdM CA が発行するサブシステム証明書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** 追跡リストに記載する必要があります。追跡リストを更新するには、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 として使用している場合は、「CA 証明書の手動更新」 の説明に従って証明書を手動で更新する必要があります。**Certmonger** サービスは、外部 CA により署名された証明書を自動的に更新することはできません。

Certmonger が 証明書の有効期限 を監視する方法の詳細は、『certmonger を使用した証明書の追跡』のシステムレベルの認証ガイド を参照してください。

自動更新が想定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確認します。

- 証明書の更新後、**certmonger** は以下のようなメッセージを記録し、更新操作が成功したか、失敗したことを示します。

```
Certificate named "NSS Certificate DB" in token "auditSigningCert cert-pki-ca" in database "/var/lib/pki-ca/alias" renew success
```

- 証明書が有効期限に近づくと、**certmonger** は次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. CA 証明書の手動更新

ipa-cacert-manage ユーティリティーを使用して、以下を手動で更新できます。

- 自己署名 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、古い証明書と同じキーペアと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を有効にするために以前のバージョンが削除されません。

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. 自己署名証明書の手動更新

1. **ipa-cacert-manage --renew** コマンドを実行します。このコマンドでは、証明書へのパスを指定する必要はありません。
2. 更新された証明書が LDAP 証明書ストアと、**/etc/pki/pki-tomcat/alias** NSS データベースに表示されるようになりました。
3. すべてのサーバーとクライアントで **ipa-certupdate** ユーティリティーを実行し、LDAP から新しい証明書に関する情報で更新します。すべてのサーバーとクライアントで **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は、常に **ipa-certupdate** を実行します。これがない場合、証明書は他のマシンに配布されません。

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。

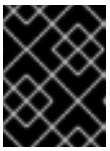
2. このコマンドは、**/var/lib/ipa/ca.crt** CSR ファイルを作成します。CSR を外部 CA に送信して、更新した CA 証明書を取得します。

3. **ipa-cacert-manage renew** 再度実行し、**--external-cert-file** オプションを使用して更新された CA 証明書と外部 CA 証明書チェーンファイルを指定します。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. 更新された CA 証明書と外部 CA 証明書チェーンが LDAP 証明書ストアと、**/etc/pki/pki-tomcat/alias/** NSS データベースに存在するようになりました。

5. すべてのサーバーとクライアントで **ipa-certupdate** ユーティリティーを実行し、LDAP から新しい証明書に関する情報で更新します。すべてのサーバーとクライアントで **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は、常に **ipa-certupdate** を実行します。これがない場合、証明書は他のマシンに配布されません。

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/
```

26.2.3. IdM がオフライン時に期限切れのシステム証明書の更新

システム証明書の期限が切れると、IdM が起動できません。IdM は、**ipa-cert-fix** ツールを使用して、このような状況であってもシステム証明書の更新に対応します。

前提条件

- ホストで **ipactl start --ignore-service-failures** コマンドを入力して、LDAP サービスが実行中であることを確認する。

手順26.1IdM サーバーで期限切れのシステム証明書の更新

1. IdM ドメインの CA で、以下を行います。

   a. **Ipa-cert-fix** ユーティリティーを起動してシステムを調整し、期限切れの証明書をリスト表示します。

```
# ipa-cert-fix
...
The following certificates will be renewed:

Dogtag sslserver certificate:
  Subject: CN=ca1.example.com,O=EXAMPLE.COM 201905222205
  Serial:  13
  Expires: 2019-05-12 05:55:47
...
Enter "yes" to proceed:
```

b. 更新プロセスを開始するには、**yes** を入力します。

```
Enter "yes" to proceed: yes
Proceeding.
Renewed Dogtag sslserver certificate:
  Subject: CN=ca1.example.com,O=EXAMPLE.COM 201905222205
  Serial:  268369925
  Expires: 2021-08-14 02:19:33
...

Becoming renewal master.
The ipa-cert-fix command was successful
```

ipa-cert-fix が期限切れの証明書をすべて更新する前に、最大 1 分かかる場合があります。

注記

更新マスターではない CA ホストで **ipa-cert-fix** ユーティリティーを実行し、ユーティリティーが共有証明書を更新すると、このホストはドメインの新しい更新マスターになります。不整合を避けるために、ドメインには常に更新マスターを 1 つだけ設定する必要があります。

c. 必要に応じて、すべてのサービスが実行中であることを確認します。

```
# ipactl status
Directory Service: RUNNING
krb5kdc Service: RUNNING
kadmin Service: RUNNING
httpd Service: RUNNING
ipa-custodia Service: RUNNING
pki-tomcatd Service: RUNNING
ipa-otpd Service: RUNNING
ipa: INFO: The ipactl command was successful
```

2. IdM ドメインの他のサーバーの場合:

a. ***--force*** パラメーターを使用して IdM を再起動します。

```
# ipactl restart --force
```

--force パラメーターを使用すると、**ipactl** ユーティリティーは個々の起動失敗を無視します。たとえば、サーバーが CA もあると、**pki-tomcat** サービスが起動に失敗します。***--force*** パラメーターを使用しているため、これが予想され、無視されます。

b. 再起動後に、**certmonger** サービスが証明書を更新することを確認します。

```
# getcert list | egrep '^Request|status:|subject:'
Request ID '20190522120745':
        status: MONITORING
        subject: CN=IPA RA,O=EXAMPLE.COM 201905222205
Request ID '20190522120834':
        status: MONITORING
        subject: CN=Certificate Authority,O=EXAMPLE.COM 201905222205
...
```

-

certmonger がレプリカ上で共有証明書を更新する前に時間がかかる場合があることに注意してください。

c. サーバーも CA の場合、上記のコマンドは、**pki-tomcat** サービスが使用する証明書の **CA_UNREACHABLE** を報告します。

```
Request ID '20190522120835':
        status: CA_UNREACHABLE
        subject: CN=ca2.example.com,O=EXAMPLE.COM 201905222205
...
```

この証明書を更新するには、**ipa-cert-fix** ユーティリティーを使用します。

```
# ipa-cert-fix
Dogtag sslserver certificate:
  Subject: CN=ca2.example.com,O=EXAMPLE.COM
  Serial:  3
  Expires: 2019-05-11 12:07:11

Enter "yes" to proceed: yes
Proceeding.
Renewed Dogtag sslserver certificate:
  Subject: CN=ca2.example.com,O=EXAMPLE.COM 201905222205
  Serial:  15
  Expires: 2019-08-14 04:25:05

The ipa-cert-fix command was successful
```

26.3. CA 証明書の手動インストール

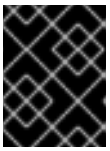
新規証明書を IdM にインストールするには、**ipa-cacert-manage install** コマンドを使用します。たとえば、このコマンドでは、有効期限の近くにある場合に、現在の証明書を変更できます。

1. **ipa-cacert-manage install** コマンドを実行し、証明書が含まれ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入れます。

```
[root@server ~]# ipa-cacert-manage install /etc/group/cert.pem
```

これで、証明書が LDAP 証明書ストアに表示されます。

2. すべてのサーバーとクライアントで **ipa-certupdate** ユーティリティーを実行し、LDAP から新しい証明書に関する情報で更新します。すべてのサーバーとクライアントで **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は、常に **ipa-certupdate** を実行します。これがない場合、証明書は他のマシンに配布されません。

ipa-cacert-manage install コマンドは、以下のオプションを使用できます。

-n

証明書のニックネームを指定します。デフォルト値は証明書のサブジェクト名です。

-t

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の詳細は、ipa-cacert-manage(1) の 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

ipa-cacert-manage 更新を使用して CA 証明書を更新して、証明書チェーンを変更できます。

自己署名 CA 証明書 → 外部署名 CA 証明書

--external-ca オプションを **ipa-cacert-manage renew** 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 CA 証明書として更新します。

このオプションを使用したコマンドの実行の詳細は、「CA 証明書の手動更新」 を参照してください。

外部署名 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 CA 証明書を自己署名の CA 証明書として更新されます。

26.5. IDM が期限切れの証明書で起動できるようにする

IdM 管理サーバーの証明書が期限切れになると、ほとんどの IdM サービスにアクセスできなくなります。基礎となる Apache サービスおよび LDAP サービスを設定し、証明書が期限切れであってもサービスへの SSL アクセスを許可できます。

期限切れの証明書でアクセスを許可する場合は、以下を行います。

- Apache、Kerberos、DNS、および LDAP サービスは引き続き操作します。これらのサービスをアクティブにすることで、ユーザーは IdM ドメインにログインできるようになります。
- アクセスに SSL を必要とするクライアントサービスは引き続き失敗します。たとえば、IdM クライアントで SSSD を必要とし、SSSD は IdM と通信するのに SSL が必要なため、**sudo** は失敗します。

重要

この手順は、一時的な回避策としてのみ意図されています。必要な証明書をできるだけ早く更新し、上記の変更を元に戻します。

1. Apache サーバーが有効な証明書を適用しないように **mod_nss** モジュールを設定します。
   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は、その属性を変更します。

```
# 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. Directory Server を再起動します。

```
# systemctl restart dirsrv.target
```

26.6. HTTP または LDAP のサードパーティーの証明書のインストール

Apache Web Server、Directory Server、またはその両方用に新しい SSL サーバー証明書をインストールすると、現在の SSL 証明書を新しい SSL 証明書に置き換えられます。これを実行するには、以下が必要です。

- プライベート SSL キー (以下の手順の**ssl.key**)
- SSL 証明書 (以下の手順の**ssl.crt**)

キーおよび証明書で使用できる形式のリストは、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ビスによって認識される CA により署名される必要があります。そうでない場合は、「CA 証明書の手動インストール」 の説明に従って、**ssl.crt** に署名した CA の CA 証明書をインストールします。

これにより、IdM は CA を認識できるため、**ssl.crt** が許可されます。

サードパーティー証明書のインストール

1. **ipa-server-certinstall** ユーティリティーを使用して証明書をインストールします。インストール先の場所を指定します。

   - **--http** は、Apache Web Server 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 証明書を両方にインストールするには、以下を行います。

```
# ipa-server-certinstall --http --dirsrv ssl.key ssl.crt
```

2. 証明書をインストールしたサーバーを再起動します。

   - Apache Web Server を再起動するには、以下を行います。

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

   - Directory Server を再起動するには、以下を実行します。

     ```
     # 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、証明書データベースに証明書が存在することを確認します。

   - Apache 証明書データベースを表示するには、以下を行います。

     ```
     # certutil -L -d /etc/httpd/alias
     ```

   - Directory Server 証明書データベースを表示するには、以下を実行します。

     ```
     # certutil -L -d /etc/dirsrv/slapd-REALM/
     ```

26.7. OCSP 応答の設定

IdM サーバーと統合されるすべての CA は、内部のオンライン証明書ステータスプロトコル (OCSP) レスポンダーを使用します。OCSP 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** で利用できます。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することができます。

注記

OCSP の詳細は、Red Hat Certificate System のドキュメントを参照してください。例: 『計画、インストール、およびデプロイメントのガイド』 の 2.2.4 証明書の取り消しおよびステータスの確認 を参照してください。

26.7.1. CRL 更新間隔の変更

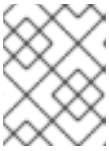
CRL ファイルは、デフォルトで 4 時間ごとに IdM CA により自動的に生成されます。この間隔を変更するには、以下を実行します。

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、60 分ごとに CRL を生成するには、次のコマンドを実行します。

   ```
   ca.crl.MasterCRL.autoUpdateInterval=60
   ```



注記

ca.crl.MasterCRL.autoUpdateInterval パラメーターを更新すると、すでにスケジュールされている CRL の次回更新後に変更が有効になります。

3. CA サーバーを起動します。

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

26.8. 既存の IDM ドメインへの CA のインストール

IdM ドメインが認証局 (CA) なしでインストールされている場合は、後で CA サービスをインストールできます。環境に応じて、IdM Certificate Server CA をインストールするか、外部 CA を使用します。

注記

サポート対象の CA 設定の詳細は、「使用する CA 設定の決定」 を参照してください。

IdM 証明書サーバーのインストール

1. 以下のコマンドを使用して、IdM Certificate Server CA をインストールします。

```
[root@ipa-server ~] ipa-ca-install
```

2. すべてのサーバーとクライアントで **ipa-certupdate** ユーティリティーを実行し、LDAP から新しい証明書に関する情報で更新します。すべてのサーバーとクライアントで **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は、常に **ipa-certupdate** を実行します。これがない場合、証明書は他のマシンに配布されません。

外部 CA のインストール

外部 CA の後続のインストールは、複数の手順で設定されます。

1. インストールを開始します。

```
[root@ipa-server ~] ipa-ca-install --external-ca
```

このステップの後に、証明書署名要求 (CSR) が保存されていることを示す情報が表示されます。CSR を外部 CA に送信し、発行した証明書を IdM サーバーにコピーします。

2. 外部 CA ファイルへの証明書および完全パスを **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. すべてのサーバーとクライアントで **ipa-certupdate** ユーティリティーを実行し、LDAP から新しい証明書に関する情報で更新します。すべてのサーバーとクライアントで **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は、常に **ipa-certupdate** を実行します。これがない場合、証明書は他のマシンに配布されません。

CA インストールは、LDAP および Web サーバーの既存のサービス証明書を、新規インストールした CA により発行された証明書に置き換えません。証明書を置き換える方法は、「Web サーバーの証明書および 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 サーバーの証明書および 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るには、以下を実行します。

1. 新しい証明書を要求します。これは、以下を使用して実行できます。

   - 統合 CA の場合: 詳細は 「ユーザー、ホスト、またはサービスの新規証明書の要求」 を参照してください。
   - 外部 CA: 秘密鍵および証明書署名要求 (CSR) を生成します。例: OpenSSL の使用:

     ```
     $ 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 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
   ```

第27章 IDM の KERBEROS PKINIT 認証

Kerberos(PKINIT) の初期認証の公開鍵暗号化は、Kerberos の事前認証メカニズムです。Red Hat Enterprise Linux 7.4 以降、Identity Management(IdM) サーバーには、Kerberos PKINIT 認証のメカニズムが含まれています。以下のセクションでは、IdM の PKINIT 実装の概要と、IdM で PKINIT の実装を設定する方法を説明します。

27.1. IDM バージョンが異なるデフォルトの PKINIT ステータス

IdM サーバーのデフォルトの PKINIT 設定は、Red Hat Enterprise Linux(RHEL) および認証局 (CA) 設定の IdM のバージョンによって異なります。表27.1「IdM バージョンのデフォルトの PKINIT 設定」 を参照してください。

表27.1 IdM バージョンのデフォルトの PKINIT 設定

RHEL のバージョン	CA 設定	PKINIT の設定
7.3 以前	CA なし	ローカル PKINIT: IdM はサーバーの内部用途でのみ PKINIT を使用します。
7.3 以前	統合 CA の場合	IdM は、統合 IdM CA が署名した証明書を使用して PKINIT の設定を試みます。 試行に失敗すると、IdM はローカルの PKINIT のみを設定します。
7.4 以降	CA なし IdM に提供されている外部 PKINIT 証明書がない	ローカル PKINIT: IdM はサーバーの内部用途でのみ PKINIT を使用します。
7.4 以降	CA なし IdM に提供される外部 PKINIT 証明書	IdM は、外部の Kerberos 鍵配布センター (KDC) 証明書と CA 証明書を使用して PKINIT を設定します。
7.4 以降	統合 CA の場合	IdM は、IdM CA が署名した証明書を使用して PKINIT を設定します。

ドメインレベル 0 では、PKINIT は無効になります。デフォルトの動作は、ローカルの PKINIT です。IdM は、サーバーでの内部目的でのみ PKINIT を使用します。*7章 ドメインレベルの表示と引き上げ*も参照してください。

27.2. 現在の PKINIT 設定の表示

IdM には、ドメインの PKINIT 設定をクエリーするのに使用できるコマンドが複数用意されています。

ドメインの PKINIT のステータスを確認するには、**ipa pkinit-status** コマンドを使用します。

```
$ ipa pkinit-status
  Server name: server1.example.com
  PKINIT status: enabled
```

```
[...output truncated...]
Server name: server2.example.com
PKINIT status: disabled
[...output truncated...]
```

ログインしているサーバーで PKINIT のステータスを確認するには、**ipa-pkinit-manage status** コマンドを使用します。

```
# ipa-pkinit-manage status
PKINIT is enabled
The ipa-pkinit-manage command was successful
```

このコマンドは、**enabled** または **disabled** として PKINIT 設定の状態を表示します。

- **Enabled**: PKINIT は、統合 IdM CA または外部 PKINIT 証明書により署名された証明書を使用して設定されます。「IdM バージョンが異なるデフォルトの PKINIT ステータス」も参照してください。
- **Disabled**: IdM は、IdM サーバーでの内部目的でのみ PKINIT を使用します。

IdM クライアントの PKINIT に対応するアクティブな Kerberos 鍵配布センター (KDC) がある IdM サーバーを表示するには、任意のサーバーで **ipa config-show** コマンドを使用します。

```
$ ipa config-show
  Maximum username length: 32
  Home directory base: /home
  Default shell: /bin/sh
  Default users group: ipausers
  [...output truncated...]
  IPA masters capable of PKINIT: server1.example.com
  [...output truncated...]
```

関連情報

- PKINIT のステータスを報告するコマンドラインツールの詳細は、**ipa help pkinit** コマンドを使用します。

27.3. IDM での PKINIT の設定

IdM サーバーが PKINIT を無効にした状態で動作している場合は、以下の手順に従って有効にします。たとえば、**--no-pkinit** オプションを **ipa-server-install** ユーティリティーまたは **ipa-replica-install** ユーティリティーで渡した場合には、PKINIT が無効になります。

前提条件

- 認証局 (CA) がインストールされているすべての IdM サーバーが、同じドメインレベルで稼働していることを確認します。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。

手順

1. サーバーで PKINIT が有効になっているかどうかを確認します。

```
# kinit admin
Password for admin@IPA.TEST:
# ipa pkinit-status --server=server.idm.example.com
```

```
----------------
1 server matched
----------------
Server name: server.idm.example.com
PKINIT status: enabled
----------------------------
Number of entries returned 1
----------------------------
```

PKINIT が無効になっている場合は、以下の出力が表示されます。

```
# ipa pkinit-status --server server.idm.example.com
----------------
0 servers matched
----------------
----------------------------
Number of entries returned 0
----------------------------
```

--server <server_fqdn> パラメーターを省略した場合は、コマンドを使用して PKINIT が有効になっているすべてのサーバー を見つけることもできます。

2. CA を使用せずに IdM を使用している場合は、次のコマンドを実行します。

   a. IdM サーバーで、Kerberos キー配布センター(KDC)証明書に署名した CA 証明書をインストールします。

   ```
   # ipa-cacert-manage install -t CT,C,C ca.pem
   ```

   b. すべての IPA ホストを更新するには、すべてのレプリカおよびクライアントで **ipa-certupdate** コマンドを繰り返します。

   ```
   # ipa-certupdate
   ```

   c. **ipa-cacert-manage list** コマンドを使用して、CA 証明書がすでに追加されているかどうかを確認します。以下に例を示します。

   ```
   # ipa-cacert-manage list
   CN=CA,O=Example Organization
   The ipa-cacert-manage command was successful
   ```

   d. **ipa-server-certinstall** ユーティリティーを使用して、外部 KDC 証明書をインストールします。KDC 証明書は以下の条件を満たしている必要があります。

   - これは、共通名 **CN=*fully_qualified_domain_name*、*certificate_subject_base*** で発行されます。
   - これには、Kerberos プリンシパル **krbtgt/*REALM_NAME*@*REALM_NAME*** が含まれます。
   - KDC 認証のオブジェクト識別子 (OID) が含まれます:**1.3.6.1.5.2.3.5**。

   ```
   # ipa-server-certinstall --kdc kdc.pem kdc.key
   # systemctl restart krb5kdc.service
   ```

e. PKINIT のステータスを参照してください。

```
# ipa pkinit-status
  Server name: server1.example.com
  PKINIT status: enabled
  [...output truncated...]
  Server name: server2.example.com
  PKINIT status: disabled
  [...output truncated...]
```

3. CA 証明書で IdM を使用している場合は、以下のように PKINIT を有効にします。

```
# ipa-pkinit-manage enable
  Configuring Kerberos KDC (krb5kdc)
  [1/1]: installing X509 Certificate for PKINIT
  Done configuring Kerberos KDC (krb5kdc).
  The ipa-pkinit-manage command was successful
```

IdM CA を使用している場合、コマンドは CA から PKINIT KDC 証明書を要求します。

関連情報

- 詳細は、**ipa-server-certinstall (1)** man ページを参照してください。

27.4. 関連情報

- Kerberos PKINIT の詳細は、MIT Kerberos ドキュメントの PKINIT 設定
- IdM での PKINIT スマートカード認証の設定に関するドキュメントは、「Identity Management での PKINIT スマートカード認証」を参照してください。

パート VI. 管理: ポリシーの管理

ここでは、パスワードポリシーの定義、**Kerberos**ドメインの管理、**sudo** ユーティリティーの使用、**ホストベースのアクセス制御** の設定、および **SELinux** ユーザーマップの定義方法を説明します。

第28章 パスワードポリシーの定義

本章では、Identity Management(IdM) のパスワードポリシーとその管理方法を説明します。

28.1. パスワードポリシーとは、なぜ有用なのか

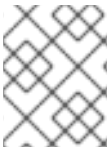
パスワードポリシー は、パスワードが満たさなければならない一連のルールです。

たとえば、パスワードポリシーでは、パスワードの最小長さと最大有効期間を定義できます。このようなポリシーの対象となる全ユーザーは、十分に長いパスワードを設定して、十分な頻度で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を検出および誤用するリスクが軽減されます。

28.2. IDM でのパスワードポリシーの仕組み

すべてのユーザーには、Identity Management (IdM) Kerberos ドメインへの認証に使用するパスワードが必要です。IdM のパスワードポリシーは、これらのユーザーパスワードが満たする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) により強制的に適用されます。

28.2.1. サポートされるパスワードポリシー属性

表28.1「パスワードポリシーの属性」 に、IdM のパスワードポリシーで定義できる属性をリストします。

表28.1 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、パスワードの有効日数の上限です。	Max lifetime = 90 ユーザーパスワードは 90 日間のみ有効です。有効期限が経過すると、IdM は変更を求めるプロンプトを表示します。
Min lifetime	パスワード変更操作間で渡す必要のある最小時間 (時間)。	Min lifetime = 1 ユーザーがパスワードの変更後に、次に変更するまでに最低でも 1 時間待機する必要があります。
History size	保存される以前のパスワードの数。ユーザーは、パスワード履歴からパスワードを再利用できません。	History size = 0 ユーザーは、以前のいずれのパスワードでも再利用できます。

属性	説明	例
Character classes	パスワードで使用する必要のある文字クラスの数。文字クラスは次のとおりです。 • 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 1 つの文字を複数回連続で使用すると、文字クラスが 1 つ減少します。以下に例を示します。 • * **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** には、大文字、小文字、数字、および -1 ペナルティの 2 つの文字クラスがあります。**1** を 繰り返し使用できます。	Character classes = 0 必要なクラスのデフォルト数は 0 です。番号を設定するには、**--minclasses** オプションを指定して **ipa pwpolicy-mod** コマンドを実行します。このコマンドは、必要な文字クラスの数を 1 に設定します。 `$ ipa pwpolicy-mod --minclasses=1` この表の下にある 重要 注記も参照してください。
Min length	パスワードの最小長。	Min length = 8 8 文字未満のパスワードは使用できません。
Max failures	IdM がユーザーアカウントをロックするまでのログイン試行の最大失敗数。「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。	Max failures = 6 ユーザーが間違ったパスワードを 7 回入力すると、IdM はユーザーアカウントをロックします。
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 (秒単位)。	Failure reset interval = 60 **Max failures** で定義されたログイン試行回数が 1 分以上経過すると、ユーザーはユーザーアカウントがロックされる心配なく再ログインを試みることができます。
Lockout duration	**Max failures** で定義された回数のログイン試行に失敗した後にユーザーアカウントがロックされる時間 (秒単位)。「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。	Lockout duration = 600 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。

重要

国際文字や記号を使用できないハードウェアセットが各種ある場合には、文字クラス要件に英語と共通記号を使用してください。パスワードの文字クラスポリシーの詳細は、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。

28.2.2. グローバルパスワードポリシー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*。グローバルポリシーとは別に、追加の *グループパスワードポリシー* を作成できます。

グローバルパスワードポリシー

初期 IdM サーバーをインストールすると、デフォルト設定でグローバルパスワードポリシーが自動的に作成されます。

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なしですべての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すべてのメンバーに適用されます。

どのユーザーに対しても、一度に有効にできるパスワードポリシーは 1 つだけです。ユーザーに複数のパスワードポリシーが割り当てられている場合は、そのうちの 1 つが優先度に基づいて優先されます。「パスワードポリシーの優先順位」を参照してください。

28.2.3. パスワードポリシーの優先順位

すべてのグループパスワードポリシーに *優先順位* が設定されています。値が小さいほど、ポリシーの優先度が高くなります。サポートされる最も低い優先度の値は **0** です。

- 複数のパスワードポリシーがユーザーに適用される場合は、優先度の値が最も小さいポリシーが優先されます。他のポリシーで定義されたすべてのルールは無視されます。
- 優先度の値が最も小さいパスワードポリシーは、ポリシーに定義されていない属性であっても、すべての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は設定されません。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は、グループポリシーより優先されることはありません。

表28.2「優先度に基づくパスワードポリシー属性の適用例」 は、ポリシーが定義されている 2 つのグループに属するユーザーを例に、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28.2 優先度に基づく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)
	↓	↓
ユーザー (グループ A およびグループ B のメンバー)	60	10

注記

ipa pwpolicy-show --user=*user_name* コマンドは、現在特定のユーザーに有効なポリシーを表示します。

28.3. 新しいパスワードポリシーの追加

新しいパスワードポリシーを追加する場合は、以下を指定する必要があります。

- ポリシーを適用するユーザーグループ (「グローバルパスワードポリシー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順位」を参照)

以下のツールを使用して、新規パスワードポリシーを追加します。

- Web UI(「Web UI: 新しいパスワードポリシーの追加」を参照してください)
- コマンドラインは (「コマンドライン: 新しいパスワードポリシーの追加」を参照してください)

Web UI: 新しい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. ユーザーグループおよび優先度を定義します。
4. **Add** をクリックして確定します。

新しいパスワードポリシーの属性を設定するには、「パスワードポリシー属性の変更」 を参照してください。

コマンドライン: 新しいパスワードポリシーの追加

1. **ipa pwpolicy-add** コマンドを使用します。ユーザーグループおよび優先度を指定します。

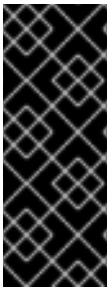
```
$ ipa pwpolicy-add
Group: group_name
Priority: priority_level
```

2. **オプション: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します。

```
$ ipa pwpolicy-find
```

新しいパスワードポリシーの属性を設定するには、「パスワードポリシー属性の変更」 を参照してください。

28.4. パスワードポリシー属性の変更



重要

パスワードポリシーを変更すると、新しいルールは新しいパスワードにのみ適用されます。変更は、既存のパスワードに遡っては適用されません。

変更を有効にするには、ユーザーが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

セキュアなユーザーパスワードに関する推奨事項は、『Security Guide』のPassword Securityを参照してください。

以下のツールを使用して、パスワードポリシーを変更します。

- Web UI は、「Web UI: パスワードポリシーの変更」を参照してください。
- コマンドラインは、「コマンドライン: パスワードポリシーの変更」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、属性の制限がないことを意味します。たとえば、最大有効期間を **0** に設定すると、ユーザーパスワードは期限切れになりません。

Web UI: パスワードポリシーの変更

1. **Policy → Password Policies** を選択します。
2. 変更するポリシーをクリックします。
3. 必要な属性を更新します。利用可能な属性の詳細は、「サポートされるパスワードポリシー属性」 を参照してください。
4. **Save** をクリックして、変更を確定します。

コマンドライン: パスワードポリシーの変更

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、パスワードの最小の長さを **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。以下に例を示します。

```
$ ipa pwpolicy-mod group_name --minlength=10
```

2. **オプション:ipa pwpolicy-show** コマンドを使用して、新しいポリシー設定を表示します。
   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
```

28.5. パスワード有効期限の変更および即時の有効化

ipa user-mod ユーティリティーまたは **ldapmodify** ユーティリティーを使用して、ユーザーパスワードの有効期限を変更できます。

ipa user-mod ユーティリティーを使用したユーザーパスワードの有効期限の変更

- 有効期限の変更をすぐに実行するには、**--password-expiration** オプションを指定して **ipa user-mod** コマンドを使用します。たとえば、UTC タイムゾーンで有効期限を **2016-02-03 20:37:34 に** 設定するには、以下を実行します。

  ```
  # ipa user-mod user_name --password-expiration='2016-02-03 20:37:34Z'
  ```

  このコマンドは、一般化された時間形式を使用し、有効期限を **20160203203734Z** に設定することも可能であることに注意してください。

ldapmodify ユーティリティーを使用したユーザーパスワードの有効期限の変更
有効期限の即時に変更を適用するには、LDAP の ***krbPasswordExpiration*** 属性値をリセットします。

単一ユーザーの有効期限を変更するには、以下を実行します。

1. 次のコマンドを使用して、ユーザーエントリーの新しい値を ***krbPasswordExpiration*** に設定します。

   ```
   # 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
   ```

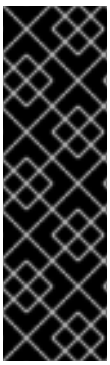
   krbPasswordExpiration 形式は、一般的な時間形式 YYYMMDDHHMMSS.0Z に従います。

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一度に編集するには、**ldapmodify** を **-f** オプションを指定して LDIF ファイルを参照します。

第29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) コンポーネントを管理する方法を説明します。



重要

Identity Management の Kerberos ポリシーを管理する場合は、**kadmin** ユーティリティーまたは **kadmin.local** ユーティリティーを使用しないでください。本ガイドで説明されているように、ネイティブの Identity Management コマンドラインツールを使用します。

上記の Kerberos ツールを使用して Identity Management ポリシーを管理しようとすると、一部の操作が Directory Server インスタンスに保存されている Identity Management 設定に影響を及ぼしません。

29.1. 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、チケットの有効期間および更新に対する制限を設定します。以下の手順を使用して、Identity Management サーバーで実行している Kerberos Key Distribution Center (KDC) の Kerberos チケットポリシーを設定できます。

29.1.1. Kerberos チケットの有効期間の判断

Identity Management サーバーが、*user_name* の代わりに Identity Management クライアントが Kerberos チケットを要求した後に付与されるチケットの有効期間を決定する際には、さまざまなパラメーターが考慮に入れられます。まず、クライアント側の評価が行われ、**kinit** コマンドと **/etc/krb5.conf** ファイルの **ticket_lifetime** 設定をもとに要求する値を計算します。その後、値はサーバー側の評価が行われる Identity Management サーバーに送信されます。要求されたライフタイムがグローバル設定が許可するものよりも短い場合、要求されたライフタイムが付与されます。それ以外の場合は、付与されるライフタイムは、グローバル設定が許可する値です。

User_name の代わりにクライアントが要求したライフタイムは、以下のようにして決定されます。

クライアント側

- **-l** オプションを使用して**kinit** コマンド自体で *user_name* に対する値を明示的に記述する場合は (以下の例を参照)、

  ```
  $ kinit user_name -l 90000
  ```

  その値 (この場合は 90000 秒) が*user_name* の代わりにクライアントによって要求されます。

- 一方、ライフタイム値が **kinit *user_name*** コマンドの引数として渡されない場合、クライアントの **/etc/krb5.conf** ファイルの **ticket_lifetime** 設定の値が*user_name* の代わりにクライアントによって使用されます。**/etc/krb5.conf** ファイルに値を指定しないと、初期チケット要求のデフォルトの IdM 値が使用されます (これは 1 日)。

サーバー側
サーバー側の 2 段階評価が行われます。

1. クライアントが要求する値は、*user_name*固有の Kerberos チケットポリシー (ポリシーが存在する場合) の**--maxlife** 設定と比較され、その 2 つのより小さい値が選択されます。*user_name* 固有の Kerberos チケットポリシーが存在しない場合、クライアントが送信した値は、Global Kerberos チケットポリシーの **--maxlife** 設定と比較され、その 2 つのより小さい値が選択され

ます。グローバル Kerberos チケットポリシーおよびユーザー固有の Kerberos チケットポリシーの詳細は、「グローバル Kerberos チケットポリシーおよびユーザー固有の Kerberos チケットポリシー」を参照してください。

2. 前の手順で選択した値は、他の 2 つの値と比較されます。

   - **/var/kerberos/krb5kdc/kdc.conf** ファイルの **max_life** 設定の値
   - 識別名 (DN) **krbPrincipalName=krbtgt/*REALM_NAME*@*REALM_NAME*,cn=*REALM_NAME*,cn=kerberos,*domain_name*** を使用した LDAP エントリーの **krbMaxTicketLife** 属性に設定された値

   これら 3 つの中の一番小さい値が、最終的に*User_name* に付与される Kerberos チケットの有効期間に選択されます。

29.1.2. グローバル Kerberos チケットポリシーおよびユーザー固有の Kerberos チケットポリシー

グローバル Kerberos チケットポリシーを再定義し、個々のユーザー固有の追加のポリシーを定義できます。

グローバル Kerberos チケットポリシー

グローバルポリシーは、Identity Management Kerberos レルム内で発行されるすべての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のみ適用されます。たとえば、ユーザー固有の Kerberos チケットポリシーでは、**admin** ユーザーにより長いチケットの最大有効期間を定義できます。

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。

29.1.3. グローバル Kerberos チケットポリシーの設定

グローバル Kerberos チケットポリシーを設定するには、以下のツールを使用できます。

- Identity Management の Web UI: 「Web UI: グローバル Kerberos チケットポリシーの設定」 を参照してください。
- コマンドラインは、「コマンドライン: グローバル Kerberos チケットポリシーの設定」を参照してください。

表29.1 サポートされる Kerberos チケットポリシー属性

属性	説明	例
Max renew	有効期限が切れた後に、ユーザーが Kerberos チケットを更新できる期間 (秒単位)。更新期間の後は、ユーザーは**kinit** ユーティリティーを使用して新しい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 チケットの有効期限が切れた後、ユーザーは次の 7 日 (604,800 秒) 以内に更新することができます。

属性	説明	例
Max life	Kerberos チケットの有効期間 (秒単位)。Kerberos チケットがアクティブな期間。	Max life = 86400 チケットは発行後 24 時間 (86,400 秒) で有効期限が切れます。

Web UI: グローバル Kerberos チケットポリシーの設定

1. **Policy → Kerberos Ticket Policy** を選択します。
2. 必要な値を定義します。
   a. **Max renew** フィールドに、Kerberos チケットの最大更新期間を入力します。
   b. **Max life** フィールドに Kerberos チケットの最大有効期間を入力します。

   図29.1 グローバル Kerberos チケットポリシーの設定

3. **Save** をクリックします。

コマンドライン: グローバル Kerberos チケットポリシーの設定

グローバル Kerberos チケットポリシーを変更するには、以下を実行します。

- **ipa krbtpolicy-mod** コマンドを使用して、以下のオプションの少なくとも 1 つを渡します。
  - **--maxrenew** - Kerberos チケットの最大更新間隔を定義します。
  - **--maxlife** - 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
```

グローバル Kerberos チケットポリシーを元のデフォルト値にリセットするには、以下を実行します。

1. **ipa krbtpolicy-reset** コマンドを使用します。
2. **オプション: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、コマンドに **--help** オプションを渡します。

29.1.4. 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するには、以下を実行します。

1. **ipa krbtpolicy-mod *user_name*** コマンドを使用して、以下のオプションの少なくとも 1 つを渡します。
   - **--maxrenew** - Kerberos チケットの最大更新間隔を定義します。
   - **--maxlife** - Kerberos チケットの最大有効期間を定義します。

   1 つの属性のみを定義すると、Identity Management は他の属性にグローバル Kerberos チケットポリシーの値を適用します。

   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:ipa krbtpolicy-show *user_name*** コマンドを使用して、指定したユーザーの現在の値を表示します。

新しいポリシーは、**kinit** ユーティリティーを使用する場合など、ユーザーが要求する次の Kerberos チケットで直ちに有効になります。

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、ユーザーに特別に定義した値を消去し、その後、Identity Management はグローバルポリシーの値を適用します。

ipa krbtpolicy-mod および **ipa krbtpolicy-reset** の詳細は、コマンドに **--help** オプションを渡します。

29.2. KERBEROS プリンシパルのキー再生成

Kerberos プリンシパルの *キー再生成* により、より大きなキーバージョン番号 (KVNO) を持つ新しいキータブエントリーが、プリンシパルのキータブに追加されます。元のエントリーはキータブに残りますが、チケットを発行するのに使用されません。

1. 要求された期間に発行された全キータブを検索します。たとえば、次のコマンドは **ldapsearch** ユーティリティーを使用して、Greenwich Mean Time(GMT) で 2016 年 1 月 1 日の 00:00 AM から 2016 年 12 月 31 日の 11:59 PM の間に作成されたホストとサービスプリンシパルをすべて表示します。

   ```
   # 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**) 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** サブツリーに保存されます。
- ***krblastpwdchange*** パラメーターは、最後の変更日で検索結果をフィルタリングします。このパラメーターでは、日付に YYYYMMDD 形式と、時刻に HHMMSS 形式を使用できます (GMT)。
- ***dn*** 属性および ***krbprincipalname*** 属性を指定すると、検索結果はエントリー名とプリンシパルに制限されます。

2. プリンシパルのキー再生成を必要とするすべてのサービスおよびホストで、**ipa-getkeytab** ユーティリティーを使用して新しいキータブ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

     --keytab (**-k**): 元のキータブ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。

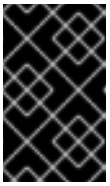
   以下に例を示します。

   - **/etc/krb5.keytab** のデフォルトロケーションにあるキータブでホストプリンシパルのキーを再生成するには、以下のコマンドを実行します。

     ```
     # ipa-getkeytab -p host/client.example.com@EXAMPLE.COM -s server.example.com -k
     /etc/krb5.keytab
     ```

   - **/etc/httpd/conf/ipa.keytab** のデフォルトロケーションにある Apache サービスのキータブのキーを再生成するには、以下のコマンドを実行します。

     ```
     # 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 などの一部のサービスは、限定された暗号化タイプのみに対応します。適切な引数を **ipa-getkeytab** コマンドに渡してキータブを適切に設定します。

3. **オプション:**プリンシパルのキーが正常に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 チケットをリスト表示します。たとえば、**/etc/krb5.keytab** のすべてのキータブエントリーをリスト表示するには、次のコマンドを実行します。

```
# 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)
```

この出力は、**client.example.com** のキータブエントリーのキーが、より高い KVNO で再生成されたことを示しています。元のキータブは、以前の KVNO で引き続きデータベースに存在します。

古いキータブに対して発行されたチケットは引き続き機能しますが、KVNO のキーを使用して新しいチケットが発行されます。これにより、システム操作の中断が回避されます。

29.3. キータブの保護

サーバーにアクセスできる他のユーザーから Kerberos キータブを保護するには、キータブへのアクセスをキータブの所有者だけに制限します。キータブを取得直後に保護することが推奨されます。

たとえば、**/etc/httpd/conf/ipa.keytab** の Apache キータブを保護するには、以下の手順を実施します。

1. ファイルの所有者を **apache** に設定します。

```
# 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。これにより、所有者に読み取り、書き込み、実行パーミッションが付与されます。

```
# chmod 0600 /etc/httpd/conf/ipa.keytab
```

29.4. キータブの削除

ホストを登録解除して再登録する場合や Kerberos 接続エラーが発生した場合などに、キータブを削除して新しいキータブを作成する必要があります。

ホストの全キータブを削除するには、**ipa-rmkeytab** ユーティリティーを使用し、以下のオプションを渡します。

- **--realm** (**-r**) で Kerberos レルムを指定します。
- キータブファイルへのパスを指定する **--keytab** (**-k**)

```
# ipa-rmkeytab --realm EXAMPLE.COM --keytab /etc/krb5.keytab
```

特定のサービスのキータブを削除するには、**--principal** (**-p**) オプションを指定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
```

29.5. 関連情報

- Identity Management サーバーがホストする Kerberos KDC の概要は、「IdM サーバーがホストするサービス」を参照してください。
- Kerberos に関する Red Hat ドキュメントは、『System-Level Authentication Guide』のUsing Kerberosを参照してください。
- Kerberos の概念に関する詳細情報は、MIT Kerberos ドキュメント を参照してください。

第30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 ドメインのすべてのシステムは、**sudo** クライアントとして設定できます。

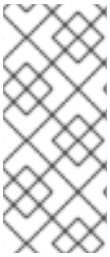
30.1. IDENTITY MANAGEMENT の SUDO ユーティリティー

sudo ユーティリティーを使用すると、指定したユーザーへの管理者アクセスが可能になります。信頼されるユーザーが、管理コマンドの前に **sudo** を付けると、この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 権限で実行されているかのように実行されます。**sudo** の詳細は、システム管理者ガイドを参照してください。

30.1.1. sudoの Identity Management LDAP スキーマ

IdM には、**sudo** エントリーに特化した LDAP スキーマがあります。スキーマは以下をサポートします。

- ホストグループと netgroups。**sudo** は netgroups のみに対応していることに注意してください。
- 複数のコマンドを含む **sudo** コマンドグループ。



注記

sudo はホストグループやコマンドグループに対応していないため、**sudo** ルールの作成時に 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 を作成します。これにより、IdM 管理者はホストグループを参照する **sudo** ルールを作成でき、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、この **sudo** 情報は LDAP 上で匿名で利用できません。そのため、IdM は **uid=sudo,cn=sysaccounts,cn=etc,$SUFFIX** でデフォルトの **sudo** ユーザーを定義します。このユーザーは、**/etc/sudo-ldap.conf** の LDAP **sudo** 設定ファイルで変更できます。

30.1.2. NIS ドメイン名の要件

netgroups および **sudo** が適切に機能するには、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 形式の netgroups と netgroups の NIS ドメイン名が必要です。ただし、IdM では、実際に NIS ドメインが存在している必要はありません。NIS サーバーをインストールする必要もあります。

注記

ipa-client-install ユーティリティーは、NIS ドメイン名を、デフォルトで IdM ドメイン名に自動的に設定します。

30.2. IDENTITY MANAGEMENT の SUDO ルール

sudo ルールを使用すると、**誰が 何を、どこで**、および **誰として** 実行できるかを定義できます。

- ユーザーが **sudo** を使用することができるユーザーは **誰** ですか。

- **sudo** で使用できるコマンドは **何** ですか。
- ユーザーが **sudo** を使用できるターゲットホストは**どこか**。
- ユーザーは、システムまたは他のユーザー ID の**誰として** タスクを実行すると想定されるか。

30.2.1. sudo ルールの外部ユーザーおよびホスト

IdM は **sudo** ルールの外部エンティティーを受け入れます。外部エンティティーとは、IdM ドメインの一部ではないユーザーまたはホストなど、IdM ドメイン外に保存されるエンティティーです。

たとえば、**sudo** ルールを使用して、IdM の IT グループのメンバーに root アクセスを付与できます。ここで、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できます。

30.2.2. sudo ルールのユーザーグループサポート

sudo を使用して、IdM のユーザーグループ全体へのアクセス権限を付与できます。IdM は、Unix グループと非 POSIX グループの両方に対応します。非 POSIX グループを作成すると、非 POSIX グループのユーザーはグループから非 POSIX パーミッションを継承するため、アクセスの問題が発生する可能性があります。

30.2.3. sudoers オプションのサポート

IdM は **sudoers** オプションに対応します。利用可能な **sudoers** オプションの完全なリストは、sudoers(5) の man ページを参照してください。

IdM では、**sudoers** オプションで空白や改行を使用できないことに注意してください。したがって、複数のオプションをコンマ区切りリストで指定する代わりに、個別に追加します。たとえば、コマンドラインから 2 つの **sudoers** オプションを追加するには、以下を実行します。

```
$ 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、1 行に長いオプションを指定するようにしてください。たとえば、コマンドラインでは、以下のようになります。

```
$ 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"
```

30.3. SUDO ポリシーを検索する場所の設定

sudo 設定用の集中 IdM データベースにより、IdM で定義される **sudo** ポリシーは、すべてのドメインホストでグローバルに利用できます。Red Hat Enterprise Linux 7.1 以降のシステムでは、SSSD を **sudo** のデータプロバイダーとして設定することで、**ipa-server-install** および **ipa-client-install** ユーティリティーは、IdM が定義したポリシーを使用するようにシステムを自動的に設定します。

sudo ポリシーを検索する場所は、**/etc/nsswitch.conf** ファイルの **sudoers** 行で定義されます。Red Hat Enterprise Linux 7.1 以降を実行している IdM システムでは、**nsswitch.conf** のデフォルトの **sudoers** 設定は以下のようになります。

```
sudoers: files sss
```

files オプションは、システムが、**/etc/sudoers** ローカル SSSD 設定ファイルで定義された **sudo** 設定を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。

30.3.1. 以前のバージョンの IdM で IdM sudo ポリシーを使用するためのホスト設定

7.1 よりも前の Red Hat Enterprise Linux のバージョンを実行している IdM システムに IdM が定義した **sudo** ポリシーを実装するには、ローカルマシンを手動で設定します。これは、SSSD または LDAP を使用して実行できます。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。

30.3.1.1. SSSD を使用した sudo ポリシーのホストへの適用

sudo ルール用に SSSD を使用する必要のある各システムで、以下の手順を実行します。

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
   ```

   この **files** オプションをそのままにすると、**sudo** で、IdM 設定について SSSD を確認する前にローカル設定を確認することができます。

2. **sudo** を、ローカルの SSSD クライアントが管理するサービスリストに追加します。

   ```
   # vim /etc/sssd/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** サービスをまだ有効にしていない場合は、このサービスを有効にして、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。以下に例を示します。

```
# echo "NISDOMAIN=example.com" >> /etc/sysconfig/network
```

これにより、NIS ドメインを持つ **/etc/sysconfig/network** および **/etc/yp.conf** ファイルが更新されます。

4. **rhel-domainname** サービスを再起動します。

```
# systemctl restart rhel-domainname.service
```

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 が操作に使用する LDAP 検索ベースは、**sssd_DOMAINNAME.log** ログに記録されます。

30.3.1.2. LDAP を使用した sudo ポリシーのホストへの適用

重要

SSSD を使用しないクライアントには 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 を使用して **sudo** ポリシーを適用する方法は、『Red Hat Enterprise Linux 6 Identity Management Guide』のApplying the sudo Policies to Hosts Using LDAP を参照してください。

LDAP ベースの設定は、主に、Red Hat Enterprise Linux 7 よりも古いバージョンの Red Hat Enterprise Linux をベースとするクライアントに使用されることが想定されます。したがって、Red Hat Enterprise Linux 6 のドキュメントのみで説明されています。

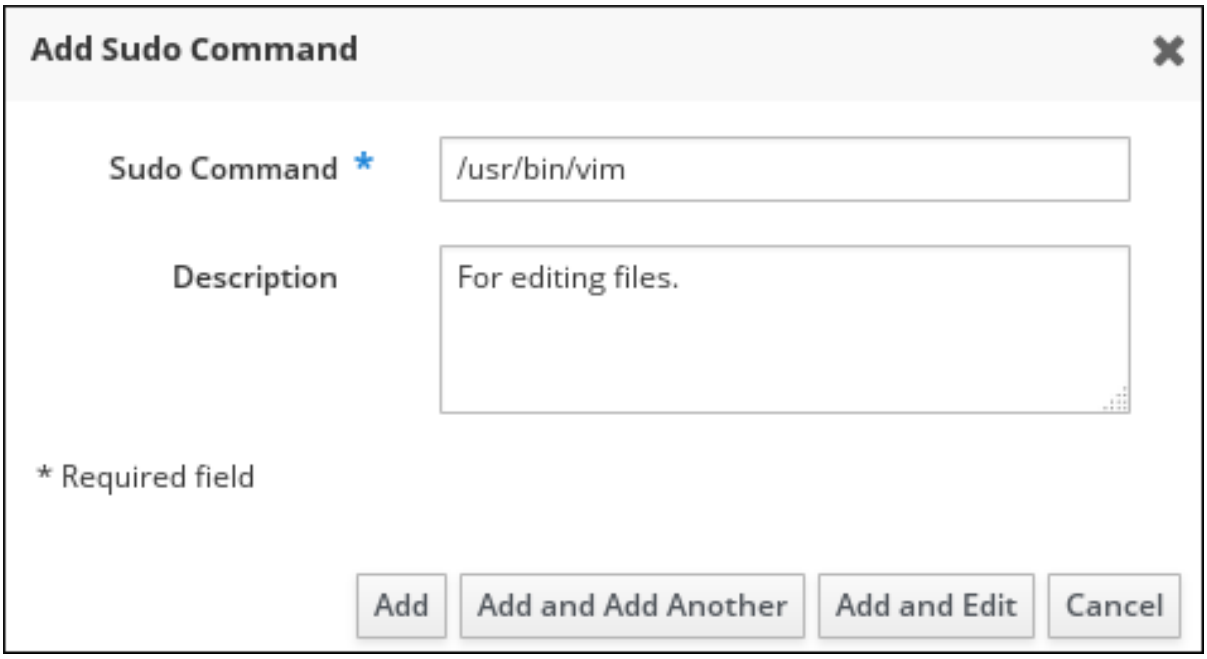
30.4. SUDO コマンド、コマンドグループ、およびルールの追加

30.4.1. 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 → Commands** をクリックします。
2. リストの上部にある **Add** をクリックします。
3. コマンドの情報を入力します。コマンド実行ファイルへの完全なシステムパスを入力します。

図30.1 新規 sudo コマンドの追加



4. **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エントリーの追加を開始するか、**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。

コマンドラインでの 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。コマンド実行ファイルへの完全なシステムパスを指定します。たとえば、**/usr/bin/less** コマンドと説明を追加するには、次のコマンドを実行します。

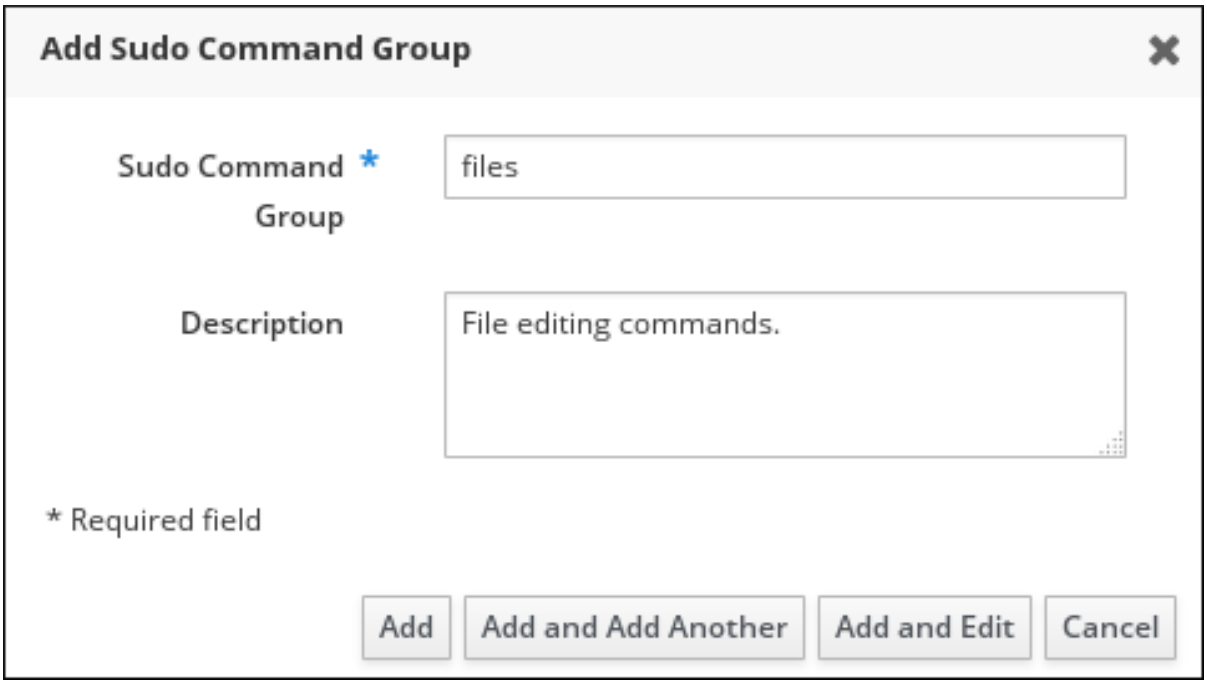
```
$ ipa sudocmd-add /usr/bin/less --desc="For reading log files"
----------------------------------
Added sudo command "/usr/bin/less"
----------------------------------
  sudo Command: /usr/bin/less
  Description: For reading log files
```

30.4.2. 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

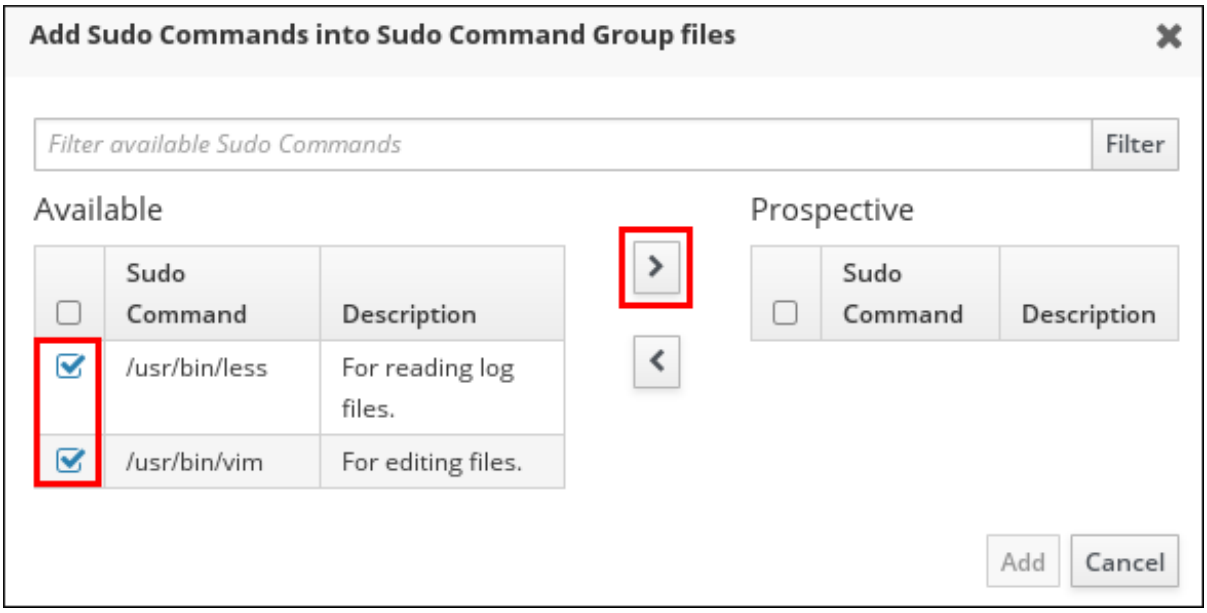
1. **Policy** タブで、**Sudo → Command Groups** をクリックします。
2. リストの上部にある **Add** をクリックします。
3. コマンドグループの情報を入力します。

図30.2 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、コマンドグループの編集を開始します。
5. **Sudo Commands** タブで **Add** をクリックして、グループに **sudo** コマンドを追加します。必要なコマンドを選択し、**>** ボタンを使用してそれらを **Prospective** 列に移動します。

図30.3 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。

コマンドラインでの sudo コマンドグループの追加

1. **ipa sudocmdgroup-add** コマンドを使用して、コマンドグループを作成します。たとえば、**files** コマンドグループを作成して説明を追加するには、次のコマンドを実行します。

```
$ 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** コマンドを追加します。「**sudo** コマンドの追加」で説明されているように、IdM にすでに追加されているコマンドのみを追加することができることに注意してください。

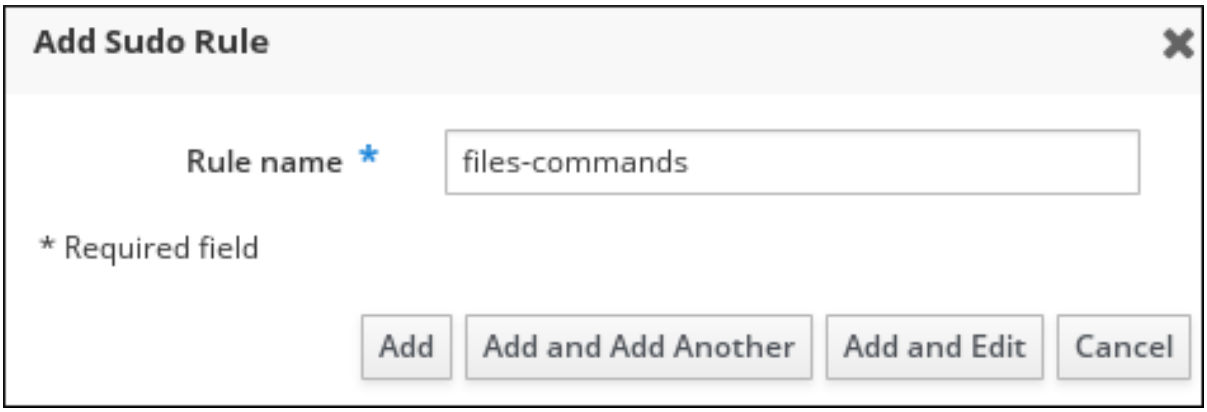
```
$ ipa sudocmdgroup-add-member files --sudocmds "/usr/bin/vim"
  sudo Command Group: files
  Description: File editing commands
  Member sudo commands: /usr/bin/vim
-------------------------
Number of members added 1
-------------------------
```

30.4.3. sudo ルールの追加

Web UI での sudo ルールの追加

1. **Policy** タブで Sudo → Rules をクリックします。
2. リストの上部にある **Add** をクリックします。
3. ルールの名前を入力します。

図30.4 新規 sudo ルールの命名



4. **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エントリーの追加を開始するか、**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。

新規の **sudo** ルールを編集する方法は、「**sudo** ルールの変更」 を参照してください。

コマンドラインでの 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、次のコマンドを実行します。

```
$ ipa sudorule-add files-commands
-------------------------------
Added Sudo Rule "files-commands"
-------------------------------
  Rule name: files-commands
  Enabled: TRUE
```

ipa sudorule-add の使用方法と、許可するオプションの詳細は、**--help** オプションを追加してコマンドを実行します。

新規の **sudo** ルールを編集する方法は、「**sudo** ルールの変更」 を参照してください。

新規の **sudo** ルールを追加し、コマンドラインから当該ルールを編集する詳細な例は、例30.1「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。

30.5. SUDO コマンドおよびコマンドグループの変更

Web UI での sudo コマンドおよびコマンドグループの変更

1. **Policy** タブで、Sudo → Sudo CommandsまたはSudo → Sudo Command Groupsをクリックします。
2. コマンドまたはコマンドグループの名前をクリックして、設定ページを表示します。
3. 必要に応じて設定を変更します。設定ページによっては、ページの上部に **Save** ボタンを使用できます。これらのページで、ボタンをクリックして変更を確認する必要があります。

コマンドラインでの sudo コマンドおよびコマンドグループの変更

コマンドまたはコマンドグループを変更するには、以下のコマンドを使用します。

- **ipa sudocmd-mod**
- **ipa sudocmdgroup-mod**

上記の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** コマンドの新しい説明を追加するには、次のコマンドを実行します。

```
$ 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
```

これらのコマンドとそれらが受け入れるオプションの詳細は、**-help**オプションを追加して実行してください。

30.6. SUDO ルールの変更

Web UI での sudo ルールの変更

1. **Policy** タブで Sudo → Rules をクリックします。
2. ルールの名前をクリックして、設定ページを表示します。
3. 必要に応じて設定を変更します。設定ページによっては、ページの上部に **Save** ボタンを使用できます。これらのページで、ボタンをクリックして変更を確認します。

sudo ルールの設定ページには、複数の設定エリアが含まれます。

General エリア

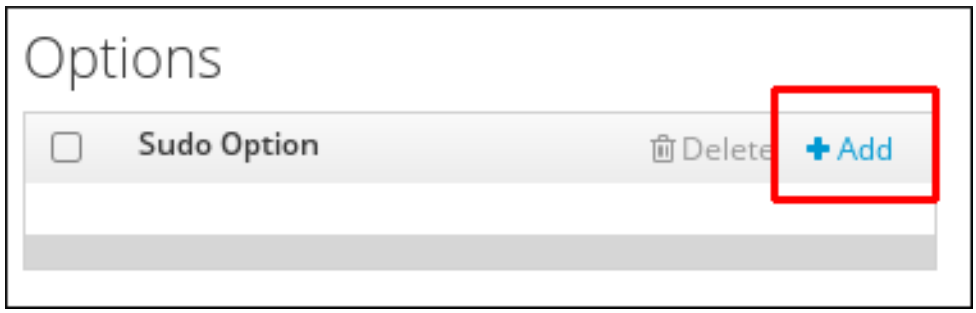
このエリアでは、ルールの説明と **sudo 順序** を変更できます。**sudo order** フィールドは整数を受け入れ、IdM がルールを評価する順番を定義します。最も高い **sudo 順序** 値を持つルールが最初に評価されます。

Options 領域

このエリアでは、**sudoers** オプションをルールに追加できます。

1. オプションリストの上にある **追加** をクリックします。

図30.5 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** がユーザーを認証するよう要求しない場合は、**!authenticate** オプションを追加します。

図30.6 sudoers オプションの入力

sudoers オプションの詳細は、sudoers(5) の 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

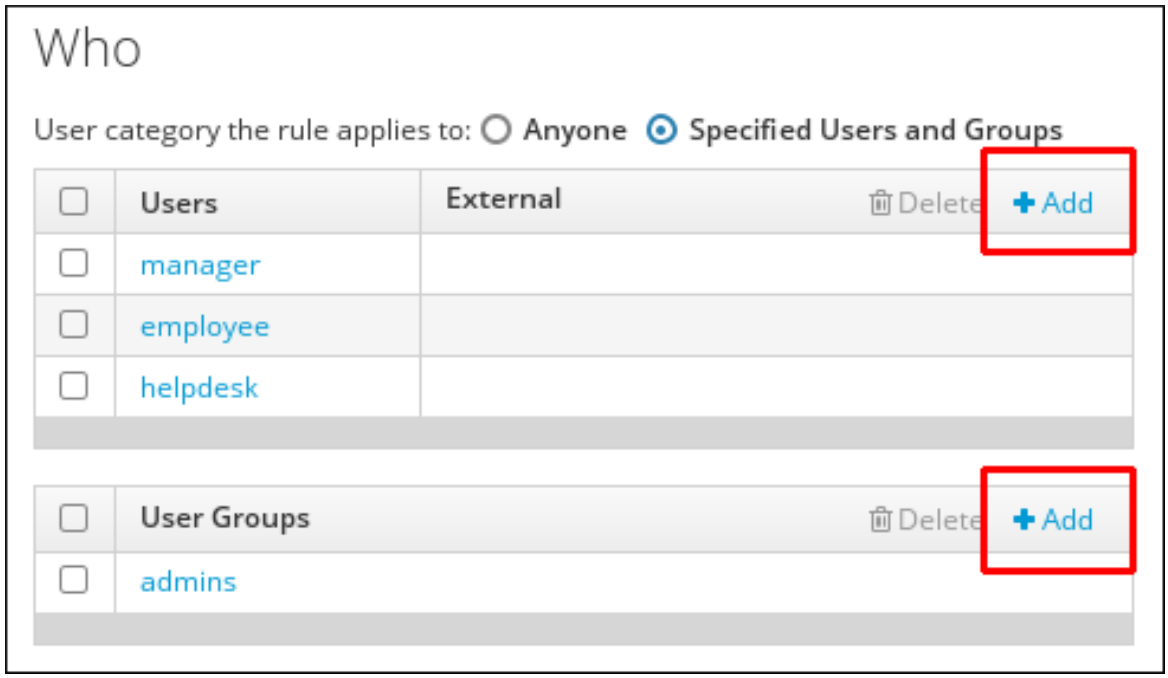
このエリアでは、**sudo** ルールが適用されるユーザーまたはユーザーグループを選択できます。これらのユーザーは、ルールで定義されているように **sudo** を使用する権利があります。

すべてのシステムユーザーがルールで定義されているように **sudo** を使用できるようにするには、**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、以下の手順に従います。

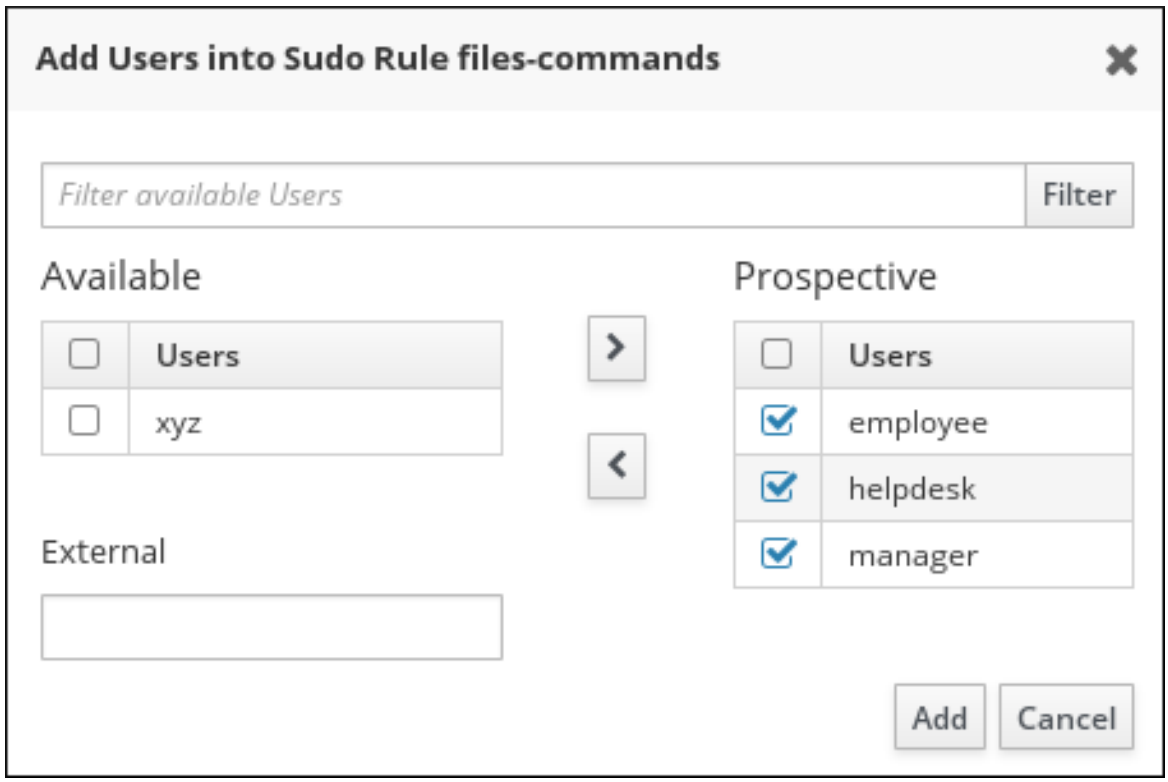
1. ユーザーまたはユーザーグループリストの上にある **追加** をクリックします。

図30.7 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には、**External** フィールドでユーザーを指定してから、> の矢印アイコンをクリックします。

図30.8 sudo ルールのユーザーの選択



3. **Add** をクリックします。

Access This Host エリア

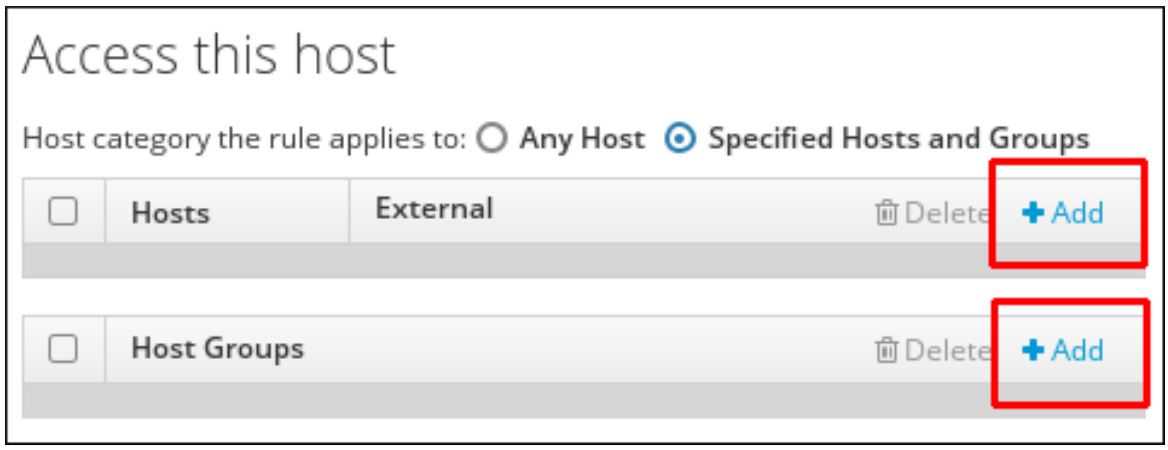
このエリアでは、**sudo** ルールが有効なホストを選択できます。これらは、ユーザーに **sudo** パーミッションが付与されるホストです。

すべてのホストでルールが有効になるように指定するには、**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、以下の手順に従います。

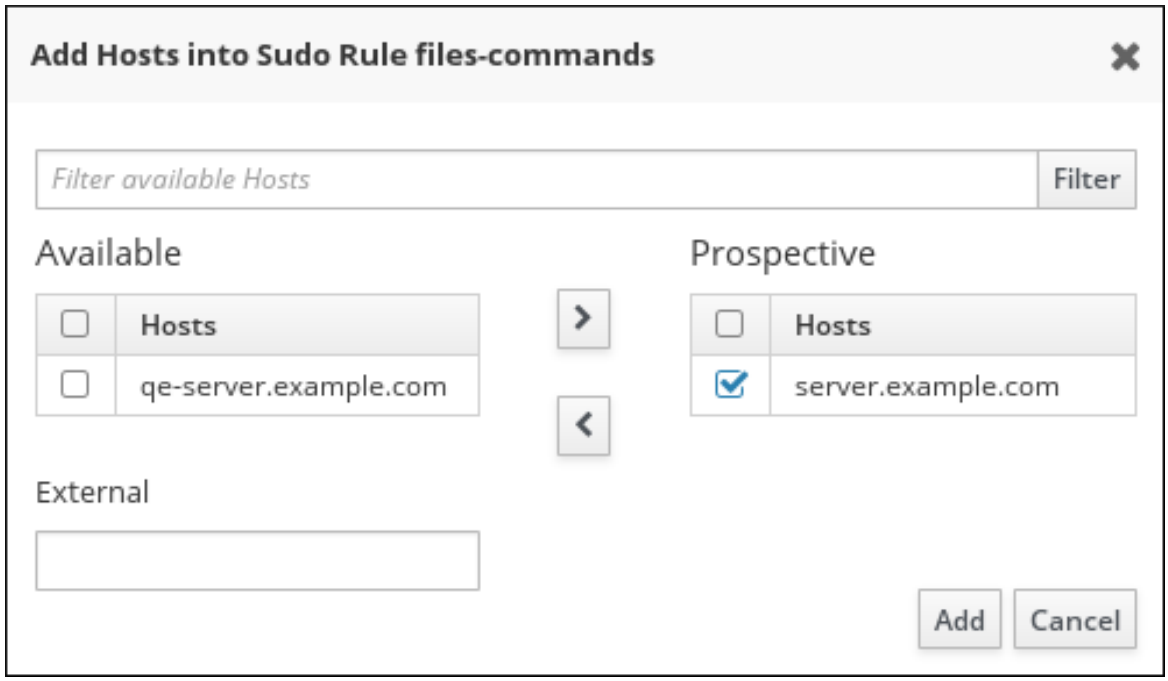
1. ホストリストの上にある **追加** をクリックします。

図30.9 ホストの sudo ルールへの追加



2. ルールに追加するホストまたはホストグループを選択し、**>** の矢印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には、**External** フィールドでホストを指定してから、**>** の矢印アイコンをクリックします。

図30.10 sudo ルールのホストの選択



3. **Add** をクリックします。

Run Commands エリア

このエリアでは、**sudo** ルールに追加するコマンドを選択できます。特定のコマンドの使用を許可または拒否するように指定できます。

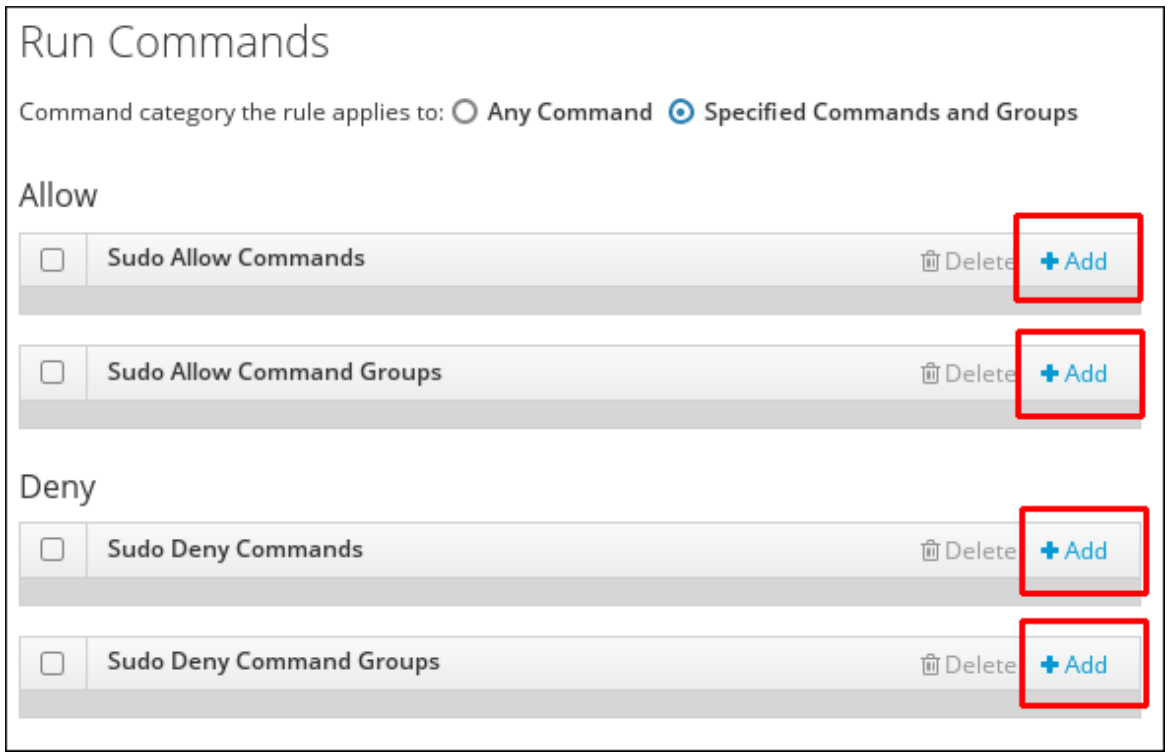
ユーザーが**sudo** で任意のコマンドを使用できることを指定するには、**Any Command** を選択します。

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、以下の手順に従います。

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。

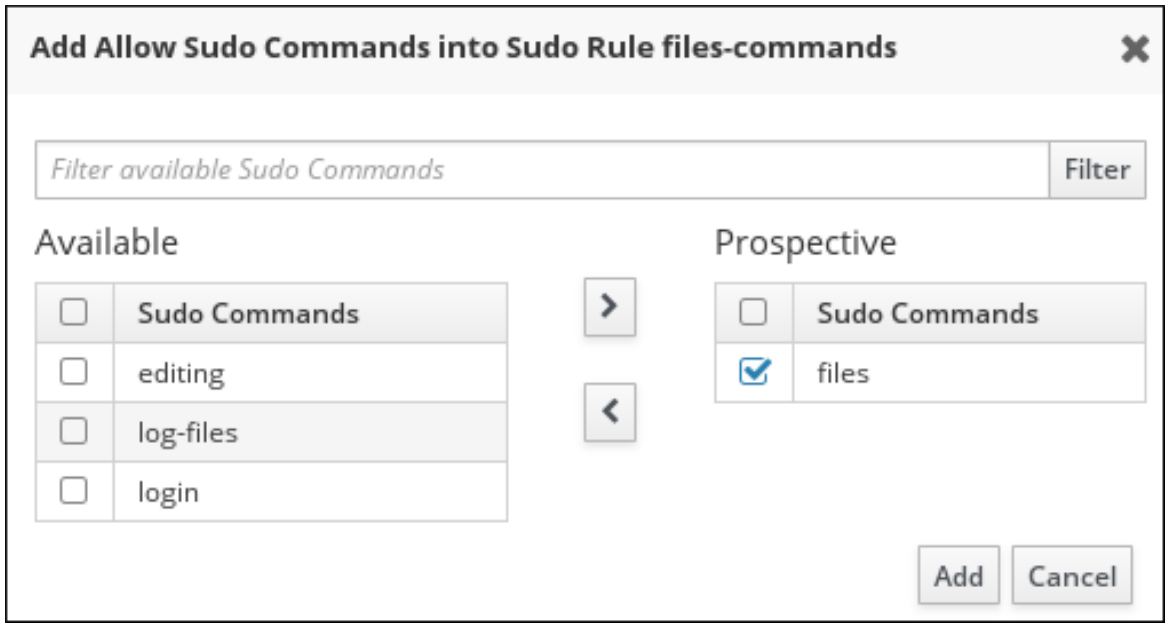
   使用できるコマンドまたはコマンドグループを指定するには、**Allow** エリアを使用します。
   拒否されたコマンドまたはコマンドグループを指定するには、**Deny** エリアを使用します。

   図30.11 sudo ルールへのコマンドの追加



2. ルールに追加するコマンドまたはコマンドグループを選択し、> の矢印ボタンをクリックして **Prospective** コラムに移動します。

   図30.12 sudo ルールのコマンドの選択



3. **Add** をクリックします。

As Whom エリア

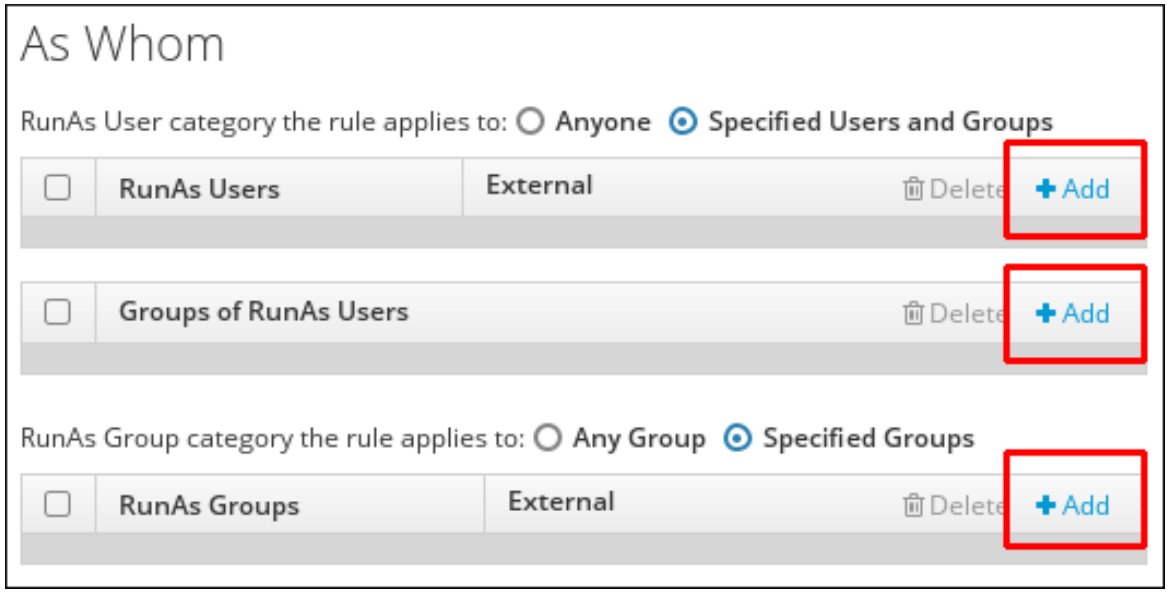
このエリアでは、指定のコマンドを特定の root 以外のユーザーとして実行するように**sudo** ルールを設定することができます。

RunAs ユーザーのグループを追加すると、そのグループのメンバーの UID を使用してコマンドが実行されることに注意してください。RunAs グループを追加すると、そのグループの GID を使用してコマンドが実行されます。

ルールがシステム上の任意のユーザーとして実行されるように指定するには、**Anyone** を選択します。ルールがシステム上の任意のグループとして実行されるように指定するには、**Any Group** を選択します。

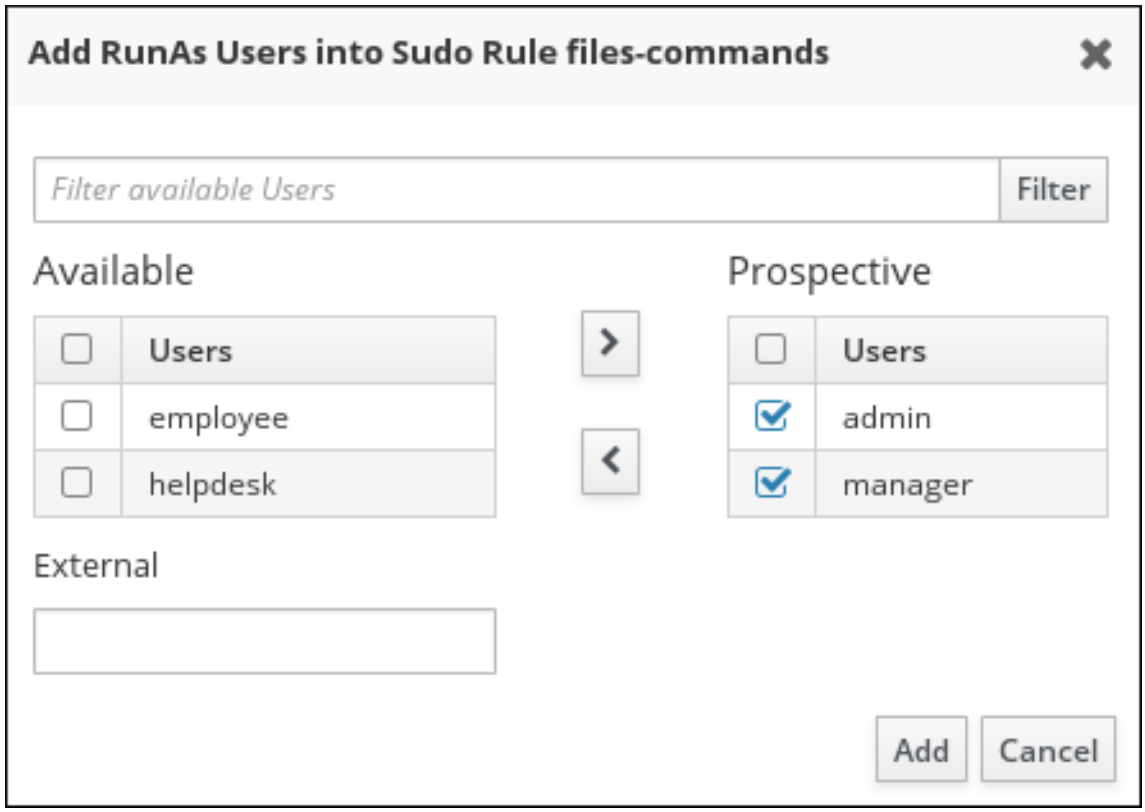
1. ユーザーリストの上にある **追加** をクリックします。

図30.13 特定のユーザーとしてコマンドを実行する sudo ルールの設定



2. 必要なユーザーまたはグループを選択し、> の矢印ボタンを使用して **Prospective** 列に移動します。外部エンティティーを追加するには、**External** フィールドで指定して > の矢印ボタンをクリックします。

図30.14 コマンドのユーザーの選択



3. **Add** をクリックします。

コマンドラインでの sudo ルールの変更

IdM コマンドラインユーティリティーを使用すると、複数の **sudo** ルール領域を設定できます。

一般的な sudo ルール管理

sudo ルールの一般的な設定を変更するには、**ipa sudorule-mod** コマンドを使用します。コマンドで使用できる最も一般的なオプションは次のとおりです。

- **sudo** ルールの説明を変更する **--desc** オプション以下に例を示します。

```
$ ipa sudorule-mod sudo_rule_name --desc="sudo_rule_description"
```

- 指定されたルールの順序を定義する **--order** オプション。以下に例を示します。

```
$ ipa sudorule-mod sudo_rule_name --order=3
```

- エンティティーのカテゴリーを指定するオプション: **--usercat** (ユーザーカテゴリー)、**--hostcat** (ホストカテゴリー)、**--cmdcat** (コマンドカテゴリー)、**--runasusercat** (run-as ユーザーカテゴリー)、および **--runasgroupcat** (run-as グループカテゴリー)このオプションは、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、または run-as グループに関連付ける **all** の値のみを受け入れます。

  たとえば、すべてのユーザーが **sudo_rule** ルールで定義されたとおりに **sudo** を使用できることを指定するには、以下を実行します。

```
$ ipa sudorule-mod sudo_rule --usercat=all
```

ルールがすでに特定のエンティティーに関連付けられている場合は、対応する **all** カテゴリーを定義する前にそのエンティティーを削除する必要があります。たとえば、以前に**sudo_rule** が **ipa sudorule-add-user** コマンドを使用して特定のユーザーに関連付けられていた場合は、最初に **ipa sudorule-remove-user** コマンドを使用してユーザーを削除する必要があります。

詳細と、**ipa sudorule-mod** で使用できるオプションの完全リストは、**--help** オプションを追加してコマンドを実行します。

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、**ファイル-コマンド** 規則に基づく **sudo** を使用するユーザーを認証する必要がないことを指定する場合は、**!authenticate** を追加します。

```
$ ipa sudorule-add-option files-commands
Sudo Option: !authenticate
-----------------------------------------------------
Added option "!authenticate" to Sudo Rule "files-commands"
-----------------------------------------------------
```

sudoers オプションの詳細は、sudoers(5) の 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、**ipa sudorule-remove-option** コマンドを使用します。以下に例を示します。

```
$ ipa sudorule-remove-option files-commands
Sudo Option: authenticate
-----------------------------------------------------------
Removed option "authenticate" from Sudo Rule "files-commands"
-----------------------------------------------------------
```

sudoを使用するパーミッションを付与されているユーザーの管理

個々のユーザーを指定するには、**ipa sudorule-add-user** コマンドに **--users** オプションを追加します。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** および **user_group** を **files-commands** ルールに追加するには、次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
-------------------------
```

個々のユーザーまたはグループを削除するには、**ipa sudorule-remove-user**を使用します。たとえば、ユーザーを削除するには、次のコマンドを実行します。

```
$ ipa sudorule-remove-user files-commands
[member user]: user
[member group]:
...
```

```
---------------------------
Number of members removed 1
---------------------------
```

ユーザーに sudo パーミッションが付与される場所の管理

ホストを指定するには、**ipa sudorule-add-host** コマンドに **--hosts** オプションを追加します。ホストグループを指定するには、**--hostgroups** を **ipa sudorule-add-host** に追加します。

たとえば、**example.com** と **host_group** を **files-commands** 規則に追加するには、次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、**ipa sudorule-remove-host** コマンドを使用します。以下に例を示します。

```
$ ipa sudorule-remove-host files-commands
[member host]: example.com
[member host group]:
...
---------------------------
Number of members removed 1
---------------------------
```

sudoで使用可能なコマンドの管理

特定のコマンドの使用を許可または拒否するように指定できます。

許可されたコマンドまたはコマンドグループを指定するには、**ipa sudorule-add-allow-command**に **--sudocmds** オプションまたは **--sudocmdgroups** オプションを追加します。拒否されたコマンドまたはコマンドグループを指定するには、**ipa sudorule-add-deny-command** コマンドに **--sudocmds** オプションまたは **--sudocmdgroups** オプションを追加します。

たとえば、**ファイル-コマンド** ルールで許可されているように **/usr/bin/less** コマンドと **ファイル** コマンドグループを追加するには、次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
-------------------------
```

ルールからコマンドまたはコマンドグループを削除するには、**ipa sudorule-remove-allow-command** コマンドまたは **ipa sudorule-remove-deny-command** コマンドを使用します。以下に例を示します。

```
$ ipa sudorule-remove-allow-command files-commands
[member sudo command]: /usr/bin/less
[member sudo command group]:
```

```
...
-------------------------
Number of members removed 1
-------------------------
```

--sudocmds は、「**sudo** コマンドの追加」 で説明されているように、IdM に追加されたコマンドのみを受け入れることに注意してください。

sudo コマンドを実行するための管理

個々のユーザーまたはユーザーの UID を、コマンドを実行する ID として使用するには、**ipa sudorule-add-runasuser** コマンドに **--users** オプションまたは **--groups** オプションを指定します。

ユーザーグループの GID をコマンドの 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。

ユーザーまたはグループを指定しないと、**sudo** コマンドが root で実行されます。

たとえば、**ユーザー** の ID を使用して、 **sudo** ルールでコマンドを実行するように指定する場合は、次のコマンドを実行します。

```
$ ipa sudorule-add-runasuser files-commands --users=user
...
RunAs Users: user
...
```

ipa sudorule-* コマンドの詳細は、**ipa help sudorule** コマンドの出力を参照するか、**--help** オプションを追加して特定のコマンドを実行してください。

例30.1 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
```

30.7. SUDO コマンド、コマンドグループ、およびルールのリスト表示と表示

Web UI での sudo コマンド、コマンドグループ、およびルールのリスト表示と表示

1. **Policy** タブで **Sudo** をクリックし、**Sudo Rules**、**Sudo Commands**、または **Sudo Command Groups** を選択します。
2. ルール、コマンドまたはコマンドグループの名前をクリックして、設定ページを表示します。

コマンドラインでの sudo コマンド、コマンドグループ、およびルールのリスト表示および表示

すべてのコマンド、コマンドグループ、およびルールのリストを表示するには、次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、次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** に関する情報を表示するには、次のコマンドを実行します。

```
$ ipa sudocmd-show /usr/bin/less
  Sudo Command: /usr/bin/less
  Description: For reading log files.
  Sudo Command Groups: files
```

これらのコマンドとそれらが受け入れるオプションの詳細は、**-help**オプションを追加して実行してください。

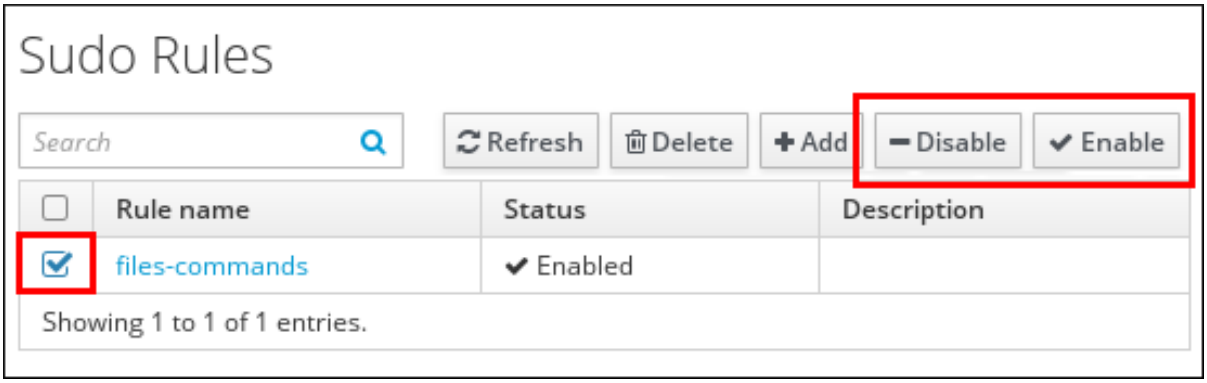
30.8. SUDO ルールの無効化および有効化

sudo ルールを無効にすると、一時的に非アクティブになります。無効なルールは IdM から削除されず、再度有効にできます。

Web UI での sudo ルールの無効化および有効化

1. **Policy** タブで Sudo → Sudo Rule をクリックします。
2. 無効にするルールを選択し、**Disable** または **Enable** をクリックします。

図30.15 sudo ルールの無効化または有効化



コマンドラインでの sudo ルールの無効化および有効化

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。

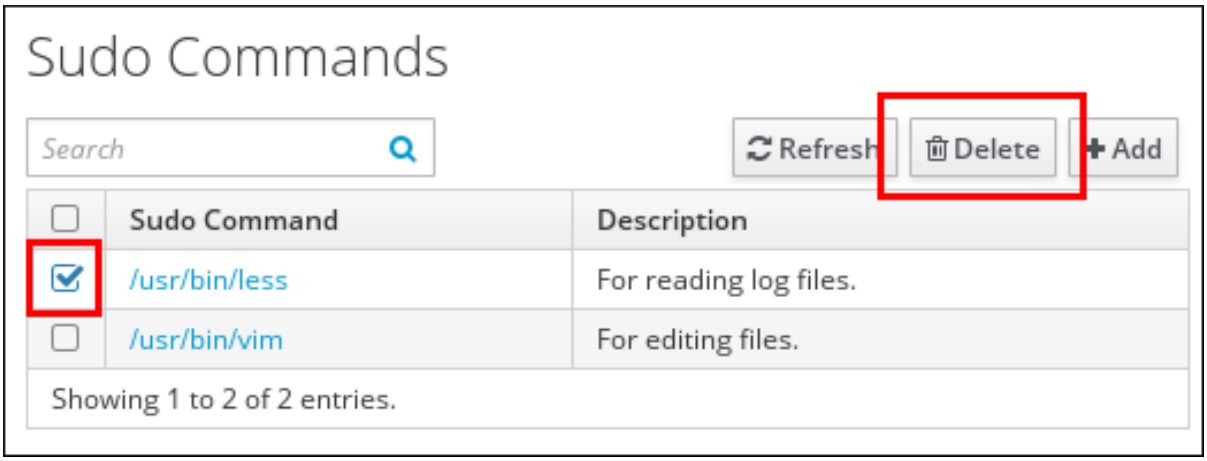
```
$ ipa sudorule-enable sudo_rule_name
---------------------------------------
Enabled Sudo Rule "sudo_rule_name"
---------------------------------------
```

30.9. 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** 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、**削除** をクリックします。

図30.16 sudo コマンドの削除



コマンドラインでの sudoのコマンド、コマンドグループ、およびルールの削除
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、次のコマンドを使用します。

- **ipa sudocmd-del**
- **ipa sudocmdgroup-del**
- **ipa sudorule-del**

これらのコマンドとそれらが受け入れるオプションの詳細は、**-help**オプションを追加して実行してください。

30.10. 関連情報

Red Hat Enterprise Linux 7 で Identity Management 環境を新しい環境に移行する際の sudo ルールのインポートおよびエクスポートの詳細は、ナレッジベースソリューション を参照してください。

第31章 ホストベースのアクセス制御の設定

本章では、Identity Management (IdM) の *ホストベースのアクセス制御* (HBAC) と、HBAC を使用して IdM ドメインでアクセス制御を管理する方法を説明します。

31.1. IDM でのホストベースのアクセス制御の仕組み

ホストベースのアクセス制御は、指定したサービス (またはサービスグループ内のサービス) を使用して、指定したホスト (またはホストグループ) にアクセスできるユーザー (またはユーザーグループ) を定義します。たとえば、以下を行うことができます。

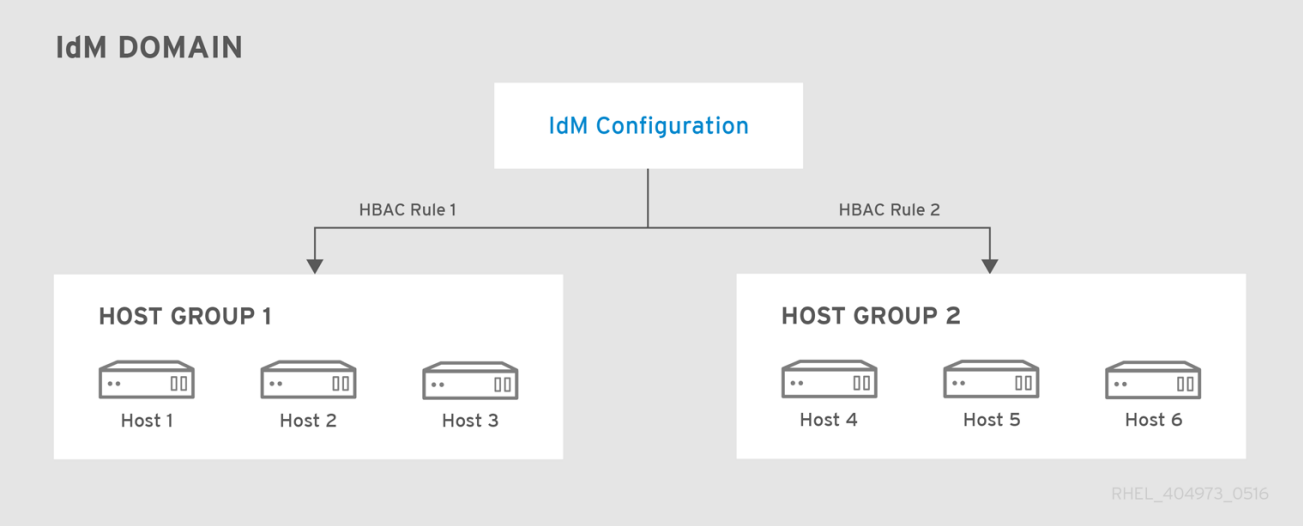
- 指定のユーザーグループのメンバーだけがドメイン内の特定のシステムにアクセスできるように制限する。
- ドメイン内のシステムにアクセスするために特定のサービスのみを使用できます。

管理者は、*HBAC ルール* と呼ばれる、一連の許可ルールを使用して、ホストベースのアクセス制御を設定します。デフォルトでは、IdM は、**low_all** という名前のデフォルトの HBAC ルールで設定されており、IdM ドメイン全体にユニバーサルアクセスを許可します。

グループへの HBAC ルールの適用

アクセス制御管理を集中化し、簡素化するには、個々のユーザー、ホスト、またはサービスの代わりに、ユーザー、ホスト、またはサービスグループ全体に HBAC ルールを適用できます。

図31.1 ホストグループとホストベースのアクセス制御

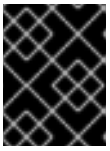


HBAC ルールをグループに適用する場合は、*自動メンバールール* の使用を検討してください。「ユーザーおよびホストの自動グループメンバーシップの定義」を参照してください。

31.2. IDM ドメインでのホストベースのアクセス制御設定

ホストベースのアクセス制御用にドメインを設定するには、次のコマンドを実行します。

1. HBAC ルールの作成
2. 新しい 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

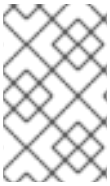
カスタムの HBAC ルールを作成する前に、**low_all** ルールを無効にしないでください。これを行うと、どのホストにもアクセスできなくなります。

31.2.1. HBAC ルールの作成

HBAC ルールを作成するには、次のコマンドを使用できます。

- IdM Web UI は、「Web UI - HBAC ルールの作成」 を参照してください。
- コマンドラインは、「コマンドライン: HBAC ルールの作成」を参照してください。

例については、「HBAC ルールの例」 を参照してください。



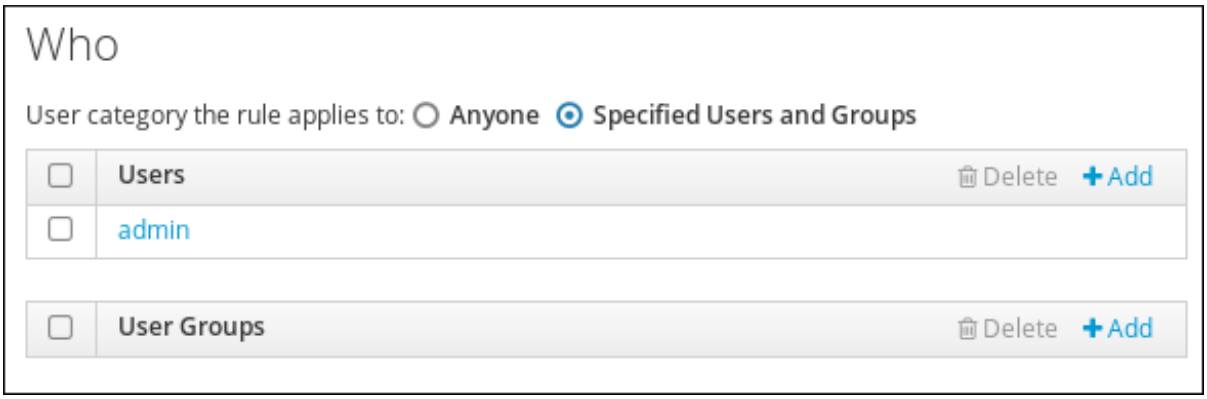
注記

IdM は、ユーザーのプライマリーグループを、IdM グループオブジェクトへのリンクの代わりに、***gidNumber*** 属性の数値として保存します。このため、HBAC ルールでは、ユーザーの補助グループだけで、プライマリーグループは参照できません。

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、**追加および編集** をクリックして、HBAC ルール設定ページに直接移動します。
4. **Who** エリアで、ターゲットユーザーを指定します。
   - 指定したユーザーまたはグループにのみ適用する場合は、**指定したユーザーとグループ** を選択します。次に、**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。
   - HBAC ルールをすべてのユーザーに適用するには、**Anyone** を選択します。

図31.2 HBAC ルールのターゲットユーザーの指定



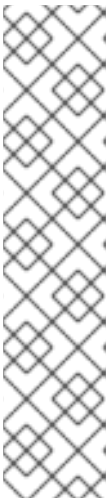
5. **Accessing** エリアで、ターゲットホストを指定します。
   - 指定したホストまたはグループにのみ HBAC ルールを適用するには、**指定したホストおよびグループ** を選択します。次に、**Add** をクリックしてホストまたはグループを追加します。

- HBAC ルールをすべてのホストに適用するには、**Any Host**を選択します。

6. **Via Service** エリアで、ターゲット HBAC サービスを指定します。

   - 指定したサービスまたはグループにのみ適用する場合は、**指定したサービスとグループ** を選択します。**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。

   - HBAC ルールをすべてのサービスに適用するには、**Any Service** を選択します。



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、デフォルトで HBAC ルールに対して設定されます。

- 現在利用可能なサービスのリストを表示するには、**ポリシー → ホストベースのアクセス制御 → HBAC サービス** を選択します。

- 現在利用可能なサービスグループのリストを表示するには、**ポリシー → ホストベースのアクセス制御 → HBAC サービスグループ** を選択します。

さらにサービスおよびサービスグループを追加するには、「カスタム HBAC サービス用の 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、ページの最上部にある **保存** が強調表示されます。この場合は、ボタンを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. ターゲットユーザーを指定します。

   - 指定したユーザーまたはグループにのみ 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、次のコマンドを実行します。

```
$ ipa hbacrule-add-user
Rule name: rule_name
[member user]:
[member group]: group_name
  Rule name: rule_name
  Enabled: TRUE
  User Groups: group_name
```

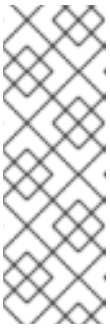
```
-------------------------
Number of members added 1
-------------------------
```

複数のユーザーまたはグループを追加するには、**--users** オプションおよび **--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
-------------------------
```

- すべてのユーザーに HBAC ルールを適用するには、**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 ルールが個々のユーザーまたはグループに関連付けられていると、**ipa hbacrule-mod --usercat=all**が失敗します。この場合は、**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. ターゲットホストを指定します。

- 指定したホストまたはグループにのみ 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、1つのホストを追加するには、次のコマンドを実行します

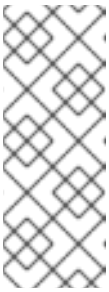
```
$ ipa hbacrule-add-host
Rule name: rule_name
[member host]: host.example.com
[member host group]:
  Rule name: rule_name
  Enabled: TRUE
  Hosts: host.example.com
-------------------------
Number of members added 1
-------------------------
```

複数のホストまたはグループを追加するには、**--hosts** オプションおよび **--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
-------------------------
```

- すべてのホストに HBAC ルールを適用するには、**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 ルールが個々のホストまたはグループに関連付けられていると、**ipa hbacrule-mod --hostcat=all**が失敗します。この場合は、**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. ターゲットの HBAC サービスを指定します。

- 指定したサービスまたはグループにのみ 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、以下のコマンドを実行します。

```
$ 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** オプションおよび **--hbacsvcgroups** オプションを使用できます。

```
$ ipa hbacrule-add-service rule_name --hbacsvcs=su --hbacsvcs=sudo
  Rule name: rule_name
```

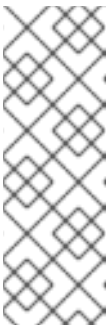
```
  Enabled: TRUE
  Services: su, sudo
-------------------------
Number of members added 2
-------------------------
```

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 ルールに対して設定されます。さらに追加するには、「カスタム HBAC サービス用の 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。

- すべてのサービスに HBAC ルールを適用するには、**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 ルールが個々のサービスまたはグループに関連付けられていると、**ipa hbacrule-mod --servicecat=all**が失敗します。この場合は、**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。

   a. **ipa hbacrule-find** コマンドを使用して、HBAC ルールが IdM に追加されていることを確認します。

   b. **ipa hbacrule-show** を実行して、HBAC ルールのプロパティーを確認します。

   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31.1 任意のサービスを使用した全ホストへの単一ユーザーアクセスの付与

admin ユーザーが任意のサービスを使用して、ドメイン内のすべてのシステムにアクセスできるようにするには、新しい HBAC ルールを作成し、以下を設定します。

- ユーザーを **admin** に
- (Web UI) ホストを **Any host**に。または **--hostcat=all** を **ipa hbacrule-add** (ルールの追加時) または **ipa hbacrule-mod** に。

- (Web UI) サービスを **Any service**に。または **--servicecat=all** を **ipa hbacrule-add** (ルールの追加時) または **ipa hbacrule-mod** に。

例31.2 特定のサービスのみをホストへのアクセスに使用できるようにする

すべてのユーザーが **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必要があることを確認するには、新しい HBAC ルールを作成し、以下を設定します。

- (Web UI) ユーザーを **Anyone**に。または **--usercat=all** を **ipa hbacrule-add** (ルールの追加時) または **ipa hbacrule-mod** に。
- **host.example.com** へのホスト
- HBAC サービスグループを **Sudo** に。これは **sudo** と関連サービスのデフォルトグループです。

31.2.2. HBAC ルールのテスト

IdM では、シミュレートシナリオを使用して、さまざまな状況で HBAC 設定をテストできます。シミュレートしたテストランを実行することで、実稼働環境に HBAC ルールをデプロイする前に、設定ミスやセキュリティーリスクを見つけることができます。

重要

プロダクションで使用する前に、カスタム HBAC ルールを常にテストしてください。

IdM では、信頼された Active Directory (AD) ユーザーに対する HBAC ルールの影響については検証され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ため、HBAC シナリオをシミュレートする場合、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。

HBAC ルールをテストするには、次のコマンドを使用できます。

- IdM Web UI は、「Web UI - HBAC ルールのテスト」 を参照してください。
- コマンドラインは、「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、テスト実行の ID が割り当てられたユーザーを指定し、**次** をクリックします。

図31.3 HBAC テストのターゲットユーザーの指定

3. **アクセス** 画面: ユーザーがアクセスしようとするホストを指定し、**次へ** をクリックします。

4. **Via Service** 画面で、ユーザーが使用するサービスを指定し、**次** をクリックします。

5. **ルール** 画面で、テストする HBAC ルールを選択し、**次へ** をクリックします。ルールを選択しないと、すべてのルールがテストされます。

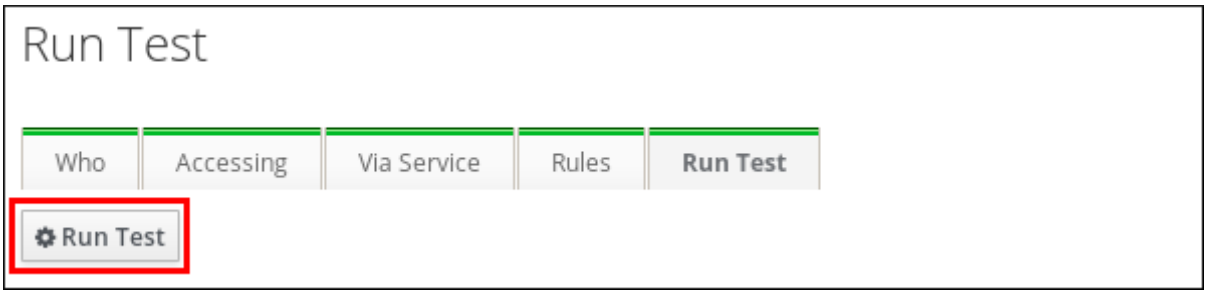
   Include Enabled を選択して、状況が **Enabled** であるすべてのルールでテストを実行します。**Include Disabled** を選択して、状況が **Disabled** であるすべてのルールでテストを実行します。HBAC ルールの状態を表示および変更するには、**ポリシー → ホストベースのアクセス制御 → HBAC ルール** を選択します。

重要

テストを複数のルールで実行すると、選択したルールの中から少なくとも 1 つがアクセスを許可していれば成功します。

6. **Run Test** 画面で **Run Test** をクリックします。

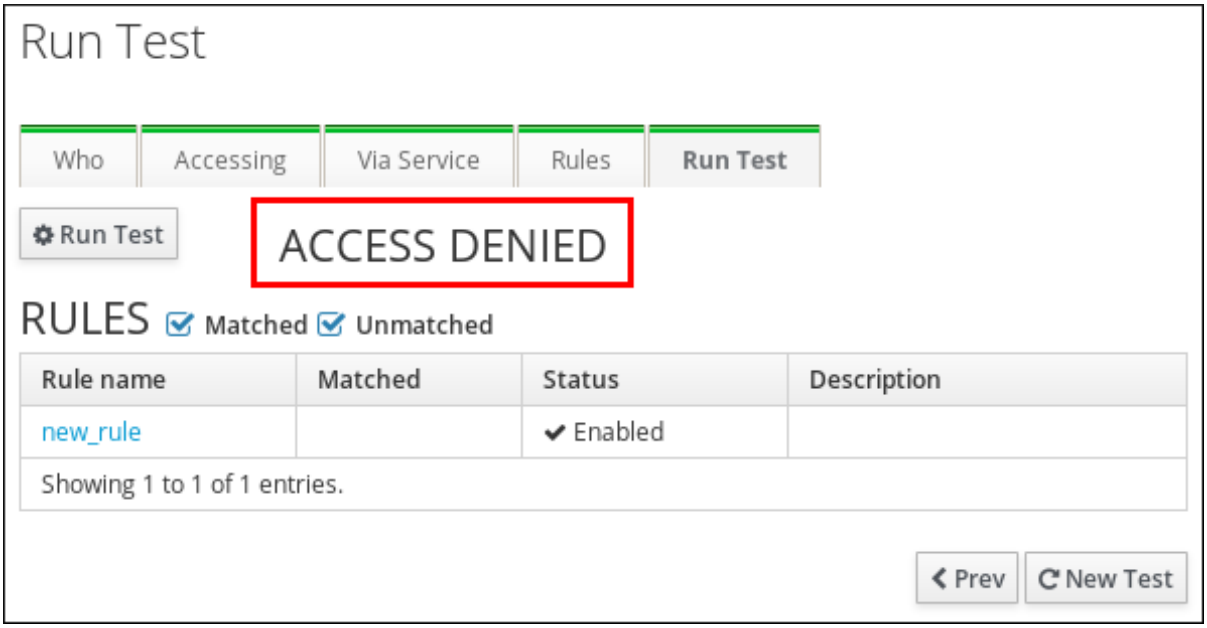
図31.4 HBAC テストの実行



7. テスト結果を確認します。

- **ACCESS DENIED** が表示された場合は、テストでそのユーザーにアクセスが許可されていませんでした。
- **ACCESS GRANTED** が表示された場合は、ホストコンピューターに正常にアクセスできたことを示しています。

図31.5 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、テストされている HBAC ルールをすべてリスト表示します。

- **Matched** を選択して、アクセスが成功したことを許可するルールを表示します。
- **Unmatched** を選択して、アクセスを阻止するルールを表示します。

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、少なくとも以下の項目を指定します。

- テストを実行する ID のユーザー
- ユーザーがアクセスを試行するホスト
- ユーザーが使用しようとするサービス

たとえば、この値を対話的に指定する場合は、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** の全 HBAC ルールでテストを実行します。別の HBAC ルールを指定するには、以下を行います。

- **--rules** オプションを使用して、1 つ以上の HBAC ルールを定義します。
- **--disable** を使用して、ステータスが **disabled** である HBAC ルールをすべてテストします。

HBAC ルールの現在の状態を表示するには、**ipa hbacrule-find** コマンドを実行します。

例31.3 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、HBAC ルール **rule2** により、**sudo** サービスを使用して **user1** が **example.com** にアクセスできなくなりました。

```
$ ipa hbactest --user=user1 --host=example.com --service=sudo --rules=rule1
---------------------
Access granted: False
---------------------
  Not matched rules: rule1
```

例31.4 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に、ルール 1 つでも正常なユーザーアクセスを許可している場合には、テストに合格します。

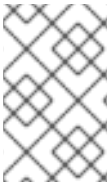
```
$ ipa hbactest --user=user1 --host=example.com --service=sudo --rules=rule1 --rules=rule2
---------------------
Access granted: True
---------------------
  Matched rules: rule2
  Not matched rules: rule1
```

出力:

- **Matched rules** は、正常なアクセスが許可されるルールをリスト表示します。
- **not match rules** は、アクセスを阻止するルールをリスト表示します。

31.2.3. HBAC ルールの無効化

HBAC ルールを無効にするとルールが非アクティブになりますが、削除はされません。HBAC ルールを無効にした場合は、後で再度有効にできます。



注記

たとえば、カスタムの HBAC ルールを初めて設定する場合は、HBAC ルールを無効にすると便利です。新しい設定がデフォルトの **low_all** HBAC ルールで上書きされないようにするには、**low_all** を無効にする必要があります。

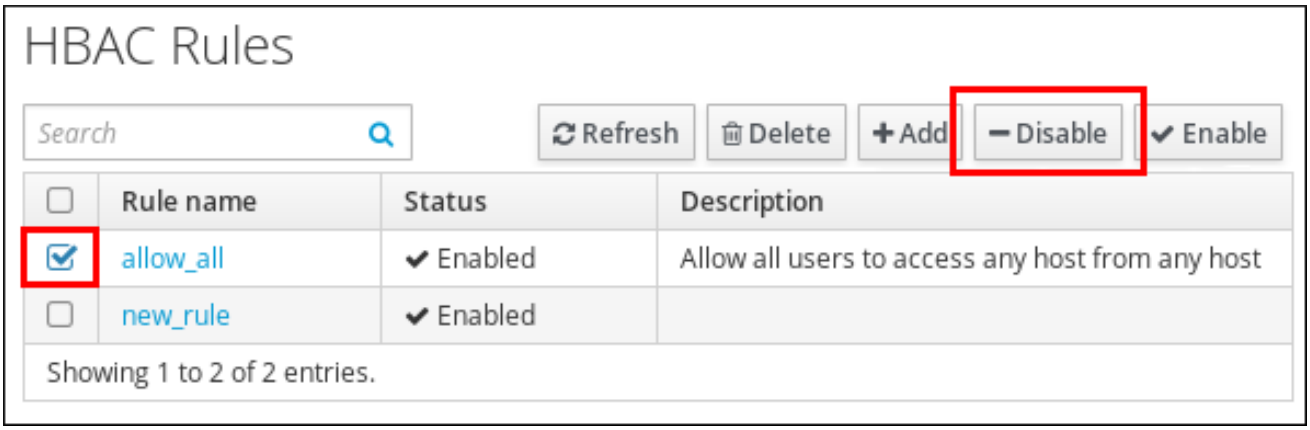
HBAC ルールを無効にするには、次のコマンドを使用できます。

- IdM Web UI は、「Web UI - HBAC ルールの無効化」 を参照してください。
- コマンドラインは、「コマンドライン: HBAC ルールの無効化」を参照してください。

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. 無効にする HBAC ルールを選択し、**Disable** をクリックします。

図31.6 allow_all 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、次のコマンドを実行します。

```
$ ipa hbacrule-disable allow_all
-----------------------------
Disabled HBAC rule "allow_all"
-----------------------------
```

31.3. カスタム HBAC サービス用の HBAC サービスエントリーの追加

最も一般的なサービスおよびサービスグループのみが、デフォルトで HBAC ルールに対して設定されます。ただし、その他のプラグ可能な認証モジュール (PAM) サービスは 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 PAM サービスを定義できます。

注記

サービスの HBAC サービスとしての追加と、ドメインへのサービスの追加は同じではありません。ドメインにサービスを追加 (「サービスエントリーおよびキータブの追加と編集」 で説明) すると、サービスは、ドメイン内のその他のリソースで認識されたリソースを利用できるようになりますが、HBAC ルールではそのサービスを使用できなくなります。

HBAC サービスエントリーを追加するには、次のコマンドを使用できます。

- IdM Web UI は、「Web UI - HBAC サービスエントリーの追加」 を参照してください。
- コマンドラインは、「コマンドライン: HBAC サービスエントリーの追加」を参照してください。

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、**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、次のコマンドを実行します。

```
$ ipa hbacsvc-add tftp
-------------------------
Added HBAC service "tftp"
-------------------------
  Service name: tftp
```

31.4. HBAC サービスグループの追加

HBAC サービスグループは、HBAC ルール管理を簡素化できます。個々のサービスを HBAC ルールに追加する代わりに、サービスグループ全体を追加できます。

HBAC サービスグループを追加するには、次のコマンドを使用できます。

- IdM Web UI は、「Web UI - HBAC サービスグループの追加」 を参照してください。
- コマンドラインは、「コマンドライン: HBAC サービスグループの追加」を参照してください。

Web UI - HBAC サービスグループの追加

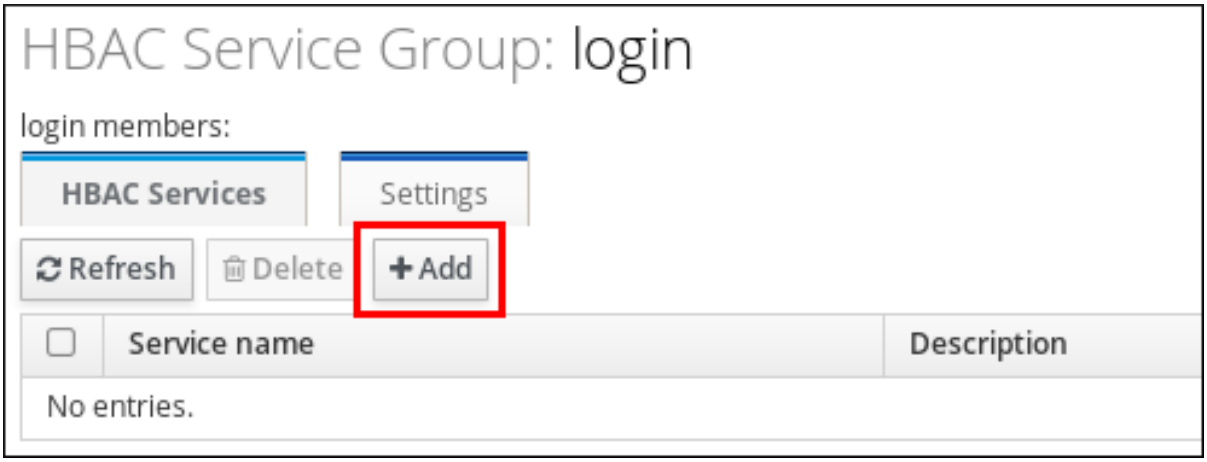
1. **Policy → Host-Based Access Control → HBAC Service Groups**を選択します。

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、**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 **追加** を選択し、HBAC サービスをグループのメンバーとして追加します。

図31.7 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 hbacsvcgroup-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。たとえば、**login** という名前のグループを追加するには、次のコマンドを実行します。

```
$ ipa hbacsvcgroup-add
Service group name: login
------------------------------
Added HBAC service group "login"
------------------------------
  Service group name: login
```

2. **ipa hbacsvcgroup-add-member** コマンドを使用して、HBAC サービスをグループのメンバーとして追加します。たとえば、**sshd** サービスを **login** グループに追加するには、次のコマンドを実行します。

```
$ ipa hbacsvcgroup-add-member
Service group name: login
[member HBAC service]: sshd
  Service group name: login
  Member HBAC service: sshd
-------------------------
Number of members added 1
-------------------------
```

第32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、他のアプリケーションからのアクセスを制限または許可する **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 Managemen 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。

32.1. IDENTITY MANAGEMENT、SELINUX、およびユーザーのマッピング

Identity Management は、システムで SELinux コンテキストを作成または変更しません。むしろ、システムの SELinux ユーザーに、ドメインの IdM ユーザーをマッピングする基準として、ターゲットホストの既存のコンテキストに一致する可能性がある文字列を使用します。

セキュリティーが強化した Linux は、システム上のその他のリソースとプロセスが対話する仕組みに必須のアクセスコントロールであるカーネルレベルを定義します。システムで想定されるプロセスの挙動と、そのセキュリティーへの影響を基に、ポリシーと呼ばれる特定のルールが設定されます。これは、主にファイルの所有権およびユーザー ID を対象とする、高レベルの任意アクセス制御とは対照的です。システムのすべてのリソースには、コンテキストが割り当てられます。リソースには、ユーザー、アプリケーション、ファイル、およびプロセスが含まれます。

システムユーザーが、SELinux **ロール** に関連付けられます。このロールには、マルチレイヤーセキュリティーコンテキスト (MLS) とマルチレイヤー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 コンテキストおよび MCS コンテキストでは、ユーザーがシステム上の特定のプロセス、ファイル、および操作にのみアクセスできるように、ユーザーを制限しています。

利用可能な SELinux ユーザーのリストを表示するには、以下のコマンドを実行します。

```
[root@server1 ~]# semanage user -l

                Labelling  MLS/       MLS/
SELinux User    Prefix     MCS Level  MCS Range                    SELinux Roles

guest_u         user       s0         s0                           guest_r
root            user       s0         s0-s0:c0.c1023               staff_r sysadm_r system_r unconfined_r
staff_u         user       s0         s0-s0:c0.c1023               staff_r sysadm_r system_r unconfined_r
sysadm_u        user       s0         s0-s0:c0.c1023               sysadm_r
system_u        user       s0         s0-s0:c0.c1023               system_r unconfined_r
unconfined_u    user       s0         s0-s0:c0.c1023               system_r unconfined_r
user_u          user       s0         s0                           user_r
xguest_u        user       s0         s0                           xguest_r
```

Red Hat Enterprise Linux の SELinux の詳細は、Red Hat Enterprise Linux 7 SELinux ユーザーおよび管理者のガイド を参照してください。

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。これは多くの状況で許容されますが、SELinux には共通の定義済みシステムユーザーと SELinux 対応サービスが独自のポリシーを定義しているため、リモートユーザーとシステムがローカルリソースにアクセスする際に問題が発生します。リモートのユーザーとサービスには、実際の SELinux ユーザーとロールの内容を知らなくても、デフォルトのゲストコンテキストを割り当てることができます。

Identity Management は、ID ドメインをローカルの SELinux サービスと統合できます。Identity Management では、IdM ユーザーを、設定した SELinux ロール**ホストごと、ホストグループごと** にマッピングするか、**HBAC ルール** に基づいてマッピングできます。SELinux および IdM ユーザーのマッピングにより、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付与されます。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。
- ユーザーに関連付けられた SELinux コンテキストは集中管理されます。
- 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ー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 ユーザーマップでは、システムの SELinux ユーザー、IdM ユーザー、および IdM ホストの 3 つの間に存在する 2 つの関係が定義されます。まず、SELinux ユーザーマップでは、SELinux ユーザーと IdM ホスト (ローカルまたはターゲットシステム) の間の関係を定義します。次に、SELinux ユーザーと IdM ユーザーの関係を定義します。

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 のマッピングルールの中核となるのは、SELinux システムユーザーです。各マップは、最初に SELinux ユーザーに関連付けられます。マッピングに利用できる SELinux ユーザーは、IdM ユーザーで設定されます。そのため、中央のユニバーサルなリストがあります。この方法で、IdM は、認識している SELinux ユーザーのセットを定義し、ログイン時に IdM ユーザーと関連付けることができます。デフォルトでは、次のようになります。

- unconfined_u (IdM ユーザーのデフォルトとしても使用されます)
- guest_u
- xguest_u
- user_u
- staff_u

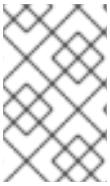
ただし、このデフォルトリストは変更でき、**ネイティブ**の SELinux ユーザー (「Identity Management、SELinux、およびユーザーのマッピング」 を参照) は、集約 IdM SELinux ユーザーリストに追加または削除できます。

IdM サーバー設定では、各 SELinux ユーザーは、ユーザー名だけでなく、その MLS および MCS の範囲である **SELinux_user:MLS[:MCS]** で設定されます。IPA サーバーは、マップの設定時にこの形式を使用して SELinux ユーザーを特定します。

IdM のユーザーおよびグループの設定は柔軟性が非常に高くなります。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ることもできます。

また、SELinux マッピングルールをホストベースのアクセス制御ルールに関連付けて、管理を容易にし、2 つの場所で同じルールを重複しないようにしてルールの同期を維持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がユーザーとホストを定義する限り、SELinux ユーザーマップに使用する

ことができます。(31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。

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。SSSD キャッシュを使用すると、マッピングがオフラインで機能できます。

32.2. SELINUX ユーザーマップの順序とデフォルト値の設定

SELinux ユーザーマップは、クライアント上の SELinux ユーザーと、IdM ユーザーの間の関連付けです。

利用可能な SELinux ユーザーマップの順序は、IdM サーバー設定の一部です。SELinux のユーザーマップの順序は、SELinux のユーザーのリストで、最も限定されているものから最も限定されていないものの順になります。SELinux ユーザーエントリーには、以下の形式が使われます。

```
SELinux_user:MLS[:MCS]
```

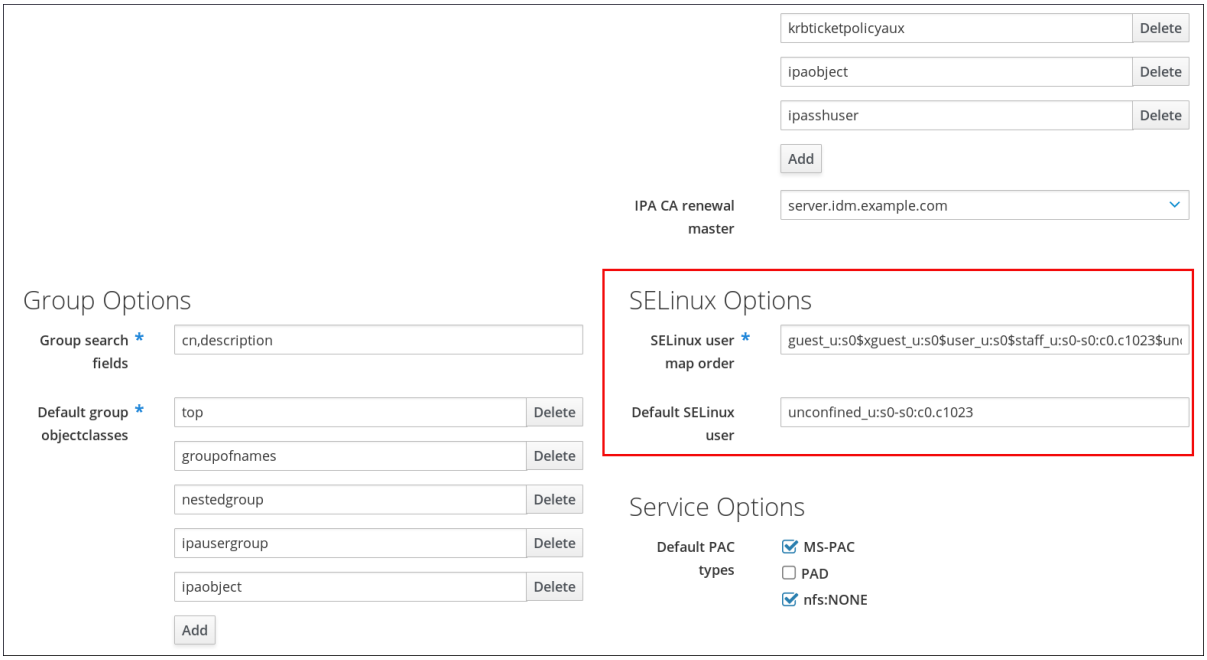
個別のユーザーエントリーは、ドル記号 ($) で区切ります。

SELinux マップを持つユーザーエントリーはないため、多くのエントリーをマッピングできない場合があります。IdM サーバー設定では、デフォルトの SELinux ユーザー (SELinux マップリスト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 ユーザーでも、SELinux コンテキストが機能します。マッピングされていない IdM ユーザーエントリーのデフォルトの SELinux ユーザーは **unconfined_u** (Red Hat Enterprise Linux のシステムユーザーのデフォルトの SELinux ユーザー) です。

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 ポリシーを定義しません。IdM ユーザーと SELinux ユーザーのマップを定義し、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。詳細は、「SELinux ユーザーおよび IdM ユーザーのマッピング」を参照してください。

32.2.1. 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リストでスクロールダウンします。
3. SELinux ユーザー設定、**SELinux ユーザーマップの順序**、**デフォルトの SELinux ユーザー**、またはその両方を編集します。



4. 変更を保存するには、ページの上部にある **Update** リンクをクリックします。

32.2.2. コマンドラインでの設定

SELinux ユーザーのリストを表示するには、IdM サーバー設定で指定した、マッピング可能なユーザーのリストを表示します。

```
[user1]@server ~]$ ipa config-show
...
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
```

SELinux ユーザー設定を編集するには、**config-mod** コマンドを使用します。

例32.1 SELinux ユーザーのリスト

マッピングに使用できる SELinux ユーザーのリストを変更する場合は、**--ipaselinuxusermaporder** を使用します。以下のように、このリストでは、SELinux ユーザーの順序は、最も高いものから、最も低いものへと振り分けられています。

```
[user1@server ~]$ ipa config-mod --ipaselinuxusermaporder="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プリストに含めないと、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 マップリストにあるユーザーに変更する必要があり、そうでない場合はマップリストを先に更新する必要があります。

例32.2 デフォルトの SELinux ユーザー

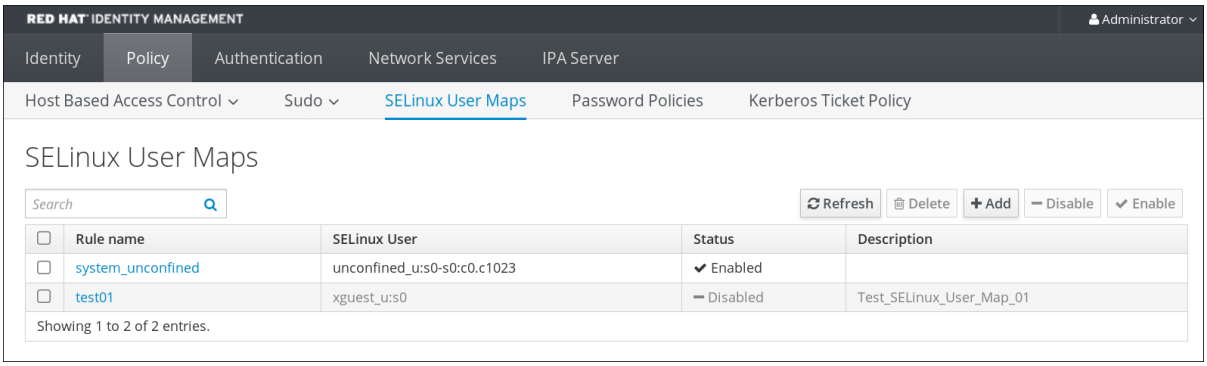
> IdM ユーザーは特定の SELinux ユーザーをアカウントにマッピングさせる必要はありません。ただし、ローカルシステムは、IdM ユーザーアカウントに使用する SELinux ユーザーの IdM エントリーを確認します。
>
> デフォルトの SELinux ユーザーを変更するには、**--ipaselinuxusermapdefault** を使用します。以下に例を示します。
>
> ```
> [user1@server ~]$ ipa config-mod --ipaselinuxusermapdefault="guest_u:s0"
> ```

32.3. SELINUX ユーザーおよび IDM ユーザーのマッピング

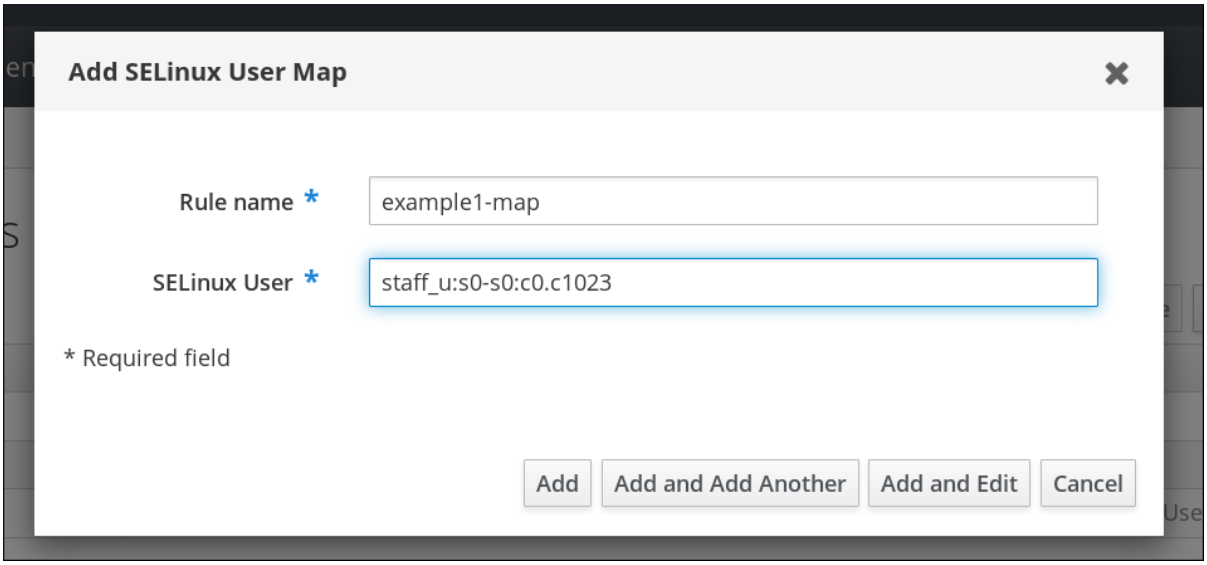
SELinux マップは、ローカルシステム上の SELinux ユーザーコンテキストをドメイン内の 単一または複数の IdM ユーザーに関連付けます。SELinux マップは、SELinux ユーザーコンテキストと IdM ユーザー/ホストのペアという 3 つの部分で設定されています。この IdM ユーザー/ホストのペアは、以下のいずれかの方法で定義できます。明示的なホストまたはホストグループ上の明示的なユーザーまたはユーザーグループに設定するか、ホストベースのアクセス制御ルールを使用して定義できます。

32.3.1. 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 Mappings** 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. マップ名と SELinux ユーザー名を入力します。SELinux ユーザーの形式は、IdM サーバー設定での表示と同じである必要があります。SELinux ユーザーの形式は、**SELinux_user:MLS[:MCS]** です。

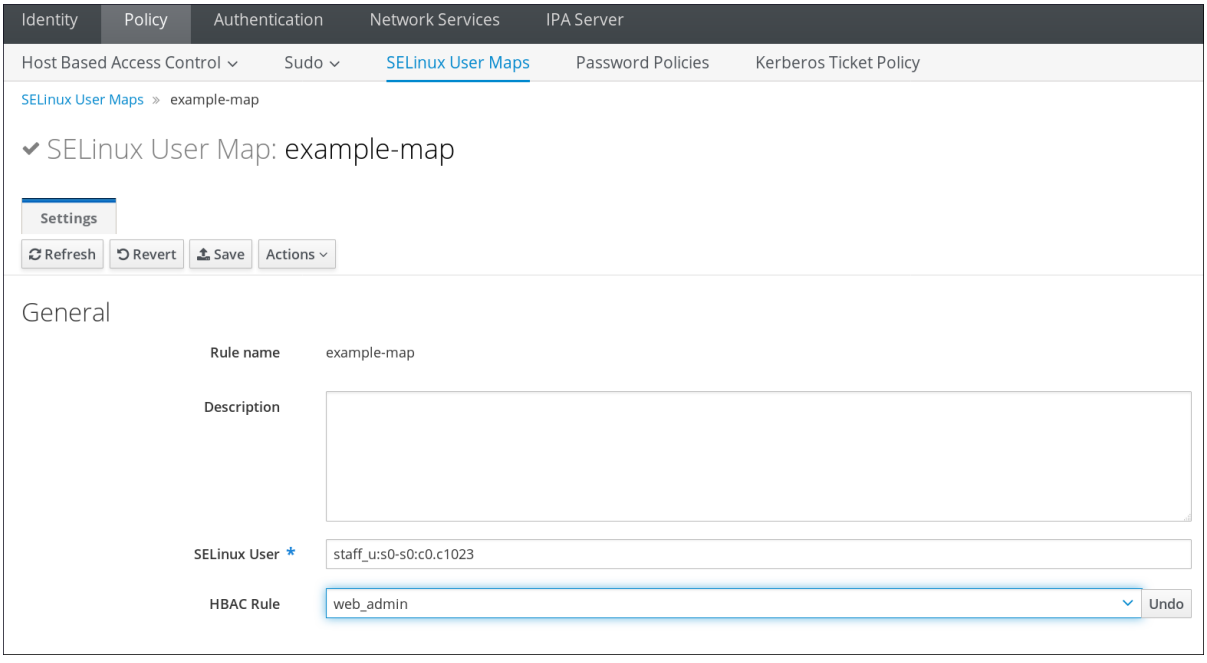


4. **Add and Edit** をクリックして、IdM ユーザー情報を追加します。

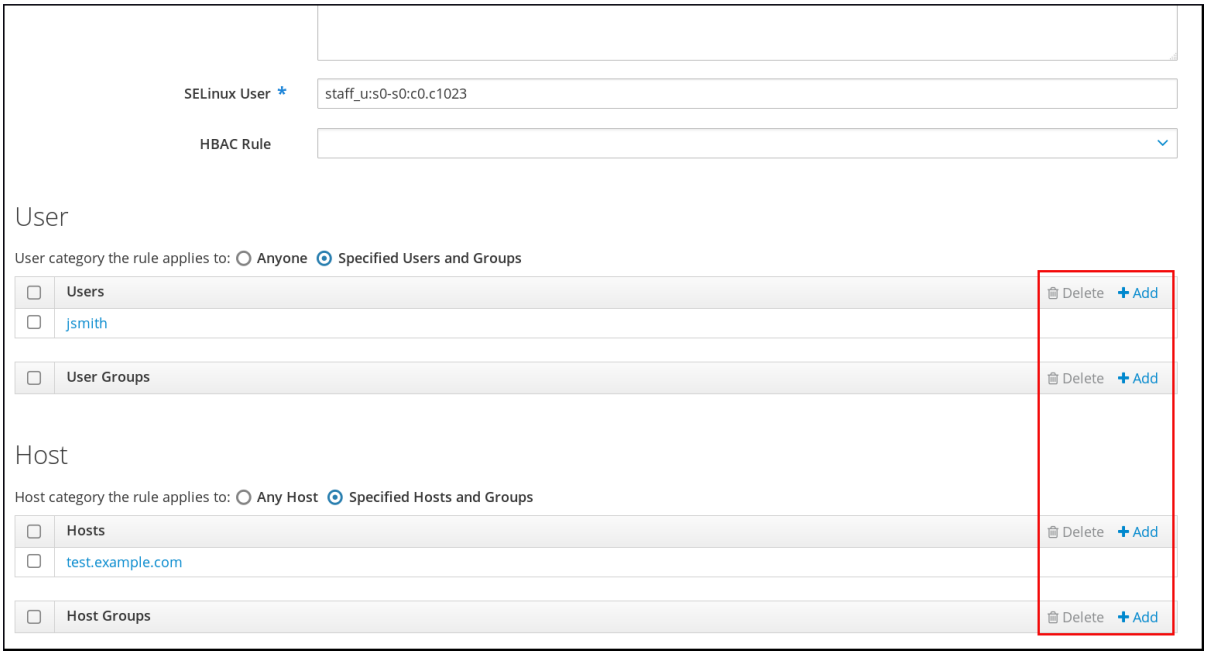
5. ホストベースのアクセス制御ルールを設定するには、設定の **General** エリアでドロップダウンメニューからルールを選択します。ホストベースのアクセス制御ルールを使用すると、リモートユーザーがターゲットマシンにアクセスする際に使用するホストでアクセス制御が導入されます。**割り当て可能なホストベースのアクセス制御ルールは、1つのみです。**

注記

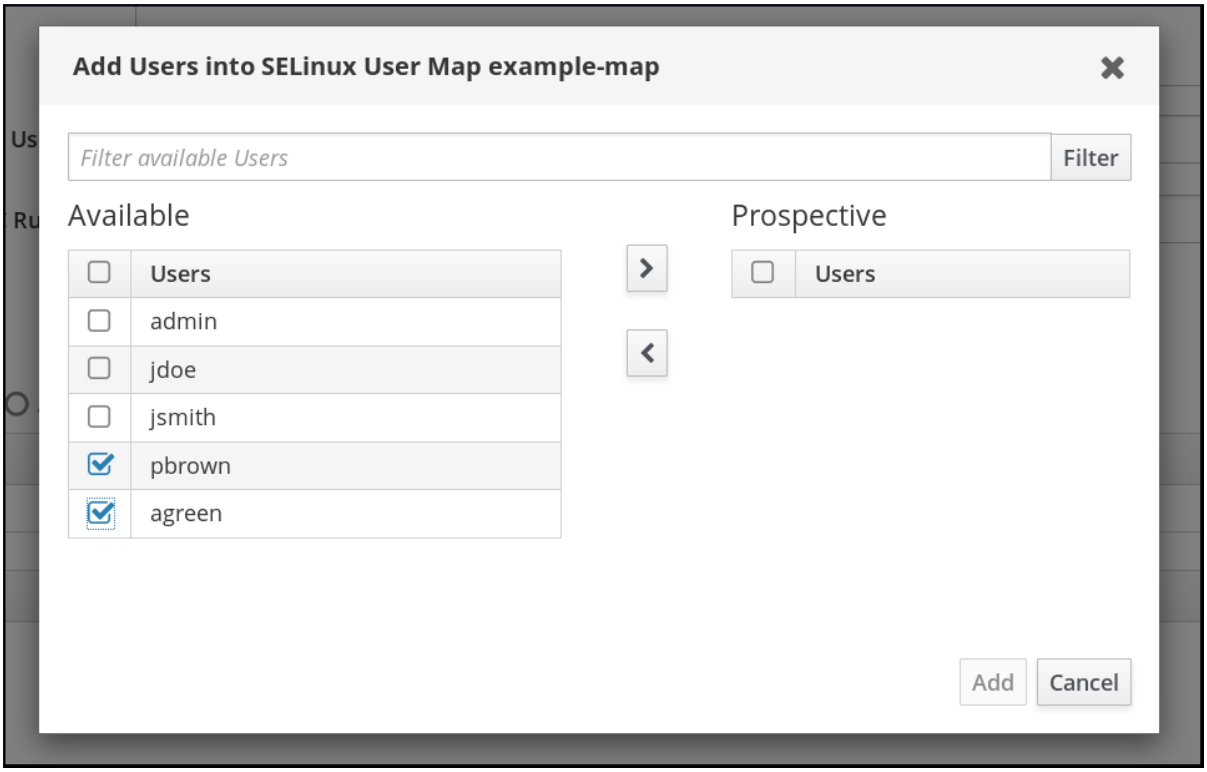
ホストベースのアクセス制御ルールには、サービスだけでなく、ユーザーとホストも含める必要があります。



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

オプションは1つのみ使用できます。ホストベースのアクセス制御ルールを指定するか、ユーザーおよびホストを手動で設定できます。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** リンクをクリックして、SELinux ユーザーマップへの変更を保存します。

32.3.2. 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 ユーザーと関連づけられたホストまたはホストグループ (**--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**):

ルールは、**selinuxusermap-add** コマンドを使用して、すべての情報を一度に追加できます。ユーザーとホストは、**selinuxusermap-add-user** および **selinuxusermap-add-host** コマンドを使用してそれぞれ作成した後にルールに追加できます。

例32.3 新規 SELinux マップの作成

この **--selinuxuser** 値は、IdM サーバー設定に表示されているとおりに SELinux ユーザー名である必要があります。SELinux ユーザーの形式は、**SELinux_user:MLS[:MCS]** です。

SELinux マッピングを有効にするには、ユーザー、ユーザーグループ、およびホストグループを指定する必要があります。user オプション、host オプション、および group オプションは、複数回使用することも、**--option={val1,val2,val3}** などのように、一度にまとめて、中括弧で囲んだコンマ区切りにして使用することもできます。

```
[user1@server ~]$ ipa selinuxusermap-add --selinuxuser="xguest_u:s0" selinux1
[user1@server ~]$ ipa selinuxusermap-add-user --users=user1 --users=user2 --users=user3
selinux1
[user1@server ~]$ ipa selinuxusermap-add-host --hosts=server.example.com --
hosts=test.example.com selinux1
```

例32.4 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 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識別します。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。

```
[user1@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、31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32.5 ユーザーを SELinux マッピングに追加する

ユーザーおよびホストは、既存のマップに追加できます。これは、**selinuxusermap-add-user** または **selinuxusermap-add-host** など、特定のコマンドを使用して行われます。

```
[user1@server ~]$ ipa selinuxusermap-add-user --users=user1 selinux1
```

selinuxusermap-mod コマンドに **--hbacrule** オプションを指定して、既存の SELinux マップを変更すると、新しい SELinux マップにより、以前の SELinux マップが上書きされます。

例32.6 ユーザーの SELinux マッピングからの削除

特定のユーザーまたはホストは、**selinuxusermap-remove-host** または **selinuxusermap-remove-user** コマンドを使用して SELinux マップから削除できます。以下に例を示します。

```
[user1@server ~]$ ipa selinuxusermap-remove-user --users=user2 selinux1
```

パート VII. 管理: ネットワークサービスの管理

本パートでは、**ID 管理** と統合されている**ドメインネームサービス** (DNS) を管理する方法と、**Automount** を使用して複数システムにわたってディレクトリーを管理、編成、およびアクセスする方法を説明します。

第33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できるため、外部 DNS サービスまたは DNS が設定された状態で使用できます。詳細は 「IdM サーバーのインストール: 概要」 および 「統合 DNS を使用するかどうかの決定」 を参照してください。

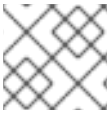
DNS サービスがドメイン内で設定されている場合、IdM は管理者に対して幅広い DNS 設定の制御や柔軟性を提供します。たとえば、ホストのエントリー、場所、レコードなどのドメインの DNS エントリーは、ネイティブの IdM ツールを使用して管理でき、クライアントは独自の DNS レコードを動的に更新できます。

BIND バージョン 9.9 で利用可能なほとんどのドキュメントとチュートリアルは、IdM DNS にも適用されます。これは、ほとんどの設定オプションが BIND と IdM で同じように機能するためです。本章では、主に BIND と IdM との間の顕著な相違点に着目します。

33.1. IDENTITY MANAGEMENT での BIND

IdM は、BIND DNS サーバーバージョン 9.9 と、データレプリケーションに使用される LDAP データベース、および GSS-TSIG プロトコルを使用した DNS 更新署名の Kerberos を統合します。[3].これにより、IdM 統合 DNS サーバーがマルチマスター操作に対応し、単一障害点を持たずにすべての IdM 統合 DNS サーバーがクライアントからの DNS 更新を受け入れられるようになるため、IdM ツールを使用して DNS 管理が便利になり、耐障害性が向上します。

デフォルトの IdM DNS 設定は、パブリック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 サーバーがパブリック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 に記載されているように、BIND サービスに該当する通常のハードニングを適用することが推奨されます。



注記

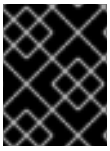
chroot 内では、IdM と統合されている BIND を実行できません。

Red Hat Enterprise Linux の DNS (Domain Name System) プロトコルの **BIND** (Berkeley Internet Name Domain) 実装には、**名前付き** の DNS サーバーが含まれています。**named-pkcs11** は、PKCS#11 暗号化標準に対するネイティブサポートありで構築された **BIND** DNS サーバーのバージョンです。

IdM と統合された BIND は、**bind-dyndb-ldap** プラグインを使用してディレクトリーサーバーと通信します。IdM は、BIND サービスの **/etc/named.conf** ファイルに **dynamic-db** 設定セクションを作成します。これにより、BIND の **named-pkcs11** サービスの **bind-dyndb-ldap** プラグインが設定されます。

標準の BIND と IdM DNS の最も注目すべき違いは、IdM がすべての DNS 情報を LDAP エントリーとして保存することです。すべてのドメイン名は LDAP エントリーとして表され、すべてのリソースレコードは LDAP エントリーの LDAP 属性として保存されます。たとえば、次の**client1.example.com.** ドメインネームには、A レコードが 3 つ、AAAA レコードが 1 つ含まれます。

```
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
```

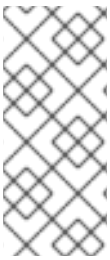


重要

DNS データまたは BIND 設定を編集するには、常に、本章で説明されている IdM ツールを使用します。

33.2. サポート対象の DNS ゾーンタイプ

IdM は、2 つの DNS ゾーンタイプ (*マスター*と *正引き*) に対応します。



注記

本ガイドでは、ゾーンタイプには BIND の用語を使用し、Microsoft Windows DNS で使用する用語とは異なります。BIND のマスターゾーンは、Microsoft Windows DNS の *正引きルックアップゾーン* と *逆引きルックアップゾーン* と同じ目的で使用されます。BIND の正引きゾーンは、Microsoft Windows DNS の *条件付きフォワーダー* と同じ目的で使用されます。

マスター DNS ゾーン

マスター DNS ゾーンには、権威 DNS データが含まれ、DNS を動的に更新できます。この動作は、標準 BIND 設定の **type master** 設定と同じです。マスターゾーンは、**ipa dnszone-*** コマンドを使用して管理されます。

標準の DNS ルールに従い、すべてのマスターゾーンに SOA レコードおよび NS レコードが含まれている必要があります。IdM では、DNS ゾーンの作成時にこれらのレコードが自動的に生成されますが、NS レコードを親ゾーンに手動でコピーして適切な委譲を作成する必要があります。

標準の BIND の動作に従って、マスターゾーンに指定された転送設定は、サーバーに権限がない名前に対するクエリーにのみ影響します。

> **例33.1 DNS 転送のシナリオ例**
>
> IdM サーバーには **test.example.** マスターゾーンが含まれています。このゾーンには、**sub.test.example.** 名前の NS 委譲レコードが含まれます。また、**test.example.** ゾーンは、**192.0.2.254** フォワーダー IP アドレスで設定されます。
>
> クライアントが **nonexistent.test.example.** の名前をクエリーすると、**NXDomain** の応答を受け取りますが、IdM サーバーはこの名前に対して権威があるため、転送は発生しません。
>
> 反対に、**sub.test.example.** の名前をクエリーすると、IdM サーバーはこの名前に対して権威がないため、設定済みのフォワーダー (**192.0.2.254**) に転送されます。

正引き DNS ゾーン

正引きの DNS ゾーンには、信頼できるデータは含まれていません。正引きの DNS ゾーンに属する名前に対して出されたクエリーはすべて、指定のフォワーダーに転送されます。この動作は、標準 BIND 設定の **type forward** 設定と同じです。正引きゾーンは、**ipa dnsforwardzone-*** コマンドを使用して管理されます。

33.3. DNS 設定の優先順位

多くの DNS 設定オプションは、3 つの異なるレベルで設定できます。

ゾーン固有の設定

IdM に定義されている特定のゾーンに固有の設定は、優先度が最も高いレベルです。**ipa dnszone-*** と **ipa dnsforwardzone-*** コマンドを使用してゾーン固有の設定を管理できます。

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は、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-*** コマンドを使用して管理します。グローバル DNS 設定で定義したオプションは、すべての IdM DNS サーバーに適用されます。

/etc/named.confの設定

IdM DNS サーバーごとに **/etc/named.conf** ファイルで定義されている設定の優先度は、最も低くなります。これは各サーバーに固有のものであり、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。その他のオプションは、上記のゾーン固有の DNS 設定およびグローバル DNS 設定のコマンドを使用して管理されます。

DNS オプションは、一度に複数レベルで設定できます。このような場合に、最も優先度が高い設定は、レベルが低い設定よりも優先されます。

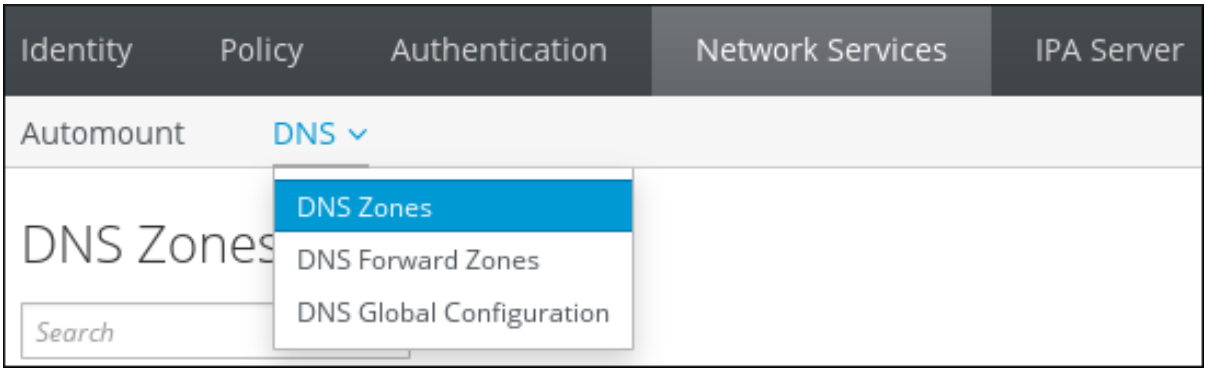
33.4. マスター DNS ゾーンの管理

33.4.1. マスター DNS ゾーンの追加および削除

Web UI でのマスター DNS ゾーンの追加

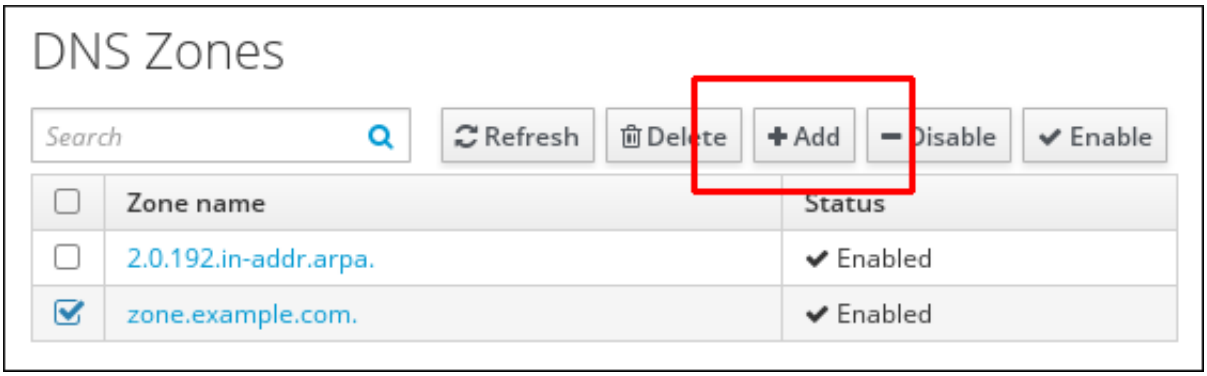
1. **Network Services** タブを開き、**DNS** サブタブを選択し、その後に **DNS Zones** セクションを選択します。

   図33.1 DNS マスターゾーンの管理

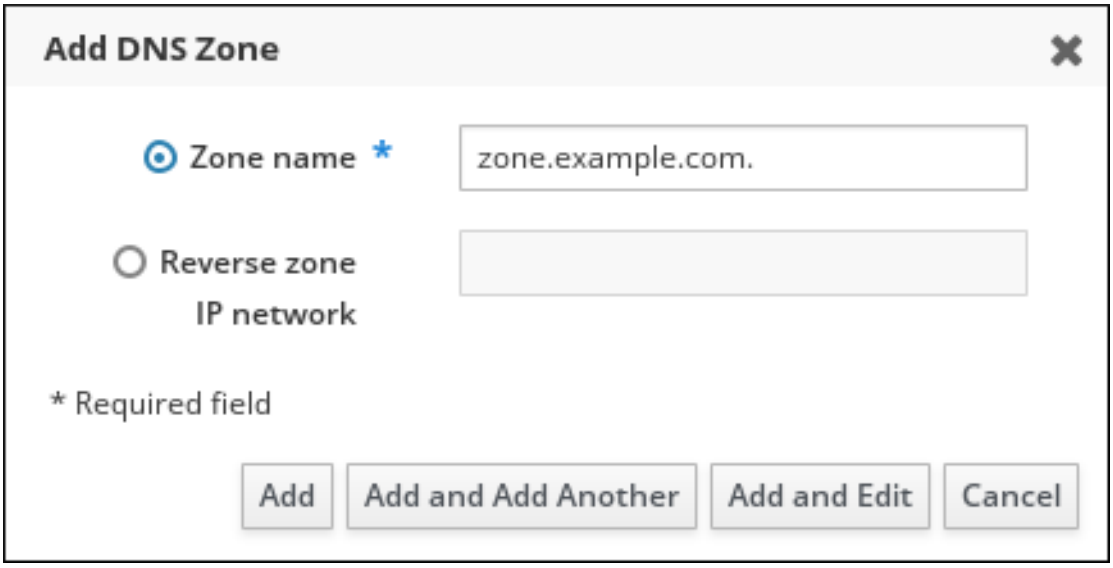
2. 新しいマスターゾーンを追加するには、すべてのゾーン のリストの上部にある **追加** をクリックします。

図33.2 マスター DNS ゾーンの追加



3. ゾーン名を指定して、**追加** をクリックします。

図33.3 新しいマスターゾーンの入力



コマンドラインでのマスター DNS ゾーンの追加

ipa dnszone-add コマンドは、新しいゾーンを DNS ドメインに追加します。新しいゾーンを追加するには、新しいサブドメイン名を指定する必要があります。サブドメイン名を直接指定するには、以下のコマンドを実行します。

```
$ ipa dnszone-add newserver.example.com
```

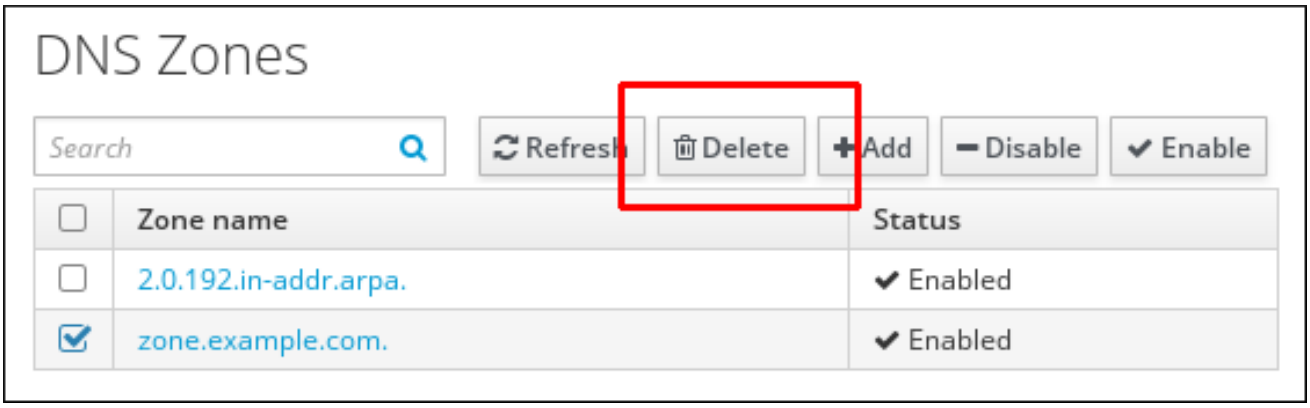
ipa dnszone-add に名前を指定しない場合には、スクリプトにより自動的に名前を求めるプロンプトが表示されます。

ipa dnszone-add コマンドでは、さまざまなコマンドラインオプションも使用できます。このオプションの完全なリストを表示するには、**ipa dnszone-add --help** コマンドを実行します。

マスター DNS ゾーンの削除

Web UI でマスター DNS ゾーンを削除する場合は、すべてのゾーンのリストでゾーン名を指定してチェックボックスを選択し、**削除** をクリックします。

図33.4 マスター DNS ゾーンの削除



コマンドラインからマスター DNS ゾーンを削除する場合は、**ipa dnszone-del** コマンドを使用します。以下に例を示します。

```
$ ipa dnszone-del server.example.com
```

33.4.2. マスター DNS ゾーンの他の設定の追加

IdM は、更新期間、転送設定、キャッシュ設定など、特定のデフォルト設定を指定して新しいゾーンを作成します。

DNS ゾーン設定の属性
利用可能なゾーン設定は 表33.1「ゾーン属性」 に記載されています。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33.1 ゾーン属性

属性	コマンドラインオプション	説明
権威ネームサーバー	**--name-server**	マスター DNS ネームサーバーのドメイン名 (別称: SOA MNAME) を設定します。 デフォルトでは、各 IdM サーバーは SOA MNAME フィールドで自己アドバタイズします。そのため、**--name-server** を使用して LDAP に保存されている値は無視されます。
管理者の電子メールアドレス	**--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**	SOA レコードにシリアル番号を設定します。IdM ではバージョン番号が自動的に設定され、この番号のユーザーによる変更は想定されていません。
SOA refresh	**--refresh**	セカンダリー DNS サーバーがプライマリー 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。

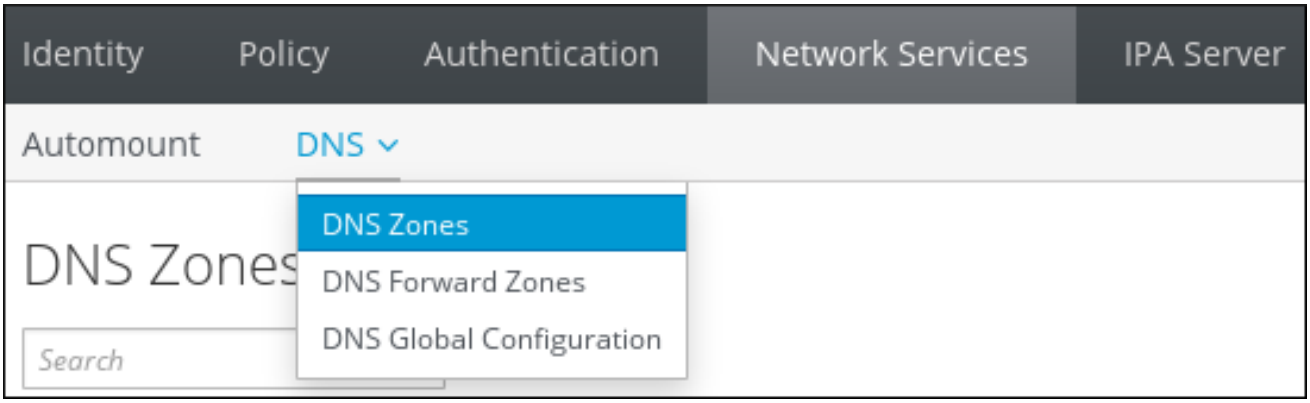
属性	コマンドラインオプション	説明
SOA retry	**--retry**	失敗した更新操作を再試行するまでに待機する時間を秒単位で設定します。
SOA expire	**--expire**	セカンダリー DNS サーバーが操作の試行を終了するまでに、更新操作を実行する時間を秒単位で設定します。
SOA minimum	**--minimum**	RFC 2308 に準拠し、ネガティブキャッシュの TTL (TTL) 値を秒単位で設定します。
SOA time to live	**--ttl**	ゾーン apex のレコードの TTL を秒単位で設定します。たとえば、**example.com** ゾーンでは、名前が **example.com** の全レコード (A、NS または SOA) は設定されますが、**test.example.com** などの他のドメイン名には影響はありません。
デフォルトの TTL	**--default-ttl**	これまでに個別の Time To Live (TTL) 値が設定されたことのないゾーンで、すべての値のネガティブキャッシュのデフォルト TTL を秒単位で設定します。変更を有効にするには、すべての 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 更新ポリシー	**--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。 更新ポリシー構文の詳細は、『BIND 9 管理者リファレンスマニュアル』 のダイナミック更新ポリシー 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。 false に設定すると、IdM クライアントマシンは IP アドレスを追加または更新できなくなる点に注意してください。詳細は、「ダイナミック DNS 更新の有効化」を参照してください。
Allow transfer	**--allow-transfer**=*string*	指定のゾーンを転送できる IP アドレスまたはネットワーク名のセミコロン区切りのリストを指定します。 デフォルトでは、ゾーン転送は無効です。**--allow-transfer** のデフォルト値は **none** です。
Allow query	**--allow-query**	DNS クエリーを発行できる IP アドレスまたはネットワーク名のセミコロン区切りのリストを指定します。
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 forwarder	**--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 ドメインで使用されるグローバル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場愛には、オプションを複数回使用します。
転送ポリシー	**--forward-policy**=none|only|first	転送ポリシーを指定します。サポート対象のポリシーに関する情報は、「フォワードポリシー」を参照してください。

Web UI でのゾーン設定編集

Web UI から DNS マスターゾーンを管理するには、**ネットワークサービス** タブを開いて、**DNS** サブタブを選択し、その後に **DNS ゾーン** セクションを選択します。

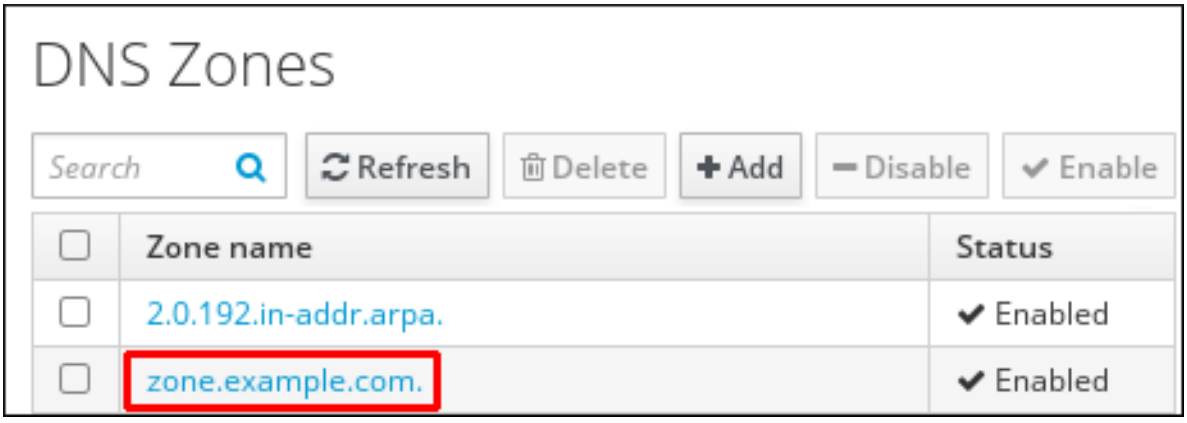
図33.5 DNS マスターゾーンの管理



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、以下を行います。

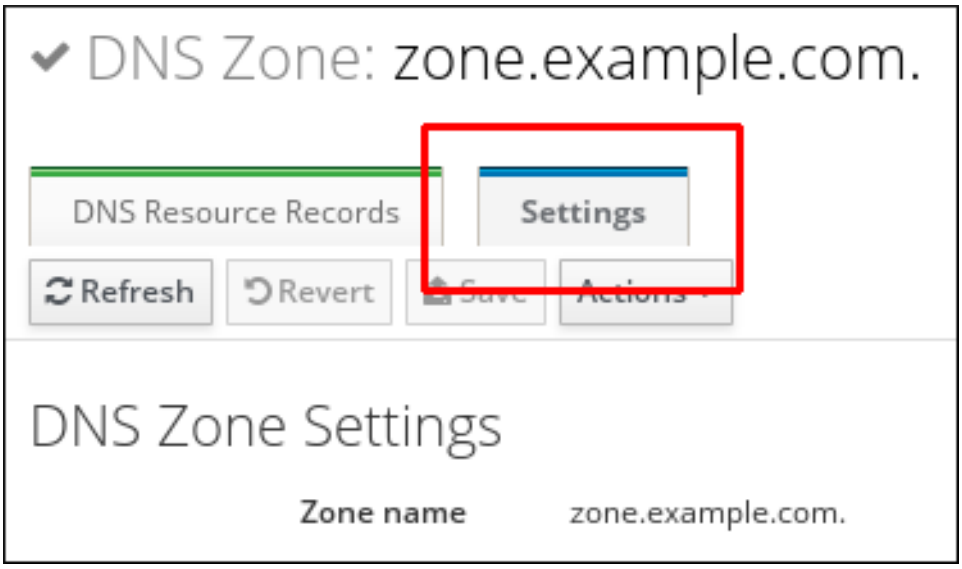
1. ゾーンの全リストからゾーン名をクリックして DNS ゾーンページを開きます。

   図33.6 マスターゾーンの編集



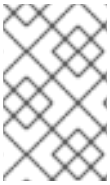
2. **設定** をクリックし、必要に応じてゾーン設定を変更します。

図33.7 マスターゾーン編集ページの Settings タブ



利用可能な設定の詳細は、表33.1「ゾーン属性」 を参照してください。

3. **Save** をクリックして、新しい設定を確定します。



注記

ゾーンのデフォルトの Time To Live (TTL) を変更する場愛には、全 IdM DNS サーバーで**named-pkcs11** サービスを再起動して、変更を適用します。他の全設定は、すぐに自動的に有効になります。

コマンドラインでのゾーン設定の編集
コマンドラインから既存のマスター DNS ゾーンを変更する場合は、**ipa dnszone-mod** コマンドを使用します。利用可能な設定の詳細は、表33.1「ゾーン属性」 を参照してください。

DNS ゾーンエントリーに属性が存在しない場合は、**ipa dnszone-mod** コマンドにより属性が追加されます。属性が存在する場合は、このコマンドは現在の値を指定された値で上書きします。

ipa dnszone-mod とそのオプションの詳細は、**ipa dnszone-mod --help** コマンドを実行します。

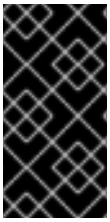
注記

ゾーンのデフォルトの Time To Live (TTL) を変更する場愛には、全 IdM DNS サーバーで**named-pkcs11** サービスを再起動して、変更を適用します。他の全設定は、すぐに自動的に有効になります。

33.4.3. ゾーン転送の有効化

ネームサーバーはゾーンの権威データを維持します。ゾーンに変更が加えられるため、DNS ドメインのネームサーバーに送信および配布される必要があります。 *ゾーン転送* では、別のサーバーにリソースレコードがすべてコピーします。

IdM は、RFC 5936 (AXFR) および RFC 1995 (IXFR) 標準に準拠するゾーン転送をサポートします。



重要

IdM が統合された DNS はマルチマスターです。IdM ゾーンの SOA シリアル番号は、IdM サーバー間で同期されません。このため、DNS スレーブサーバーが IdM マスターサーバーを 1 台だけ使用するように設定します。こうすることで、同期されていない SOA シリアル番号が原因でゾーン転送が失敗しないようにします。

UI でのゾーン転送の有効化

「Web UI でのゾーン設定編集」 の説明に従って DNS ゾーンページを開き、**設定** タブに切り替えます。

Allow transfer で、ゾーンレコードを転送するネームサーバーを指定します。

図33.8 ゾーン転送の有効化

DNS ゾーンページの上部にある **Save** をクリックして、新しい設定を確定します。

コマンドラインでのゾーン転送の有効化

コマンドラインからゾーン転送を有効にするには、**ipa dnszone-mod** コマンドに **--allow-transfer** オプションを追加します。**--allow-transfer** を使用して、ゾーンレコードを転送するネームサーバーのリストを指定します。以下に例を示します。

```
[user@server ~]$ ipa dnszone-mod --allow-transfer="192.0.2.1;198.51.100.1;203.0.113.1"
example.com
```

BIND サービスでゾーン転送を有効にすると、IdM DNS ゾーンは、**dig** ユーティリティーなどのクライアントから名前を使用して転送できます。

```
[root@server ~]# dig @ipa-server zone_name AXFR
```

33.4.4. DNS ゾーンへのレコードの追加

IdM は、さまざまなレコードタイプに対応します。以下の 4 つが最も頻繁に使用されます。

A

これは、ホスト名および通常の IPv4 アドレスの基本マップです。A レコードのレコード名は、**www** などのホスト名です。A レコードの **IP アドレス** 値は、**192.0.2.1** などの IPv4 アドレスです。

A レコードの詳細は、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 アドレス** は、**2001: DB8::1111** など、標準の 16 進数の IPv6 アドレスです。

AAAA レコードの詳細は RFC 3596 を参照してください。

SRV

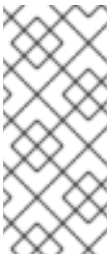
*サービス (SRV) リソースレコード*は、サービス名を、その特定サービスを提供するサーバーの DNS 名にマッピングします。たとえば、このタイプのレコードは LDAP ディレクトリーのようなサービスを管理するサーバーに、このサービスをマッピングします。

SRV レコードのレコード名は、**_ldap._tcp** など、***_service._protocol*** の形式を取ります。SRV レコードの設定オプションには、ターゲットサービスの優先順位、加重、ポート番号、およびホスト名が含まれます。

SRV レコードの詳細は、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。人間が判別可能な形式の逆アドレスは、通常の IP とまったく逆で、**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 で拡張) で指定の標準形式を使用する必要があります。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33.8「PTR レコード」 を参照してください。

注記

また、IPv6 アドレスの逆引きゾーンは、**.ip6.arpa.** ドメインのゾーンを使用して設定できます。IPv6 逆引きゾーンの詳細は、RFC 3596 を参照してください。

DNS リソースレコードの追加時には、レコードの多くで異なるデータが必要になることに注意してください。たとえば、CNAME レコードにはホスト名が必要ですが、A レコードには IP アドレスが必要です。Web UI では、新しいレコードを追加するフォームのフィールドが自動的に更新され、現在選択されているレコードタイプに必要なデータが反映されます。

DNS ワイルドカードのサポート

IdM は、ワイルドカードとして DNS ゾーン内の特別なレコード * に対応します。

例33.2 DNS ワイルドカードの結果のデモンストレーション

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 レコードですが、このホストの A レコードはありません。

- **demo.example.com** のレコードがありません。

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します。以下の結果が返されます。

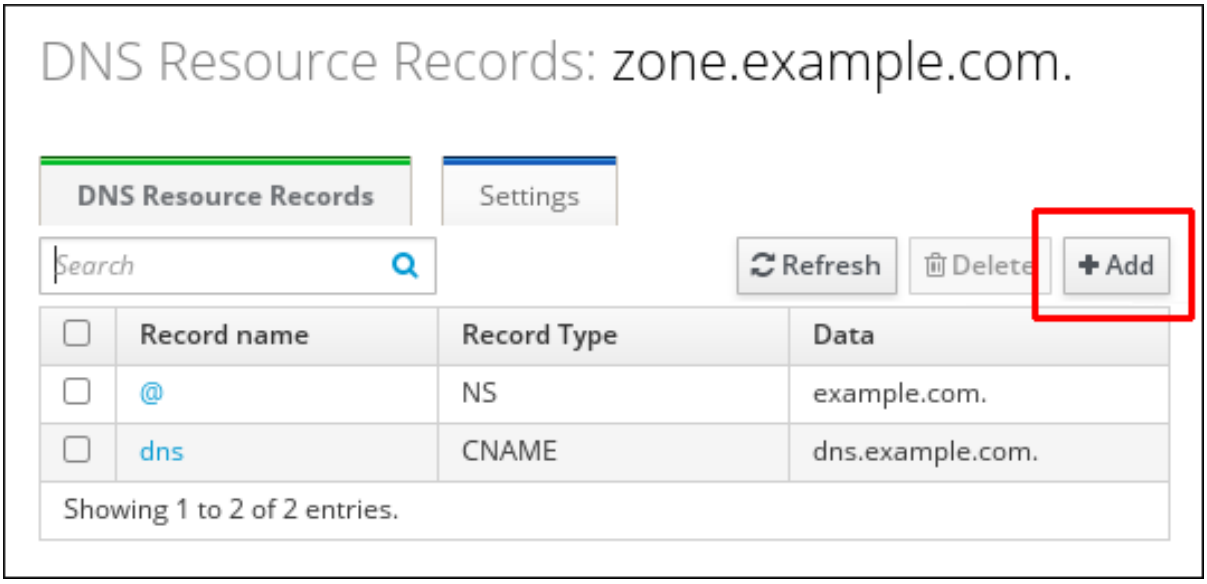
```
# host -t MX mail.example.com.
mail.example.com mail is handled by 10 server.example.com.

# 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

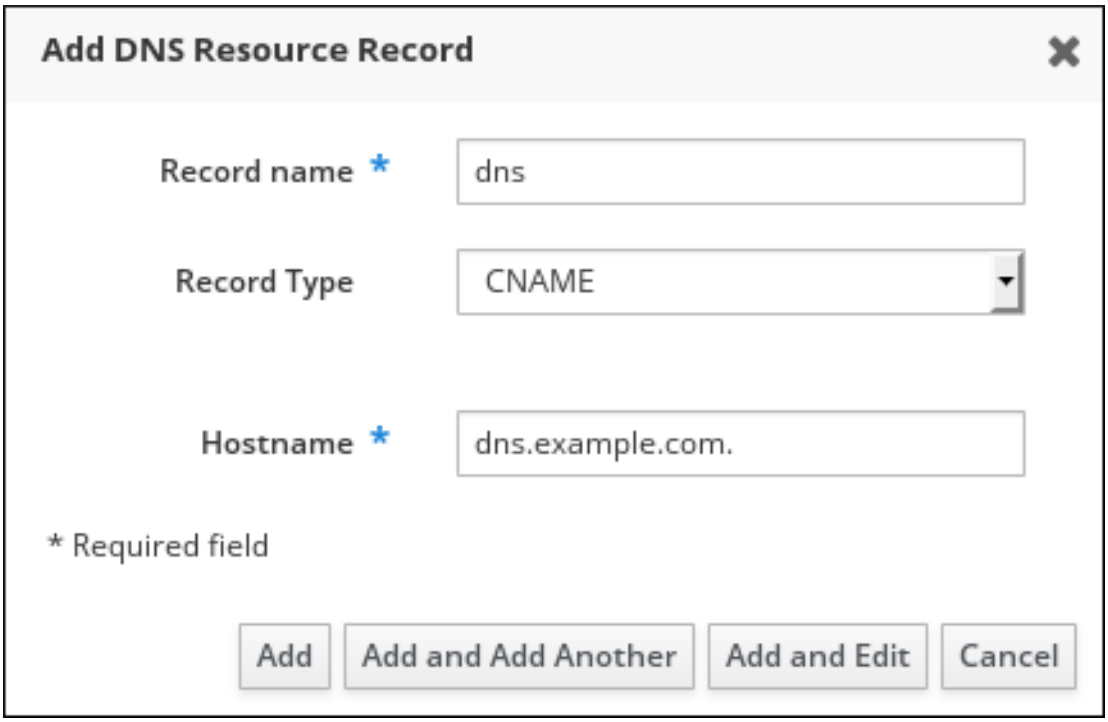
1. 「Web UI でのゾーン設定編集」 の説明に従って、DNS ゾーンページを開きます。
2. **DNS Resource Record** セクションで、**Add** をクリックして新規レコードを追加します。

図33.9 新しい DNS リソースレコードの追加



3. 作成するレコードのタイプを選択し、必要に応じて他のフィールドにも入力します。

図33.10 新しい DNS リソースレコードの定義



4. **Add** をクリックして、新規レコードを確定します。

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任意の種類の DNS リソースレコードを追加するには、**ipa dnsrecord-add** コマンドを使用します。このコマンドは、以下の構文に従います。

```
$ 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、新しい DNS リソースレコードの識別子です。

表33.2「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、同じコマンド呼び出しでオプションを複数回使用するか、**--option={val1,val2,val3}**のように、中括弧内のコンマ区切りリストにオプションをリストすることで設定できます。

ipa dnsrecord-add の使用方法および IdM で対応している DNS レコードタイプに関する詳細は、 **ipa dnsrecord-add --help** コマンドを実行します。

表33.2 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=AAAARECORD	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指定します。

"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 (.) として指定される場合があります。

33.4.5. コマンドラインから DNS リソースレコードを追加または変更する例

例33.3 IPv4 レコードの追加

以下の例では、**192.0.2.123**のアドレスを含めて、**www.example.com** レコードを作成します。

```
$ ipa dnsrecord-add example.com www --a-rec 192.0.2.123
```

例33.4 IPv4 ワイルドカードレコードの追加

この例では、IP アドレスが **192.0.2.123** のワイルドカード A レコードを作成します。

```
$ ipa dnsrecord-add example.com "*" --a-rec 192.0.2.123
```

例33.5 IPv4 レコードの変更

レコードの作成時に、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 レコードを変更する時に、**--a-record** オプションを使用して A 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33.6 IPv6 レコードの追加

以下の例では、**2001: db8::1231:5675**の IP アドレスを含めて、**www.example.com** レコードを作成します。

```
$ ipa dnsrecord-add example.com www --aaaa-rec 2001:db8::1231:5675
```

例33.7 SRV レコードの追加

以下の例では、**_ldap._tcp** は、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** オプションは、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。

以下に例を示します。

```
[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."
```

加重値 (この例では **51** と **49**) が最大 100 まで加算され、特定のレコードが使用される確率を % で表します。

例33.8 PTR レコード

逆引き DNS レコードを追加する時には、他の DNS レコードの追加の方法と比べ、**ipa dnsrecord-add** コマンドで使用するゾーン名は、逆になります。

```
$ ipa dnsrecord-add reverseNetworkIpAddress hostIpAddress --ptr-rec FQDN
```

通常、*hostIpAddress* は、指定のネットワークにおける IP アドレスの最後のオクテットです。

たとえば、IPv4 アドレスが 192.0.2.4 の**server4.example.com** に PTR レコードを追加します。

```
$ ipa dnsrecord-add 2.0.192.in-addr.arpa 4 --ptr-rec server4.example.com.
```

次の例では、**0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa** に逆引き DNS エントリーを追加します。IP アドレスが **2001:DB8::1111** の **server2.example.com** ホストの IPv 6 逆引きゾーン。

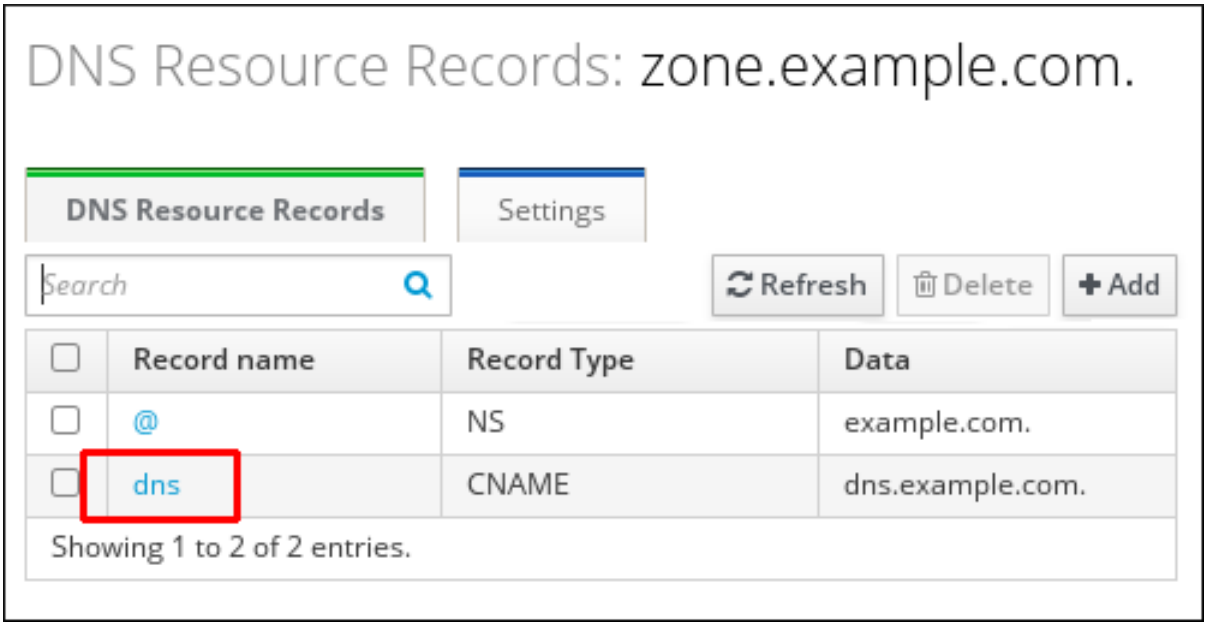
```
$ ipa dnsrecord-add 0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa. 1.1.1.0.0.0.0.0.0.0.0.0.0.0.0.0 --ptr-rec server2.example.com.
```

33.4.6. DNS ゾーンからレコードを削除する

Web UI でのレコードの削除
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。

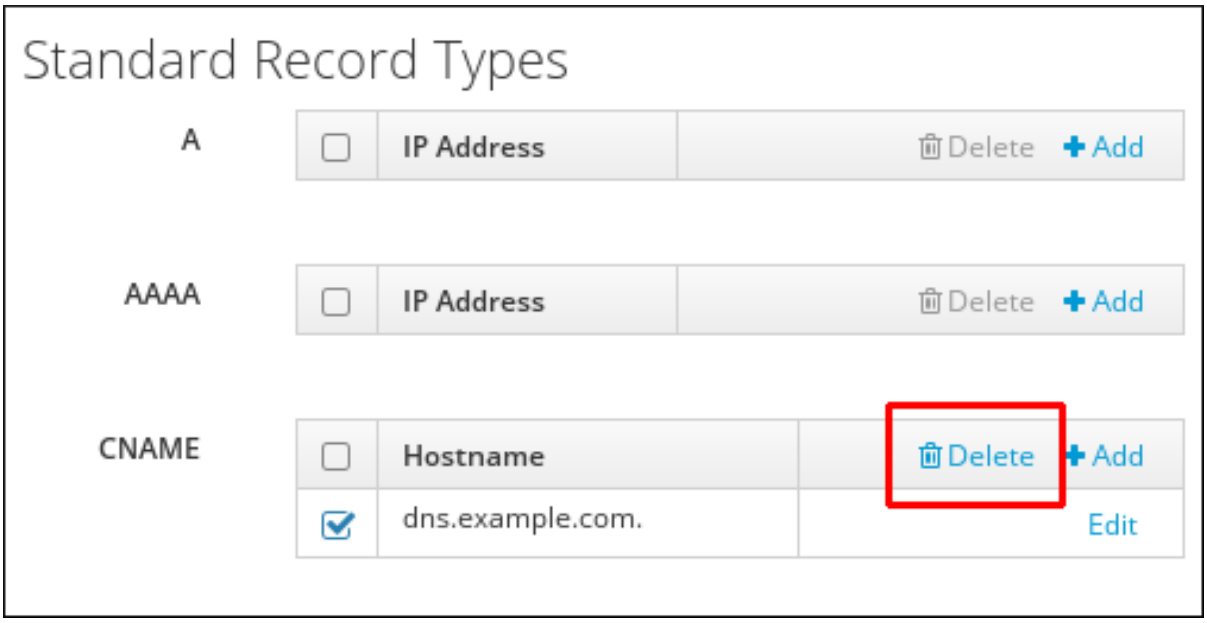
1. 「Web UI でのゾーン設定編集」 の説明に従って、DNS ゾーンページを開きます。
2. **DNS Resource Record** のセクションで、リソースレコードの名前をクリックします。

図33.11 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の横にあるチェックボックスを選択します。

図33.12 DNS リソースレコードの削除

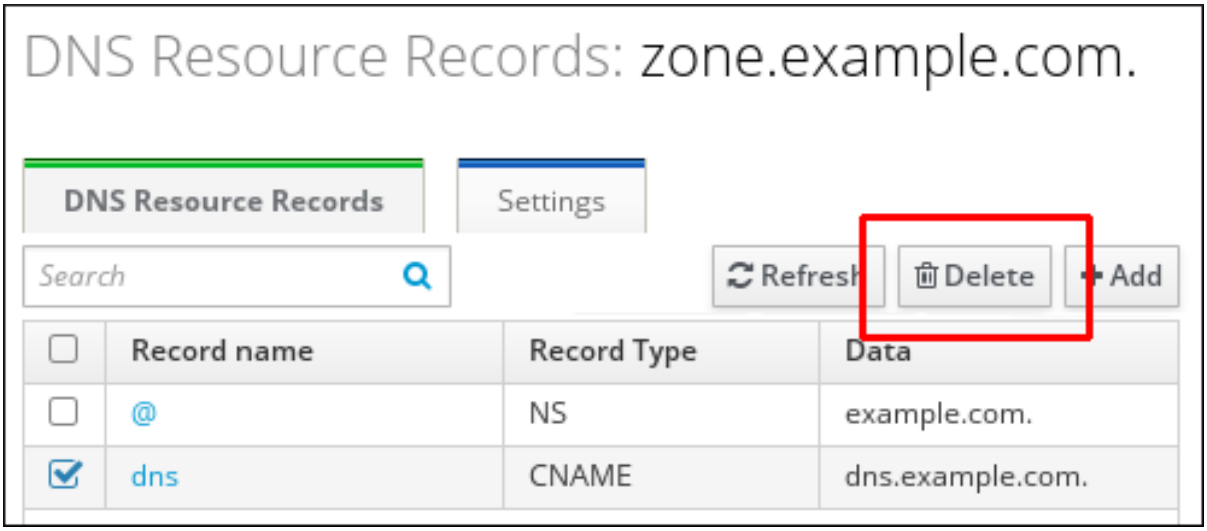


この後、選択したレコードタイプのみが削除され、その他の設定はそのまま残ります。

ゾーン内のリソースのレコードをすべて削除するには、次のコマンドを実行します。

1. 「Web UI でのゾーン設定編集」 の説明に従って、DNS ゾーンページを開きます。
2. **DNS リソースレコード** セクションで、削除するリソースレコード名のチェックボックスを選択し、ゾーンレコードリストの上部にある **削除** をクリックします。

図33.13 全リソースレコードの削除



この後、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の削除

ゾーンからレコードを削除するには **ipa dnsrecord-del** コマンドを使用して、***--recordType*-rec** オプションでレコードの値を指定して追加します。

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** コマンドを実行すると、削除するレコードについての情報の入力が求められます。**--del-all** オプションを指定してコマンドを実行すると、ゾーンに関連するレコードがすべて削除されることに注意してください。

ipa dnsrecord-del の使用方法と、このコマンドで使用できるオプションの全リストに関する詳細は、**ipa dnsrecord-del --help** コマンドを実行します。

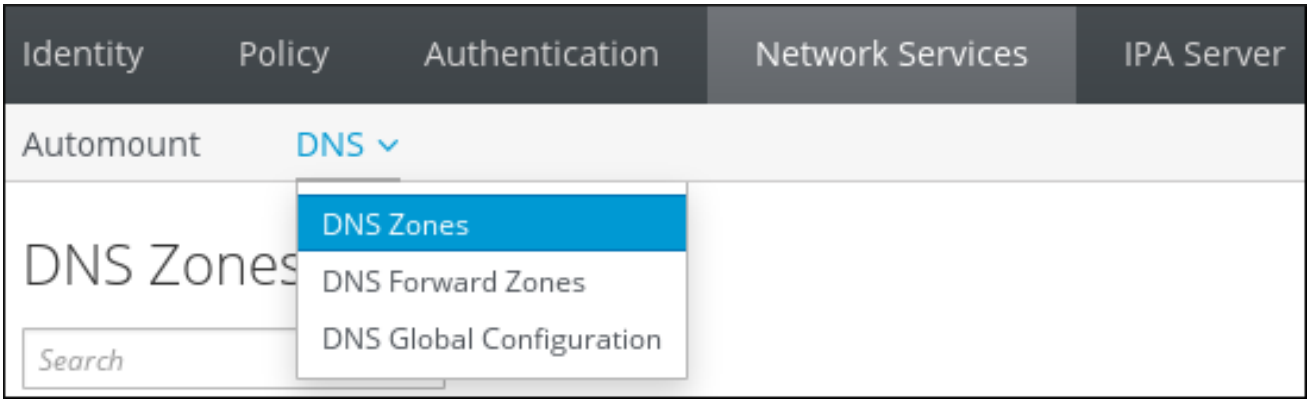
33.4.7. ゾーンの無効化と有効化

IdM を使用すると、管理者による DNS ゾーンの無効化/有効化が可能です。「マスター DNS ゾーンの削除」 で説明されているように DNS ゾーンを削除すると、ゾーンエントリーと関連するすべての設定が完全に削除されます。ゾーンを無効にすると、IdM からそのゾーンを完全に削除することなく、そのゾーンがアクティビティーから削除されます。無効ゾーンは再度有効にすることもできます。

Web UI でのゾーンの無効化と有効化

Web UI から DNS ゾーンを管理するには、**ネットワークサービス** タブを開き、**DNS** サブタブを選択してから、**DNS ゾーン** セクションを選択します。

図33.14 DNS ゾーンの管理



ゾーンを無効にするには、ゾーン名の横にあるチェックボックスを選択し、**Disable** をクリックします。

図33.15 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、ゾーン名の横にあるチェックボックスを選択し、**有効** をクリックします。

コマンドラインからの DNS ゾーンの無効化と有効化

コマンドラインから DNS ゾーンを無効にするには、**ipa dnszone-disable** コマンドを使用します。以下に例を示します。

```
[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** コマンドを使用します。

33.5. 動的 DNS 更新の管理

33.5.1. ダイナミック DNS 更新の有効化

動的 DNS 更新は、IdM の新規 DNS ゾーンに対してデフォルトでは無効となっています。動的更新を無効にすると、**ipa-client-install** スクリプトでは、新規クライアントを指定する DNS レコードを追加できません。

注記

動的更新を有効にすると、セキュリティーリスクが発生する可能性があります。ただし、使用環境で動的更新が可能である場合には、この更新方法を使用すると、クライアントのインストールが簡素化されます。

動的更新を有効にするには、以下が必要です。

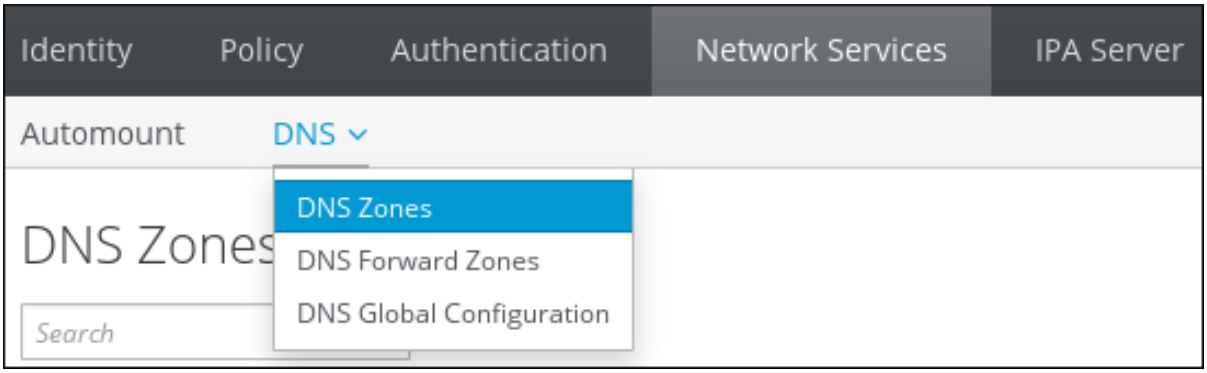
- 動的更新を許可するように DNS ゾーンを設定する必要があります。
- ローカルクライアントは、動的更新を送信するように設定する必要があります。

33.5.1.1. 動的更新を許可するための DNS ゾーンの設定

Web UI での動的 DNS 更新の有効化

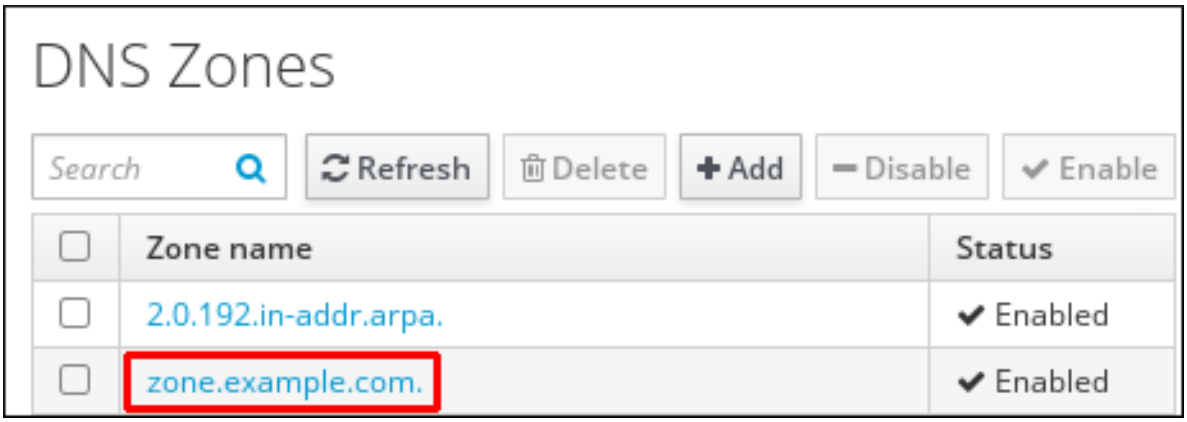
1. **Network Services** タブを開き、**DNS** サブタブを選択し、その後に **DNS Zones** セクションを選択します。

   図33.16 DNS ゾーンの管理

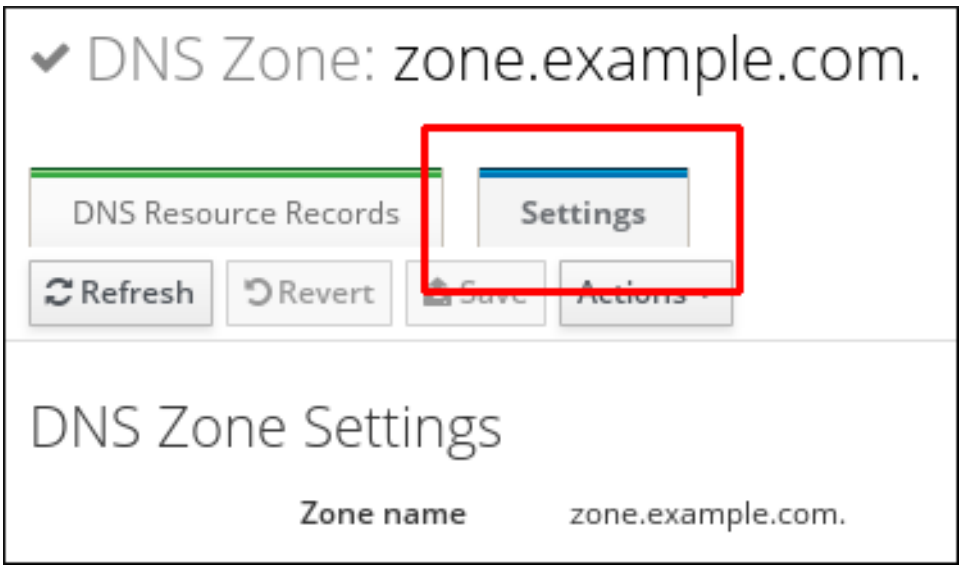
2. ゾーンの全リストからゾーン名をクリックして DNS ゾーンページを開きます。

図33.17 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33.18 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33.19 ダイナミック DNS 更新の有効化

5. ページ上部の **Save** をクリックして、新しい設定を確認します。

コマンドラインでの動的 DNS 更新の有効化

コマンドラインから DNS ゾーンへの動的な更新を可能にするには、**--dynamic-update=TRUE** オプションを指定して **ipa dnszone-mod** コマンドを使用します。以下に例を示します。

```
[user@server ~]$ ipa dnszone-mod server.example.com --dynamic-update=TRUE
```

33.5.1.2. 動的更新を送信するためのクライアントの設定

クライアントは、**ipa-client-install** スクリプトで **--enable-dns-updates** を使用して、ドメインに登録すると DNS 更新を送信するように自動的に設定されます。

```
[root@client ~]# ipa-client-install --enable-dns-updates
```

DNS ゾーンの SOA 設定には、レコードに設定された TTL (Time to Live) 値があります。ただし、動的更新の TTL は、System Security Service Daemon (SSSD) によりローカルシステムで管理されます。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。デフォルトは 1200 秒です。

1. SSSD 設定ファイルを開きます。

   ```
   [root@server ~]# vim /etc/sssd/sssd.conf
   ```

2. IdM ドメインのドメインセクションを検索します。

   ```
   [domain/ipa.example.com]
   ```

3. 動的更新がクライアントで有効になっていない場合は、***dyndns_update*** を true に設定します。

   ```
   dyndns_update = true
   ```

4. ***dyndns_ttl*** パラメーターを追加または変更して、値を秒単位で設定します。

   ```
   dyndns_ttl = 2400
   ```

33.5.2. A/AAAA レコードと PTR レコードの同期

AAA レコードと AAA レコードは、逆引きゾーンで PTR レコードとは別に設定されます。これらのレコードは個別に設定されるため、対応する PTR レコードがない場合は A/AAAA レコードが存在し、その逆も可能です。

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、「ダイナミック DNS 更新の有効化」で説明されています。
- マスターの正引きゾーンおよび逆引きゾーンでは、PTR 同期を有効にする必要があります。
- 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

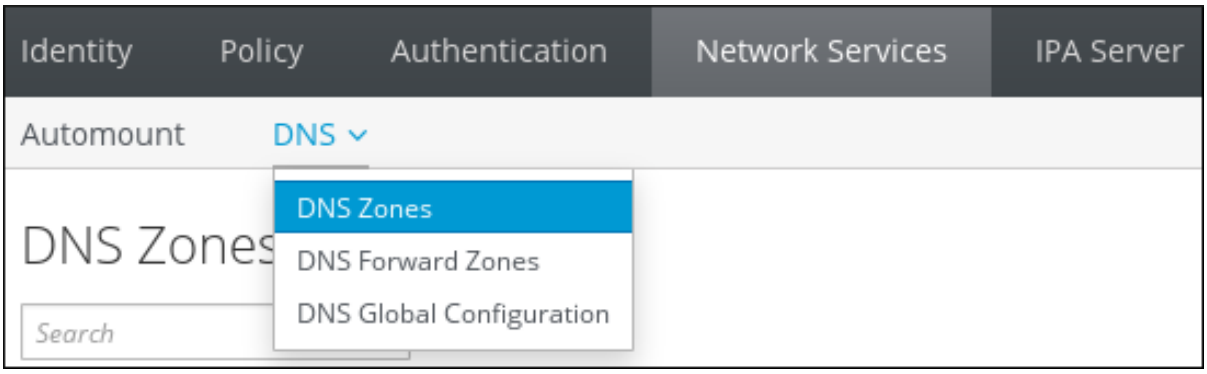
クライアントシステムは、自身の IP アドレスを更新できます。つまり、危険にさらされたクライアントを使用して IP アドレスを変更すると、PTR レコードの上書きが可能になります。

33.5.2.1. Web UI での PTR レコードの同期設定

PTR レコードの同期は、PTR レコードが存在する逆引き DNS ゾーンではなく、A レコードまたは 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ります。

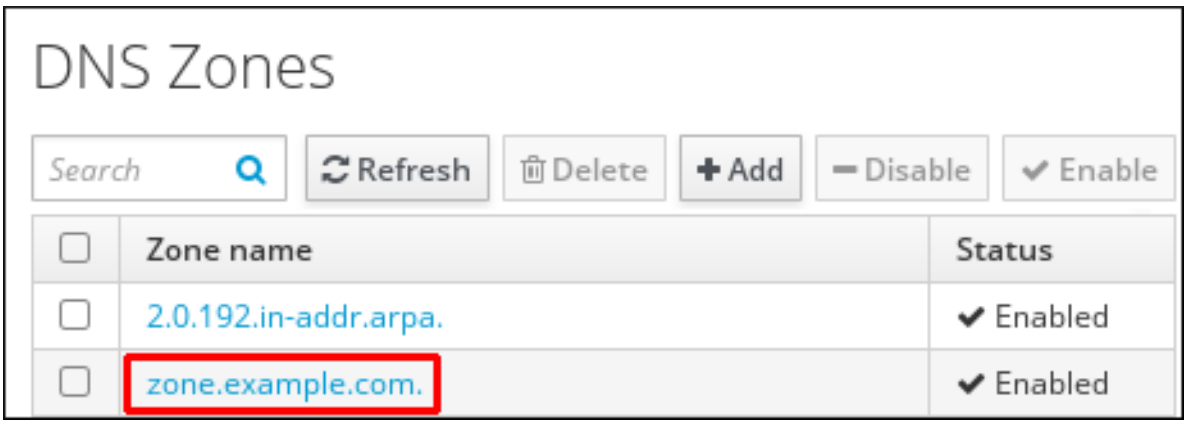
1. **Network Services** タブを開き、**DNS** サブタブを選択し、その後に **DNS Zones** セクションを選択します。

図33.20 DNS ゾーンの管理



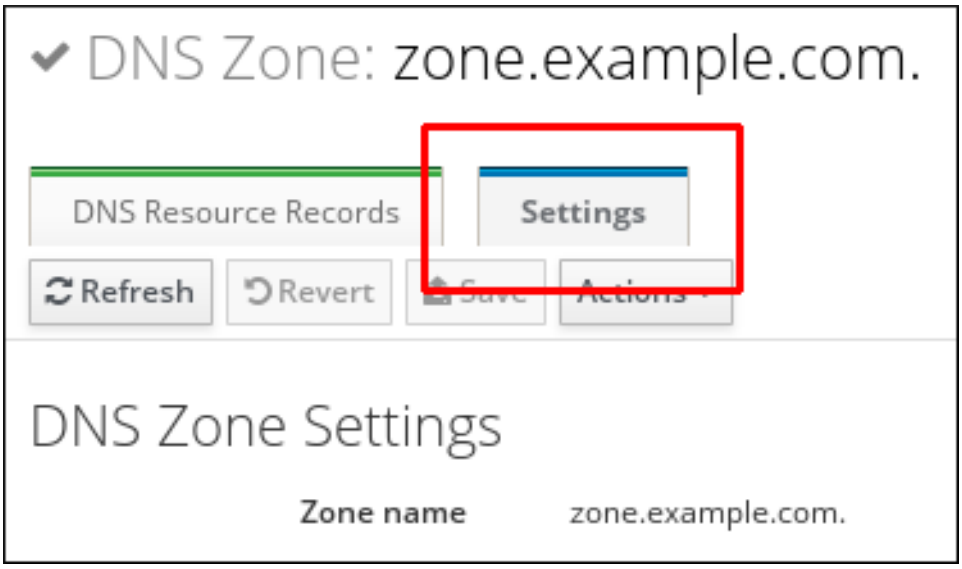
2. ゾーンの全リストからゾーン名をクリックして DNS ゾーンページを開きます。

図33.21 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33.22 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33.23 PTR 同期の有効化

5. ページ上部の **Save** をクリックして、新しい設定を確認します。

33.5.2.2. コマンドラインを使用した PTR レコードの同期設定

コマンドラインを使用して、特定のゾーン、またはすべてのゾーンに対してグローバルに PTR レコードの同期を設定できます。

33.5.2.2.1. 特定のゾーンでの PTR レコードの同期設定

たとえば、**idm.example.com** 正引きゾーンに PTR レコード同期を設定するには、次のコマンドを実行します。

1. 正引きゾーンの動的更新を有効にします。

```
# ipa dnszone-mod idm.example.com. --dynamic-update=TRUE
```

2. 正引きゾーンの更新ポリシーを設定します。

```
# ipa dnszone-mod idm.example.com. --update-policy='grant IDM.EXAMPLE.COM krb5-self
* A; grant IDM.EXAMPLE.COM krb5-self * AAAA; grant IDM.EXAMPLE.COM krb5-self *
SSHFP;'
```

3. 正引きゾーンの PTR レコード同期を有効にします。

```
# ipa dnszone-mod idm.example.com. --allow-sync-ptr=True
```

4. 逆引きゾーンの動的更新を有効にします。

```
# ipa dnszone-mod 2.0.192.in-addr.arpa. --dynamic-update=TRUE
```

33.5.2.2.2. すべてのゾーンでの PTR レコードの同期のグローバル設定

以下の手順のいずれかを使用して、IdM が管理するすべてのゾーンで PTR 同期を有効にできます。

- すべてのサーバーのゾーンで PTR 同期を同時に有効にするには、次のコマンドを実行します。

```
# ipa dnsconfig-mod --allow-sync-ptr=true
```

- サーバーごとの同期を有効にするには、次のコマンドを実行します。

1. **/etc/named.conf** 内の **dyndb "ipa" "/usr/lib64/bind/ldap.so"** に、***sync_ptr yes;*** 設定を追加します。

```
dyndb "ipa" "/usr/lib64/bind/ldap.so" {
  ...
  sync_ptr yes;
};
```

2. IdM を再起動します。

```
# ipactl restart
```

3. DNS サービスがインストールされている各 IdM サーバーでこの手順を繰り返します。

33.5.3. DNS 動的更新ポリシーの更新

IdM サーバーが管理する DNS ドメインは、RFC 3007 に従って DNS 動的更新を受け入れることができます。[4].

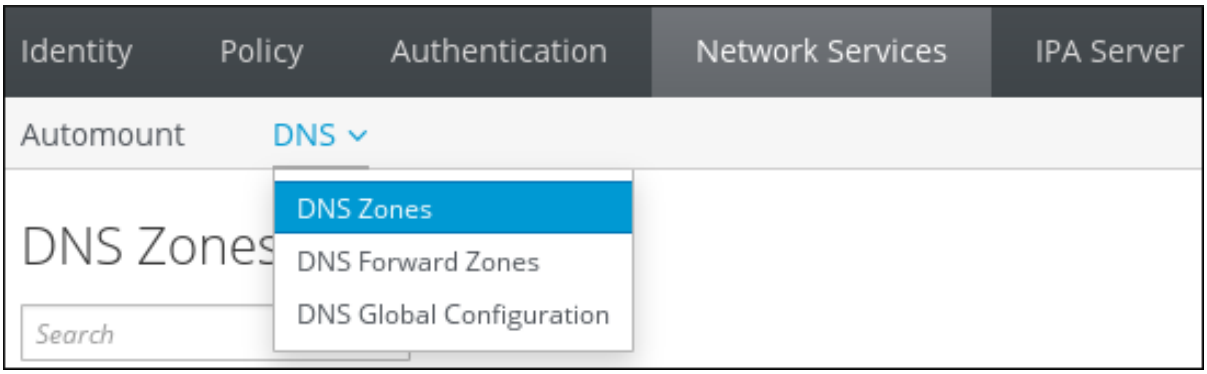
特定のクライアントが修正可能なレコードを判断するルールは、**/etc/named.conf** ファイルの **更新ポリシー** 文と同じ構文に従います。動的更新ポリシーの詳細は、BIND 9 のドキュメント を参照してください。

DNS ゾーンで動的 DNS 更新が無効になっていると、動的更新ポリシーステートメントを反映せずに、すべての DNS 更新が拒否されることに注意してください。動的 DNS 更新を有効にする方法は、「ダイナミック DNS 更新の有効化」 を参照してください。

Web UI での DNS 更新ポリシーの更新

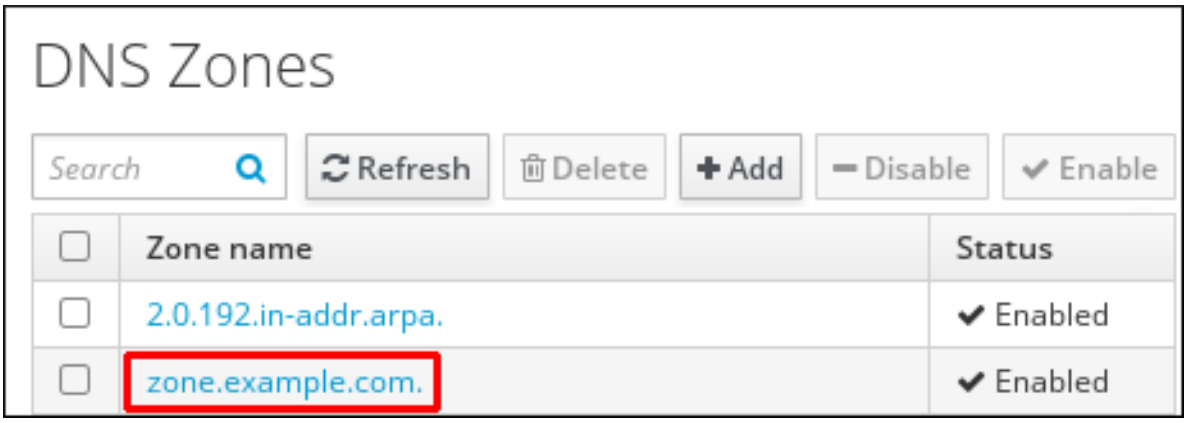
1. **Network Services** タブを開き、**DNS** サブタブを選択し、その後に **DNS Zones** セクションを選択します。

図33.24 DNS ゾーンの管理



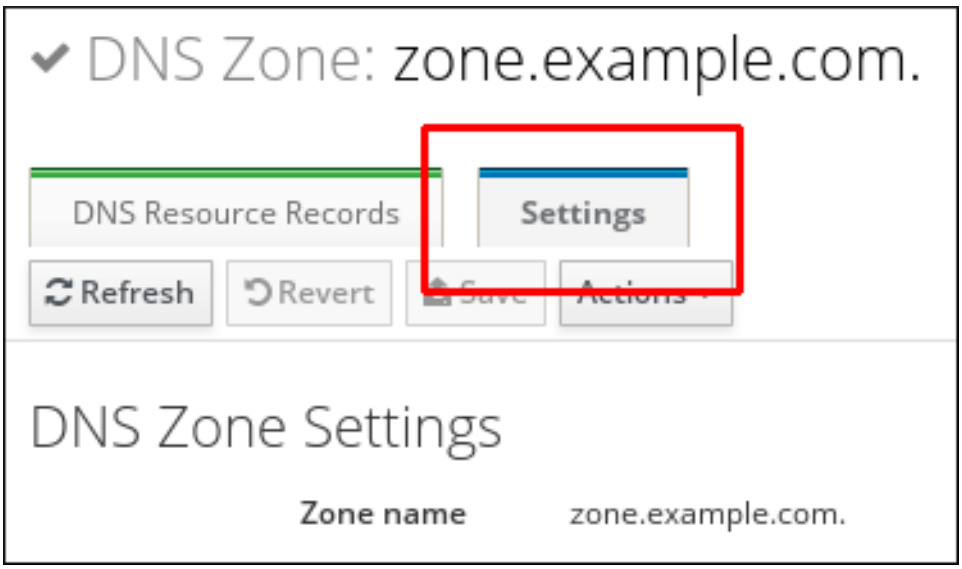
2. ゾーンの全リストからゾーン名をクリックして DNS ゾーンページを開きます。

図33.25 DNS ゾーンの編集



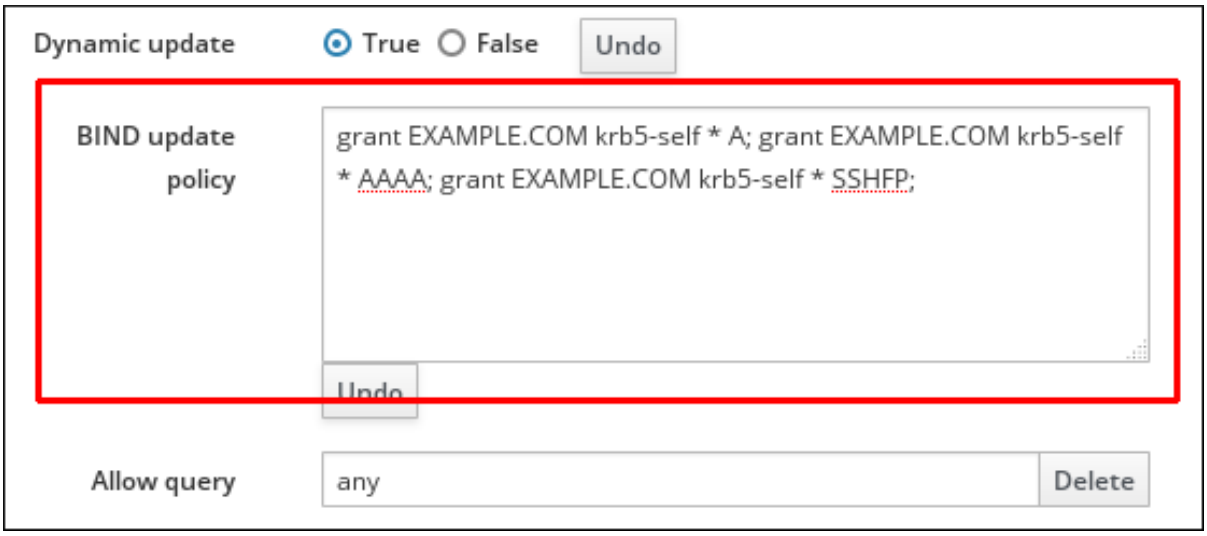
3. **Settings** をクリックして DNS ゾーン設定タブに切り替えます。

図33.26 マスターゾーン編集ページの Settings タブ



4. **BIND 更新ポリシー** テキストボックスに、セミコロンで区切ったリストで必要な更新ポリシーを設定します。

図33.27 DNS 更新ポリシーの設定



5. DNS ゾーンページの上部にある **Save** をクリックして、新しい設定を確定します。

コマンドラインでの 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、そのオプションの後にくる文でアクセス制御ルールを追加します。以下に例を示します。

```
$ 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;"
```

33.6. DNS 転送の管理

DNS 転送は、DNS クエリーへの応答に影響を及ぼします。デフォルトでは、IdM と統合された BIND サービスは、信頼できる DNS サーバーおよび再帰的な DNS サーバーの両方として機能するように設定されています。

IdM サーバーが権威のある DNS ゾーンに所属する名前のクエリーを DNS クライアントが出した場合に、BIND は設定済みのゾーンに含まれ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 サーバーが権威のない名前のクエリーを DNS クライアントが出した場合に、BIND は他の DNS サーバーを使用し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 がインターネット上のルートサーバーにクエリーを出し、再帰解決アルゴリズムを使用して DNS クエリーに応答します。

BIND を使用して他の DNS サーバーに直接問い合わせて、インターネットで利用可能なデータをもとに再帰を実行することは推奨されません。ユースケースには以下が含まれます。

- *分割 DNS* 設定 (*DNS ビュー* 設定 とも呼ばれます)。DNS サーバーがさまざまなクライアントに異なる応答を返す設定です。スプリット DNS 設定とは一般的に、会社のネットワーク内で一部の DNS 名が利用できますが、外部からは利用できない環境です。
- ファイアウォールがインターネット上の DNS へのアクセスを制限する設定。
- DNS レベルでの集中フィルタリングまたはロギングを使用した設定
- ローカルの DNS キャッシュへの転送を使用した設定。これにより、ネットワークトラフィックが最適化されます。

このような設定では、BIND はパブリックインターネットで完全再帰を使用しません。代わりに、別の DNS サーバー (*フォワーダー* と呼ばれます) を使用してクエリーを解決します。フォワーダーを使用するように BIND を設定すると、クエリーと応答が IdM サーバーとフォワーダーの間で送受信され、IdM サーバーが権威データ以外の DNS キャッシュとして機能します。

フォワードポリシー

IdM は、*first* および *only* の BIND 転送ポリシーと、IdM 固有の転送ポリシー *none* をサポートします。

forward first (デフォルト)

DNS クエリーは設定済みの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が原因でクエリーに失敗すると、BIND はインターネット上のサーバーを使用して再帰解決にフォールバックします。Forward first ポリシーはデフォルトのポリシーです。トラフィックの最適化に適しています。

Forward only

DNS クエリーは設定済みの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が原因でクエリーに失敗すると、BIND はエラーをクライアントに返します。分割された DNS 設定の環境では、forward only ポリシーが推奨されます。

None: 転送の無効化

DNS クエリーは転送されません。グローバル転送設定をゾーン別にオーバーライドする場合にのみ、転送の無効化は有用です。このオプションは、IdM の BIND 設定で空のフォワーダーリストを指定するのと同じです。

転送で IdM サーバーおよびその他の DNS サーバーのデータが結合されない

転送を使用して、IdM のデータを、その他の DNS サーバーのデータと組み合わせることはできません。IdM DNS のマスターゾーン内にある特定のサブゾーンのクエリーのみを転送できます。「IdM DNS マスターゾーンのゾーン委譲」を参照してください。

デフォルトでは、IdM サーバーが権威サーバーとなっているゾーンに、クエリーされた DNS 名が所属する場合には、BIND サービスは、クエリーを別のサーバーに転送しません。このような場合は、クエリーされた DNS 名が IdM データベースに見つからない場合は、**NXDOMAIN** との応答が返されます。転送は使用されません。

> **例33.9 サンプルシナリオ**
>
> IdM サーバーは、**test.example** の権威サーバーです。DNS ゾーン。BIND は、IP アドレス **192.0.2.254** でクエリーを DNS サーバーに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** のクエリーを送信する場合DNS 名である BIND は、IdM サーバーが **test.example.** ゾーンの権威サーバーであることを検出して、クエリーを **192.0.2.254.** サーバーには転送しません。その結果、DNS クライアントは **NXDomain** の応答を受け取り、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通知します。

IdM DNS マスターゾーンのゾーン委譲

IdM DNS では、マスターゾーンの特定のサブゾーンに対するクエリーを転送できます。たとえば、IdM DNS がゾーン**idm.example.com**を処理する場合は、*sub_zone1*.**idm.example.com** サブゾーンの権限を別の DNS サーバーに委譲できます。この動作を実現するには、上述のように転送を使用する必要があります。また、サブゾーンを別の DNS サーバーに委譲するネームサーバーレコードも使用する必要があります。以下の例では、*sub_zone1* がサブゾーンで、**192.0.2.1** は、サブゾーンが委譲される DNS サーバーの IP アドレスです。

```
$ ipa dnsrecord-add idm.example.com. sub_zone1 --ns-rec=192.0.2.1
```

正引きゾーンを追加すると、以下のようになります。

```
$ ipa dnsforwardzone-add  sub_zone1.idm.example.com. --forwarder 192.0.2.1
```

33.6.1. 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」で説明されているように、IdM サーバーに権限がない場合の全 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、次の 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できます。

ipa dnsconfig-mod コマンドまたは IdM Web UI の使用

このネイティブの IdM ツールを使用して指定した設定は、すべての IdM DNS サーバーにすぐに適用されます。「DNS 設定の優先順位」 で説明しているように、グローバル DNS 設定は、**/etc/named.conf** ファイルで定義されているローカル設定よりも優先されます。

/etc/named.conf ファイルの編集により

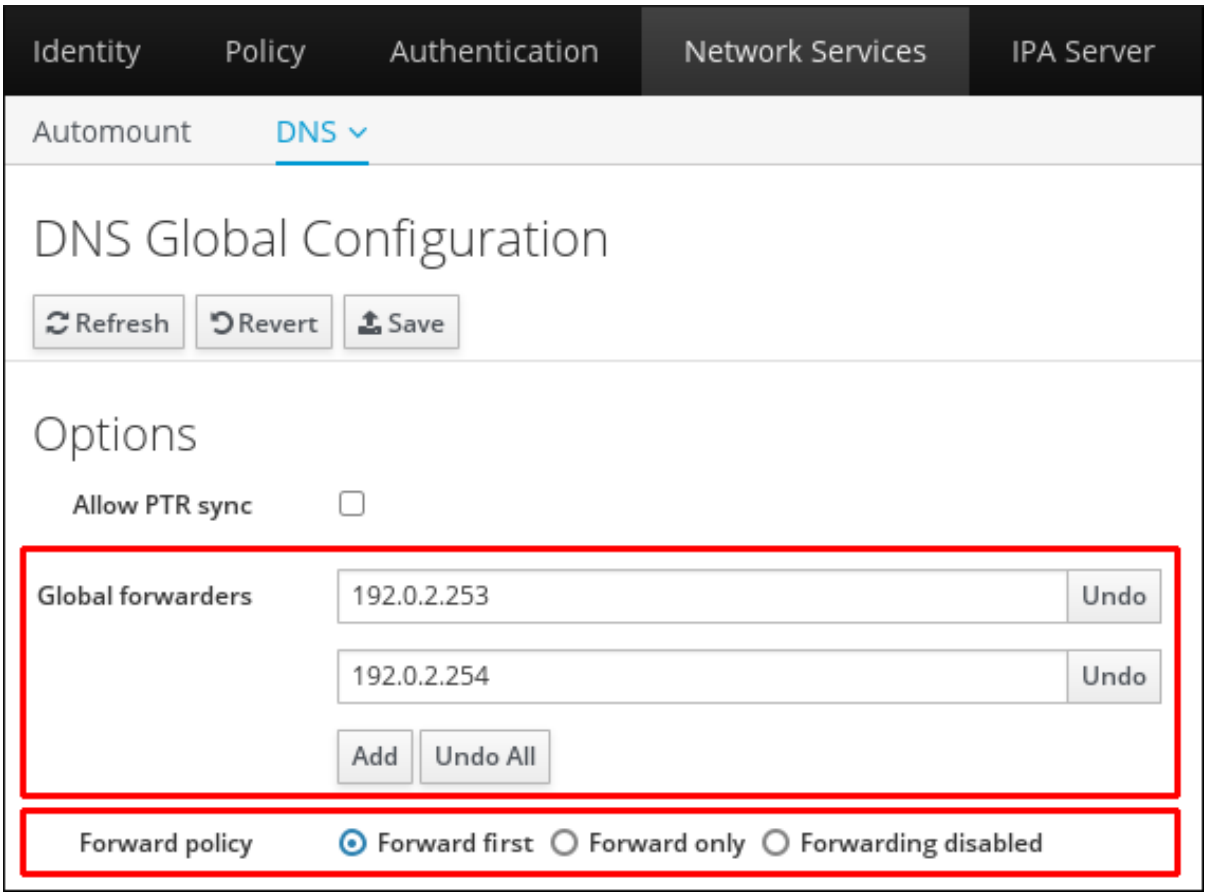
すべての IdM DNS サーバーで**/etc/named.conf**を手動で編集すると、サーバーごとに異なるグローバルフォワーダーとポリシーを使用できるようになります。**/etc/named.conf** を変更した後は、再起動が必要です。

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、次のコマンドを実行します。

1. **ネットワークサービス** タブをクリックし、**DNS** サブタブを選択してから、**DNS グローバル設定** セクションを選択します。
2. 新しい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。新しい転送ポリシーを定義するには、使用可能なポリシーのリストからポリシーを選択します。

図33.28 Web UI でのグローバル DNS 設定の編集



3. **Save** をクリックして、新しい設定を確定します。

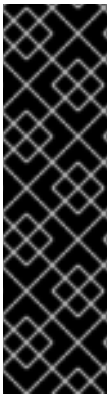
コマンドラインでのフォワーダーの設定
コマンドラインからフォワーダーのグローバルリストを設定するには、**ipa dnsconfig-mod** コマンドを使用します。LDAP データを編集することで、DNS グローバル設定を編集します。**ipa dnsconfig-mod** コマンドとそのオプションは、IdM DNS サーバーすべてを一度に設定して、ローカル設定を上書きします。

たとえば、**ipa dnsconfig-mod** を使用してグローバルフォワーダーのリストを編集する場合は、次のコマンドを実行します。

```
[user@server ~]$ ipa dnsconfig-mod --forwarder=192.0.2.254
  Global forwarders: 192.0.2.254
```

33.6.2. 正引きゾーンの設定

正引きゾーンには権限データが含まれておらず、ネームサーバーに、特定のゾーンに属する名前のクエリーのみを、設定されたフォワーダーに転送するように指示します。



重要

絶対に必要な場合を除き、正引きゾーンは使用しないでください。グローバル転送設定の上書きだけに使用を限定します。ほとんどの場合、「グローバルフォワーダーの設定」 で説明されている**グローバル転送のみ**を設定すれば十分で、正引きゾーンは必要ありません。

正引きゾーンは、標準的な解決策ではないので、正引きゾーンを使用すると予期しない問題のある動作が発生する可能性があります。新しい DNS ゾーンを作成する場合には、Red Hat は、NS レコードで標準の DNS 委譲を常に使用し、正引きゾーンを回避することを推奨します。

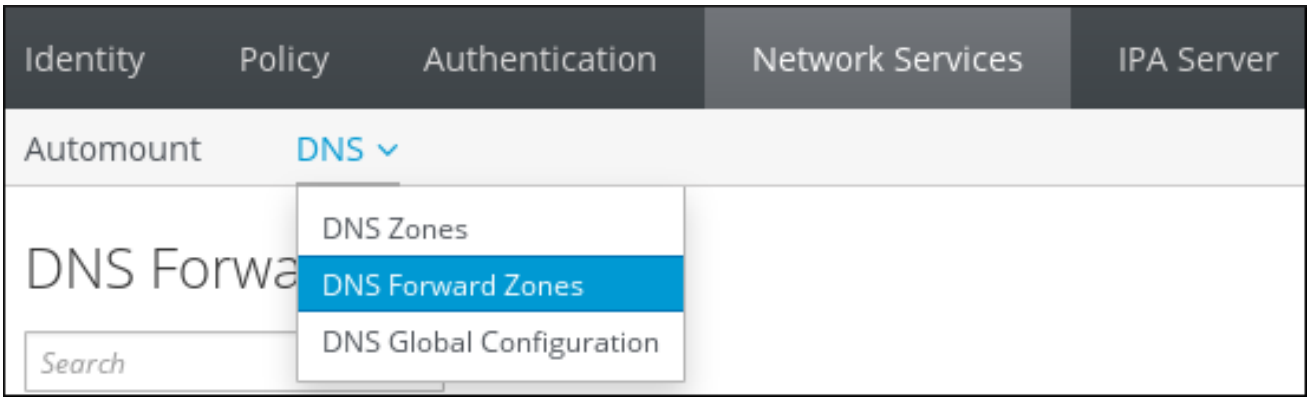
サポート対象の転送ポリシーに関する情報は、「フォワードポリシー」 を参照してください。

BIND サービスの詳細は、Red Hat Enterprise Linux ネットワークガイド、**/usr/share/doc/bind-*version_number*/** ディレクトリーに含まれる BIND 9 管理者リファレンスマニュアル、または外部ソースを参照してください。 [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、**ネットワークサービス** タブをクリックし、**DNS** サブタブを選択してから、**DNS 正引きゾーン** セクションを選択します。

図33.29 DNS 正引きゾーンの管理



DNS 正引きゾーン セクションでは、正引きゾーンに関するすべての必要な操作 (現在の正引きゾーンのリストの表示、新しい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、正引きゾーンの表示、正引きゾーンごとのフォワーダーと転送ポリシーの変更の許可、正引きゾーンの無効化または有効化) を処理できます。

コマンドラインで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**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される **ipa dnszone-*** コマンドと同じように動作します。

ipa dnsforwardzone-* コマンドには、複数のオプション (特に **--forwarder**、**--forward-policy**、**--name-from-ip**) があります。利用可能なオプションの詳細は、表33.1「ゾーン属性」 を参照するか、**--help** オプションを追加してコマンドを実行します。以下に例を示します。

```
ipa dnsforwardzone-add --help
```

正引きゾーンの追加

dnsforwardzone-add コマンドを使用して、新しい正引きゾーンを追加します。転送ポリシーが **none** に設定されていない場合は、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
```

正引きゾーンの変更

dnsforwardzone-mod コマンドを使用して正引きゾーンを変更します。転送ポリシーが **none** でない場合は、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。変更は、以下のようなさまざまな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 を実行すると、指定した正引きゾーンの情報が表示されます。

```
[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 を実行して、指定した正引きゾーンを特定します。

```
[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

指定した正引きゾーンは、**dnsforwardzone-del** で削除します。

```
[user@server ~]$ ipa dnsforwardzone-del zone.test.

----------------------------
Deleted forward DNS zone "zone.test."
----------------------------
```

正引きゾーンの有効化と無効化

dnsforwardzone-enable コマンドおよび **dnsforwardzone-disable** コマンドを使用して、正引きゾーンを有効または無効にします。正引きゾーンはデフォルトで有効になっていることに注意してください。

```
[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 コマンドおよび **dnsforwardzone-remove-permission** コマンドを使用して、システム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.
```

33.7. 逆引き DNS ゾーンの管理

逆引き DNS ゾーンは、次の 2 つの方法で識別できます。

- ゾーン名 (***reverse_ipv4_address*.in-addr.arpa** または ***reverse_ipv6_address*.ip6.arpa** の形式)。

  逆引きの IP アドレスは、IP アドレスのコンポーネントの順序を逆にすることで作成されます。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 (末尾はピリオド)。

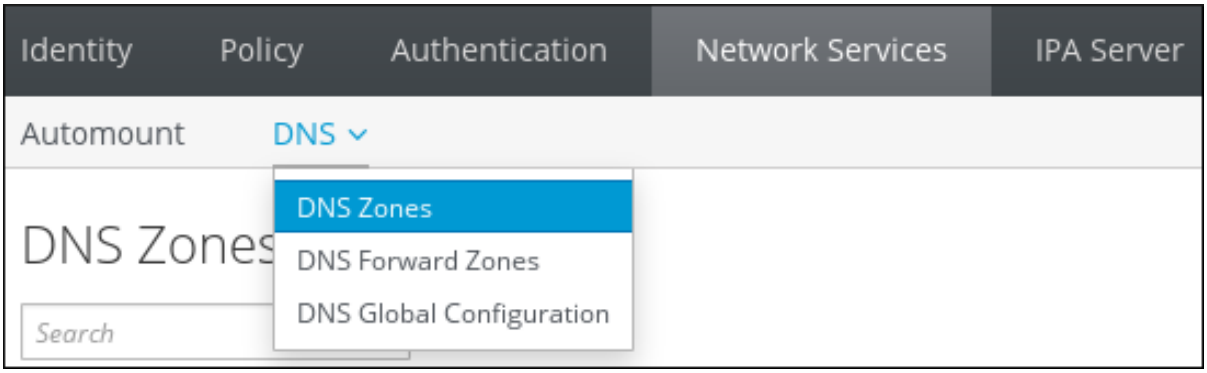
- ***network_ip_address*/*subnet_mask_bit_count*** の形式でのネットワークアドレス。

  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

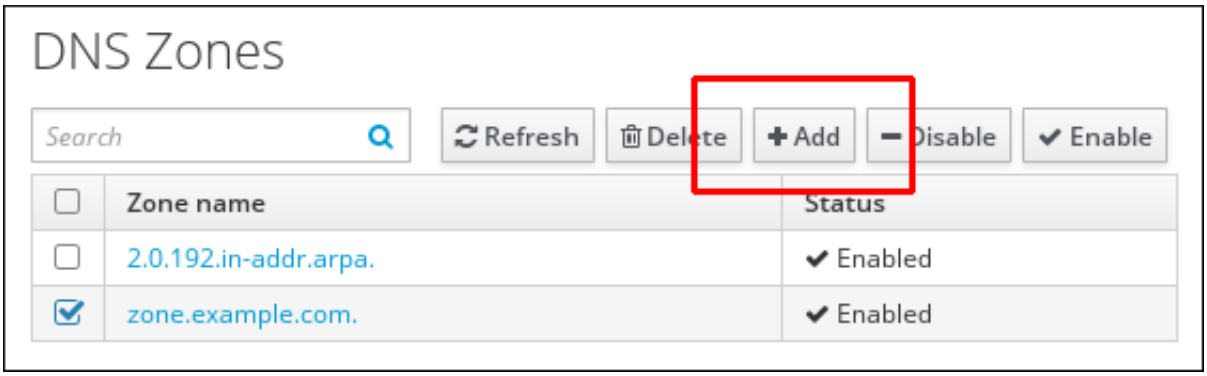
1. **Network Services** タブを開き、**DNS** サブタブを選択し、その後に **DNS Zones** セクションを選択します。

   図33.30 DNS ゾーンの管理



2. すべてのゾーンリストの上部にある **追加** をクリックします。

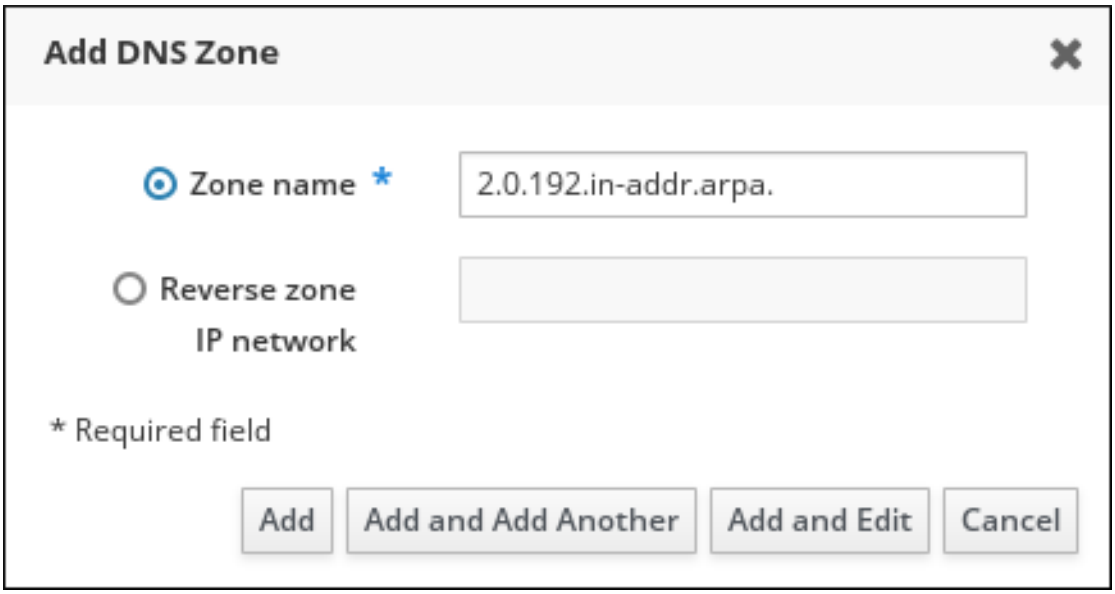
   図33.31 逆引き DNS ゾーンの追加



3. ゾーン名または逆引きゾーンの IP ネットワークを入力します。

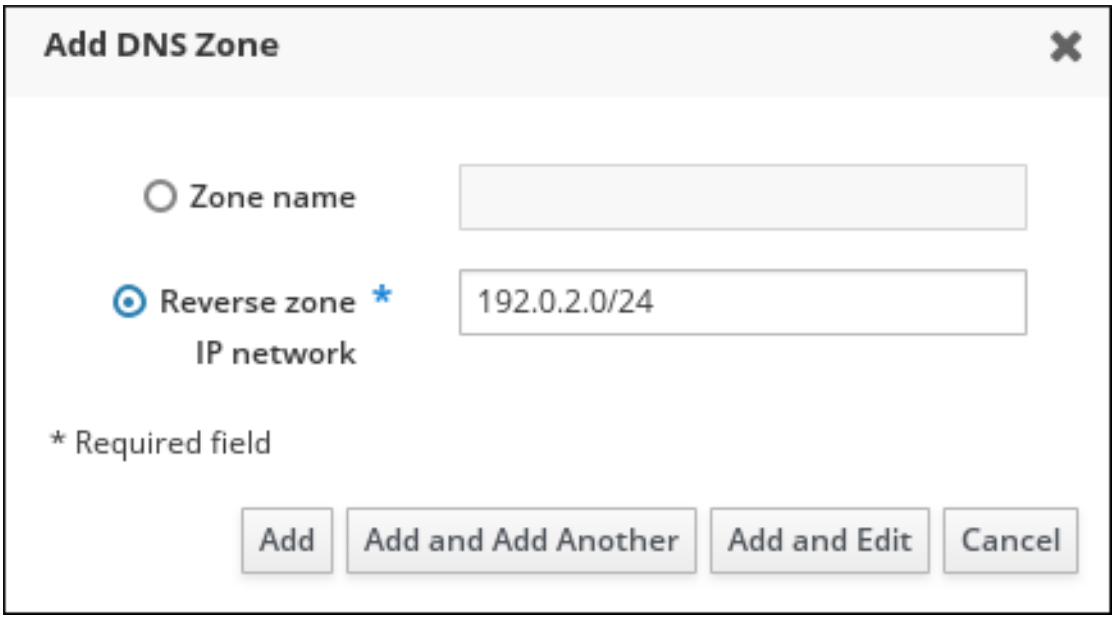
   a. たとえば、ゾーン名を使用して逆引き DNS ゾーンを追加する場合は、次のコマンドを実行します。

図33.32 名前での逆引きゾーンの作成



b. または、逆引きゾーンの IP ネットワークで逆引き DNS ゾーンを追加する場合は、次のコマンドを実行します。

図33.33 IP ネットワークでの逆引きゾーンの作成



逆引きゾーン IP ネットワーク フィールドのバリデーターは、入力中に無効なネットワークアドレスがあると警告を出します。ネットワークアドレスをすべて入力すると、この警告は表示されなくなります。

4. **Add** をクリックして、新しい逆引きゾーンを確認します。

コマンドラインでの逆引き DNS ゾーンの追加

コマンドラインから逆引きの DNS ゾーンを作成する場合は、**ipa dnszone-add** コマンドを使用します。

たとえば、ゾーン名を使用して逆引きゾーンを作成する場合は、次のコマンドを実行します。

```
[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場合は、次のコマンドを実行します。

```
[user@server ~]$ ipa dnszone-add --name-from-ip=192.0.2.0/24
```

他の逆引き DNS ゾーンの管理操作
「マスター DNS ゾーンの管理」 その他のゾーン管理操作 (一部の操作は、DNS ゾーンの編集や無効化、有効化などの逆引き DNS ゾーン管理にも該当) を説明します。

33.8. 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変更時、または **--allow-query** オプションを指定した **ipa dnszone-mod** を使用してクエリーを発行できるクライアントのリストを設定するときに設定できます。

以下に例を示します。

```
[user@server ~]$ ipa dnszone-mod --allow-query=192.0.2.0/24;2001:DB8::/32;203.0.113.1
example.com
```

デフォルトの**--allow-query** の値は **任意** で、これにより、任意のクライアントによるゾーンのクエリーが可能になります。

33.9. DNS の場所

33.9.1. DNS ベースのサービス検出

DNS ベースのサービス検出は、クライアントが DNS プロトコルを使用するプロセスで、**LDAP** や **Kerberos** など、特定のサービスを提供するネットワークでサーバーを見つけ出します。一般的な操作の1つとして、クライアントが最寄りのネットワークインフラストラクチャー内にある認証サーバーを特定できるようにすることが挙げられます。理由は、スループットが向上してネットワークレイテンシーが短縮されるので全体的なコスト削減を図ることができるためです。

サービス検出の主な利点は以下のとおりです。

- 近くにあるサーバーの名前を明示的に設定する必要がない。
- DNS サーバーをポリシーの中央プロバイダーとして使用する。同じ 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と優先順序に関する同じポリシーにアクセスできます。

IdM ドメインでは、LDAP、Kerberos、およびその他のサービスの DNS サービスレコード (SRV レコード) が存在します。たとえば、次のコマンドは、IdM DNS ドメインで TCP ベースの Kerberos サービスを提供するホストの DNS サーバーをクエリーします。

例33.10 DNS の場所に関する独立した結果

```
$ dig -t SRV +short _kerberos._tcp.idm.example.com
0 100 88 idmserver-01.idm.example.com.
0 100 88 idmserver-02.idm.example.com.
```

出力には、以下の情報が含まれます。

- **0** (優先度): ターゲットホストの優先度。値が小さいほど優先度が高くなります。
- **100** (加重)。優先順位が同じエントリーの相対的な重みを指定します。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
- サービスを提供するホストの正規名。

上記の例では、2 つのホスト名が返され、どちらも同じ優先順位と重みでした。この場合には、クライアントは結果リストから無作為にエントリーを使用します。

代わりに、クライアントが、DNS の場所に設定された DNS サーバーをクエリーすると、出力が異なります。場所が割り当てられた IdM サーバーの場合は、カスタマイズした値が返されます。以下の例では、クライアントは場所 **germany** の DNS サーバーをクエリーします。

例33.11 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、ローカルサーバーを優先する DNS の場所固有の SRV レコードを参照する DNS エイリアス (CNAME) を自動的に返します。この CNAME レコードは、出力の最初の行に表示されます。上記の例では、**idmserver-01.idm.example.com** ホストの優先度の値が最も小さいため、優先順位が高くなります。**idmserver-02.idm.example.com** の優先度の値が高く、推奨されるホストが使用できない場合にバックアップとしてのみ使用されます。

33.9.2. DNS の場所のデプロイに関する考慮事項

プライマリー IdM DNS ドメインに対して権威を持つ IdM DNS サーバーの場合には、IdM は場所固有の SRV レコードを生成できます。各 IdM DNS サーバーはロケーション固有の SRV レコードを生成するため、DNS の場所ごとに 1 つ以上の 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 の場所に対するアフィニティーは、クライアントが受け取った DNS レコードでのみ定義されます。そのため、DNS のサービス検出を行うクライアントが、IdM DNS サーバーからの場所固有のレコードを解決した場合には、IdM DNS サーバーと IdM 以外の DNS スレーブサーバーと recursor を組み合わせることができます。

IdM サービスおよび IdM DNS サービス以外のほとんどのデプロイメントでは、DNS recursor はラウンドトリップタイム (RTT) メトリックを使用して、最寄りの IdM DNS サーバーを自動的に選択します。通常、IdM DNS サーバーを使用するクライアントが、最寄りの DNS の場所のレコードを取得し、最寄りの DNS サーバーの最適なセットを使用するようになります。

33.9.2.1. DNS の Time to live (TTL)

クライアントは、ゾーンの設定に指定された期間の DNS リソースレコードをキャッシュできます。このキャッシュにより、クライアントは Time to Live (TTL) 値の有効期限が切れるまで変更を受け取れない場合があります。IdM におけるデフォルトの TTL 値は **1 day** です。

クライアントコンピューターがサイト間でローミングする場合には、IdM DNS ゾーンの TTL 値を調整する必要があります。この値は、クライアントがサイト間のローミングに必要とする時間よりも低い値に設定します。これにより、別のサイトに再接続する前にクライアントでキャッシュされた DNS エントリーが期限切れになり、DNS サーバーに対してクエリーを実行し、場所固有の SRV レコードを更新します。

DNS ゾーンのデフォルトの TTL を変更する方法は、「マスター DNS ゾーンの他の設定の追加」 を参照してください。

33.9.3. DNS の場所の作成

Web UI での DNS 場所の作成

1. **IPA Server** タブを開き、**Topology** サブタブを選択します。
2. ナビゲーションバーの **IPA の場所** をクリックします。
3. ロケーションリストの上部にある **追加** をクリックします。
4. ロケーション名を入力します。
5. **追加** ボタンをクリックして場所を保存します。

追加する場所についても手順を繰り返します。

コマンドラインでの DNS の場所の作成

たとえば、新しい場所 **germany** を作成するには、以下を入力します。

```
[root@server ~]# ipa location-add germany
----------------------------
Added IPA location "germany"
----------------------------
  Location name: germany
```

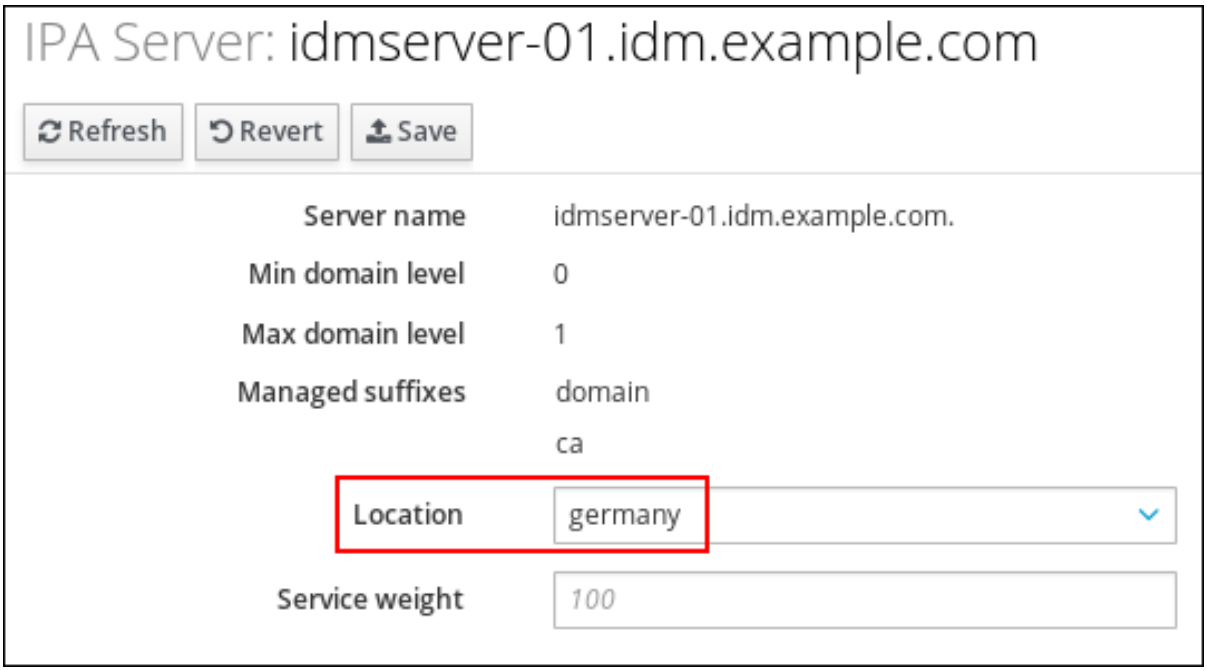
追加するすべての場所でこの手順を繰り返します。

33.9.4. DNS の場所への IdM サーバーの割り当て

Web UI での DNS の場所への IdM サーバーの割り当て

1. **IPA Server** タブを開き、**Topology** サブタブを選択します。
2. ナビゲーションにある **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、必要に応じてサービスの加重を設定します。

図33.34 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. 前の手順で DNS の場所を割り当てたホストで **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追加の IdM サーバーに対して、この手順を繰り返します。

コマンドラインでの DNS の場所への IdM サーバーの割り当て

1. オプション: 設定済みの DNS の場所をすべて表示します。

```
[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、場所 **germany** を *idmserver-01.idm.example.com* サーバーに割り当てるには、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. 前の手順で DNS の場所を割り当てたホストで **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追加の IdM サーバーに対して、この手順を繰り返します。

33.9.5. IdM クライアントが同じ場所にある IdM サーバーを使用するように設定する手順

IdM サーバーは、「DNS の場所への IdM サーバーの割り当て」 の説明に従って DNS の場所に割り当てられます。これで、IdM サーバーと同じ場所にある DNS サーバーを使用するようにクライアントを設定できます。

- DHCP サーバーが DNS サーバーの IP アドレスをクライアントに割り当てる場合は、DHCP サービスを設定します。DHCP サービスで DNS サーバーを割り当てる方法は、DHCP サービスのドキュメントを参照してください。
- クライアントに DHCP サーバーから DNS サーバーの IP アドレスが割り当てられない場合は、クライアントのネットワーク設定で IP を手動で設定します。Red Hat Enterprise Linux でネットワークを設定する方法は、『Red Hat Enterprise Linux ネットワークガイド』のネットワーク接続の設定セクションを参照してください。

注記

別のロケーションに割り当てられた DNS サーバーを使用するようにクライアントを設定すると、クライアントは両方の場所にある IdM サーバーに接続します。

例33.12 クライアントの場所により変化するネームサーバーエントリー

以下の例は、場所が異なるクライアントの **/etc/resolv.conf** ファイルにあるさまざまなネームサーバーエントリーを示しています。

プラハのクライアント:

```
nameserver 10.10.0.1
nameserver 10.10.0.2
```

パリのクライアント:

```
nameserver 10.50.0.1
nameserver 10.50.0.3
```

オスロのクライアント:

```
nameserver 10.30.0.1
```

ベルリンのクライアント:

```
nameserver 10.30.0.1
```

- 各 DNS サーバーが IdM の場所に割り当てられている場合に、クライアントはその場所にある IdM サーバーを使用します。

33.10. 外部 DNS の使用時の DNS レコードのシステム的な更新

外部 DNS を使用する場合には、Identity Management は、トポロジーの変更後に DNS レコードを自動的に更新しません。以下の手順では、外部 DNS サービスが管理する DNS レコードを体系的に更新する方法を説明します。これにより、手動の DNS 更新が必要なくなります。

基本的な概要は、「Identity Management で外部 DNS の更新」 を参照してください。

手順および例は、以下を参照してください。

- GUI を使用して外部 DNS 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 レコードを管理する場合: 「コマンドライン: **nsupdate**を使用した外部 DNS レコードの更新」

33.10.1. Identity Management で外部 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。

トポロジーの変更後に、DNS レコードを更新する必要があります。以下に例を示します。

- レプリカのインストールまたはアンインストール後
- Identity Management サーバーに CA、DNS、KRA、または Active Directory の信頼をインストールした後

33.10.2. GUI: 外部 DNS レコードの更新

1. 更新が必要な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 DNS GUI を使用して、レコードを更新します。

33.10.3. コマンドライン: **nsupdate**を使用した外部 DNS レコードの更新

本セクションでは、**nsupdate** ユーティリティーを使用して外部 DNS レコードを手動で更新する方法を説明します。スクリプトでこのセクションのコマンドを使用して、プロセスを自動化することもできます。

nsupdateの DNS レコードでファイルの生成

1. **--out** を指定して **ipa dns-update-system-records --dry-run** コマンドを実行します。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** ユーティリティーが許可する形式で、必要な 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。

   - レコードを更新するゾーンの自動検出
   - ゾーンの権威サーバーの自動検出

   標準以外の DNS 設定を使用しているか、ゾーンの委譲がない場合は、**nsupdate** が正しいゾーンとサーバーを見つけられない可能性があります。この場合は、生成されるファイルの先頭に以下のオプションを追加します。

   - **server** は、**nsupdate** がレコードを送信する信頼できる DNS サーバーのサーバー名またはポートを指定します。
   - **zone** は **nsupdate** がレコードを配置するゾーンの名前を指定します。

   以下に例を示します。

```
$ 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 更新要求の送信

nsupdate を使用して要求を送信する場合は、要求を適切にセキュリティー保護してください。以下のメカニズムを使用して、要求を保護できます。

Transaction Signature (TSIG) プロトコル

TSIG を使用すると、共有キーで **nsupdate** を使用できます。手順33.1「TSIG を使用した **nsupdate** 要求のセキュアな送信」を参照してください。

TSIG の GSS アルゴリズム (GSS-TSIG)

GSS-TSIG は、GSS-API インターフェイスを使用して秘密の TSIG 鍵を取得します。GSS-TSIG は、TSIG プロトコルの拡張機能です。手順33.2「GSS-TSIG を使用した **nsupdate** 要求のセキュアな送信」を参照してください。

手順33.1 TSIG を使用した nsupdate 要求のセキュアな送信

1. 以下の前提条件を満たしていることを確認します。

- TSIG に DNS サーバーを設定する必要があります。サーバー設定の例 (BIND、PowerDNS) 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、共有鍵が必要です。

2. **nsupdate** を実行し、以下のいずれかのオプションを使用して共有シークレットを指定します。
   - **-k**: TSIG 認証キーを指定します。

     ```
     $ nsupdate -k tsig_key.file dns_records_file.nsupdate
     ```

   - **-y**: 鍵の名前と Base64 でエンコードされた共有秘密鍵から署名を生成します。

     ```
     $ nsupdate -y algorithm:keyname:secret dns_records_file.nsupdate
     ```

手順33.2 GSS-TSIG を使用した nsupdate 要求のセキュアな送信

1. 以下の前提条件を満たしていることを確認します。
   - DNS サーバーは GSS-TSIG 用に設定する必要があります。サーバー設定の例 (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ていることを前提としています。

2. DNS 更新要求を送信し、レコードを更新できるプリンシパルで認証し、**-g** オプションで **nsupdate** を実行して 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
```

関連情報

- nsupdate(8) の man ページ
- RFC 2845 では TSIG プロトコルが説明されています。
- RFC 3645 は GSS-TSIG アルゴリズムを記述します。

33.11. 既存のサーバーへの DNS サービスのインストール

DNS サービスは、インストールされていない IdM サーバーにインストールできます。これを行うには、ipa-server-dns 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場合は、「統合 DNS を使用したサーバーのインストール」の説明に従い、**ipa-server-install** ユーティリティーを使用して DNS をインストールするのと同じ原則に従います。

ipa-dns-install の詳細は、ipa-dns-install(1) の man ページを参照してください。

33.11.1. ネームサーバーの追加設定

33.11.1.1. ネームサーバーの追加設定

IdM は、新しく設定した IdM DNS サーバーを、**/etc/resolv.conf** ファイル内の DNS サーバーのリストに追加します。IdM サーバーが利用できなくなった場合に備えて、バックアップサーバーとして別の DNS サーバーを手動で追加することが推奨されます。以下に例を示します。

```
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 の設定方法は、resolv.conf(5) man ページを参照してください。

---

[3] GSS-TSIG の詳細は RFC 3545 を参照してください。

[4] RFC 3007 の完全なテキストは、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34章 AUTOMOUNT の使用

自動マウントは、複数のシステムにわたってディレクトリーを管理、整理、およびアクセスする方法です。Automount は、ディレクトリーへのアクセスが要求されるたびに、その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、ユーザーのホームディレクトリーでは特に重要です。「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、自動マウントは、内部 LDAP ディレクトリー、および設定されている場合は DNS サービスでも機能します。

34.1. 自動マウントと IDM

Automount は、これらの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。すべてのディレクトリーは、**マウントポイント** または **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。

自動マウント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。必要に応じて、複数の **auto.master** 設定ファイルが別々のサーバーの場所にある可能性があります。

autofs ユーティリティーはサーバーで設定され、そのサーバーが IdM ドメインのクライアントである場合は、自動マウントのすべての設定情報が IdM ディレクトリーに保存されます。個別のテキストファイルに格納されるのではなく、**autofs** 設定 (マップ、場所、およびキー) が LDAP エントリーとして格納されます。たとえば、デフォルトのマップファイルは以下のように **auto.master** 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連携しますが、**autofs** 自体は設定または設定されません。

新しい場所は **cn=automount,dc=example,dc=com** 下のコンテナーエントリーとして追加され、各マップとキーは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、自動マウントは IdM とネイティブで機能します。自動マウント設定は、IdM ツールで管理できます。

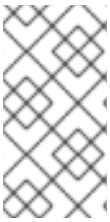
- **場所**用の**ipa automountlocation*** コマンド
- 直接および間接**マップ**用の**ipa automountmap***コマンド
- **鍵** 用の **ipa automountkey*** コマンド。

自動マウントが IdM ドメイン内で機能するには、NFS サーバーを IdM クライアントとして設定する必要があります。NFS の設定は、Red Hat Enterprise Linux ストレージ管理ガイド に記載されています。

34.2. 自動マウントの設定

Identity Management で自動マウントエントリー (場所やマップなど) を設定するには、既存の autofs/NFS サーバーが必要です。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、1 人以上のユーザーに対して**/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ことを確認します。NFS がすでに正常に機能していることを確認すると、後で IdM 自動マウント設定エラーが発生してもトラブルシュートが容易になります。

34.2.1. NFS の自動設定

システムを IdM クライアント (設定の一部としてドメインクライアントとして設定された IdM サーバーおよびレプリカを含む) として設定すると、**autofs** を設定し、IdM ドメインを NFS ドメインとして使用し、**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 NFS 設定ファイル (**/etc/sysconfig/nfs** および **/etc/idmapd.conf**) を自動的に設定します。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
```

この **ipa-client-automount** ユーティリティーは、NFS の設定とともに、外部 IdM ストアにアクセスできない場合に、SSSD が自動マウントマップをキャッシュするように設定します。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 は、対応しているサービスのリスト (**services = nss,pam,autofs...**) に追加され、空の設定エントリー (**[autofs]**) が指定されます。

- Name Service Switch (NSS) サービス情報が更新され、自動マウント情報についてまず 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。

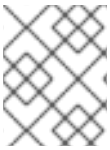
```
automount: sss files
```

クライアントによる自動マウント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がいくつかあることがあります。この場合は、**--no-sssd** オプションを使用して **ipa-client-automount** コマンドを実行できます。これにより、必要なすべての NFS 設定ファイルが変更されますが、SSSD 設定は変更されません。

```
[root@server ~]# ipa-client-automount --no-sssd
```

--no-sssd を使用する場合は、**ipa-client-automount** が更新する設定ファイルのリストが異なります。

- このコマンドにより、**/etc/sysconfig/nfs** ではなく **/etc/sysconfig/autofs** が更新されます。
- このコマンドは、IdM LDAP 設定で **/etc/autofs_ldap_auth.conf** を設定します。
- このコマンドは、自動マウントマップに LDAP サービスを使用するように **/etc/nsswitch.conf** を設定します。



注記

この **ipa-client-automount** コマンドは 1 回のみ実行できます。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。

34.2.2. SSSD および Identity Management を使用するように autofs を手動で設定

1. autofs が検索するスキーマ属性を指定するには、**/etc/sysconfig/autofs** ファイルを編集します。

```
#
# 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、自動マウントサービスが 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. autofs が IdM LDAP サーバーによるクライアント認証を許可するように **/etc/autofs_ldap_auth.conf** ファイルを編集します。

   - ***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"
    />
```

   必要に応じて **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。

   a. SSSD 設定ファイルを開きます。

```
[root@server ~]# vim /etc/sssd/sssd.conf
```

   b. autofs サービスを、SSSD が処理するサービスリストに追加します。

```
[sssd]
services = nss,pam,autofs
```

   c. **[autofs]** セクションを新規作成します。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]
```

   d. 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. SSSD が自動マウント設定のソースとしてリスト表示されるように、**/etc/nsswitch.conf** ファイルを確認します。

```
automount: sss files
```

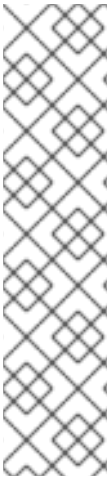
7. Restart autofs:

```
[root@server ~]# systemctl restart autofs.service
```

8. ユーザーの **/home** ディレクトリーをリスト表示して、設定をテストします。

```
[root@server ~]# ls /home/userName
```

リモートファイルシステムをマウントしない場合は、**/var/log/messages** ファイルでエラーを確認します。必要に応じて、***LOGGING*** パラメーターを **debug** に設定して、**/etc/sysconfig/autofs** ファイルのデバッグレベルを増やします。



注記

自動マウントで問題がある場合は、IdM インスタンスの 389 Directory Server アクセスログで自動マウント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 のアクセスログと自動マウント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。

34.2.3. Solaris での Automount の設定

注記

Solaris は、Identity Management で使用されるスキーマとは異なるスキーマを autofs 設定に使用します。Identity Management は、389 Directory Server 向けに定義される 2307bis 形式の自動マウントスキーマを使用します (IdM の内部 Directory Server インスタンスで使用されます)。

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** ファイルを編集し、以下のパラメーターを設定します。

```
NFS_CLIENT_VERSMAX=3
```

2. **ldapclient** コマンドを使用して、LDAP を使用するようホストを設定します。

-

```
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=aut
omount,dc=example,dc=com?one
    -a
serviceSearchDescriptor=auto_home:automountMapName=auto_home,cn=location,cn=auto
mount,dc=example,dc=com?one
    -a objectClassMap=shadow:shadowAccount=posixAccount
    -a searchTimelimit=15
    -a bindTimeLimit=5
```

3. **自動マウント** を有効にします。

```
# svcadm enable svc:/system/filesystem/autofs
```

4. 設定をテストします。

   a. LDAP 設定を確認します。

```
# ldapclient -l auto_master

dn:
automountkey=/home,automountmapname=auto.master,cn=location,cn=automount,dc=e
xample,dc=com
objectClass: automount
objectClass: top
automountKey: /home
automountInformation: auto.home
```

   b. ユーザーの /**home** ディレクトリーをリスト表示します。

```
# ls /home/userName
```

34.3. KERBEROS 対応の NFS サーバーのセットアップ

1. Red Hat Enterprise Linux 5 クライアントなど、いずれかの NFS クライアントが弱い暗号化のみをサポートしている場合は、以下を行います。

   a. IdM サーバーの Kerberos 設定を更新して、弱い **des-cbc-crc** 暗号化タイプを有効にします。

```
$ ldapmodify -x -D "cn=directory manager" -w password -h ipaserver.example.com -p
389

dn: cn=REALM_NAME,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
```

b. NFS サーバーで、NFS サーバーの **/etc/krb5.conf** ファイルに次のエントリーを追加して、弱い暗号化サポートを有効にします。

```
allow_weak_crypto = true
```

2. Kerberos チケットを取得します。

```
[root@nfs-server ~]# kinit admin
```

3. 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。

4. NFS サービスエントリーを作成します。

```
[root@nfs-server ~]# ipa service-add nfs/nfs-server.example.com
```

詳細な情報は、「サービスエントリーおよびキータブの追加と編集」 を参照してください。

5. **/etc/krb5.keytab** ファイルにキーを保存する次の **ipa-getkeytab** コマンドを使用して、NFS サーバーの NFS サービスキータブを取得します。

```
[root@nfs-server ~]# ipa-getkeytab -s ipaserver.example.com -p nfs/nfs-server.example.com
-k /etc/krb5.keytab
```

NFS クライアントのいずれかが弱い暗号化のみに対応している場合は、さらにコマンドに **-e des-cbc-crc** オプションを渡して、DES 暗号化キータブを要求します。

6. サービスエントリーを確認して、NFS サービスがキータブを使用して IdM で適切に設定されていることを確認します。

```
[root@nfs-server ~]# ipa service-show nfs/nfs-server.example.com
  Principal name: nfs/nfs-server.example.com@IDM.EXAMPLE.COM
  Principal alias: nfs/nfs-server.example.com@IDM.EXAMPLE.COM
  Keytab: True
  Managed by: nfs-server.example.com
```

7. nfs-utils パッケージをインストールします。

```
[root@nfs-server ~]# yum install nfs-utils
```

8. **ipa-client-automount** ユーティリティーを実行して、NFS 設定を設定します。

```
[root@nfs-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/sysconfig/nfs
Configured /etc/idmapd.conf
Started rpcidmapd
Started rpcgssd
Restarting sssd, waiting for it to become available.
Started autofs
```

デフォルトでは、このコマンドはセキュアな NFS を有効にし、**/etc/idmapd.conf** ファイルの ***Domain*** パラメーターを IdM DNS ドメインに設定します。別のドメインを使用する場合は、 ***--idmap-domain domain_name*** パラメーターを使用して指定します。

9. システムの起動時に自動起動するように **nfs-idmapd** を設定します。

```
# systemctl enable nfs-idmapd
```

10. **/etc/exports** ファイルを編集し、Kerberos セキュリティー設定 **krb5p** で共有を追加します。

```
/export  *(rw,sec=krb5:krb5i:krb5p)
/home  *(rw,sec=krb5:krb5i:krb5p)
```

この例では、Kerberos 認証が有効になっている読み書きモードで、**/export** ディレクトリーおよび **/home** ディレクトリーを共有します。

11. 共有ディレクトリーを再エクスポートします。

```
[root@nfs-server ~]# exportfs -rav
```

12. 必要に応じて、NFS サーバーを NFS クライアントとして設定します。「Kerberos 対応の NFS クライアントのセットアップ」を参照してください。

34.4. KERBEROS 対応の NFS クライアントのセットアップ

1. NFS クライアントが Red Hat Enterprise Linux 5 クライアントなどの弱い暗号化のみに対応している場合は、サーバーの **/etc/krb5.conf** ファイルに次のエントリーを設定して、弱い暗号化を許可します。

```
allow_weak_crypto = true
```

2. NFS クライアントが IdM ドメインの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」の説明に従って必要なホストエントリーを設定します。

3. nfs-utils パッケージをインストールします。

```
[root@nfs-client ~]# yum install nfs-utils
```

4. IdM ツールを実行する前に、Kerberos チケットを取得します。

```
[root@nfs-client ~]# kinit admin
```

5. **ipa-client-automount** ユーティリティーを実行して、NFS 設定を設定します。

```
[root@nfs-client ~] 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/sysconfig/nfs
Configured /etc/idmapd.conf
Started rpcidmapd
Started rpcgssd
Restarting sssd, waiting for it to become available.
Started autofs
```

デフォルトでは、これにより **/etc/sysconfig/nfs** ファイルでセキュアな NFS が有効になり、**/etc/idmapd.conf** ファイルの ***Domain*** パラメーターで IdM DNS ドメインが設定されます。

6. システムの起動時に自動起動するようにサービスを設定します。

```
[root@nfs-client ~]# systemctl enable rpc-gssd.service
[root@nfs-client ~]# systemctl enable rpcbind.service
```

7. **/etc/fstab** ファイルに次のエントリーを追加して、システムの起動時に **nfs-server.example.com** ホストから NFS 共有をマウントします。

```
nfs-server.example.com:/export  /mnt          nfs4  sec=krb5p,rw
nfs-server.example.com:/home    /home  nfs4  sec=krb5p,rw
```

この設定により、Red Hat Enterprise Linux が **/export** 共有を **/mnt** に、**/home** 共有を **/home** ディレクトリーにマウントするように設定されます。

8. マウントポイントが存在しない場合は作成します。

```
# mkdir -p /mnt/
# mkdir -p /home
```

9. NFS 共有をマウントします。

```
[root@nfs-client ~]# mount /mnt/
[root@nfs-client ~]# mount /home
```

このコマンドは、**/etc/fstab** エントリーからの情報を使用します。

10. Kerberos チケットを更新するように SSSD を設定します。

    a. **/etc/sssd/sssd.conf** ファイルの IdM ドメインセクションで次のパラメーターを設定して、SSSD がチケットを自動的に更新するように設定します。

    ```
    [domain/EXAMPLE.COM]
    ...
    krb5_renewable_lifetime = 50d
    krb5_renew_interval = 3600
    ```

    b. SSSD を再起動します。

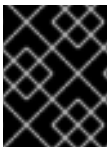
```
[root@nfs-client ~]# systemctl restart sssd
```

重要

pam_oddjob_mkhomedir モジュールは、NFS 共有でのホームディレクトリーの自動作成に対応していません。したがって、ホームディレクトリーを含む共有のルートにあるサーバーに、ホームディレクトリーを手動で作成する必要があります。

34.5. 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、場所は複数のマップを保存できます。また、場所には複数のマップを保存できます。場所のエントリーは、マップエントリーのコンテナーとしてのみ機能します。それ自体は、自動マウント設定ではありません。

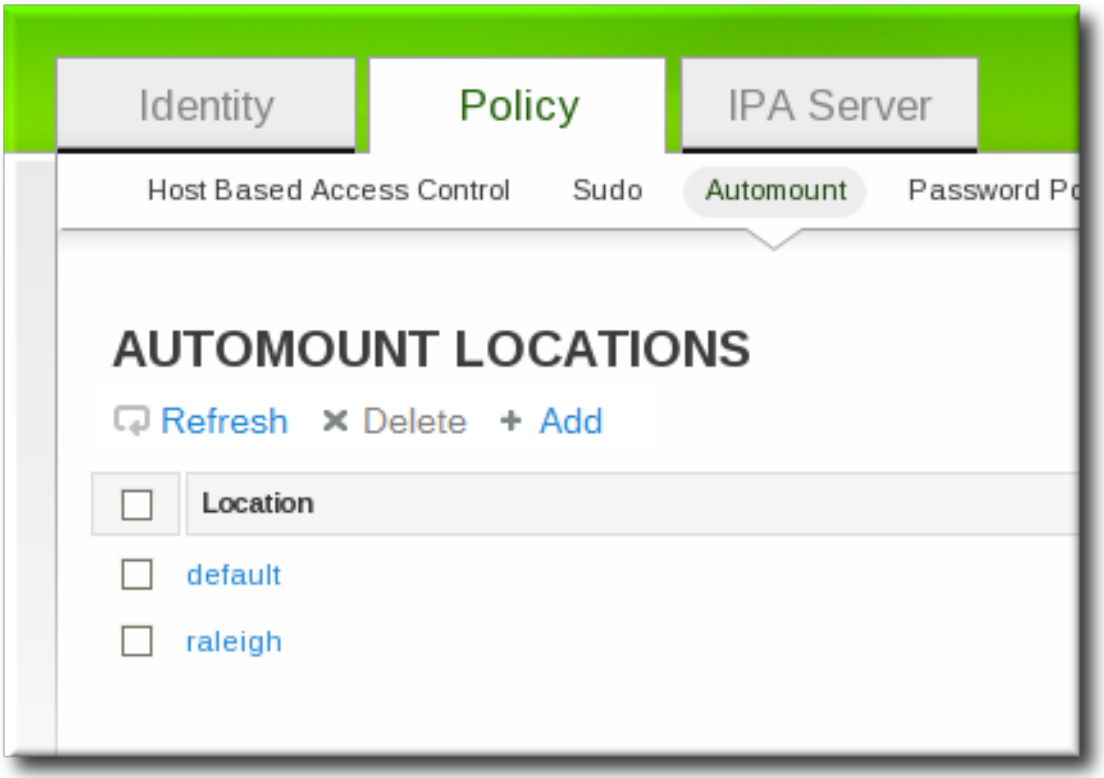


重要

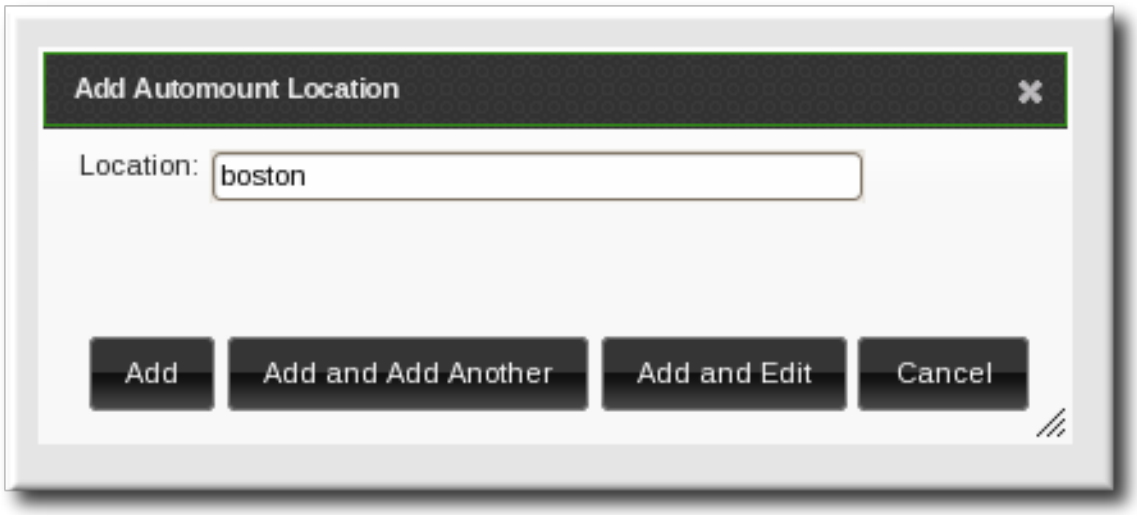
Identity Management は autofs を設定または設定しません。これは個別に行う必要があります。Identity Management は、既存の autofs デプロイメントと動作します。

34.5.1. 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. 自動マウントの場所リストの上部にある **Add** リンクをクリックします。



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。

34.5.2. 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** を使用して場所名を指定します。

```
$ ipa automountlocation-add location
```

以下に例を示します。

```
$ ipa automountlocation-add raleigh
---------------------------------
Added automount location "raleigh"
---------------------------------
  Location: raleigh
```

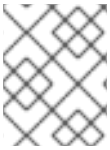
新しい場所が作成されると、2 つのマップ **auto.master** および **auto.direct** が自動的に作成されます。**auto.master** は、その場所のすべての自動マウントマップのルートマップです。**auto.direct** は、ダイレクトマウント用のデフォルトのマップで、**/-** にマウントされます。

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:
```

34.6. 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、ダイレクトマップと間接マップの両方に対応します。



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 を設定または設定しません。これは個別に行う必要があります。Identity Management は、既存の autofs デプロイメントと動作します。

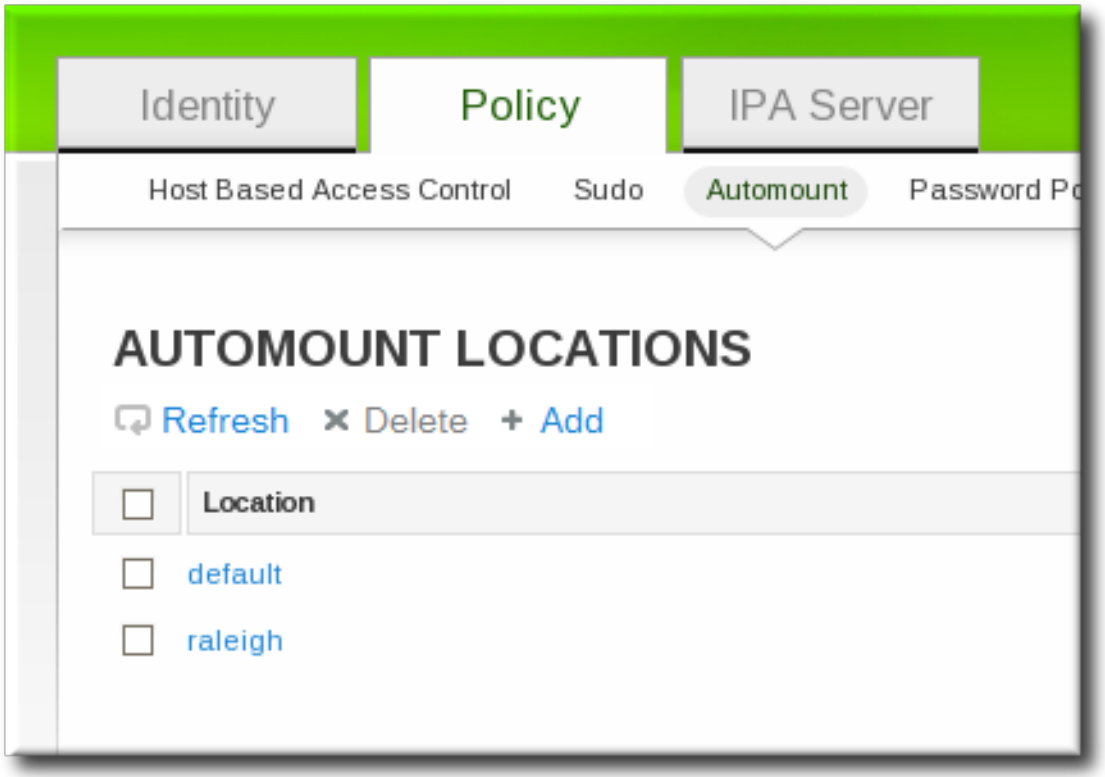
34.6.1. ダイレクトマップの設定

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。

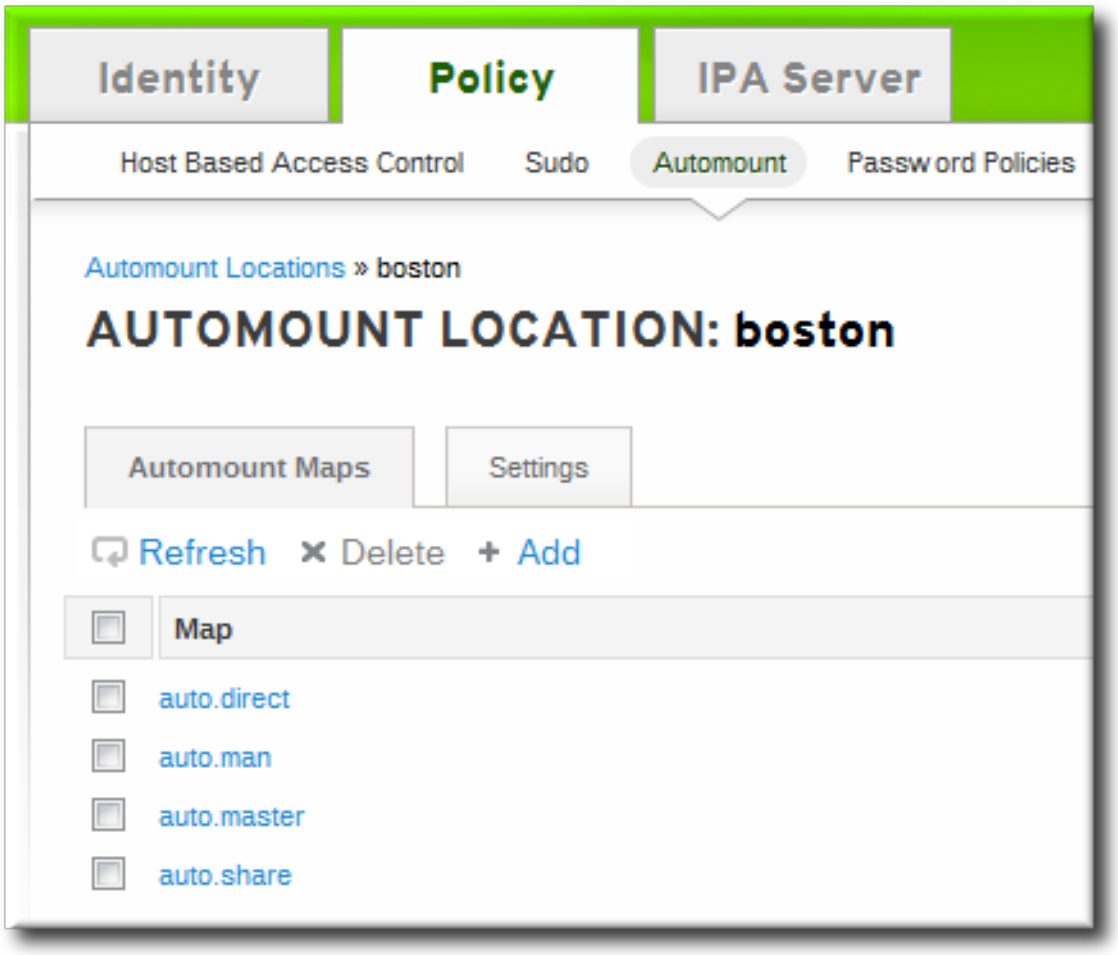
```
---------------------------
/etc/auto.direct:
/shared/man server.example.com:/shared/man
```

34.6.1.1. Web UI でのダイレクトマップの設定

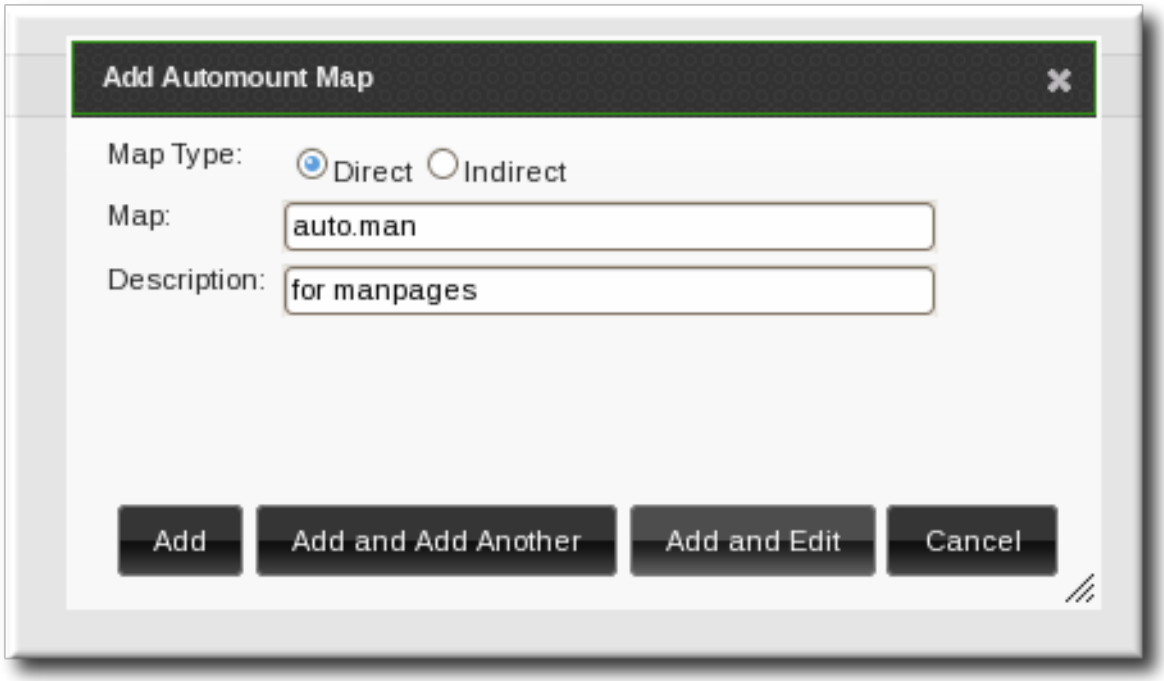
1. **Policy** タブをクリックします。
2. **Automount** サブタブをクリックします。
3. マップの追加先となる automount の場所の名前をクリックします。



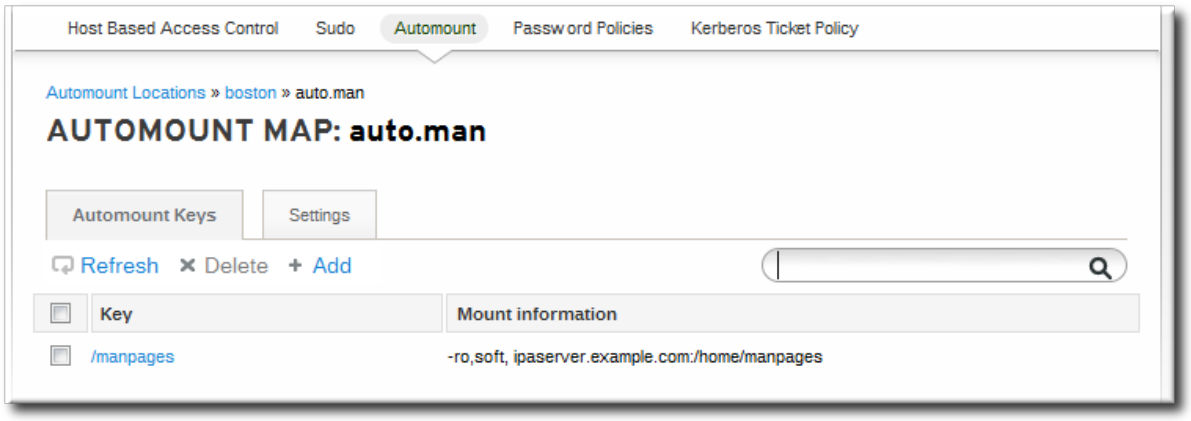
4. **Automount Maps** タブで **Add** をクリックして新規マップを作成します。



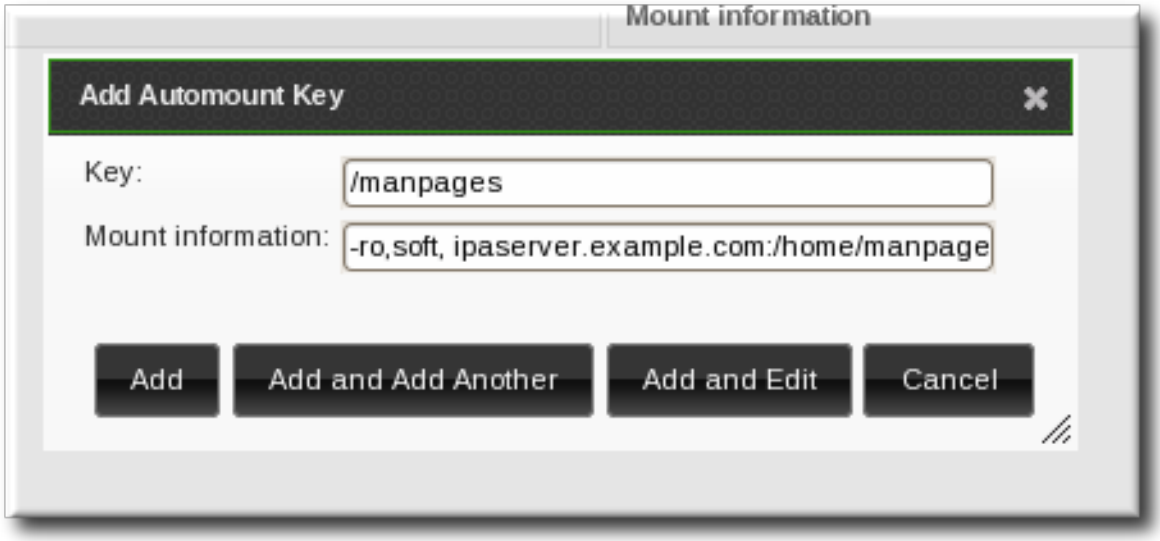
5. ポップアップウィンドウで **Direct** ラジオボタンを選択し、新規マップの名前を入力します。



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。

34.6.1.2. 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。**--key** オプションはマウントポイントを特定し、**--info** がディレクトリーのネットワークの場所と、使用する **mount** オプションを指定します。以下に例を示します。

```
$ 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、**ldapclient** コマンドを使用してダイレクトマップおよびキーを追加して、LDAP エントリーを直接追加します。

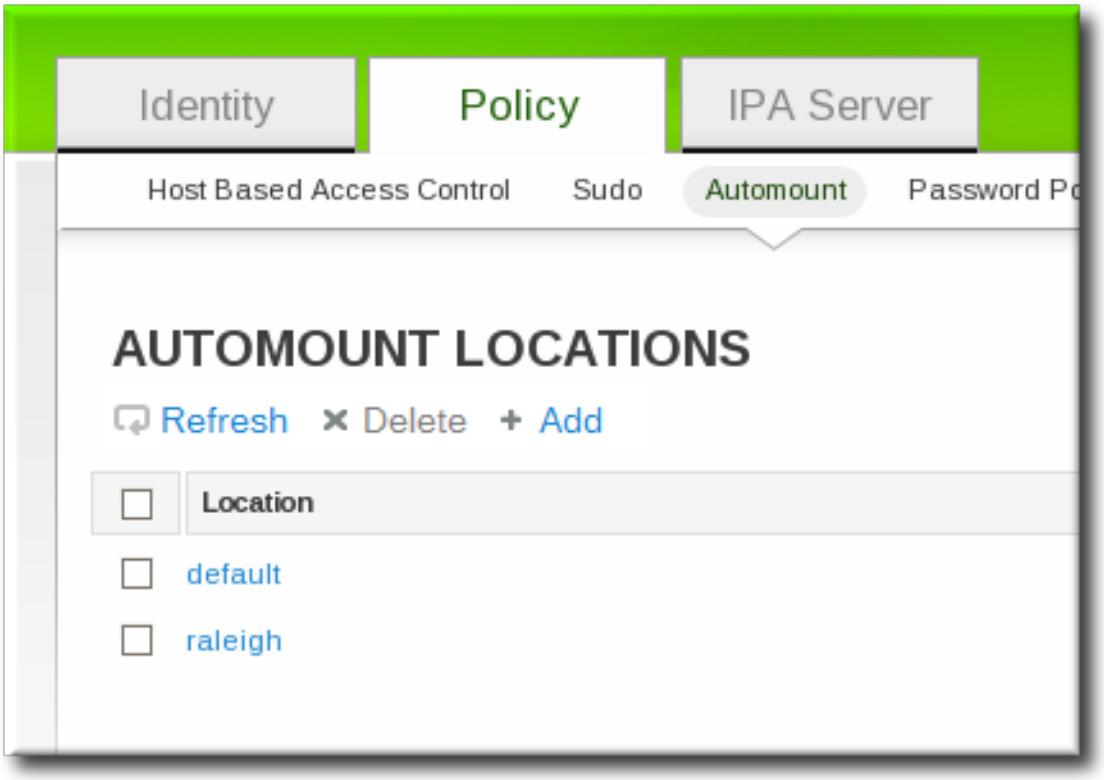
```
ldapclient -a
serviceSearchDescriptor=auto_direct:automountMapName=auto.direct,cn=location,cn=automount,dc
=example,dc=com?one
```

34.6.2. 間接マップの設定

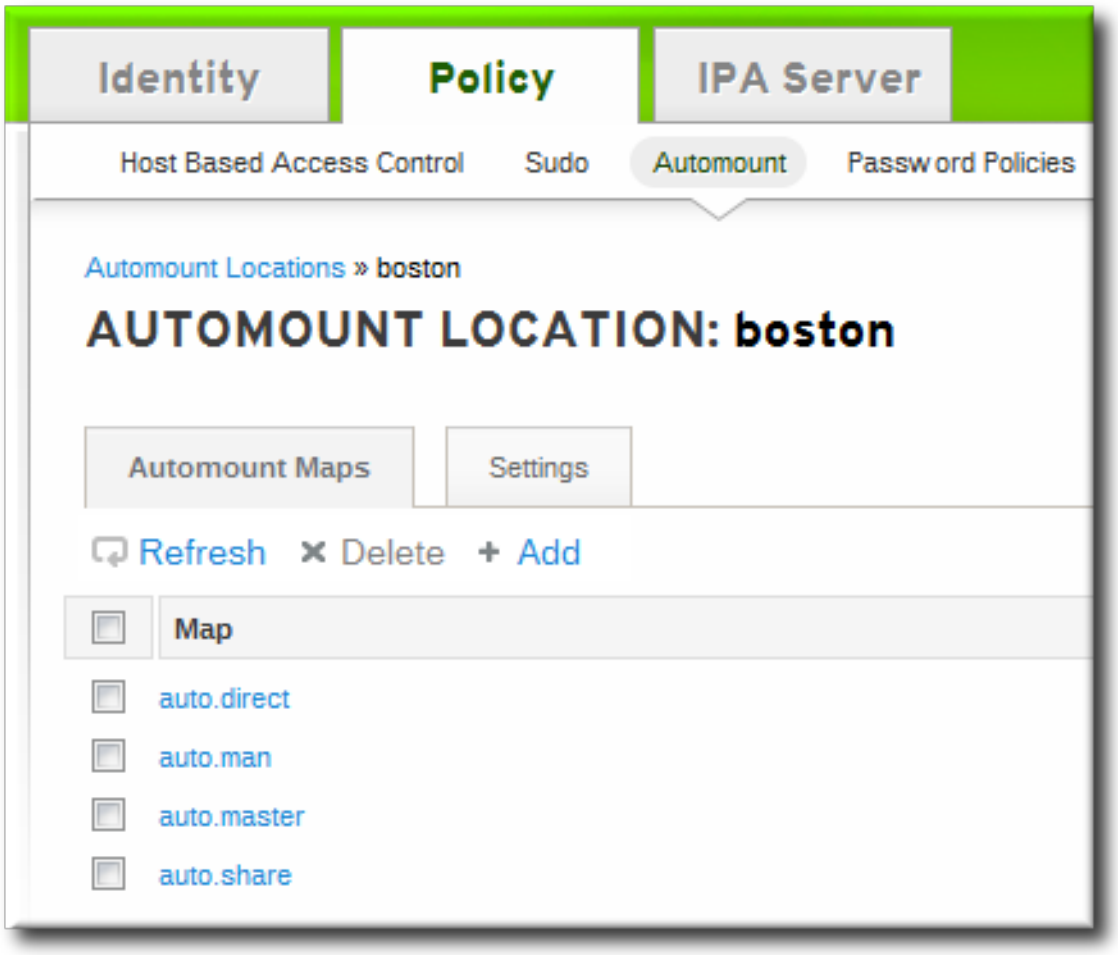
間接マップは、基本的にマップの相対パスを指定するものです。親エントリーがすべての間接マップのベースディレクトリーを設定します。間接マップキーはサブディレクトリーを設定します。間接マップの場所がロードされたときに常に、キーがベースディレクトリーに追加されます。たとえば、ベースディレクトリーが **/docs** で、キーが **man** の場合は、マップは **/docs/man** になります。

34.6.2.1. Web UI での間接マップの設定

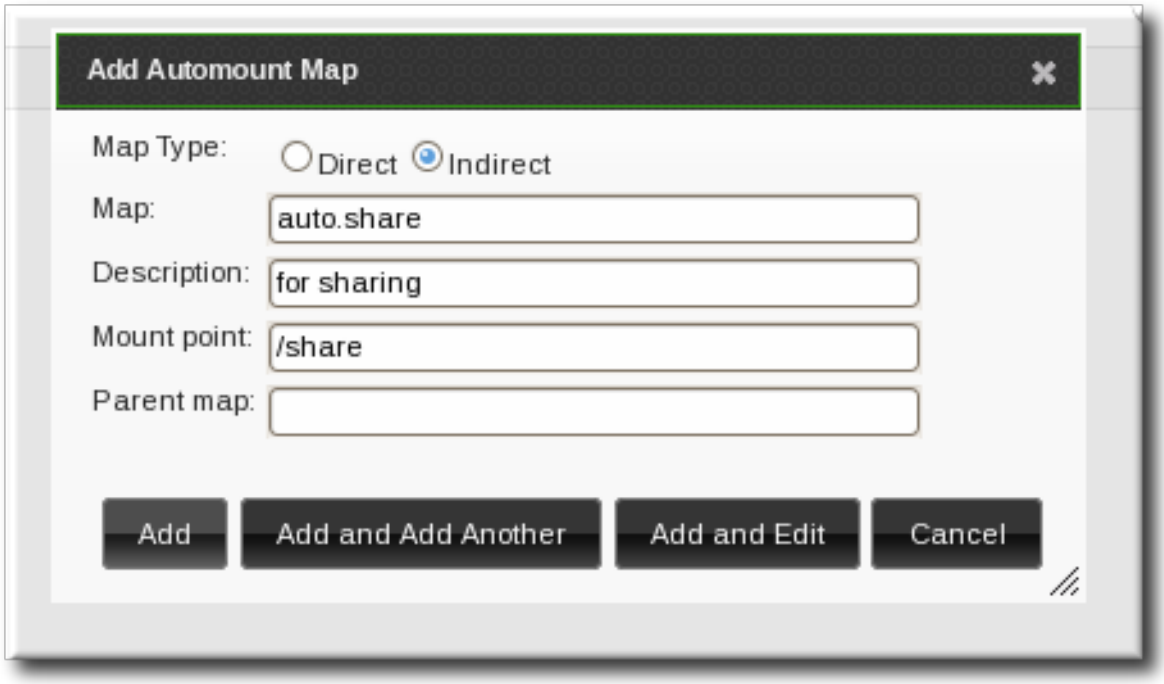
1. **Policy** タブをクリックします。
2. **Automount** サブタブをクリックします。
3. マップの追加先となる automount の場所の名前をクリックします。



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。

34.6.2.2. 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** コマンドを使用して、ベースエントリーを設定するための間接マップを作成します。**--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ーを設定します。デフォルトの親エントリーは **auto.master** ですが 、使用すべき別のマップが存在する場合は、**--parentmap** オプションで指定できます。

   ```
   $ ipa automountmap-add-indirect location mapName --mount=directory [--
   parentmap=mapName]
   ```

   以下に例を示します。

   ```
   $ 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. 設定を確認するには、**automountlocation-tofiles** で、その場所ファイルリストを確認します。

   ```
   $ 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** コマンドを使用して間接マップを追加し、LDAP エントリーを直接追加します。

```
ldapclient -a
serviceSearchDescriptor=auto_share:automountMapName=auto.share,cn=location,cn=automount,dc
=example,dc=com?one
```

34.6.3. 自動マウントマップのインポート

既存の自動マウントマップがある場合は、それを IdM 自動マウント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 自動マウント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。この **--continuous** オプションでは、**automountlocation-import** コマンドに対して、エラーが発生した場合でも、マップファイルを継続するように指示します。

以下に例を示します。

```
$ ipa automountlocation-import raleigh /etc/custom.map
```

パート VIII. セキュリティーの強化

本章では、セキュアに **Identity Management** を使用する方法を推奨します。

第35章 IDENTITY MANAGEMENT の TLS 設定

本書は、Red Hat Enterprise Linux 7.3 以降で、TLS プロトコルバージョン 1.2 を必要とするように Identity Management サーバーを設定する方法を説明します。

TLS 1.2 は、以前のバージョンの TLS よりも安全性が高いと見なされています。IdM サーバーを、セキュリティー要件が高い環境にデプロイする場合は、TLS 1.2 よりも安全でないプロトコルを使用した通信を禁止するように設定できます。

重要

TLS 1.2 を使用するすべての IdM サーバーでこの手順を繰り返します。

35.1. HTTPD デーモンの設定

1. **/etc/httpd/conf.d/nss.conf** を開き、**NSSProtocol** エントリーおよび **NSSCipherSuite** エントリーに次の値を設定します。

```
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
```

   または、次のコマンドを使用して、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
```

35.2. DIRECTORY SERVER コンポーネントの設定

ldapmodify ユーティリティーを使用して DS を自動的に設定するには、以下を実行します。

1. **ldapmodify** を使用して、設定の変更を行います。

```
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
```

DS (Directory Server) を手動で設定します。

1. DS を停止します。

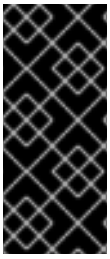
```
# systemctl stop dirsrv@EXAMPLE-COM.service
```

2. **/etc/dirsrv/slapd-*EXAMPLE-COM*/dse.ldif** を開き、**cn=encryption,cn=config** エントリーを変更して以下を設定します。

```
sslVersionMin: TLS1.2
```

3. DS を起動します。

```
# systemctl start dirsrv@EXAMPLE-COM.service
```



重要

dse.ldif ファイルを手動で編集する場合は、事前にサーバーをシャットダウンしていることを確認してください。DS は、システムの起動時に一度だけこのファイルを読み込むため、LDAP 経由で設定を変更すると、サーバーの実行中に手動で行った変更がすべて失われます。**dse.ldif** ファイルの編集は、動的に変更できない属性の変更のみに推奨されます。

35.3. 証明書サーバー (CS) 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** ファイルを開きます。**sslVersionRangeStream** パラメーターおよび **sslVersionRangeDatagram** パラメーターの発生をすべて、以下の値に設定します。

```
sslVersionRangeStream="tls1_2:tls1_2"
sslVersionRangeDatagram="tls1_2:tls1_2"
```

または、以下のコマンドを使用して、値を置き換えます。

```
# 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
```

35.4. 結果

Identity Management サーバーは、TLS 1.2 を必要とするように設定されています。以前のバージョンの TLS にのみ対応する Identity Management クライアントは、Identity Management サーバーと通信できなくなります。

第36章 匿名バインドの無効化

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。

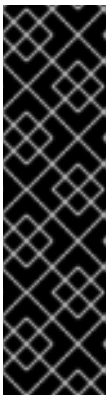
LDAP ツールを使用して ***nsslapd-allow-anonymous-access*** 属性をリセットすることで、389 Directory Server インスタンスで匿名バインドを無効にできます。

警告

特定のクライアントは、匿名バインドを使用して IdM 設定を検出します。また、compat ツリーは、認証を使用していない従来のクライアントでは機能しない可能性があります。

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、匿名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 では、ディレクトリーデータへのアクセス**なしで**、ルート DSE および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、新しい設定を読み込みます。

```
# systemctl restart dirsrv.target
```

関連情報

- 『Red Hat Directory Server 管理ガイド』 の 『コマンドラインを使用したエントリーの管理』 セクション

パート IX. パフォーマンスチューニング

本章では、**Identity Management** のパフォーマンスを最適化するための推奨プラクティスを説明します。

第37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

ユーザー追加の 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多数のエントリーを追加すると、時間がかかる可能性があります。本章では、プロビジョニングができるだけ早く完了するように、プロセスをチューニングする方法を説明します。

手順の一環として、以下を行います。

- Identity Management (IdM) は、LDIF ファイルからプロビジョニングされるエントリーを読み込み、それらをターゲットの 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の特定の属性にカスタム値を設定し、MemberOf および Schema Compatibility プラグインを無効にします。この手順には、Memberof が無効になっているのに対応するため、プロビジョニングされたエントリーで**fixup-memberof.pl** プラグインを実行することが含まれます。

この手順は、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、およびホストベースのアクセス制御 (HBAC) ルールのエントリータイプをプロビジョニングするように設計され、テストされています。

一括プロビジョニングの推奨事項および前提条件

推奨事項:

- 多数のエントリー (10,000 以上) をプロビジョニングする場合は、LDAP クライアントがエントリーがプロビジョニングされている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しないようにしてください。たとえば、サーバーのポート 389 および 636 を無効にし、LDAPI を使用して Unix ソケットで機能させることで、これを実現できます。

  理由 MemberOf プラグインはサーバーで無効になっているため、サーバー上のメンバーシップ情報は無効です。

- プロビジョニング中に実行する必要がないアプリケーションを停止します。

  理由: これは、マシン上で可能な限り多くのメモリーを確保するのに役立ちます。空きメモリーはファイルシステムキャッシュにより使用されるため、プロビジョニングのパフォーマンスが向上します。

  以下の手順には、IdM サービスを停止し、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ことに注意してください。IdM サービス、とりわけ **tomcat** は大量のメモリーを消費しますが、プロビジョニング時には使用されません。

- 1 台のサーバーのみを使用した新規の IdM デプロイメントで手順を実行します。レプリカは、プロビジョニングが完了しないと作成できません。

  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速くなります。サーバーが複数存在するデプロイメントでは、レプリカの情報が大幅に古くなります。

前提条件:

- プロビジョニングするエントリーを含む 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て LDIF ファイルを作成します。

  LDIF 形式の詳細は、『Red Hat Directory Server 10 管理ガイド』 の LDIF ファイル形式の概要 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とデータベースのロック

     ```
     # ldapsearch -D "cn=directory manager" -w secret -b "cn=config,cn=ldbm
     database,cn=plugins,cn=config" nsslapd-dbcachesize nsslapd-db-locks

     ...
     nsslapd-dbcachesize: 10000000
     nsslapd-db-locks: 50000
     ...
     ```

   - エントリーキャッシュのサイズと DN キャッシュのサイズ

     ```
     # ldapsearch -D "cn=directory manager" -w secret -b "cn=userRoot,cn=ldbm
     database,cn=plugins,cn=config" nsslapd-cachememsize nsslapd-dncachememsize

     ...
     nsslapd-cachememsize: 10485760
     nsslapd-dncachememsize: 10485760
     ...
     ```

2. 取得した値を書き留めておきます。プロビジョニングが終了すると、パラメーターはこの値にリセットされます。

データベース、ドメインエントリー、および DN キャッシュサイズの調整

データベースのキャッシュサイズの場合は、次のコマンドを実行します。

1. 必要な値を判断します。

   推奨値は、通常 200MB から 500MB です。ユースケースに適した値は、システムで利用可能なメモリーによって異なります。

   - 8GB 以上のメモリー → 500 MB
   - 8GB - 4GB のメモリー → 200 MB
   - 4GB 未満のメモリー → 100 MB

2. 以下のテンプレートを使用して、決定された値を設定します。

   ```
   dn: cn=config,cn=ldbm 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37.1「**ldapmodify** を使用した LDAP 属性の変更」 を参照してください。

例37.1 ldapmodify を使用した LDAP 属性の変更

1. **ldapmodify** コマンドを実行し、文を追加して属性値を変更します。以下に例を示します。

```
# ldapmodify -D "cn=directory manager" -w secret -x
dn: cn=config,cn=ldbm database,cn=plugins,cn=config
changetype: modify
replace: nsslapd-dbcachesize
nsslapd-dbcachesize: 200000000
```

2. **Ctrl**+**D** を押して、変更をサーバーに送信します。操作が正常に終了すると、以下のメッセージが表示されます。

```
modifying entry "cn=config,cn=ldbm database,cn=plugins,cn=config"
```

ドメインエントリーキャッシュのサイズの場合は、次のコマンドを実行します。

1. 必要な値を判断します。

   推奨値は 100MB から 400MB です。適切な値は、使用しているシステムで利用可能なメモリーによって異なります。

   - 4 GB 以上のメモリー → 400 MB
   - 2GB - 4GB のメモリー → 200 MB
   - 2GB 未満のメモリー → 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は、全エントリー (グループとメンバー) が収まるようにエントリーキャッシュを大きく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、決定された値を設定します。

```
dn: cn=userRoot,cn=ldbm database,cn=plugins,cn=config
changetype: modify
replace: nsslapd-cachememsize
nsslapd-cachememsize: entry_cache_size_in_bytes
```

ドメイン名 (DN) キャッシュサイズの場合は、次のコマンドを実行します。

1. パフォーマンスを最大化するために、DN キャッシュは、プロビジョニングされたエントリーのすべての DN に適合することが推奨されます。ユースケースに適した値を見積もるには、以下のコマンドを実行します。

   a. ファイル内のすべての DN エントリーの数を指定します。DN エントリーは **dn:** で始まる行にあります。たとえば **# grep**、**sed**、および **wc** を使用する場合は、以下を実行します。

```
# grep '^dn: ' ldif_file | sed 's/^dn: //' | wc -l
92200
```

   b. LDIF ファイル内のすべての DN エントリー文字列のサイズを決定します。

```
# grep '^dn: ' ldif_file | sed 's/^dn: //' | wc -c
9802460
```

   c. 平均の DN サイズを取得する: すべての DN エントリー文字列のサイズを、ファイル内のすべての DN エントリーの数で除算します。

例: 9,802,460 / 92,200 ≈ 106

d. 平均のメモリーサイズを取得します。平均の DN サイズの倍数 2 を指定して、結果に 32 を加算します。

例: (106 * 2) + 32 = 244

e. 適切な DN キャッシュサイズを取得します。平均メモリーサイズに、LDIF ファイルの DN エントリーの合計数を乗算します。

例: 244 * 92,200 = 22,496,800

2. 以下のテンプレートを使用して、決定された値を設定します。

```
dn: cn=userRoot,cn=ldbm 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とデータベースのロックの調整

1. MemberOf プラグインおよび 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、プロビジョニングの速度が大幅に改善されます。スキーマの互換性を無効にすると、操作時間が短縮されます。

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37.1「**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 (「一括プロビジョニングの推奨事項および前提条件」 で推奨されている) 場合は、コンテンツ同期および 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
```

追加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が向上します。

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します。

```
# ipactl stop
```

次の手順でデータベース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。後でもう一度、再起動します。

4. データベースのロック数を調整します。適切な値は、プロビジョニングされたエントリーの半数になります。

- 最小値は 10,000 です。
- 最大値は 200,000 です。

DS が停止したので、**/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
```

メンバーシップを計算すると、IdM は多数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数が多いほど、プロビジョニングに必要なロックが多くなります。

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 インスタンスに新しいエントリーをインポートするには、次のコマンドを実行します。たとえば、**ldapadd** ユーティリティーを使用する場合は、以下を実行します。

```
# ldapadd -D "binddn" -y password_file -f ldif_file
```

ldapadd の使用方法は、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と元の属性値の復元

1. MemberOf の有効化

```
dn: cn=MemberOf Plugin,cn=plugins,cn=config
changetype: modify
replace: nsslapd-pluginEnabled
nsslapd-pluginEnabled: on
```

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37.1「**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
```

前の手順で MemberOf を有効にしたため、この時点で DS を再起動する必要があります。

3. **(objectClass=*)** フィルターを使用して **fixup-memberof.pl** スクリプトを実行し、プロビジョニングされたすべてのエントリーで **memberOf** 属性を再生成して更新します。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時に MemberOf プラグインが無効になっていたため、**fixup-memberof.pl** の実行が必要です。プロビジョニングを続行するには、スクリプトが正常に完了している必要があります。

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. 「不要なサービスの無効化とデータベースのロックの調整」 でコンテンツ同期プラグインおよび Retro Changelog プラグインを無効にした場合は、再度有効にします。

```
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」 でバックアップを作成したデータベースキャッシュ、エントリーキャッシュ、および 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」 でバックアップを作成したデータベースロックの元の値を復元します。DS が停止したので、**/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 など、すべての IdM サービスが開始します。

第38章 IDENTITY MANAGEMENT におけるフェイルオーバー、負荷分散、および高可用性

Identity Management (IdM) には、LDAP ID ドメインおよび証明書のレプリケーションなどの独自のフェイルオーバー、負荷分散、および高可用性機能、ならびに**システムセキュリティーサービスデーモン** (SSSD) によるサービス検出とフェイルオーバーのサポートが同梱されます。

IdM には以下の機能が備わっています。

- クライアント側のフェイルオーバー機能
- サーバー側のサービスの可用性

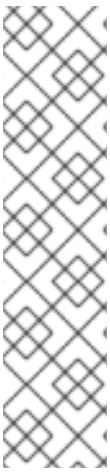
クライアント側のフェイルオーバー機能
SSSD は、クライアントが自動的に検出した DNS サーバーからサービス (SRV) リソースレコードを取得します。**SSSD** は、SRV レコードに基づいて、利用可能な IdM サーバーのリスト (これらのサーバーの接続性に関する情報を含む) を保持します。IdM サーバーがオフラインになるか、過負荷になると、SSSD は他のどのサーバーと通信するかをすでに認識しています。

DNS 自動検出が利用できない場合は、IdM クライアントを設定して、IdM サーバーの固定リストを使用して、障害発生時に SRV レコードを取得するようにします。

IdM クライアントのインストール時に、インストーラーは、クライアントのホスト名の親であるすべてのドメインについて、**_ldap._tcp.*DOMAIN*** DNS SRV レコードを検索します。この方法で、インストーラーは、クライアントとの通信に最も便利な IdM サーバーのホスト名を取得し、そのドメインを使用してクライアントコンポーネントを設定します。

サーバー側のサービスの可用性
IdM を使用すると、地理的に分散しているデータセンターでサーバーを複製できます。このため、IdM クライアントと、アクセス可能な最寄りのサーバーとの間のパスが短くなります。サーバーを複製すると、より多くのクライアントで負荷を分散し、スケーリングできます。

IdM レプリケーションメカニズムでは、アクティブ/アクティブのサービスの可用性を実現できます。すべての IdM レプリカのサービスは、同時に利用できます。



注記

IdM とその他の負荷分散を組み合わせる場合に、HA ソフトウェアは推奨されません。サードパーティーの高可用性 (HA) ソリューションの多くは、アクティブ/パッシブのシナリオを想定し、IdM の可用性に対して不要なサービスの中断を発生させます。他のソリューションでは、クラスター化されたサービスごとに仮想 IP または単一のホスト名を使用します。このような方法はすべて、通常、IdM ソリューションが提供するタイプのサービスの可用性では適切に機能しません。また、Kerberos との統合性が非常に低く、デプロイメントのセキュリティーと安定性が全体的に低下します。

また、特にこのようなサービスの可用性が高く、HA 機能を提供するためにネットワーク設定を変更するソリューションを使用する場合は、関連のないその他のサービスの IdM マスターへのデプロイは推奨されません。

認証に Kerberos を使用する場合のロードバランサーの使用方法は、今回のブログ投稿 を参照してください。

パート X. 移行

本章では、他のソリューションから**ID 管理** にデプロイメントを移行する際の推奨プラクティスを説明します。

第39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 ID 検索用に LDAP サーバーをデプロイしてあるので、次にバックエンドを Identity Management に移行する必要があります。IdM 移行ツールを使用して、データを失うことなく、パスワードやグループなどのユーザーアカウントを転送します。また、クライアントでの高価な設定更新を回避する場合もあります。

ここで説明する移行プロセスは、LDAP に 1 つ、IdM に 1 つの名前空間がある単純な導入シナリオを想定しています。複数の名前空間やカスタムスキーマなど、より複雑な環境では、Red Hat サポートサービスにお問い合わせください。

39.1. LDAP から IDM への移行の概要

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。

クライアントが Identity Management をどのように使用するかを決定する部分が、移行で最もコストがかかります。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。

つぎに、パスワードの移行方法の計画です。Identity Management では、パスワードに加えて、すべてのユーザーアカウントに 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。

39.1.1. 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

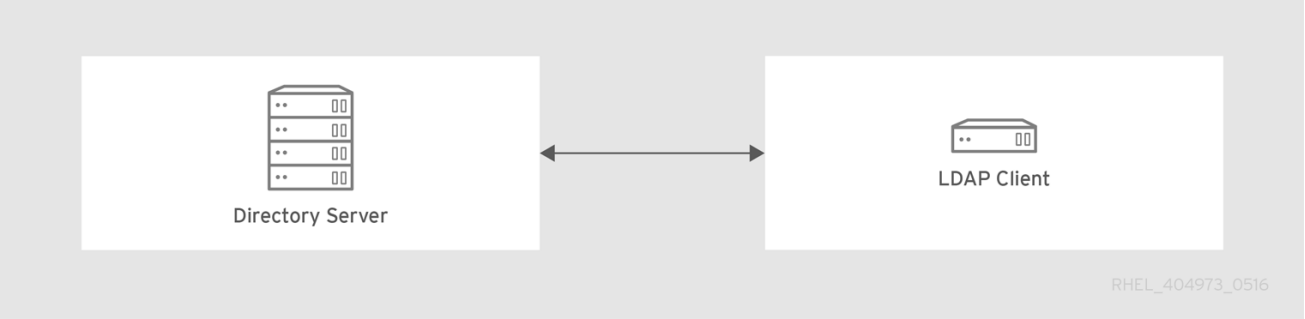
異なるクライアント設定は **相互に排他的とはなりません**。ほとんどの環境でクライアントが IdM ドメインへの接続に使用する方法はクライアントによって異なっています。管理者は各クライアント別に最適となるシナリオを決定しなければなりません。

39.1.1.1. 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39.1 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 ライブラリーを使用して LDAP サービスに直接接続を行います。これらのライブラリーにより、クライアントは、**/etc/passwd** または **/etc/shadow** にデーが格納されて**いるかのように** LDAP ディレクトリーからユーザー情報を取得できます。(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。)

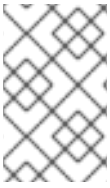
LDAP ディレクトリーと IdM サーバーの間には特にスキーマサポートとディレクトリーツリーに構造的な違いがあります。(これらの相違点の詳細は、「Identity Management と標準 LDAP ディレクトリーの比較」を参照してください。 こうした違いはデータ (特にエントリー名に影響するディレクトリーツリー) には影響する可能性がありますが **クライアントの設定** にはほとんど影響しないため、Identity Management にクライアントを移行させる上では実際にはほとんど影響がありません。

39.1.1.2. Red Hat Enterprise Linux クライアントの推奨設定

Red Hat Enterprise Linux には、SSSD (**System Security Services Daemon**) と呼ばれるサービスがあります。SSSD は、特別な PAM ライブラリーおよび NSS ライブラリー (**pam_sss** および **nss_ss**) を使用して、SSSD を Identity Management と密接に統合し、Identity Management の完全な認証およびアイデンティティー機能を利用できます。このライブラリーによって SSSD と ∏ の緊密な統合が行われ、∏ の認証機能および ID 機能をフル活用することが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 『System-Level Authentication Guide』 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。

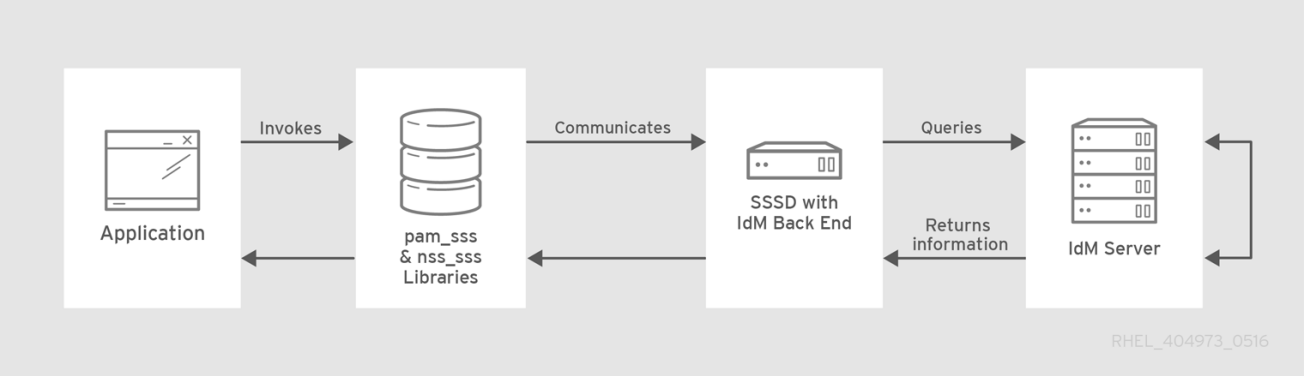
SSSD は、そのバックエンド機能すべてに Identity Management を使用できます。LDAP ID の汎用プロバイダーや Kerberos 認証とは異なり、多岐に渡る Identity Management の機能性をすべて利用することができます。たとえば、SSSD では 日常的な運用時に Identity Management でセキュリティー機能やホストベースのアクセス制御ルールを有効化させることができます。



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39.2 IdM バックエンドのあるクライアントおよび SSSD



ipa-client-install スクリプトは、その 4 つのバックエンドサービスすべてに IdM を使用するよう SSSD を自動的に設定するため、Red Hat Enterprise Linux クライアントはデフォルトで推奨される設定で設定されます。

注記

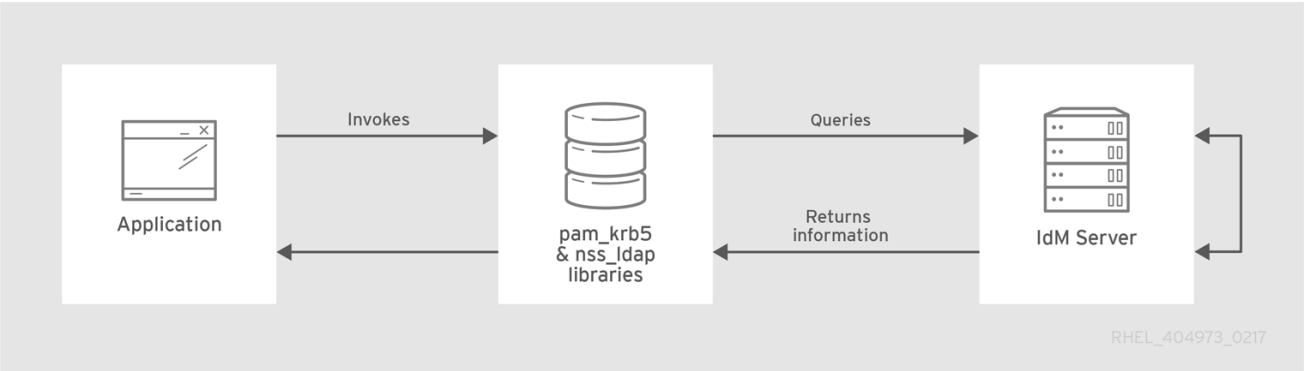
このクライアント設定は、最新バージョンの SSSD と **ipa-client** に対応している Red Hat Enterprise Linux 6.1 以降および Red Hat Enterprise Linux 5.7 以降でのみサポートされています。「推奨設定以外で対応している設定」 の説明に従って、Red Hat Enterprise Linux の古いバージョンを設定できます。

39.1.1.3. 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、古い Red Hat Enterprise Linux バージョン (6.1 および 5.6) は SSSD をサポートしますが、アイデンティティープロバイダーとして IdM に対応していない古いバージョンがあります。

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 検索用 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39.3 LDAP および Kerberos を使用するクライアントおよび IdM



Red Hat Enterprise Linux クライアントが古いバージョンの SSSD を使用している場合は、引き続き IdM サーバーをアイデンティティープロバイダーとその Kerberos 認証ドメインとして使用するように SSSD を設定できます。これは、『システムレベルの認証ガイド』の SSSD 設定を参照してください。

IdM ドメインクライアントは、**nss_ldap** および **pam_krb5** を使用して IdM サーバーに接続するように設定できます。共通する設定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、一般的には、クライアントで可能な限り安全な設定を使用することがベストプラクティスとなります。これは、ID の SSSD または LDAP、および認証の Kerberos を意味します。

39.1.2. パスワード移行のプランニング

LDAP から Identity Management への移行に影響を及ぼす可能性がある最も注目すべき問題は、ユーザーパスワードの移行です。

Identity Management (デフォルトでは) は認証に Kerberos を使用し、各ユーザーには、標準のユーザーパスワードに加えて、Identity Management Directory Server に保存されている Kerberos ハッシュが必要です。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。ユーザーを作成すると、パスワードがハッシュされ、Identity Management に保存される前に、平文で利用できるようになります。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

ユーザーは、IdM ドメインに対して認証したり、IdM リソースにアクセスしたり、Kerberos ハッシュがなくなるまでできません。

ユーザーが Kerberos ハッシュを持たない場合[6]ユーザーアカウントがある場合でも、そのユーザーは IdM ドメインにログインできません。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スですが、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。

39.1.2.1. 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 ハッシュでパスワードが作成されます。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。

39.1.2.2. 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。

39.1.2.3. 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、IdM サーバーに対して Kerberos 認証の実行を試みます。
3. ユーザーがシステムに存在していも 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、セキュアな接続でプレーンテキストの LDAP バインドを実行します。
5. IdM はこのバインド要求をインターセプトします。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ティティーに格納します。
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、エンティティー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ため、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。

39.1.2.4. 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ティティー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。

ユーザーが LDAP サーバーから IdM サーバーに接続すると、クリアテキストのパスワードは移行されません。ID 管理では、クリアテキストのパスワードは許可されません。Kerberos プリンシパルはユーザーに作成され、キータブは true に設定されます。また、パスワードは期限が切れたときに設定されます。つまり、Identity Management では、次回ログイン時にパスワードをリセットする必要があります。

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」および「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。

39.1.2.5. 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はじめて IdM ドメインでへの **kinit** を試行したときに自動的に行われます。

```
[jsmith@server ~]$ kinit
Password for jsmith@EXAMPLE.COM:
Password expired.  You must change it now.
Enter new password:
Enter it again:
```

39.1.3. 移行における考慮事項と要件

LDAP サーバーから Identity Management への移行を計画しているため、LDAP 環境が Identity Management の移行スクリプトで機能できることを確認してください。

39.1.3.1. 移行に対応している LDAP サーバー

LDAP サーバーから Identity Management への移行プロセスは、特別なスクリプト **ipa migrate-ds** を使用して移行を実行します。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 でテスト済みです。

注記

Microsoft Active Directory の場合、移行用スクリプトを使用した移行には**対応していません**。これは、LDAPv3-コンプライアントディレクトリーではないためです。Active Directory からの移行については、Red Hat Professional Services にお問い合わせください。

39.1.3.2. 移行環境に関する要件

Red Hat Directory Server と Identity Management には多くの異なる設定シナリオがあり、これらのシナリオのいずれかが移行プロセスに影響を及ぼす可能性があります。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が、1 つの IdM レルムに移行中です。統合はありません。
- ユーザーパスワードは、LDAP ディレクトリーにハッシュ形式で保存されます。サポートされるハッシュのリストは、**passwordStorageScheme** 属性 (表 19.2パスワードポリシー関連の属性、『Red Hat Directory Server 10 管理ガイド』) 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。クライアントマシンは、**pam_ldap** または **nss_ldap** を使用して LDAP サーバーに接続するように設定されます。
- エントリーは標準の LDAP スキーマのみを使用します。カスタムオブジェクトクラスまたはカスタム属性を含むエントリーは、Identity Managment に移行されません。

39.1.3.3. 移行 - IdM のシステム要件

中程度のサイズのディレクトリー (約 10,000 ユーザー、および 10 グループ) では、移行を続行するのに十分な強力なターゲットシステム (IdM システム) が必要です。移行の最小要件は以下のとおりです。

- 4 コア
- 4 GB のメモリー
- 30GB のディスク領域

- SASL バッファーサイズの 2MB (IdM サーバーのデフォルト)

  移行エラーが発生した場合は、バッファーサイズを大きくします。

  ```
  [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 をバイト単位で設定します。

39.1.3.4. sudo ルールに関する考慮事項

sudo を LDAP で使用している場合は、LDAP に保存されている **sudo**ルールを手動で移行する必要があります。Red Hat は、IdM のネットグループをホストグループとして再作成することを推奨します。IdM は、SSSD **sudo**プロバイダーを使用しない**sudo**設定で、従来のネットグループとしてホストグループを自動的に表示します。

39.1.3.5. 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 は特定の **ipa migrate-ds** コマンドを使用して移行プロセスを進めます。**ipa migrate-ds** を使用する場合は、**--bind-dn** オプションで指定するリモートシステムユーザーに、 ***userPassword*** 属性への読み取りアクセスが必要です。読み取りアクセスがないと、パスワードが移行されません。

Identity Management サーバーは移行モードで実行するように設定してから、移行スクリプトを使用することができます。詳細は、「LDAP サーバーの Identity Management への移行」 を参照してください。

39.1.3.6. 移行パフォーマンスの改善

LDAP の移行は、基本的には、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操作です。インポート操作のパフォーマンスを向上させるために、389 Directory Server インスタンスをチューニングすると、移行パフォーマンス全体を改善できます。

インポートのパフォーマンスに直接影響を与えるパラメーターは、以下の 2 つです。

- ***nsslapd-cachememsize*** 属性。エントリーキャッシュに使用できるサイズを定義します。これは、キャッシュメモリーの合計サイズの 80% に自動的に設定されるバッファーです。大量のインポート操作では、多数のエントリーや、より大きな属性のエントリーをより効率的に処理するために、このパラメーター (またはメモリーキャッシュ自体) を増やすことができます。

  ldapmodify を使用して属性を変更する方法は、『Red Hat Directory Server 10 パフォーマンスチューニングガイド』 の エントリーキャッシュサイズの設定 を参照してください。

- システム**ulimit**設定オプションは、システムユーザーに許可されるプロセスの最大数を設定します。大規模なデータベースの処理が制限を超える可能性があります。これが発生した場合は、値を上げます。

```
[root@server ~]# ulimit -u 4096
```

詳細は、Red Hat Directory Server の 『パフォーマンスチューニングガイド』 (https://access.redhat.com/documentation/ja-jp/red_hat_directory_server/11/html-single/performance_tuning_guide/index) を参照してください。

39.1.3.7. 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。

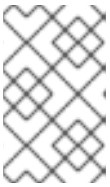
クライアントを先に移行する場合は SSSD を使用してクライアント設定を変更し、IdM サーバーを設定します。

1. SSSD をディプロイします。
2. クライアントが現在の LDAP サーバーに接続し IdM にフェイ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 **ipa migrate-ds** スクリプトを使用してユーザーデータを移行します。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、クライアントが Identity Management に透過的にフェイルオーバーできるようにします。

サーバーの移行では、LDAP から Identity Management への移行が最初に行われます。

1. IdM サーバーをインストールします。
2. IdM **ipa migrate-ds** スクリプトを使用してユーザーデータを移行します。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:**SSSD をディプロイします。
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動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。

39.2. IPA MIGRATE-DS を使用する例

データの移行は、**ipa migrate-ds** コマンドを使用して行われます。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
```

移行されたエントリー

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 コマンドを使用すると、データの識別およびエクスポート方法をカスタマイズできます。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。詳細については、**--help** をコマンドに渡します。

バインド DN

デフォルトでは、DN "**cn=Directory Manager**" は、リモート LDAP ディレクトリーにバインドするために使用されます。**--bind-dn** オプションをコマンドに渡して、カスタムバインド DN を指定します。詳細は 「移行ツール」 を参照してください。

コンテキストの変更の命名

Directory Server の命名コンテキストが Identity Management で使用されるものと異なる場合は、オブジェクトのベース DN が変換されます。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。**--base-dn** を **ipa migrate-ds** に渡して、移行に使用するリモート LDAP サーバーのベース DN を設定します。

39.2.1. 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ーエントリーになります。**migrate-ds** コマンドでオプションが渡されていない場合、ユーティリティーは、指定の LDAP ディレクトリーが **ou=People** および **ou=Groups** 構造を使用していることを前提とします。

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)。管理者が、ソース LDAP サーバーの別のユーザーまたはグループのサブツリーの RDN を指定できるようにするには、以下の 2 つのオプションがあります。

- **--user-container**
- **--group-container**

注記

いずれの場合もサブツーを 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。指定したコンテナーのエントリーのみが移行されます。
- **subtree**: 指定したコンテナーおよびすべてのサブコンテナーのエントリーが移行されます。
- **base**: 指定されたオブジェクト自体のみが移行されます。

39.2.2. 特定のエントリーのみを包含または除外

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を持つすべてのユーザーエントリーと、**groupOfUniqueNames** オブジェクトクラスまたは **groupOfNames** オブジェクトクラスを持つすべての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、これによりカスタム **fullTimeEmployee** オブジェクトクラスを持つユーザーのみが移行されます。

```
[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という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
```

exclude ステートメントは、***uid*** でパターンに一致するユーザーと、***cn*** 属性でパターン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
```

39.2.3. 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。帯域幅とネットワークの制約、または属性データが相互に関連しなくなったために、現実的な状況ではない場合があります。たとえば、ユーザーが IdM ドメインに参加する際に新しいユーザー証明書を割り当てる場合は、***userCertificate*** 属性を移行する必要はありません。

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用し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**

たとえば、ユーザーの ***userCertificate*** 属性および **strongAuthenticationUser** オブジェクトクラスとグループの **groupOfCertificates** オブジェクトクラスを除外するには、次のコマンドを実行します。

```
[root@ipaserver ~]# ipa migrate-ds --user-ignore-attribute=userCertificate --user-ignore-objectclass=strongAuthenticationUser --group-ignore-objectclass=groupOfCertificates ldap://ldap.example.com:389
```

注記

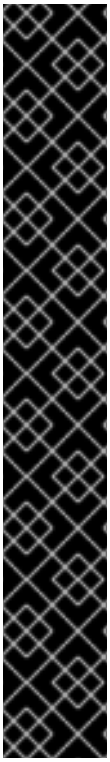
必要な属性が無視されていないか必ず確認します。また、オブジェクトクラスを除外する場合、そのオブジェクトクラスでしか対応しない属性はすべて除外するようにしてください。

39.2.4. 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、およびその他のネットワーク ID を定義します。ただし、移行のソースとして使用される LDAP サーバーが、代わりに RFC2307 スキーマを使用する場合は、**--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
```

39.3. 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。移行が正しく完了したことを確認するには、次のコマンドを実行します。

- **ipa user-add** コマンドを使用して IdM にテストユーザーを作成し、移行したユーザーの出力をテストユーザーと比較します。移行したユーザーに、テストユーザーに存在する属性およびオブジェクトクラスの最小セットが含まれていることを確認します。

```
$ ipa user-add TEST_USER
```

- IdM にあるように、移行したユーザーの出力を、元の LDAP サーバーにあるように、ソースユーザーと比較します。インポートした属性が倍にならず、期待値になっていることを確認してください。

```
$ ipa user-show --all TEST_USER
```

1. 既存の LDAP ディレクトリーとは異なるマシンに、カスタム LDAP ディレクトリースキーマなど、IdM サーバーをインストールします。

注記

カスタムユーザースキーマまたはカスタムグループスキーマの IdM でのサポートは限られています。互換性のないオブジェクト定義があると、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
```

移行中に compat ツリーによるデータが必要な場合は、この手順は必要ありません。

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できるように設定します。

```
[root@server ~]# ipa config-mod --enable-migration=TRUE
```

5. IdM 移行スクリプト **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** を使用する例」 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。

前の手順で compat プラグインを無効にしていない場合は、**--with-compat** を **ipa migrate-ds** に渡します。

情報のエクスポートが完了すると、命名コンテキストが異なる場合に、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. 移行前に compat プラグインが無効になっていた場合は、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:**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。詳細は、『System-Level Authentication Guide』 の Configuring a Kerberos Client を参照してください。

10. ハッシュした Kerberos パスワードを生成する方法には、2 つあります。「パスワード移行のプランニング」 の説明にように、いずれの場合も、ユーザーとの対話なしでユーザーパスワードが移行されます。

    a. SSSD の使用:

        i. SSSD がインストールされ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 クライアントでは、この **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
```

    b. IdM 移行 Web ページの使用

        i. 移行 Web ページを使用して IdM にログインするようにユーザーに指示します。

```
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一重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。

39.4. SSL 経由での移行

移行時に LDAP と IdM との間でデータ転送を暗号化するには、以下のコマンドを実行します。

1. リモート LDAP サーバー証明書を発行した CA の証明書を、IdM サーバーのファイルに保存します。たとえば、**/etc/ipa/remote.crt** です。

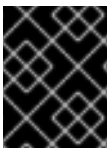
2. 「LDAP サーバーの Identity Management への移行」 に記載の手順に従ってください。ただし、移行時に暗号化された LDAP 接続の場合、URL で ***ldaps*** プロトコルを使用し、コマンドに **--ca-cert-file** オプションを渡します。以下に例を示します。

```
[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 LDAP 認証を使用することが可能です。つまり、Kerberos ハッシュはユーザーには必要ありません。ただし、これにより Identity Management の機能が制限されるため、推奨されません。

第40章 非 RHEL LINUX ディストリビューション上の FREEIPA から RHEL 7 上の IDM への移行

非 RHEL Linux ディストリビューション上の FreeIPA デプロイメントを RHEL 7 サーバー上の Identity Management (IdM) デプロイメントに移行するには、最初に新しい RHEL 7 IdM 認証局 (CA) レプリカを既存の FreeIPA 環境に追加し、証明書関連の機能をレプリカに移行してから、非 RHEL FreeIPA サーバーを廃止する必要があります。



重要

Convert2RHEL ツールを使用した、非 RHEL FreeIPA サーバーから RHEL 7 IdM サーバーへのインプレース変換の実行はサポートされていません。

前提条件

- 非 RHEL FreeIPA 認証局 (CA) 更新サーバーのドメインレベルを確認している。詳細は、現在のドメインレベルの表示 を参照してください。
- 新しい CA 更新サーバーとなるシステムに RHEL 7.9 がインストールされている。

手順

移行を実行するには、Red Hat Enterprise Linux 6 からバージョン 7 への Identity Management の移行 と同じ手順に従います。ただし、ここでは非 RHEL FreeIPA CA サーバーが RHEL 6 サーバーの役割を果たします。

1. 元の非 RHEL CA 更新サーバーが FreeIPA バージョン 3.1 以前を実行している場合は、Identity Management スキーマを更新 します。インストールされている FreeIPA のバージョンを表示するには、***ipa --version*** コマンドを使用します。
2. RHEL 7 サーバーを設定し、非 RHEL Linux ディストリビューション上の現在の FreeIPA 環境に IdM レプリカとして追加します。ドメインレベルが 0 の場合は、RHEL 7 レプリカのインストール を参照してください。ドメインレベルが 1 の場合は、 レプリカの作成: 概要 で説明されている手順に従います。
3. RHEL 7 レプリカを CA 更新サーバーにし、非 RHEL サーバーでの証明書失効リスト (CRL) の生成を停止し、CRL 要求を RHEL 7 レプリカにリダイレクトします。詳細は、CA サービスの Red Hat Enterprise Linux 7 サーバーへの移行 を参照してください。
4. 元の非 RHEL FreeIPA CA 更新サーバーを停止して、新しい RHEL 7 サーバーへのドメイン検索を実施します。詳細は、Red Hat Enterprise Linux 6 サーバーの停止 を参照してください。
5. 他の RHEL 7 システムに新しいレプリカをインストールし、非 RHEL サーバーの使用を停止します。詳細は、マスター CA サーバーの移行後の次のステップ を参照してください。

重要

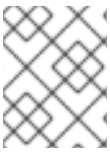
Red Hat では、トポロジー内に 1 つの RHEL メジャーバージョンの IdM レプリカのみを含めることを推奨しています。このため、古いサーバーの使用はすみやかに停止してください。

関連情報

- Red Hat Enterprise Linux 6 からバージョン 7 への Identity Management の移行

付録A トラブルシューティング: 一般的なガイドライン

この付録では、ログやサービスの状態をクエリーするなどして、問題の根本原因を特定する一般的な手順を説明します。



注記

特定の問題とそのソリューションのリストは、付録B *トラブルシューティング - 特定の問題のソリューション* を参照してください。

この問題が発生したときにどのような操作をしていましたか？

- **ipa** ユーティリティーを使用したコマンドの実行
- **kinit**を使用した認証
- IdM Web UI の認証
- スマートカードによる認証
- サービスの起動

問題の原因となっている IdM の特定領域がわかっている場合は、以下のリンクを参照してください。

- DNS
- レプリケーション

本書で問題の特定、解決ができない場合には、カスタマーケースを起票してください。その際、ケースレポートに、このトラブルシューティング手順で判断した、主なエラー出力も含めるようにしてください。Contacting Red Hat Technical Support も参照してください。

A.1. IPA ユーティリティーの実行時に障害の調査

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ます。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ます。これにより、JSON の応答と要求が表示されます。

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」 に、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操作するコンポーネントを示します。このようなコンポーネントをクエリーすると、問題が発生した場所と、その原因を調査するのに役立ちます。

1. 次のユーティリティーを使用します。
   - IdM サーバーまたはクライアントの DNS 解決を確認する **ホスト**
   - IdM サーバーが利用可能かどうかを確認する **ping**
   - **iptable** を使用して、IdM サーバーの現在のファイアウォール設定を確認します。
   - 現在の時間を確認する **日付**
   - 「ポートの要件」で記載されているように、必要なポートへの接続は **nc** です。

このユーティリティーの使用方法は、man ページを参照してください。

2. **KRB5_TRACE** には、trace-logging の出力を **/dev/stdout** に送信するための **/dev/stdout** ファイルを設定します。

```
$ KRB5_TRACE=/dev/stdout ipa cert-find
```

Kerberos キー配布センター (KDC) のログを確認します (**/var/log/krb5kdc.log**)。

3. Apache エラーログを確認します。

   a. サーバーでデバッグレベルを有効にする: **/etc/ipa/server.conf** ファイルを開き、**[glob]** セクションに ***debug=True*** オプションを追加します。

   b. **httpd** サービスを再起動します。

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

   c. 再度失敗したコマンドを実行します。

   d. サーバーの **httpd** エラーログ (**/var/log/httpd/error_log**) を確認します。

-vvv オプションを指定してコマンドを実行し、HTTP 要求および応答を表示します。

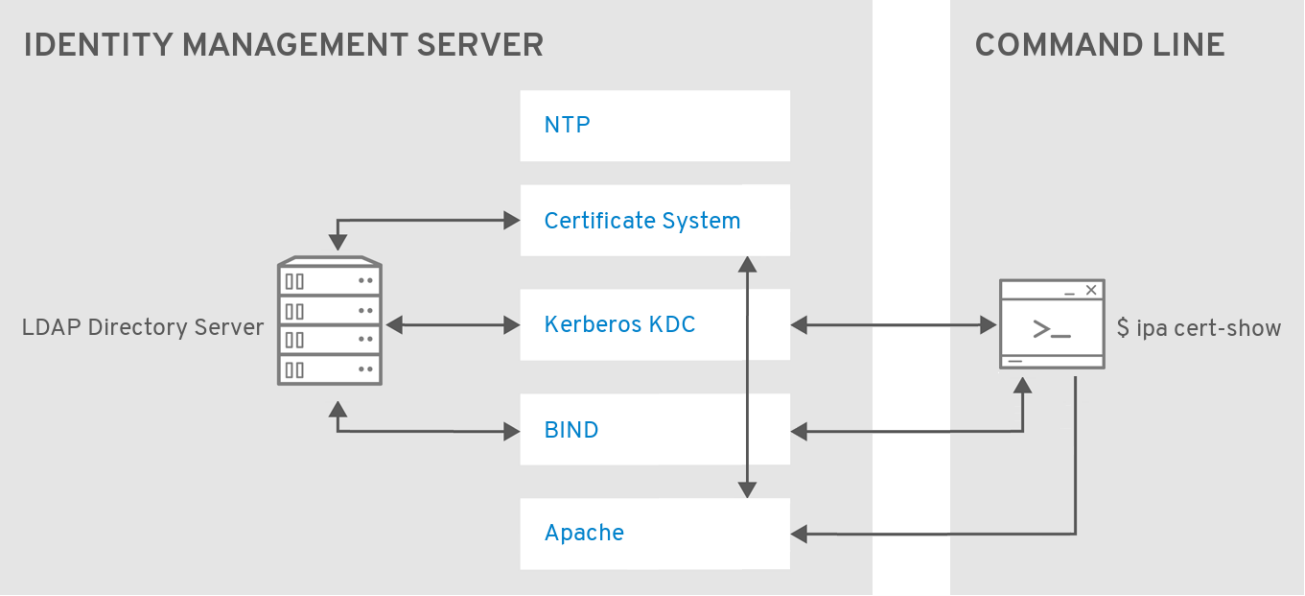
4. Apache アクセスログ (**/var/log/httpd/access_log**) を確認します。

証明書システムコンポーネントのログを確認します。

- **/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** を実行して、**ジャーナル**ログを確認します。

5. ディレクトリーサーバーのアクセスログ (**/var/log/dirsrv/slapd-*IPA-EXAMPLE-COM*/access**) を確認します。

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

- Identity Management のログファイルの詳細は、「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 認証の失敗の調査

一般的なトラブルシューティング

1. IdM クライアントで、**kinit** プロセスからのデバッグメッセージを表示します。

```
$ KRB5_TRACE=/dev/stdout kinit admin
```

2. 以下を確認します。

   - サーバーと、影響を受けるクライアントの両方で、クライアント転送レコードが正しいこと。

```
# host client_fully_qualified_domain_name
```

   - サーバーと、影響を受けるクライアントの両方で、サーバー転送レコードが正しいこと。

```
# host server_fully_qualified_domain_name
```

```
# host server_IP_address
```

   host *server_IP_address* は、完全修飾のホスト名を返します。ホスト名の末尾にはピリオドを使用します。以下に例を示します。

```
server.example.com.
```

3. クライアントで **/etc/hosts** ファイルを確認し、以下を確認します。

   - ファイル内のすべてのサーバーエントリーが正しいこと。

- すべてのサーバーエントリーで、最初の名前は完全修飾ドメイン名となっていること。

「/etc/hosts ファイル」も参照してください。

4. 「ホスト名および DNS 設定」 の他の条件を満たしていることを確認してください。

5. IdM サーバーで、**krb5kdc** サービスおよび **dirsrv** サービスが実行してい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) のログを確認します (**/var/log/krb5kdc.log**)。

7. KDC が、**/etc/krb5.conf** ファイルにハードコードされている場合 (ファイルが明示的に KDC ディレクティブを設定し、**dns_lookup_kdc = false** 設定を使用する場合) は、マスターサーバーごとに **ipactl のステータス** コマンドを使用します。コマンドで、KDC としてリスト表示されている各サーバーの IdM サービスのステータスを確認します。

```
# 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 認証が **Cannot find KDC for realm "EXAMPLE.COM" while getting initial credentials** というエラーで失敗した場合は、KDC がサーバーで実行していないか、クライアントが DNS を誤って設定していることを示しています。このような状況では、以下の手順を試してください。

1. **/etc/krb5.conf** ファイルで DNS 検出が有効になっている場合 (**dns_lookup_kdc = true** 設定) は、**dig** ユーティリティーを使用して、以下のレコードが解決可能かどうかを確認します。

```
$ dig -t TXT _kerberos.ipa.example.com
$ dig -t SRV _kerberos._udp.ipa.example.com
$ dig -t SRV _kerberos._tcp.ipa.example.com
```

以下の例では、上記の **dig** コマンドのいずれかが失敗しています。

```
; <<>> 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、**名前** サービスがマスターサーバーで実行していないことを示していました。

2. DNS ルックアップが失敗した場合は、「DNS のトラブルシューティング」 の手順に進みます。

関連情報

- Identity Management のログファイルの詳細は、「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 認証エラーの調査

1. **kinit** ユーティリティーを使用して、コマンドラインからユーザーを認証できることを確認します。認証に失敗した場合は、「**kinit** 認証の失敗の調査」 も併せて参照してください。
2. 影響を受けるサーバーで、**httpd** サービスと **dirsrv** サービスが実行していることを確認します。

```
# systemctl status httpd.service
# systemctl status dirsrv@IPA-EXAMPLE-COM.service
```

3. 関連する SELinux アクセスベクトルキャッシュ (AVC) メッセージが **/var/log/audit/audit.log** ファイルおよび **/var/log/messages** ファイルに記録されていないことを確認します。

   AVC メッセージの解決方法は、Red Hat ナレッジベースのCLI での SELinux の基本的なトラブルシューティングを参照してください。
4. 認証元のブラウザーで cookie が有効になっていることを確認します。
5. IdM サーバーと、認証しているシステムの時間差が、最大 5 分であることを確認してください。
6. Apache エラーログ **/var/log/httpd/error_log** を確認します。
7. 認証プロセスの詳細なロギングを有効にして、問題の診断に役立てます。Firefox で詳細なログを有効にする方法は、『システムレベル認証ガイド』 の Firefox Kerberos 設定のトラブルシューティング を参照してください。

証明書の使用中にログインできない場合は、以下の点に注意してください。

1. **/etc/httpd/conf.d/nss.conf** で、***LogLevel*** 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. 証明書を使用してログインし直してください。
4. Apache エラーログ **/var/log/httpd/error_log** を確認します。

   ログには、**mod_lookup_identity** モジュールが記録したメッセージと、ログイン試行時にモジュールとユーザーが問題なくマッチしたことを示す情報が表示されます。

関連情報

- Identity Management のログファイルの詳細は、「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. スマートカードの認証エラーの調査

1. **/etc/sssd/sssd.conf** を開き、**debug_level** を **2** に設定します。

2. **sssd_pam.log** ファイルおよび **sssd_*EXAMPLE.COM.* log** ファイルを確認します。ファイルにタイムアウトのエラーメッセージが表示される場合は、「スマートカード認証のタイムアウトエラーメッセージの表示」 を参照してください。

A.5. サービスの起動に失敗した理由の調査

1. 起動に失敗したサービスのログを確認します。「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、ディレクトリーサーバーのログは**/var/log/dirsrv/slapd-*IPA-EXAMPLE-COM*/errors** にあります。

2. サービスを実行しているサーバーに、完全修飾ドメイン名 (FQDN) があることを確認してください。「サーバーのホスト名の確認」を参照してください。

3. **/etc/hosts** ファイルに、サービスを実行しているサーバーのエントリーが含まれる場合は、完全修飾ドメイン名が最初に記載されていることを確認してください。「**/etc/hosts** ファイル」も参照してください。

4. 「ホスト名および DNS 設定」 の他の条件を満たしていることを確認してください。

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を判断します。たとえば、**dirsrv** サービスチケットの場合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 と一致していることを確認します。

   b. 上記のサービスキータブに表示されているキー (KVNO) のバージョンが、サーバーキータブの KVNO と一致していることを確認してください。サーバーのキータブを表示するには、次のコマンドを実行します、

```
$ kinit admin
$ kvno ldap/server.example.com@EXAMPLE.COM
```

   c. クライアントの正引き (A、AAAA、またはその両方) レコードと逆引きレコードが、表示されているシステム名とサービスプリンシパルに一致することを確認します。

6. クライアントの正引き (A、AAAA、またはその両方) レコード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のシステム時間の誤差が、最大 5 分であることを確認します。

8. IdM 管理サーバーの証明書の期限が切れると、サービスが起動できなくなることがあります。ご使用のケースでこれが原因であるかを確認するには、次のコマンドを実行します。

a. **getcert リスト** コマンドを使用して、**certmonger** ユーティリティーが追跡する証明書をすべてリスト表示します。

b. この出力で、IdM 管理証明書 (**ldap** サーバー証明書および **httpd** サーバー証明書) を見つけます。

c. **status** と **expires** というラベルが付けられたフィールドを確認します。

```
# 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/pwdfile.txt'
 certificate: type=NSSDB,location='/etc/dirsrv/slapd-IPA-EXAMPLE-
COM',nickname='Server-Cert',token='NSS Certificate DB'
 CA: IPA
 issuer: CN=Certificate Authority,O=IPA.EXAMPLE.COM
 subject: CN=ipa.example.com,O=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/pwdfile.txt'
 certificate: type=NSSDB,location='/etc/httpd/alias',nickname='Server-Cert',token='NSS
Certificate DB'
 CA: IPA
 issuer: CN=Certificate Authority,O=IPA.EXAMPLE.COM
 subject: CN=ipa.example.com,O=IPA.EXAMPLE.COM
 expires: 2019-04-22 13:05:35 UTC
[... output truncated ...]
```

証明書が期限切れになっても起動する必要がある場合は、「IdM が期限切れの証明書で起動できるようにする」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 の問題の多くは、設定の誤りが原因で発生します。したがって、「ホスト名および DNS 設定」 の条件を満たしていることを確認してください。

2. **dig** ユーティリティーを使用して、DNS サーバーから応答を確認します。

```
# 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** ユーティリティーを使用して DNS 名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** コマンドを使用して、LDAP 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 ツールを使用して DNS を管理する方法は、33章*DNS の管理* を参照してください。

5. BIND を再起動して、LDAP と強制的に再同期します。

```
$ systemctl restart named-pkcs11
```

6. 必要な DNS レコードのリストを取得します。

```
$ ipa dns-update-system-records --dry-run
```

dig ユーティリティーを使用して、表示されているレコードが DNS に存在するかどうかを確認します。Identity Management DNS を使用する場合は、**ipa dns-update-system-records** コマンドを使用して、足り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

2 つ以上のサーバーでレプリケーションをテストします (「新規レプリカのテスト」を参照)。ある IdM サーバーで行った変更が、別のサーバーに複製されない場合は、次のコマンドを実行します。

1. 「ホスト名および DNS 設定」 の条件を満たしていることを確認してください。

2. 両方のサーバーが、互いの 正引き/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. 両サーバーの誤差が、最大 5 分であることを確認してください。

4. 両方のサーバーのディレクトリーサーバーエラーログ (**/var/log/dirsrv/slapd-*SERVER-EXAMPLE-COM*/errors**) を確認します。

5. Kerberos に関連するエラーが表示された場合は、Directory Server のキータブが正しいことと、それを使用して別のサーバーにクエリーできることを確認します (この例では **server2**)。

```
[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

- Identity Management のログファイルの詳細は、「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 - 特定の問題のソリューション

トラブルシューティングのアドバイス

- サーバー (「Identity Management サーバー」 を参照)
- レプリカ (「Identity Management レプリカ」 を参照)
- クライアント (「Identity Management クライアント」 を参照)
- 認証 (「ログインと認証の問題」を参照)
- vault (「Vault」 を参照)

B.1. IDENTITY MANAGEMENT サーバー

B.1.1. 外部 CA のインストールに失敗する

ipa-server-install --external-ca コマンドが、次のエラーにより失敗します。

```
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 を実行すると、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** 環境変数が原因でサーバーをインストールできません。

解決方法:

1. 次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. 失敗した IdM サーバーのインストールを削除します。

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

4. **ipa-server-install --external-ca** を再度実行します。

B.1.2. named デーモンの起動失敗

統合 DNS を使用して IdM サーバーをインストールした後、**named-pkcs11** が起動しません。**/var/log/messages** ファイルには、**named-pkcs11** サービスおよび **ldap.so** ライブラリーに関連するエラーメッセージが含まれています。

```
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 がインストールされており、**named-pkcs11** が起動しないようになっています。

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 サービスを再起動します。

```
# ipactl restart
```

B.1.3. IPv6 が無効になっているシステムにサーバーをインストールできない

IPv6 が無効になっているシステムに 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時に次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実行には、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:
システムで IPv6 を有効にします。詳しくは、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には？を参照してください。

IPv6 は、Red Hat Enterprise Linux 7 システムでデフォルトで有効になっている点に注意してください。

B.2. IDENTITY MANAGEMENT レプリカ

本ガイドでは、Red Hat Enterprise Linux における Identity Management のレプリケーションの一般的な問題を説明します。

関連情報:

- レプリケーションの動作をテストする方法は、「新規レプリカのテスト」 を参照してください。

- レプリケーションの競合を解決する方法に関するアドバイスは、「レプリカ合意の作成と削除」 を参照してください 。詳細は、Directory Server 『管理ガイド のセクション 15.26 の Common Replication Conflicts』 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行中のステータスを表示します。これは、レプリケーションの問題のトラブルシューティングに役立ちます。詳細は、Directory Server 『管理ガイド の セクション 15.24、レプリケーションの状態の監視 を参照してください』。

- 2 つの Directory Server インスタンスが同期しているかどうかを確認するには 、Directory Server 『管理ガイドのセクション 15.25 の "Comparing Two Directory Server Instances" 』 を参照してください。

B.2.1. 新しいレプリカに対する AD ユーザーの認証に失敗する

Identity Management - Active Directory 信頼設定に新しいレプリカをインストールした後に IdM レプリカに対する Active Directory (AD) ユーザーの認証に失敗します。

エラー内容:
レプリカが、信頼コントローラーでも信頼エージェントでもありません。これが原因で、AD の信頼から情報を提供することができません。

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。『Windows Integration Guide』のTrust Controllers and Trust Agentsを参照してください。

B.2.2. レプリカが Directory Server ログの SASL エラー、GSS-API エラー、および Kerberos エラーで起動する

レプリカが起動すると、一連の SASL バインドエラーが Directory Server (DS) ログに記録されます。エラーでは、認証情報キャッシュが見つからないため、GSS-API 接続に失敗したと表示されます。

```
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、サーバーが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表示されることもあります。

```
set_krb5_creds - Could not get initial credentials for principal [ldap/ replica1.example.com] in keytab
[WRFILE:/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、Kerberos 認証情報キャッシュをメモリーに保持します。IdM レプリカの停止など、DS プロセスが終了すると、認証情報キャッシュは破棄されます。

レプリカが再起動すると、KDC サーバーが起動する前に DS が起動します。この起動順序のため、DS の起動時に、Kerberos 認証情報が認証情報キャッシュに保存されません。これがエラーの原因になります。

初期障害の後、DS は、KDC の起動後に GSS-API 接続の確立を再試行します。2 回目の試行に成功して、レプリカが期待どおりに機能することを確認します。

GSS-API 接続が正常に確立され、レプリカが期待どおりに機能する場合は**上記の起動エラーは無視できます。**以下のメッセージは、接続が成功したことを示しています。

```
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

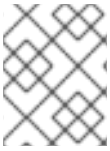
新しいレプリカを設定すると、一連の証明書エラーでインストールが失敗し、その後に DNS 正引きレコードが逆引きアドレスと一致しないことを示す DNS エラーが表示されます。

```
ipa: DEBUG: approved_usage = SSLServer intended_usage = SSLServer
ipa: DEBUG: cert valid True for "CN=replica.example.com,O=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が使用されています。DNS 規格ではこのような設定が許可されていますが、IdM レプリカのインストールに失敗します。

解決方法:
「正引きおよび逆引き DNS 設定の確認」 の説明に従って、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** で適用できます。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。

証明書のシリアル番号が見つからないことを示すエラーが、複製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のレプリカ合意が削除されましたが、データのレプリカ合意が依然として存在します。レプリカはいずれも依然として証明書を発行していますが、証明書に関する情報は複製されなくなりました。

状況例:

1. レプリカ A がホストに証明書を発行します。
2. レプリカには証明書のレプリケーション合意が確立されていないため、証明書はレプリカ B にレプリケートされません。
3. ユーザーがレプリカ B を使用してホストを管理しようとします。

4. レプリカ B は、ホストの証明書のシリアル番号を検証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ホストに関する情報がありますが、証明書ディレクトリーにホスト証明書がないためです。

解決方法:

1. **ipa-csreplica-manage connect** を使用して、2 つのレプリカ間で証明書サーバーのレプリケーションを有効にします。「レプリカ合意の作成と削除」を参照してください。

2. レプリカのいずれかを別のレプリカから初期化しなおして、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

初期化しなおすと、初期化したレプリカのデータが、別のレプリカのデータで上書きされます。情報の一部が失われる可能性があります。

B.2.5. Replica Update Vector (RUV) エラーの削除

注記

このソリューションは、ドメインレベル **0** で適用できます。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。

IdM トポロジーからレプリカを削除すると、古くなった RUV レコードが残りのレプリカ 1 つ以上に存在するようになります。

考えられる原因:

- 「レプリカ合意の削除」 で説明されているように、レプリカ合意を最初に適切に削除せずに、レプリカが削除されました。
- 別のレプリカがオフラインのときに、レプリカが削除されました。

エラー内容:
その他のレプリカでは、削除されたレプリカからも更新を受け取ることが想定されています。

注記

レプリカを削除する正しい手順は、「レプリカの削除」 に記載されています。

解決方法:
更新を受け取ることが予想されるレプリカの RUV レコードを削除します。

1. **ipa-replica-manage list-ruv** を使用して、古い RUV に関する詳細をリスト表示します。このコマンドは、レプリカ ID を表示します。

```
# 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*** コマンドを使用して、破損した RUV を消去します。このコマンドは、指定したレプリカに関連付けられている RUV を削除します。

   古い RUV を持つすべてのレプリカに対してこのコマンドを繰り返します。以下に例を示します。

```
# 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 の使用には特別な注意を払って続行してください。有効なレプリカ ID を指定してコマンドを実行すると、レプリカデータベース内のそのレプリカに関連するすべてのデータが破損します。

この場合は、「レプリカの再初期化」の説明に従って、別のレプリカからレプリカを再初期化します。

3. **ipa-replica-manage list-ruv** を再度実行します。

   - このコマンドで破損した RUV が表示されなくなった場合は、レコードが正常に消去されています。
   - このコマンドで、依然として破損した RUV が表示される場合は、このタスクを使用して手動で削除します。

```
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
```

RUV を消去するレプリカがわからない場合は、次のコマンドを実行します。

1. すべてのサーバーで、アクティブなレプリカ ID がないかを検索します。破損していない、信頼できるレプリカ ID のリストを作成します。

   有効なレプリカの ID を確認するには、トポロジー内のすべてのノードに対してこの LDAP クエリーを実行します。

```
# ldapsearch -p 389 -h ldM_node -D "cn=directory manager" -W -b "cn=config" "
(objectclass=nsds5replica)" nsDS5ReplicaId
```

2. すべてのサーバーで **ipa-replica-manage list-ruv** を実行します。破損していないレプリカ ID のリストにないレプリカ ID がある場合は注意してください。

3. 破損したすべてのレプリカ ID に対して **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** で適用できます。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。

CA がサーバー 1 台しかインストールされていませんでした。このサーバーに障害が発生して、CA サーバーがなくなってしまいました。

エラー内容:
IdM ドメインの CA 設定が利用できなくなりました。

解決方法:
使用可能な元の CA サーバーのバックアップがある場合は、サーバーを復元して、レプリカに CA をインストールできます。

1. バックアップから CA サーバーを復元します。詳細は「バックアップの復元」を参照してください。

   これにより、CA サーバーがレプリカで使用できるようになります。

2. レプリカの競合を回避するため、初期サーバーとレプリカとの間のレプリカ合意を削除します。「レプリカ合意の作成と削除」を参照してください。

3. レプリカに CA を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. 元の CA サーバーを停止します。「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と、サーバーに障害が発生したときに CA 設定が失われ、復旧できません。

B.3. IDENTITY MANAGEMENT クライアント

本セクションでは、Red Hat Enterprise Linux における IdM で発生する一般的なクライアントの問題を説明します。

関連情報:

- **/etc/sssd.conf**を検証するには、『システムレベル認証ガイド』のSSSD 設定の検証を参照してください。

B.3.1. 外部 DNS の使用時にクライアントが逆引き参照を解決できない

外部 DNS サーバーが、IdM サーバーの間違ったホスト名を返します。IdM サーバーに関連する次のエラーが、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 サーバーの間違ったホスト名を返すか、まったく応答を返しません。

解決方法:

1. DNS 設定を確認し、IdM が使用する DNS ドメインが適切に委譲されていることを確認します。詳細は「ホスト名および DNS 設定」を参照してください。

2. 逆引き (PTR) DNS レコードの設定を確認します。詳細は33章*DNS の管理*を参照してください。

B.3.2. クライアントは DNS ゾーンに追加されません。

ipa-client-install ユーティリティーを実行していると、**nsupdate** ユーティリティーがクライアントを DNS ゾーンに追加できません。

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 委譲の設定を確認します。詳細は「ホスト名および DNS 設定」を参照してください。

2. IdM ゾーンで動的更新が許可されていることを確認します。詳細は「ダイナミック DNS 更新の有効化」を参照してください。

IdM での DNS の管理の詳細は、「逆引き DNS ゾーンの管理」 を参照してください。Red Hat Enterprise Linux で DNS を管理する方法は、『ネットワークガイド』 のゾーンファイルの編集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。**getent passwd admin** コマンドなどを使用してユーザーおよびグループインフォメーションにアクセスしようとすると、失敗します。

エラー内容:
クライアント認証の問題は、多くの場合、SSSD (System Security Services Daemon) サービスの問題を示しています。

解決方法:
/var/log/sssd/ ディレクトリーの SSSD ログを確認します。ディレクトリーには、**sssd_*example.com*.log** などの DNS ドメインのログファイルが含まれています。

ログに十分な情報が含まれていない場合は、ログレベルを上げます。

1. **/etc/sssd/sssd.conf** で **[domain/*example.com*]** を探します。**debug_level** を調整して、詳細をログに記録します。

```
debug_level = 9
```

2. **sssd** サービスを再起動します。

```
# systemctl start sssd
```

3. **sssd_*example.com*. log** を再度確認します。このファイルには、より多くのエラーメッセージが含まれるようになりました。

B.4. ログインと認証の問題

B.4.1. ipa コマンドの実行時の Kerberos GSS 障害

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と、Kerberos エラーが発生します。以下に例を示します。

```
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 設定を確認します。

- IdM サーバーの DNS 要件については、「ホスト名および DNS 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 およびレルム設定』 を参照してください。

B.4.2. GSS-API の使用時に SSH 接続が失敗する

ユーザーは、SSH を使用して IdM マシンにログインできません。

エラー内容:
SSH が GSS-API をセキュリティー方法として使用して IdM リソースに接続しようとすると、GSS-API が最初に DNS レコードを検証します。SSH の問題は多くの場合、逆引き DNS エントリーが間違っていることが原因で発生します。レコードが正しくないと、SSH が IdM リソースを特定できません。

解決方法:
「ホスト名および DNS 設定」 の説明に従って DNS 設定を確認します。

一時的な回避策として、SSH 設定で逆引き DNS 参照を無効にすることもできます。これを行うには、**/etc/ssh/ssh_config** で ***GSSAPITrustDNS*** を **no** に設定します。逆引きの DNS レコードを使用する代わりに、SSH は指定したユーザー名を GSS-API に直接渡します。

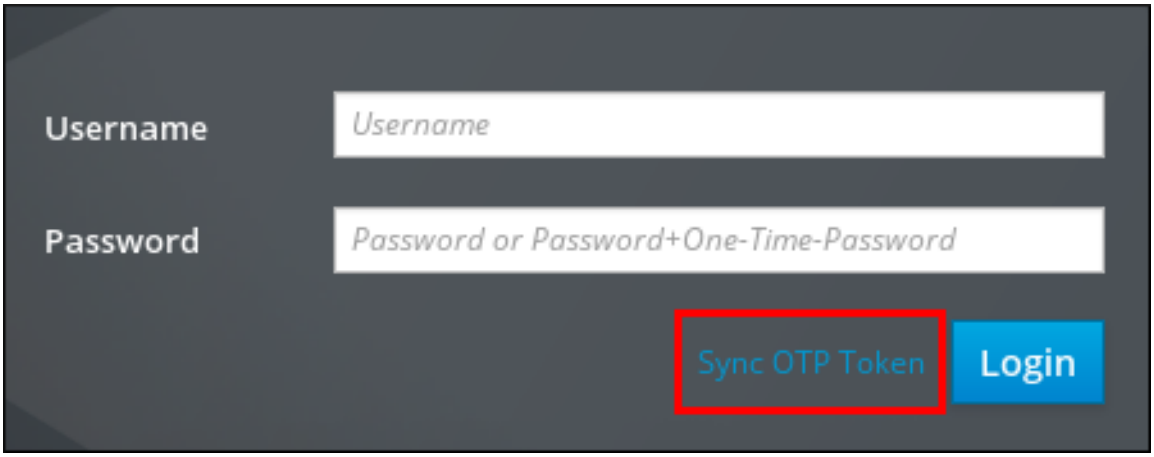
B.4.3. 同期されていない OTP トークン

トークンが同期されていないため、OTP を使用した認証に失敗します。

解決方法:
トークンを再同期します。どのユーザーも、トークンの種類、およびトークン設定を変更する権限があるかどうかに関係なく、トークンを再同期できます。

1. IdM Web UI で、ログインページで **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 **ipa otptoken-sync** コマンドを実行します。

2. トークンを再同期するために必要な情報を指定します。たとえば、IdM は、標準パスワードと、トークンにより生成された後続のトークンコード 2 回提示するように求められます。

注記

再同期は、標準パスワードが期限切れになっても機能します。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ら、IdM にログインして、パスワードの変更を求めるシステムプロンプトを表示します。

B.4.4. スマートカード認証のタイムアウトエラーメッセージの表示

sssd_pam.log ファイルおよび **sssd_*EXAMPLE.COM*.log** ファイルには、以下のようなタイムアウト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または OCSP (Online Certificate Status Protocol) を使用する場合は、スマートカードでユーザーを認証できるように、デフォルト値の調整が必要になる場合があります。

解決方法:
ユーザーを認証するサーバーおよびクライアントで、**/etc/sssd/sssd.conf** ファイルーを変更します。

1. **[pam]** で、**p11_child_timeout** を 60 秒に増やします。
2. **[domain/*EXAMPLE.COM*]** で、**krb5_auth_timeout** 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 サーバーに到達可能であることを確認してください。OCSP サーバーに直接到達できない場合は、以下のオプションを **/etc/sssd/sssd.conf** に追加して、プロキシー OCSP サーバーを設定します。

   ```
   certificate_verification = ocsp_default_responder=http://ocsp.proxy.url,
   ocsp_default_responder_signing_cert=nickname
   ```

   nickname は、**/etc/pki/nssdb/** ディレクトリー内の OCSP 署名証明書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は、sssd.conf(5) の man ページを参照してください。
4. SSSD を再起動します。

   ```
   # systemctl restart sssd.service
   ```

B.5. VAULT

B.5.1. 十分な追加権限がないことが原因で Vault にユーザーがアクセスできない

ユーザーは、自分のユーザー Vault にアクセスしたり、新しいユーザー Vault を追加したりできません。以下のエラーメッセージが表示されます。

```
ipa: ERROR: Insufficient access: Insufficient 'add' privilege to add the entry
'cn=testvault,cn=user,cn=users,cn=vaults,cn=kra,dc=example,dc=com'.
```

エラー内容:
ユーザーの格納域コンテナーは、別のユーザーが所有しています。通常、この状態は **admin** などの別のユーザーが、最初のユーザーのユーザー Vault を初めて作成した後に発生します。最初のユーザーは、自分の格納域コンテナーの Vault にはアクセスできません。

解決方法:
目的のユーザーを、Vault コンテナーの所有者として追加します。

1. **admin** としてログインします。

   ```
   $ kinit admin
   ```
2. *ユーザー* をコンテナーの所有者として追加します。

   ```
   $ ipa vaultcontainer-add-owner --user=user --users=user
     Owner users: admin, user
     Vault user: user
   ------------------------
   Number of owners added 1
   ------------------------
   ```

admin と *ユーザー* の両方がコンテナーの所有者であるため、ユーザーの vault コンテナーにアクセスできるようになりました。

3. **オプション:**ユーザーが、新しいユーザー vault を作成できるようになったことを確認します。

```
$ kinit user
$ ipa vault-add testvault2
----------------------
Added vault "testvault2"
----------------------
```

関連情報

- 「ユーザーの個人シークレットの保存」

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の設定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	説明
/etc/ipa/	メインの IdM 設定ディレクトリー。
/etc/ipa/default.conf	IdM の主な設定ファイル。サーバーとクライアントの起動時、およびユーザーが **ipa** ユーティリティーを使用する際に参照されます。
/etc/ipa/server.conf	オプションの設定ファイル (デフォルトでは存在しません)。IdM サーバーの起動時に参照されます。 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。
/etc/ipa/cli.conf	オプションの設定ファイル (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しているときに参照されます。 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。
/etc/ipa/ca.crt	IdM サーバーの CA が発行する CA 証明書。
~/.ipa/	IdM コマンドの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/** でユーザー固有の**default.conf** ファイル、**server.conf** ファイル、または **cli.conf** ファイルを作成して、個別の設定オーバーライドを指定できます。
/etc/sssd/sssd.conf	IdM ドメインおよび SSSD が使用する IdM サービスの設定。
/usr/share/sssd/sssd.api.d/sssd-ipa.conf	IdM 関連の SSSD オプションとその値のスキーマ。
/etc/gssproxy/	GSS-Proxy 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、GSS-API サービスごとのファイルと、一般的な**/etc/gssproxy/gssproxy.conf** ファイルが含まれています。
/etc/certmonger/certmonger.conf	この設定ファイルには、証明書の期限切れを監視する certmonger デーモンのデフォルト設定が含まれています。

ディレクトリーまたはファイル	説明
/etc/custodia/custodia.conf	IdM アプリケーションの秘密を管理する Custodia サービスの設定ファイル。

表C.2 システムサービス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	説明
/etc/sysconfig/	**systemd**-specific files

表C.3 Web UI 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	説明
/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 ホストで使用される設定ファイル
/etc/httpd/conf.d/ipa-rewrite.conf	
/etc/httpd/conf/ipa.keytab	Web サーバーが使用するキータブファイル。
/usr/share/ipa/	Web UI が使用するすべての HTML ファイル、スクリプト、およびスタイルシートのディレクトリー。
/usr/share/ipa/ipa.conf	
/usr/share/ipa/updates/	IdM の LDAP データ、設定、および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に UI が使用するスクリプトが含まれます。
/etc/httpd/conf.d/ipa-pki-proxy.conf	Web サーバーから証明書システムへのブリッジングの設定ファイル。

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	説明
/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	説明
/var/lib/dirsrv/slapd-*REALM_NAME*/	IdM サーバーが使用する Directory Server インスタンスに関連付けられたデータベース
/etc/sysconfig/dirsrv	**dirsrv systemd** サービスの IdM 固有の設定。
/etc/dirsrv/slapd-*REALM_NAME*/	IdM サーバーが使用する Directory Server インスタンスに関連付けられ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およびディレクトリー

ディレクトリーまたはファイル	説明
/etc/pki/pki-tomcat/ca/	IdM CA インスタンスのメインディレクトリー。
/var/lib/pki/pki-tomcat/conf/ca/CS.cfg	IdM CA インスタンスの主な設定ファイル。

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	説明
~/.cache/ipa/	IdM クライアントのサーバーごとの API スキーマが含まれます。IdM は、クライアントの API スキーマを 1 時間キャッシュします。

表C.8 システムバックアップ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	説明
/var/lib/ipa/sysrestore/	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すべての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** の両方)、および NTP の元の **.conf** ファイルが含まれます。

ディレクトリーまたはファイル	説明
/var/lib/ipa-client/sysrestore/	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、これは SSSD 認証サービスの **sssd.conf** ファイルです。

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	説明
/var/log/ipaserver-install.log	IdM サーバーのインストールログ。
/var/log/ipareplica-install.log	IdM レプリカのインストールログ
/var/log/ipaclient-install.log	IdM クライアントのインストールログ
/var/log/sssd/	SSSD のログファイル。
~/.ipa/log/cli.log	**ipa** ユーティリティーによる XML-RPC の呼び出しと応答で返されるエラーのログファイル。これは、IdM ユーザーとはユーザー名が異なるツールを実行する**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。
/etc/logrotate.d/	DNS、SSSD、Apache、Tomcat、および Kerberos のログローテーションのポリシー
/etc/pki/pki-tomcat/logging.properties	これは、**/usr/share/pki/server/conf/logging.properties** でのデフォルトの認証局ロギング設定を指しています。

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	説明
/var/log/httpd/	Apache Web サーバーのログファイル。
/var/log/httpd/access_log	これは、Apache サーバーの標準的なアクセスログおよびエラーログです。IdM の Web UI および XML-RPC コマンドラインインターフェイスは Apache を使用するため、IdM に固有のメッセージは、Apache メッセージとともに記録されます。
/var/log/httpd/error_log	
詳細は、Apache ドキュメントのログファイル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	説明
/var/log/pki/pki-ca-spawn.*time_of_installation*.log	IdM KRA のインストールログ
/var/log/pki/pki-kra-spawn.*time_of_installation*.log	IdM KRA のインストールログ
/var/log/pki/pki-tomcat/	PKI 操作ログのトップレベルディレクトリー。CA ログおよび KRA ログが含まれます。
/var/log/pki/pki-tomcat/ca/	証明書の操作に関連するログを含むディレクトリー。IdM では、このログは証明書を使用するサービスプリンシパル、ホスト、およびその他のエンティティーに使用されます。
/var/log/pki/pki-tomcat/kra	KRA に関連するログを含むディレクトリー。
/var/log/messages	証明書のエラーメッセージがその他のシステムメッセージに含まれます。
詳しくは、Red Hat 証明書システム『管理ガイド』のサブシステムログの設定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	説明
/var/log/dirsrv/slapd-*REALM_NAME*/	IdM サーバーが使用する Directory Server インスタンスに関連付けられたログファイルここに記録されるほとんどの運用データは、サーバーとレプリカの相互作用に関連しています。
/var/log/dirsrv/slapd-*REALM_NAME*/access	ドメインの Directory Server インスタンスに対して試行されたアクセスと操作の詳細を記載します。
/var/log/dirsrv/slapd-*REALM_NAME*/errors	
/var/log/dirsrv/slapd-*REALM_NAME*/audit	Directory Server 設定で監査が有効になっている場合に行うすべての Directory Server 操作の監査証跡が含まれます。
詳細は、Red Hat Directory Server ドキュメントのサーバーおよびデータベースのアクティビティーの監視 およびログファイルの参照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	説明
/var/log/krb5kdc.log	これは、Kerberos KDC サーバーの主なログファイルです。

ディレクトリーまたはファイル	説明
/var/log/kadmind.log	Kerberos 管理システムサーバーの主なログファイルです。
これらのファイルの場所は、**krb5.conf** ファイルで設定されます。システムによっては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	説明
/var/log/messages	その他のシステムメッセージに DNS エラーメッセージが含まれます。 このファイルの DNS ロギングは、デフォルトでは有効になりません。これを有効にするには、**# /usr/sbin/rndc querylog** コマンドを実行します。ロギングを無効にするには、コマンドを再度実行します。

表C.15 Custodia ログファイル

ディレクトリーまたはファイル	説明
/var/log/custodia/	Custodia サービスのログファイルディレクトリー。

関連情報

- **journalctl** ユーティリティーの使用方法は、『システム管理者ガイド』 の ジャーナルの使用 を参照してください。**journalctl** を使用すると、**システムド** ユニットファイルのログ出力を表示できます。

C.3. IDM ドメインサービスとログローテーション

複数の IdM ドメインサービスは、システム **logrotate** サービスを使用してログローテーションおよび圧縮を処理します。

- **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/** ディレクトリーに保存されます。

例C.1 /etc/logrotate.d/httpdにあるデフォルトのhttpdログローテーションファイル

```
/var/log/httpd/*log {
```

```
    missingok
    notifempty
    sharedscripts
    delaycompress
    postrotate
        /sbin/service httpd reload > /dev/null 2>/dev/null || true
    endscript
}
```

警告

ほとんどのサービスの**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、およびデフォルトのパーミッションで新しいログファイルが作成されます。ただし、**名前** および **tomcat** のファイルでは、特殊な ***作成*** ルールにより、ユーザーおよびグループの所有権と同様に明示的なパーミッションで、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およびグループ、またはパーミッションを変更しないでください。これは、IdM 操作と SELinux 設定の両方に必要です。ログローテーションポリシーまたはファイルの所有権を変更すると、IdM ドメインサービスに障害が発生する可能性があります。

関連情報

- IdM がバックエンドとして使用し、Dogtag Certificate System が使用する 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 10 管理ガイド』 の サブシステムログの設定 を参照してください。
- 圧縮設定やログファイルのサイズなど、ログローテーションに指定可能な設定の詳細は、『システム管理者ガイド』 のLog Rotation または 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

ここでは、ドメインレベルの **0** でのレプリカの管理を説明します (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 **1** でのレプリカの管理については、以下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

レプリカの作成プロセス中、**ipa-replica-prepare** ユーティリティーは、**/var/lib/ipa/** ディレクトリーに、レプリカサーバーにちなんで名前が付けられた*レプリカ情報ファイル*を作成します。レプリカ情報ファイルは、マスターサーバーのレルムと設定情報を含む GPG で暗号化したファイルです。

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る情報に基づいて Directory Server インスタンスを設定し、*レプリカの初期化* プロセスを開始します。このプロセス中に、スクリプトが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、レプリカを作成した特定のマシンにのみインストールできます。このファイルは、複数のレプリカを複数のマシンで作成する場合には使用できません。

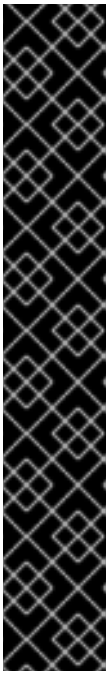
D.2. レプリカの作成

以下のセクションでは、最も注目すべきレプリカのインストールシナリオを説明します。

- 手順と例は合わせて使用してください。CA、DNS、およびその他のコマンドラインオプションを同時に使用できます。以下のセクションの例は、各設定エリアに必要なものを明確にするために、別々に説明します。
- **ipa-replica-install** ユーティリティーは、他の多くのオプションも受け付けます。完全なリストについては、ipa-replica-install(1) の man ページ。

D.2.1. DNS を使用しないレプリカのインストール

1. **マスターの IdM サーバーで、ipa-replica-prepare** ユーティリティーを実行し、**レプリカ** マシンの完全修飾ドメインネーム (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

.company など、単一ラベルのドメイン名を使用しないでください。IdM ドメインは、トップレベルドメインと、1 つ以上のサブドメイン (example.com や company.example.com など) で設定する必要があります。

完全修飾ドメイン名は、以下の条件を満たす必要があります。

- 数字、アルファベット文字、およびハイフン (-) のみが使用される有効な DNS 名である。ホスト名でアンダーライン (_) を使用すると DNS が正常に動作しません。
- すべてが小文字である。大文字は使用できません。
- 完全修飾ドメイン名は、ループバックアドレスを解決できません。**127.0.0.1** ではなく、マシンの公開 IP アドレスを解決する必要があります。

その他の推奨命名プラクティスは『Red Hat Enterprise Linux Security Guide』のRecommended Naming Practices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** でレプリカマシンの IP アドレスを指定します。インストールスクリプトは、レプリカの逆引きゾーンを設定するかどうかを尋ねられます。IdM サーバーが統合 DNS で設定されている場合に限り、**--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。

プロンプトが表示されたら、初期マスターサーバーの DM (Directory Manager) パスワードを入力します。**ipa-replica-prepare** の出力には、レプリカインフォメーションファイルの場所が表示されます。以下に例を示します。

```
[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 に追加できるその他のオプションは、ipa-replica-prepare(1) の man ページを参照してください。

2. **レプリカマシン**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. 初期サーバーからレプリカ情報ファイルを、レプリカマシンにコピーします。

```
[root@server ~]# scp /var/lib/ipa/replica-info-replica.example.com.gpg
root@replica:/var/lib/ipa/
```

4. **レプリカマシンで、ipa-replica-install** ユーティリティーを実行し、レプリカ情報ファイルの場所を追加して、レプリカの初期化プロセスを開始します。プロンプトが表示されたら、元のマスターサーバーの Directory Manager と admin のパスワードを入力し、レプリカのインストールスクリプトが完了するまで待ちます。

```
[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

インストールするレプリカファイルが現在のホスト名と一致しない場合は、レプリカのインストールスクリプトにより警告メッセージが表示され、確認するように求められます。場合によっては、マルチホームマシンなど、一致しないホスト名で続行することを確認できます。

ipa-replica-install に追加できるコマンドラインオプションは、ipa-replica-prepare (1) man ページを参照してください。**--ip-address** で使用できるオプションの 1 つに **ipa-replica-install** オプションがあります。**ipa-replica-install** に追加すると、**--ip-address** は、ローカルインターフェイスに関連付けられた IP アドレスだけを許可します。

D.2.2. DNS のある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には、「DNS を使用しないレプリカのインストール」で説明されている DNS を使用せずにインストールする手順に従いますが、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 のある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 の実行後に、適切な DNS エントリーが作成されたことを確認し、必要に応じて他の DNS サーバーをバックアップサーバーとして追加します。詳細は「DNS のあるレプリカのインストール」を参照してください。

D.2.3. さまざまな CA 設定を使用したレプリカのインストール

警告

Red Hat は、複数のサーバーに CA サービスをインストールすることを強く推奨します。CA サービスを含む初期サーバーのレプリカのインストールは、「CA のあるレプリカのインストール」 を参照してください。

CA を 1 台のサーバーにのみインストールすると、CA サーバーが失敗した場合に CA 設定を復元できる機会なしに CA 設定が失われるリスクがあります。詳細は「失われた CA サーバーの復旧」を参照してください。

証明書システム CA がインストールされているサーバーからのレプリカのインストール
初期サーバーを、統合 Red Hat 証明書システムインスタンスで設定する際に、レプリカに CA を設定するには (ルート CA であるかどうか、外部 CA の下位にあるかどうかに関係なく)、「DNS を使用しないレプリカのインストール」で説明されている基本的なインストール手順に従いますが、**ipa-replica-install** ユーティリティーに **--setup-ca** を追加します。**--setup-ca** オプションは、最初のサーバーの設定から CA 設定をコピーします。

```
[root@replica ~]# ipa-replica-install /var/lib/ipa/replica-info-replica.example.com.gpg --setup-ca
```

証明書システム 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
CA なしのレプリカのインストールでは、「DNS を使用しないレプリカのインストール」で説明されている基本的な手順に従いますが、最初のサーバーで **ipa-replica-prepare** ユーティリティーを実行する場合は次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 の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. 追加のレプリカ合意の追加

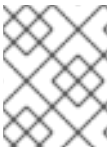
ipa-replica-install を使用してレプリカをインストールすると、マスターサーバーとレプリカとの間に初期のレプリカ合意が作成されます。レプリカを別のサーバーまたはレプリカに接続するには、**ipa-replica-manage** ユーティリティーを使用して合意を追加します。

マスターサーバーと新しいレプリカに 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。別のサーバーまたはレプリカに CA レプリカ合意を追加するには、**ipa-csreplica-manage** ユーティリティーを使用します。

別のレプリカ合意を追加する方法は、「レプリカおよびレプリカ合意の管理」 を参照してください。

D.3. レプリカおよびレプリカ合意の管理

本章では、レプリカ合意の詳細と、その管理方法を説明します。



注記

追加のレプリカ合意を設定する際のガイドラインは、「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. レプリカ合意の説明

レプリカは、データをコピーする*レプリカ合意*で結合されます。レプリカ合意は双方向です。最初のレプリカからサーバーから別のレプリカにデータが複製されるだけでなく、別ののレプリカから最初のレプリカにもデータが複製されます。

注記

最初のレプリカ合意は、**ipa-replica-install** スクリプトにより 2 つのレプリカ間で設定されます。初期レプリカのインストールの詳細は、*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 種類のレプリカ合意に対応します。

- ユーザー、グループ、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。このような合意は、**ipa-replica-manage** ユーティリティーを使用して管理できます。
- 証明書サーバーのデータを複製するレプリカ合意。このような合意は、**ipa-csreplica-manage** ユーティリティーを使用して管理できます。
- ユーザー情報を Active Directory サーバーと複製する同期合意。この合意については、本書では扱っていません。IdM および Active Directory の同期に関するドキュメントは、『Windows 統合ガイド』 の Active Directory および Identity Management ユーザーの同期 を参照してください。

ipa-replica-manage ユーティリティーと **ipa-csreplica-manage** ユーティリティーでは、同じ形式と引数が使用されます。本章の以下のセクションでは、このユーティリティーを使用して実行する最も注目すべきレプリケーション管理操作を説明します。ユーティリティーの詳細は、ipa-replica-manage(1) および ipa-csreplica-manage(1) の man ページを参照してください。

D.3.2. レプリカ合意のリスト表示

現在レプリカに設定されているディレクトリーデータのレプリカ合意のリストを表示するには、**ipa-replica-manage list** を実行します。

1. 引数を指定せずに **ipa-replica-manage list** を実行して、レプリケーショントポロジー内のすべてのレプリカをリスト表示します。この出力で、必要なレプリカを特定します。

```
$ 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、レプリカ合意をリスト表示します。

```
$ ipa-replica-manage list server1.example.com
server2.example.com: replica
server3.example.com: replica
```

この出力には、**server1.example.com** が更新を送信するレプリカが表示されます。

証明書サーバーのレプリカ合意のリストを表示するには、**ipa-csreplica-manage list** を使用します。

D.3.3. レプリカ合意の作成と削除

レプリカ合意の作成
新しい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
```

このコマンドは、*server1.example.com* から *server2.example.com*、*server2.example.com* から *server1.example.com* に向かう新しい双方向レプリカ合意を作成します。

ipa-replica-manage connect でサーバーを 1 つだけ指定すると、IdM はローカルホストと、指定されたサーバーとの間にレプリカ合意を作成します。

新しい証明書サーバーのレプリカ合意を作成するには、**ipa-csreplica-manage connect** コマンドを使用します。

レプリカ合意の削除
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
```

このコマンドは、*server1.example.com* から *server4.example.com* へ、*server4.example.com* から *server1.example.com* へのレプリケーションを無効にします。

ipa-replica-manage disconnect は、レプリカ合意のみを削除します。これにより、両方のサーバーが Identity Management レプリケーショントポロジーに残ります。すべてのレプリカ合意と、レプリカ関連のデータを削除するには、**ipa-replica-manage del** コマンドを使用します。これにより、レプリカが Identity Management ドメインから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-csreplica-manage disconnect** コマンドを使用します。同様に、2 台のサーバー間ですべての証明書のレプリカ合意およびデータを削除する場合は、**ipa-csreplica-manage del** を使用します。

D.3.4. 手動レプリケーション更新の開始

直接レプリカ合意が直接、結ばれているレプリカ間でのデータ変更は、ほぼ即座に複製されます。ただし、直接レプリカ合意ぶ参加していないレプリカは、更新は即座に受け取りません。

状況によっては、計画外のレプリケーション更新を手動で開始する必要があります。たとえば、レプリ

カをオフラインにしてメンテナンスする前に、計画更新待ちのキューに入っていた変更をすべて、別のレプリカに送信する必要があります。この状況では、レプリカをオフラインにする前に、手動のレプリケーション更新を開始できます。

レプリケーションの更新を手動で開始するには、**ipa-replica-manage force-sync** コマンドを使用します。コマンドを実行するローカルホストは、更新を受信するレプリカです。更新を送信するレプリカを指定するには、**--from** を使用します。

```
$ ipa-replica-manage force-sync --from server1.example.com
```

証明書サーバーデータのレプリケーション更新を開始するには、**ipa-csreplica-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

レプリカが長時間オフラインになっている場合や、データベースが破損している場合は、*再初期化*できます。再初期化は、「レプリカの作成: 概要」 で説明されている初期化と似ています。再初期化でじゃ、更新されたデータセットでレプリカを更新します。たとえば、バックアップからの権限のある復元を行う必要がある場合に、再初期化を使用できます。

注記

通常のレプリケーション更新待ちや、手動のレプリケーション更新の開始をしても、このような状況には役に立ちません。このレプリカの更新時には、レプリカは変更されたエントリーのみを互いに送信します。再初期化とは異なり、レプリケーションの更新ではデータベース全体が更新されません。

レプリカでデータレプリカ合意 (data replication agreement) を再初期化するには、**ipa-replica-manage re-initialize** を使用します。コマンドを実行するローカルホストは、再初期化されたレプリカです。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初期化しなおすには、**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、トポロジーから IdM レプリカが削除されるため、IdM 要求を処理できなくなります。また、ホストマシン自体も IdM ドメインから削除します。

レプリカを削除するには、レプリカで以下の手順を実行します。

1. IdM ドメインのレプリカ合意をすべてリスト表示します。この出力では、レプリカのホスト名を書き留めておきます。

```
$ 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** コマンドを使用して、レプリカに設定したすべての合意と、レプリカに関するすべてのデータを削除します。

```
$ ipa-replica-manage del server3.example.com
```

3. レプリカが独自の CA を使用して設定されている場合は、**ipa-csreplica-manage del** コマンドを使用して、すべての証明書サーバーのレプリカ合意を削除します。

```
$ ipa-csreplica-manage del server3.example.com
```

注記

この手順は、レプリカ自体が IdM CA で設定されている場合にのみ必要です。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

IdM デプロイメントで組み込み認証局 (CA) を使用する場合は、IdM CA サーバーの 1 つがマスター CA として機能します。これは、CA サブシステム証明書の更新を管理し、証明書失効リスト (CRL) を生成します。デフォルトでは、マスター CA は、システム管理者が **ipa-server-install** コマンドまたは **ipa-ca-install** コマンドを使用して CA ロールをインストールした最初のサーバーです。

マスター CA サーバーをオフラインにするか、使用を停止する場合は、レプリカを*プロモート*して、マスター CA としての地位を確立します。

- レプリカが、CA サブシステムの証明書の更新を処理するように設定されていることを確認します。「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。
- CRL を生成するようにレプリカを設定します。「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

証明書の更新を処理するサーバーを変更するには、IdM サーバーで次の手順を使用します。

1. どのサーバーが現在の更新マスターであるかを判断します。

- Red Hat Enterprise Linux 7.3 以降

```
$ ipa config-show | grep "CA renewal master"
IPA CA renewal master: server.example.com
```

- Red Hat Enterprise Linux 7.2 以前

```
$ 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
```

```
...
# 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=com
...
```

どちらの例でも、**server.example.com** は最新の更新マスターです。

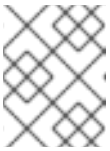
2. 証明書の更新を処理するように別のサーバーを設定するには、次のコマンドを実行します。

   - Red Hat Enterprise Linux 7.4 以降

     ```
     # ipa config-mod --ca-renewal-master-server new_server.example.com
     ```

   - Red Hat Enterprise Linux 7.3 以前

     ```
     # ipa-csreplica-manage set-renewal-master
     ```



注記

このコマンドは、コマンドを実行するサーバーを新しい更新マスターとして設定します。

また、このコマンドは以前の CA を更新マスターからクローンに自動的に再設定します。

付録E IDENTITY MANAGEMENT サーバーポートに関する考慮事項

E.1. IDENTITY MANAGEMENT コンポーネントおよび関連するサービス

表E.1「Identity Management コンポーネントおよび関連するサービス」 各 Identity Management サービスが外部に公開するポートのリストを表示します。

表E.1 Identity Management コンポーネントおよび関連するサービス

コンポーネント	サービス	アクセスが許可されているポート
Identity Management フレームワーク*	Apache ベースの Web サービスとその他のサービスへのルート	HTTPS ポート **443(TCP/TCP6)**
LDAP ディレクトリーサーバー	389-ds のインスタンス	ポート **389 (TCP/TCP6)** - 通常の LDAP トラフィック。接続を保護するには StartTLS 拡張機能または SASL GSSAPI を使用します。 ポート **636(TCP/TCP6)**: SSL 経由の通常の LDAP トラフィック ポート **389 (UDP)** - Active Directory サービスとの統合を容易にする接続なしの LDAP アクセス
Kerberos キー配布センター*	krb5kdc	ポート **88 (TCP/TCP6 and UDP/UDP6)**: 通常の Kerberos トラフィック ポート **464 (TCP/TCP6 および UDP/UDP6)** - Kerberos パスワード変更プロトコルアクセス
Kerberos 管理者ーモン *	kadmind	ポート **749 (TCP/TCP6)**: 内部で使用される Kerberos リモート管理プロトコル
Custodia 鍵管理*	custodia	HTTPS ポート **443(TCP/TCP6)**: Identity Management フレームワークの一環
System Security Services Daemon*	sssd	HTTPS ポート **443(TCP/TCP6)**: Identity Management フレームワークの一環
MS-KDCP プロキシー **	HTTPS 経由での Kerberos へのプロキシーアクセス	HTTPS ポート **443(TCP/TCP6)**: Identity Management フレームワークの一環

コンポーネント	サービス	アクセスが許可されているポート
認証局	Tomcat 上部の Dogtag インスタンス	HTTPS ポート **443(TCP/TCP6)**: Identity Management フレームワークの一環 ポート**80 (TCP/TCP6)** 経由の HTTP アクセス。ただし、Identity Management に設定された Apache ルールに従ってポート**8080 (TCP/TCP6)** に内部でリダイレクトされます。取得される情報は、OCSP レスポンダーと証明書のステータス (証明書失効リスト) です。 ポート **8443(TCP/TCP6)** での HTTPS 内部アクセス: CA 管理用 IPA マスターでは、内部的に、ポート **8005 および 8009 (TCP/TCP6)** が *127.0.0.1* and *::1* ローカルインターフェイスアドレス上の証明書サービスのコンポーネントを実行するのに使用されます。
DNS	named	ポート **53 (TCP/TCP6 and UDP/UDP6)**: 標準の DNS リゾルバー ポート **953 (TCP/TCP6)** - *127.0.0.1* および *::1* のローカルインターフェイスアドレスでの BIND サービスリモート制御
Active Directory の統合	Samba サービス (smbd、winbindd)	ポート **135(TCP/TCP6)**: DCE RPC エンドポイントマッパー (smbd デーモン) ポート **138 (TCP/TCP6)**、NetBIOS Datagram service (任意。nmbd デーモンの実行が必要) ポート **139 (TCP/TCP6)**、NetBIOS セッションサービス (smbd デーモン) ポート **445(TCP/TCP6)**: TCP/TCP6 経由の SMB プロトコル (smbd デーモン) DCE RPC エンドポイントサービスで動的に開かれたポート **49152-65535 (TCP/TCP6)**
認証局 vault	Dogtag インスタンスの KRA コンポーネント	HTTPS ポート **443(TCP/TCP6)**: Identity Management フレームワークの一環 ポート**80 (TCP/TCP6)** 経由の HTTP アクセス、ただし、Apache ルールによりポート**8080 (TCP/TCP6)** に内部でリダイレクト - OCSP レスポンダーおよび証明書のステータス (証明書失効一覧) ポート **8443(TCP/TCP6)** での HTTPS 内部アクセス: CA 管理用 IPA マスターでは、内部的に、ポート **8005 および 8009 (TCP/TCP6)** が *127.0.0.1* and *::1* ローカルインターフェイスアドレス上の証明書サービスのコンポーネントを実行するのに使用されます。

* Identity Management デプロイメントの場合はすべて、アスタリスクが付いたサービスは必須です。

** MS-KDCP プロキシーコンポーネントはオプションですが、デフォルトで有効になっています。

付録F IDM への主な変更点

特定の IdM バージョンでは、新しいコマンドが実装されていたり、既存のコマンドが置き換えられていたりします。また、設定手順やインストール手順が大幅に変更する場合があります。この付録では、最も重要な変更を説明します。

変更の詳細なリストは、各バージョンの Red Hat Enterprise Linux (RHEL) 7 リリースノートを参照してください。

RHEL 7.7 で実行している IdM 4.6

- **ipa-cert-fix** ユーティリティーは、IdM がオフライン時のシステム証明書更新用に追加されました。詳細は、「IdM がオフライン時に期限切れのシステム証明書の更新」を参照してください。

- IdM が、証明書の SAN 拡張における IP アドレスに対応するようになりました。特定の状況では、管理者は、SAN (Subject Alternative Name) 拡張機能の IP アドレスが割り当てられた証明書を発行する必要があります。今回のリリース以降、アドレスが IdM DNS サービスで管理され、サブジェクトのホストまたはサービスプリンシパルに関連付けられている場合に、管理者は SAN 拡張機能に IP アドレスを設定できます。

- IdM を使用する場合に、.company などの単一ラベルのドメイン名の使用できなくなりました。IdM ドメインは、トップレベルドメインと、1 つ以上のサブドメイン (example.com や company.example.com など) で設定する必要があります。

- このリリースにおける詳細な変更は、『Red Hat Enterprise Linux 7.7 リリースノート 』の以下のセクションを参照してください。

  - 『新機能 - 認証および相互運用性』

  - 『主なバグ修正 - 認証および相互運用性』

RHEL 7.6 で実行している IdM 4.6

- このリリースの変更点は、『Red Hat Enterprise Linux 7.6 リリースノート 』の以下のセクションを参照してください。

  - 『新機能 - 認証および相互運用性』

  - 『主なバグ修正 - 認証および相互運用性』

RHEL 7.5 で実行している IdM 4.5

- このリリースの変更点は、『Red Hat Enterprise Linux 7.5 リリースノート 』の以下のセクションを参照してください。

  - 『新機能 - 認証および相互運用性』

  - 『主なバグ修正 - 認証および相互運用性』

RHEL 7.4 で実行している IdM 4.5

- このバージョンでは、クライアントの HTTPS 接続用の SSL バックエンドが、NSS (Network Security Services) から OpenSSL に変更されました。これにより、登録機関 (RA) は証明書を NSS データベースではなく **/var/lib/ipa/** ディレクトリーに保存するようになりました。

- このリリースでの詳細な変更は、『Red Hat Enterprise Linux 7.4 リリースノート 』の以下のセクションを参照してください。
  - 『新機能 - 認証および相互運用性』
  - 『主なバグ修正 - 認証および相互運用性』

RHEL 7.3 で実行している IdM 4.4

- 新しい **ipa replica-manage clean-dangling-ruv** コマンドを使用すると、管理者は、インストールされていないレプリカから関連する更新ベクトル (RUV) をすべて削除できます。
- 新しい **ipa server-del** コマンドを使用すると、管理者は IdM サーバーをアンインストールできます。
- 今回のバージョンでは、管理者は次のコマンドを使用して、IdM 認証局 (CA) を管理できます。
  - **ipa ca-add**
  - **ipd ca-del**
  - **ipa ca-enable**
  - **ipa ca-disble**
  - **ipa ca-find**
  - **ipa ca-mod**
  - **ipa ca-show**
- 今回のバージョンでは、**ipa-replica manage** コマンドに代わる次のコマンドを使用して、レプリカ合意を管理します。
  - **ipa topology-configure**
  - **ipa topologysegment-mod**
  - **ipa topologysegment-del**
  - **ipa topologysuffix-add**
  - **ipa topologysuffix-show**
  - **ipa topologysuffix-verify**
- 今回のバージョンでは、管理者は次のコマンドを使用して、**cn=masters,cn=ipa,cn=etc,*domain_suffix*** エントリーに保存されている IdM サーバーのリストを表示できます。
  - **ipa server-find**
  - **ipa server-show**
- certmonger ヘルパースクリプトが**/usr/lib64/ipa/certmonger/** から **/usr/libexec/ipa/certmonger/** ディレクトリーに移動しました。

- 今回のバージョンでは、ドメインレベルと、ドメインレベルの表示と設定を行う以下のコマンドが導入されました。
  - **ipa domainlevel-set**
  - **ipa domainlevel-show**
- 今回のリリースでの詳細な変更は、『Red Hat Enterprise Linux 7.3 リリースノート 』の以下のセクションを参照してください。
  - 『新機能 - 認証および相互運用性』
  - 『主なバグ修正 - 認証および相互運用性』

RHEL 7.2 で実行している IdM 4.2

- 複数の証明書プロファイルとユーザー証明書に対応 - Identity Management では、単一のサーバー証明書プロファイルにのみ対応する代わりに、サーバーやその他の証明書を発行する複数のプロファイルに対応するようになりました。プロファイルは Directory Server に保存され、IdM レプリカ間で共有されます。さらに、管理者が個々のユーザーに証明書を発行できるようになりました。以前は、ホストおよびサービスに証明書を発行することしかできませんでした。
- このリリースでのその他の変更点は、『Red Hat Enterprise Linux 7.2 リリースノート 』の『新機能 - 認証および相互運用性』セクションを参照してください。

RHEL 7.1 で実行している IdM 4.1

- 今回のバージョンでは、**ipa-getkeytab -r** コマンドの代わりに、次のコマンドを使用してキータブを取得し、取得権限を設定します。
  - **ipa-host-allow-retrieve-keytab**
  - **ipa-host-disallow-retrieve-keytab**
  - **ipa-host-allow-create-keytab**
  - **ipa-host-disallow-create-keytab**
  - **ipa-service-allow-retrieve-keytab**
  - **ipa-service-disallow-retrieve-keytab**
  - **ipa-service-allow-create-keytab**
  - **ipa-service-disallow-create-keytab**
- このリリースでのその他の変更点は、『Red Hat Enterprise Linux 7.1 リリースノート 』の『新機能 - 認証および相互運用性』セクションを参照してください。

RHEL 7.0 で実行している IdM 3.3

- このリリースの変更点は、『Red Hat Enterprise Linux 7.0 リリースノート 』の 『新機能 - 認証および相互運用性』セクションを参照してください。

付録G 更新履歴

改訂番号はこのマニュアルの編集に関するものであり、Red Hat Enterprise Linux の バージョン番号とは関係ありません。

改訂 7.0-53 Tue Feb 16 2021 Florian Delehaye
IdM サービスや設定ファイルなど、さまざまな内容を明確化しました。

改訂 7.0-52 Tue Sep 29 2020 Florian Delehaye
7.9 GA 公開用ドキュメントバージョン

改訂 7.0-51 Tue Mar 31 2020 Florian Delehaye
7.8 GA 公開用ドキュメントバージョン

改訂 7.0-50 Wed Aug 28 2019 Marc Muehlfeld
『IdM 付録の主な変更点』 の追加。複数のマイナー更新。

改訂 7.0-49 Tue Aug 06 2019 Marc Muehlfeld
7.7 GA 公開用ドキュメントバージョン

改訂 7.0-48 Fri Jun 21 2019 Marc Muehlfeld
『IdM がオフライン時に期限切れのシステム証明書の更新』 セクションの追加。

改訂 7.0-47 Thu Jun 13 2019 Marc Muehlfeld
非表示のレプリカの設定に関するコンテンツの追加。

改訂 7.0-46 Wed Jun 04 2019 Marc Muehlfeld
『最後に成功した Kerberos 認証の追跡の有効化』セクションの追加。若干の編集。

改訂 7.0-45 Tue Apr 09 2019 Marc Muehlfeld
『Web UI セッションの長さ』 を追加。認証インジケーターに関するセクション 2 つ、若干の編集を複数追加。

改訂 7.0-44 Thu Nov 22 2018 Filip Hanzelka
『IdM サーバーのインストールおよびアンインストール』 の章に、『Identity Management コンポーネントおよび関連するサービス』 と、マイナーな編集を追加。

改訂 7.0-43 Mon Oct 29 2018 Lucie Maňásková
7.6 GA 公開用ドキュメントの準備。

改訂 7.0-42 Tue Jun 26 2018 Lucie Maňásková
『統合 IdM CA を使用した証明書の管理』 を更新。その他の更新。

改訂 7.0-41 Fri Apr 23 2018 Filip Hanzelka
『Kerberos チケットの有効期間の判断』 を追加。その他の若干の修正。

改訂 7.0-40 Fri Apr 6 2018 Lucie Maňásková
7.5 GA 公開用ドキュメントの準備。

改訂 7.0-39 Wed Mar 14 2018 Filip Hanzelka
マイナー更新。

改訂 7.0-38 Wed Feb 28 2018 Lucie Maňásková
マイナー更新。

改訂 7.0-37 Mon Feb 12 2018 Aneta Šteflová Petrová
『十分な追加権限がないことが原因で Vault にユーザーがアクセスできない』 を追加。その他の若干の修正。

改訂 7.0-36 Mon Jan 29 2018 Aneta Šteflová Petrová
『SELinux ユーザーマップの定義』 を更新。その他の若干の修正。

改訂 7.0-35 Fri Dec 15 2017 Aneta Šteflová Petrová
『ホストの管理』 を更新。その他の若干の修正。

改訂 7.0-34 Mon Dec 4 2017 Aneta Šteflová Petrová
『IdM の Kerberos PKINIT 認証』 を追加。『IdM ユーザーのアクセス制御の定義』 を更新。その他の若干の修正。

改訂 7.0-33 Mon Nov 20 2017 Aneta Šteflová Petrová
『ユーザースおよびグループスキーマ』および 『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
若干の修正。

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複数更新。『Identity Management のスマートカード認証』 への若干更新。

改訂 7.0-30 Mon Aug 28 2017 Aneta Šteflová Petrová
『Identity Management でのスマートカード認証』 および 『Identity Management の設定ファイルおよびディレクトリー』 を更新。

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、および automember を管理およびマージ。その他の若干の更新

改訂 7.0-27 Mon Apr 10 2017 Aneta Šteflová Petrová
Identity Management の TLS 設定を追加。若干の修正および更新。

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考慮事項とパスワードリセットの有効化に関する注意点を追加。その他の若干の更新

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、および HBAC に関する情報を更新。さまざまな章のその他の更新。

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。その他の若干の更新

改訂 7.0-23 Tue Oct 18 2016 Aneta Šteflová Petrová
7.3 GA リリースのバージョン

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
概要を更新。その他の若干の修正。

改訂 7.0-19 Tue Jun 28 2016 Aneta Petrová
図を更新。概要の章に、IdM を使用する利点についてセクションを追加。その他の若干の修正および調整

改訂 7.0-18 Fri Jun 10 2016 Aneta Petrová
概要、サーバーのインストール、およびトラブルシューティングの章を更新。その他の修正。

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、および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 セクションを更新。PAM サービスのドメイン制限をシステムレベル認証ガイドに移動。

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。インストール章にアンインストール手順を移動。その他の若干の更新

改訂 7.0-13 Thu Nov 19 2015 Aneta Petrová
証明書プロファイル管理およびレプリカのマスターへのプロモートを若干更新。

改訂 7.0-12 Fri Nov 13 2015 Aneta Petrová
7.2 GA リリースのバージョン向けに DNS およびその他のセクションを更新。

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 リリース向けバージョン。

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して再構築。

改訂 7.0-4 Wed Jun 11 2014 Ella Deon Ballard
初期リリース。