Red Hat Enterprise Linux 6

Identity Management ガイド

Linux ベースのインフラストラクチャーの ID および承認ポリシーの管理

Last Updated: 2023-03-02

Red Hat Enterprise Linux 6 Identity Management ガイド

Linux ベースのインフラストラクチャーの ID および承認ポリシーの管理

Enter your first name here. Enter your surname here.
Enter your organisation's name here. Enter your organisational division here.
Enter your email address here.

法律上の通知

概要

ユーザーおよびマシン両方のアイデンティティー管理およびポリシー管理は、多くのエンタープライズ環境おいて中核的な機能となっています。IPA 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。本書では、サーバーとクライアントの両方など、IPA ドメインのインストール、設定、および管理におけるすべての側面について説明します。このガイドは、IT および システム管理者用です。

目次

第1章 IDENTITY MANAGEMENT の概要

Red Hat Enterprise Linux IdM を使用して、ネイティブの Linux ツールを使用して、ID ストア、集中認証、Kerberos および DNS サービスのドメイン制御、および認可ポリシーを作成します。Identity Management は ID/ポリシー/認証が集約化されたソフトウェアが新たに導入されましたが、Identity Management は Linux/Unix ドメインをサポートする唯一のオプションの 1 つです。

Identity Management は、PAM、LDAP、Kerberos、DNS、NTP、証明書サービスなど、標準定義の一般的なネットワークサービスを統一し、Red Hat Enterprise Linux システムがドメインコントローラーとして機能できるようにします。

Identity Management は、Kerberos や DNS などの一元管理されたサービスを共有するサーバーおよびクライアントが含まれるドメインを定義します。本章では、最初に Identity Management の概要を説明します。本章では、ドメイン内でこれらのサービスがどのように連携し、またサーバーとクライアントがどのようにインタラクションを取るかについても説明します。

1.1. IDM V.LDAP: より集約的なサービスタイプ

最も基本的なレベルでは、Red Hat Identity Management は Linux および Unix マシンのドメインコントローラーを指します。Identity Management は、制御サーバーおよび登録されたクライアントマシンを使用してドメインを定義します。これにより、Linux/Unix 環境でネイティブの Linux アプリケーションやプロトコルを使用して、これまで利用できなかった集中構造が提供されます。

1.1.1. Identity Management の作業定義

セキュリティー情報は通常、ユーザー、マシン、およびサービスの **アイデンティティー** に関係があります。アイデンティティーの確認が済むと、サービスおよびリソースへのアクセスを制御できます。

IT 管理者は、効率性、リスク管理、管理の容易化ができるように、ID をできる限り一元管理し、認証ポリシーおよび認可ポリシーで I D 管理を統一しようと試みます。これまで、Linux 環境では、この集中管理の確立が非常に困難でした。ドメインを定義するプロトコル (NIS や Kerberos など) には多数ありますが、他にデータ(LDAP など) を保存したり、未だにアクセス権限 (sudo など) を管理したりするアプリケーションもあります 。これらのアプリケーションはいずれも、相互操作ができないだけでなく、異なる管理ツールを使用します。各アプリケーションは別々に、ローカルで管理される必要がありました。アイデンティティーポリシーを一貫性を持って取得するには、手動で設定ファイルをコピーするか、ID およびポリシーを管理するプロプライエタリーアプリケーションの開発を試みる方法しかありません。

Identity Management は、管理のオーバーヘッドを単純化することが目的です。ユーザー、マシン、サービス、およびポリシーはすべて、同じツールを使用して 1 つの場所で設定されます。IdM でドメインが作成されるので、複数のマシンはすべてそのドメインに参加して同じ設定とリソースを使用できます。ユーザーは一度だけサービスにログインするだけで済み、また管理者は単一のユーザーアカウントを管理するだけで済みます。

IdM は以下の 3 点を行います。

- Linux ベースおよび Linux 制御のドメインを作成する。IdM サーバーおよび IdM クライアントはどちらも Linux または Unix マシンです。IdM は Active Directory ドメインとデータを同期して Windows サーバーとの統合を可能にしますが、Windows マシンの管理ツールではないので Windows クライアントはサポート対象ではありません。Identity Management は、Linux ドメインの管理ツールです。

- ID 管理と ID ポリシーを一元化する。

- 既存のネイティブの Linux アプリケーションおよびプロトコル上に構築する。IdM には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 管理者から信頼されているだけでなく、Linux システムで十分に確立されています。

このように、Identity Management を使用することで、管理者は新しい作業を行うのではなく、作業内容を改善できるようになります。以下に、いくつか例を示します。

1 つの極端な例として、**制御レベルの低い** 環境が挙げられます。Little Example Corp. には複数の Linux サーバーと Unix サーバーがありますが、各サーバーは別々に管理されています。パスワードはすべてローカルマシンに保存されるので、集約された ID または認証プロセスはありません。Tim (IT 管理者) は、すべてのマシンでユーザーを管理し、認証ポリシーと承認ポリシーを別々に設定してローカルパスワードを管理する必要があります。IdM を使用すると、作業が整理されます。ユーザー、パスワード、およびポリシーストアを簡単に集約する方法があり、Tim (IT 管理者) は、マシン 1 台 (IdM サーバー) のみで ID を管理して、ユーザーとポリシーを同じように全マシンに適用します。ホストベースのアクセス制御、委譲などのルールを使用すると、ノートパソコンやリモートユーザーに異なるアクセスレベルを設定することもできます。

中間の例として、**制御レベルが中規模** の環境が挙げられます。Mid-Example Corp. には Linux および Unix の複数のサーバーがありますが、Bill (IT 管理者) は、マシンの NIS ドメイン、ユーザー用の LDAP ディレクトリー、認証用の Kerberos を作成して、詳細にわたる制御レベルを管理しようとしています。この環境は適切に管理されていますが、異なるツールを使用して各アプリケーションは別々に管理する必要があります。また、インフラストラクチャーに新しいマシンが追加されたり、サービスがオフラインになると必ず、すべてのサービスを手動で更新する必要があります。このような場合に、IdM を使用すると、簡素化されたツールセット 1 つで、さまざまなアプリケーションをすべてシームレスに統合するため、管理のオーバーヘッドが大幅に削減されます。また、ドメイン内のすべてのマシンにシングルサインオンサービスを実装することもできます。

反対の極端な例として、**制御のない** 環境が挙げられます。Big Example Corp では、システムの大半が Windows ベースで、密接に統合された Active Directory フォレストで管理されています。ただし、開発、実稼働などのチームには、Linux システムや Unix システムが多数あり、これらのシステムは基本的に Windows が制御する環境から除外されます。IdM は、Active Directory フォレストでは利用できないネイティブツールおよびアプリケーションを使用して、Linux/Unix サーバーをネイティブで管理できるようにします。さらに、IdM は Windows に対応しているため、Active Directory と IdM との間でデータを同期し、集中管理されたユーザーストアを確保できます。

IdM は、ID 管理という非常に一般的であり、また非常に特殊な問題に、非常に簡単なソリューションを提供します。

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

Identity Management に最も近いものは、389 Directory Server などの標準 LDAP ディレクトリーですが、実際の機能と、**意図する機能** にはいくつかの大きな違いがあります。

まず、ディレクトリーサービスとは何かを理解すると役立ちます。ディレクトリーサービスとは、情報を格納するソフトウェア、ハードウェア、およびプロセスの集合のことです。ディレクトリーサービスは、非常に具体的な情報 (例: ホスト名の情報を格納するためディレクトリーサービス) ですが、汎用ディレクトリーサービスはあらゆる種類の情報を保管して取得できます。389 Directory Server などの LDAP ディレクトリーは汎用ディレクトリーです。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、ビルのエントリーをサポートする柔軟なスキーマがあり、このスキーマをカスタマイズしてほぼすべてのエントリーを定義できます。389 Directory Server のような LDAP サーバーは、拡張性があるため、他のアプリケーションのデータを格納するバックエンドとして頻繁に使用されます。389 directory Server には情報を格納するだけでなく、整理します。LDAP ディレクトリーは、階層構造 (**ディレクトリーツリー**) を使用して、エントリーをルートエントリー (サフィックス)、中間エントリーまたはコンテナーエントリー (サブツリーまたはブランチ)、およびリーフエントリー (実際のデータ) に整理します。ディレクトリーツリーには、多くの分岐点を持つ非常に複雑なものと、分岐点が少ない非常に単純な (フラット) ものがあります。

LDAP ディレクトリーの主な機能は汎用性で、さまざまなアプリケーションに適合するようにできます。

一方、ID 管理には非常に特殊な目的があり、非常に特殊なアプリケーションに適合します。これは一般的な LDAP ディレクトリーやバックエンド、ポリシーサーバーではなく、これは汎用性はありません。

Identity Management は、アイデンティティー (ユーザーおよびマシン) および、アイデンティティーとそのインタラクションに関連するポリシーに重点を置いています。IdM は LDAP バックエンドを使用してデータを保管しますが、ID 関連のエントリーの特定セットやその詳細を定義する高度にカスタマイズされた特定のスキーマセットがあります。IdM には、エントリータイプや、その目的に関連のある関係が少ししかないので、比較的フラットで、シンプルなディレクトリーツリーが使用されています。また、ID の管理といった特定の目的でしかデプロイできないので、IdM サーバーのデプロイ方法にはルールや制限があります。

また、IdM には制限があるので、管理作業が大幅に簡素化されます。IT インフラストラクチャー全体で明確にロールが定義されているだけでなく、インストールプロセスはシンプルで、コマンドも統一されています。IdM ドメインは、設定、参加、管理が簡単で、特にエンタープライズ全体でのシングルサインオンなどのアイデンティティー/認証タスクといった機能も、より汎用的なディレクトリーに比べ、IdM では実現しやすくなっています。

表1.1 Identity Management と 389 Directory Server の比較

	389 ディレクトリーサーバー	ID 管理
使用方法	汎用	単一のドメイン、ID 管理にフォーカス
柔軟性	高度にカスタマイズ可能	ID と認証にフォーカスする際に制限あり
スキーマ	デフォルトの LDAP スキーマ	ID 管理向けに最適化された特別なスキーマ
ディレクトリーツリー	標準および柔軟な階層	階層が固定されたフラットツリー
認証	LDAP	Kerberos または Kerberos および LDAP
Active Directory の同期	双方向	一方向、Active Directory から Identity Management
パスワードポリシー	LDAP ベース	Kerberos ベース
ユーザーツール	Java コンソールおよび標準の LDAP ユーティリティー	Web ベースの UI および特別な Python コマンドラインツール

389 Directory Server などの LDAP ディレクトリーは柔軟性と適応性があるので、任意数のアプリケーションのバックエンドとして最適です。LDAP ディレクトリーの主な目的は、データを効率的に保存して取得することです。

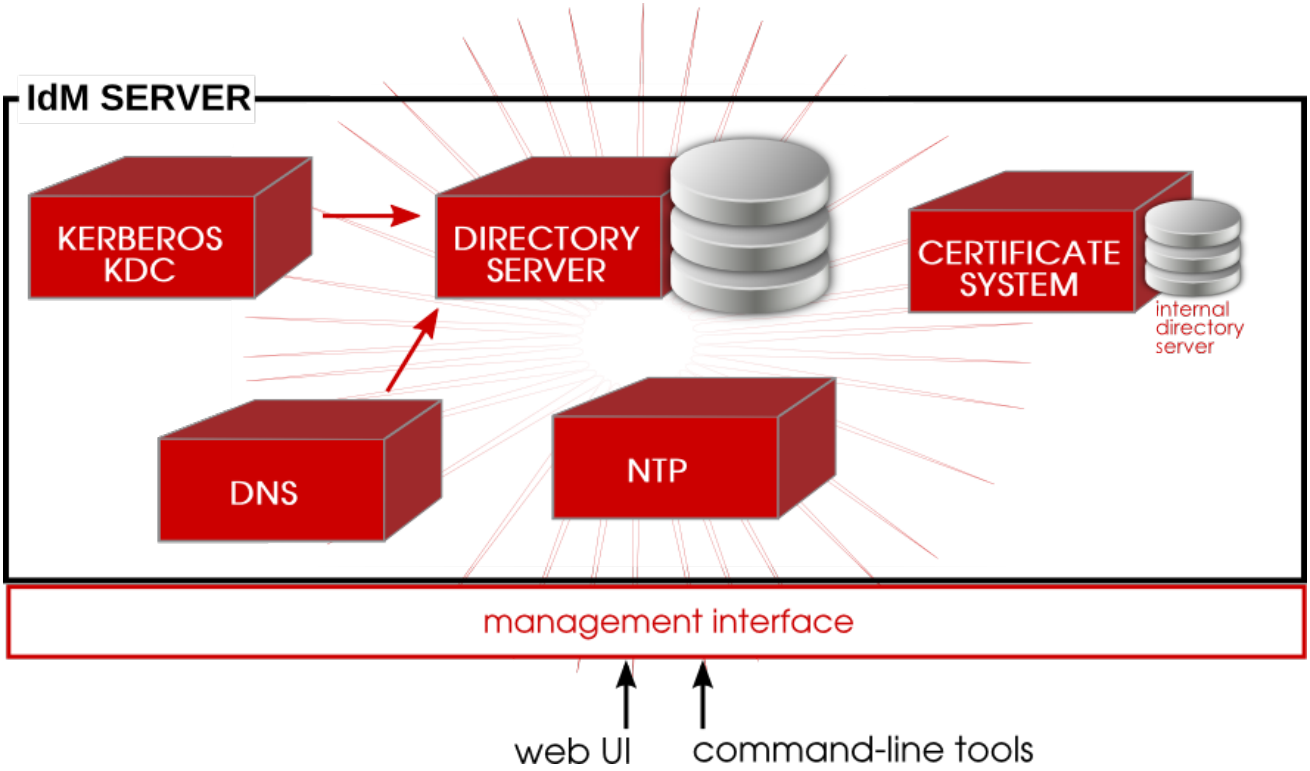
IdM は非常に異なる隙間を埋めます。IdM は、1 つのタスクを非常に効果的に実行するように最適化されています。ユーザー情報、認証および認可ポリシー、およびホスト情報などのアクセスに関する情報などを保存します。その唯一の目的は、アイデンティティーを管理することです。

1.2. LINUX サービスの統合

Identity Management を使用すると、異種ではあるが関連性のある Linux サービスを単一の管理環境に統合されます。その単一の管理環境から、ホストマシンをそれらのサービスのドメインに配置するシンプルかつ簡単な方法を確立します。

IdM サーバーは、基本的には ID サーバーおよび認証サーバーです。プライマリー IdM サーバーは基本的にドメインコントローラーで、認証に Kerberos サーバーと KDC を使用します。LDAP バックエンドには、ユーザー、クライアントマシン、およびドメイン設定を含むすべてのドメイン情報が含まれます。

図1.1 IdM サーバー: サービスの統合



コア ID/認証機能をサポートをするために、その他のサービスが含まれています。DNS は、マシンの検出や、ドメイン内の他のクライアントへの接続に使用されます。NTP を使用してすべてのドメインクロックを同期し、ログ、証明書、および操作が期待どおりに実行されるようにします。証明書サービスは、Kerberos 対応のサービスに証明書を提供します。これらの追加サービスは、すべて IdM サーバーの制御下で機能します。

IdM サーバーには、IdM 関連の全サービスの管理に使用するツールセットもあります。LDAP サーバー、KDC、DNS 設定を個別に管理するのではなく、ローカルマシン上にある異なるツールを使用する IdM には、単一の管理ツールセット (CLI および Web UI) があり、ドメインを集約的に、まとめて管理できるようにします。

1.2.1. 認証: Kerberos KDC

Kerberos は認証プロトコルです。Kerberos は共通鍵暗号を使用して、ユーザーに対して **チケット** を生成します。Kerberos 対応のサービスはチケットのキャッシュ (**キータブ**) を確認して、有効なチケットでユーザーを認証します。

他のマシン上のサービスにアクセスする場合でも、パスワードがネットワークで送信されないので、Kerberos 認証は通常のパスワードベースの認証よりもはるかに安全です。からです。

Identity Management では、Kerberos 管理サーバーが IdM ドメインコントローラーで設定され、すべての Kerberos データが IdM のバックエンド Directory Server に保存されます。Directory Server インスタンスは、Kerberos データのアクセス制御を定義して、有効化します。

注記

IdM Kerberos サーバーは、そのデータすべてが Directory Server インスタンスに保存されるため、Kerberos ツールではなく IdM ツールを使用して管理されます。KDC は Directory Server を認識しないため、Kerberos ツールで KDC を管理しても IdM 設定には影響はありません。

1.2.2. データストレージ: 389 Directory Server

Identity Management には、内部の 389 Directory Server インスタンスが含まれます。IdM にあるすべての Kerberos 情報、ユーザーアカウント、グループ、サービス、ポリシー情報、DNS ゾーン、ホストエントリーなどの情報は、この 389 Directory Server インスタンスに保存されます。

389 Directory Server は **マルチマスターレプリケーション** をサポートするので、複数のサーバーを設定すると、サーバー間で相互に対話することができます。合意は、初期サーバーと追加の **レプリカ** の間で自動的に設定されます。

1.2.3. 認証: Dogtag Certificate System

Kerberos は、証明書と、認証のキータブを使用でき、サービスによってはセキュアな通信を行うために証明書が必要です。Identity Management には、サーバーの認証局や Dogtag Certificate System が含まれます。この CA は、IdM ドメイン内のサーバー、レプリカ、ホストおよびサービスに証明書を発行します。

CA は root CA に指定することも、別の外部 CA で定義したポリシー (対象の CA に **従属** させる) を指定することできます。IdM サーバーの設定時に CA が Root CA か従属 CA かが決定されます。

1.2.4. サーバー/クライアントの検出: DNS

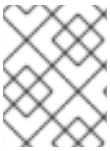
Identity Management はドメインを定義します。ドメイン内では、異なるユーザーおよびサービスが含まれる複数のマシンがあり、それぞれ共有リソースにアクセスし、共有の ID、認証、ポリシー設定を使用します。クライアントは、IdM サーバーとして相互に通信できる必要があります。また、Kerberos などのサービスでは、ホスト名により、プリンシパル ID が特定されます。

ホスト名は、**ドメインネームサービス** (DNS) を使用して IP アドレスに関連付けられます。DNS はホスト名を IP アドレスに、IP アドレスをホスト名にマッピングして、ホストを検索する必要がある場合にクライアントが使用できるリソースを提供します。クライアントが IdM ドメインに登録されると、DNS サービス検出を使用して、ドメイン内の IdM サーバーを特定し、ドメイン内のすべてのサービスおよびクライアントを特定します。

クライアントインストールツールは、サービス検出に IdM ドメインを使用するように、ローカルの System Security Services Daemon (SSSD) を自動的に設定します。SSSD は、すでに DNS を使用して LDAP/TCP サービスおよび Kerberos/UDP サービスを検索するので、クライアントインストールではドメイン名を指定するだけで済みます。SSSD サービス検出については、『Red Hat Enterprise Linux デプロイメントガイド』の「SSSD」の章で説明しています。

サーバーで、インストールスクリプトを使用して、DNS サービス検出クエリーに対応するのはどれかを指定する DNS ファイルを設定します。デフォルトでは、DNS 検出は TCP の LDAP サービスと、

UDP および TCP にあるさまざまな Kerberos サービスにクエリーを実行します。作成された DNS ファイルについては、「既存の DNS 設定での IdM および DNS サービス検出の使用」 で説明します。



注記

IdM サーバーが DNS サービスをホスト **せずに**、DNS サービスを使用するように IdM ドメインを設定することは技術的には可能ですが、これは推奨されません。

通常は、DNS サーバーを複数設定し、それぞれが特定のドメイン内のマシンの管理リソースとして機能します。IdM サーバーを DNS サーバーとして指定するのは任意ですが、強く推奨します。IdM サーバーが DNS を管理する場合には、DNS ゾーンと IdM クライアントが密接に統合され、ネイティブの IdM ツールを使用して IdM クライアントと DNS 設定を管理できます。IdM サーバーが DNS サーバーであっても、他の外部 DNS サーバーも使用できます。

1.2.5. 管理: SSSD

SSSD (System Security Services Daemon) は、認証情報をキャッシュするプラットフォームアプリケーションです。システム認証の多くは、ローカルで設定されているので、サービスは、ローカルユーザーストアをチェックして、ユーザーと認証情報を判断する必要があります。SSSD を使用することで、ローカルサービスは SSSD のローカルキャッシュをチェックできます。このキャッシュは、Identity Management を含むさまざまなリモートアイデンティティープロバイダーから取得できます。

SSSD は、ユーザー名とパスワード、Kerberos プリンシパルおよびキータブ、IPA サーバーで定義された sudo ルール、Identity Management ドメインおよびシステムが使用する SSH 鍵をキャッシュできます。SSSD を使用すると、管理者には大きな利点が 2 つあります。全 ID 設定を 1 つのアプリケーション (IdM サーバー) に集約できる点、システムまたは IdM サーバーが利用できなくなった場合に、ローカルシステムに外部情報をキャッシュして、通常の認証操作を続行できる点です。

SSSD は、IdM クライアントのインストールと管理スクリプトによって自動的に設定されるので、ドメイン設定が変更されても、システム設定を手動で更新する必要はありません。

SSSD では、Windows Active Directory と同様に、ユーザー名属性またはユーザープリンシパル名 (UPN) 属性のいずれかでログインできます。

SSSD は、**case_sensitive** オプションで **true**、**false**、および **preserve** の値をサポートします。**preserve** 値が有効にな場合には、入力内容は大文字と小文字に関係なく一致しますが、出力内容は常にサーバーと同じものを使用します。SSSD は、設定された UID フィールドの大文字、小文字設定を保持します。

SSSD は、バックグラウンドでキャッシュされた特定のエントリーを更新でき、バックエンドは常に更新されているため、エントリーが即時に返されます。現在、ユーザー、グループ、および netgroups のエントリーがサポートされています。

1.2.6. 管理: NTP

多くのサービスでは、特定の差異内でサーバーとクライアントが同一のシステムタイムを保持している必要があります。たとえば、Kerberos チケットはタイムスタンプを使用して有効性を判断します。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。

Network Time Protocol (NTP) を使用して、ネットワーク上でクロックが同期されます。中央サーバーは、信頼できるクロックとして機能し、対象の NTP サーバーを参照するすべてのクライアントが、同じ時刻を使用するように同期します。

IdM サーバーがドメインの NTP サーバーである場合には、他の操作が実行される前に、常に時刻と日

付が同期されます。これにより、パスワードの有効期限、チケット、証明書の有効期限、アカウントのロックアウトの設定、エントリー作成日など、日付関連のサービスがすべて想定通りに機能させることができます。

デフォルトでは、IdM サーバーは、ドメインの NTP サーバーとして機能します。他の NTP サーバーは、ホストに使用することもできます。

1.3. サーバーとクライアント間の関係

Identity Management 自体は、**ドメイン** (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) を定義します。この共有設定により、ドメイン内のマシン (およびユーザー) が相互を認識して共同操作ができるようになります。マシン間の認識機能を使用して、Windows システムと Linux システムの統合などのプラットフォーム間の互換性や、インフラストラクチャー全体のシングルサインオンを有効にできます。

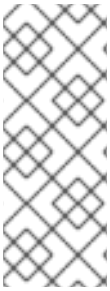
1.3.1. IdM サーバーおよびレプリカの概要

Identity Management は、ユーザーおよびマシン ID およびドメイン全体のポリシーの情報のマスターストアであるサーバーを特定することで機能します。これらのサーバーは、認証局、NTP、Kerberos、SSH、DNS などのドメイン関連のサービスをホストします。このサーバーは、アイデンティティーおよびポリシー情報の中央リポジトリーとしても機能します。

クライアントは、(SSSD および Kerberos を介して) ファイル共有、サービス、リモートマシン、認証などのドメインリソースにアクセスを試みると、IdM サーバーと間接的に対話します。

前述のとおり、IdM サーバーは、多くの関連サービスのコントローラーとなります。これらのサービスの多くが **サポート** されますが、そのほとんどは **必須** ではありません。たとえば、サーバーに CA、DNS サーバー、または NTP サーバーを追加することも、これらのサービスなしでインストールすることもできます。

IdM サーバーの設定が済むと、その設定をコピーして、別の IdM サーバーのベースとして使用できます。IdM サーバーをコピーすると、そのコピーは **レプリカ** と呼ばれます。



注記

IdM サーバーと IdM レプリカの実際の相違点は、サーバーが新規インストールされているかどうかだけです。ドメイン設定を定義するので、レプリカは既存のサーバーおよびドメイン設定をもとに作成されます。

インスタンスが設定されると、IdM ドメイン内における機能や動作の面で、サーバーとレプリカは基本的に同じです。

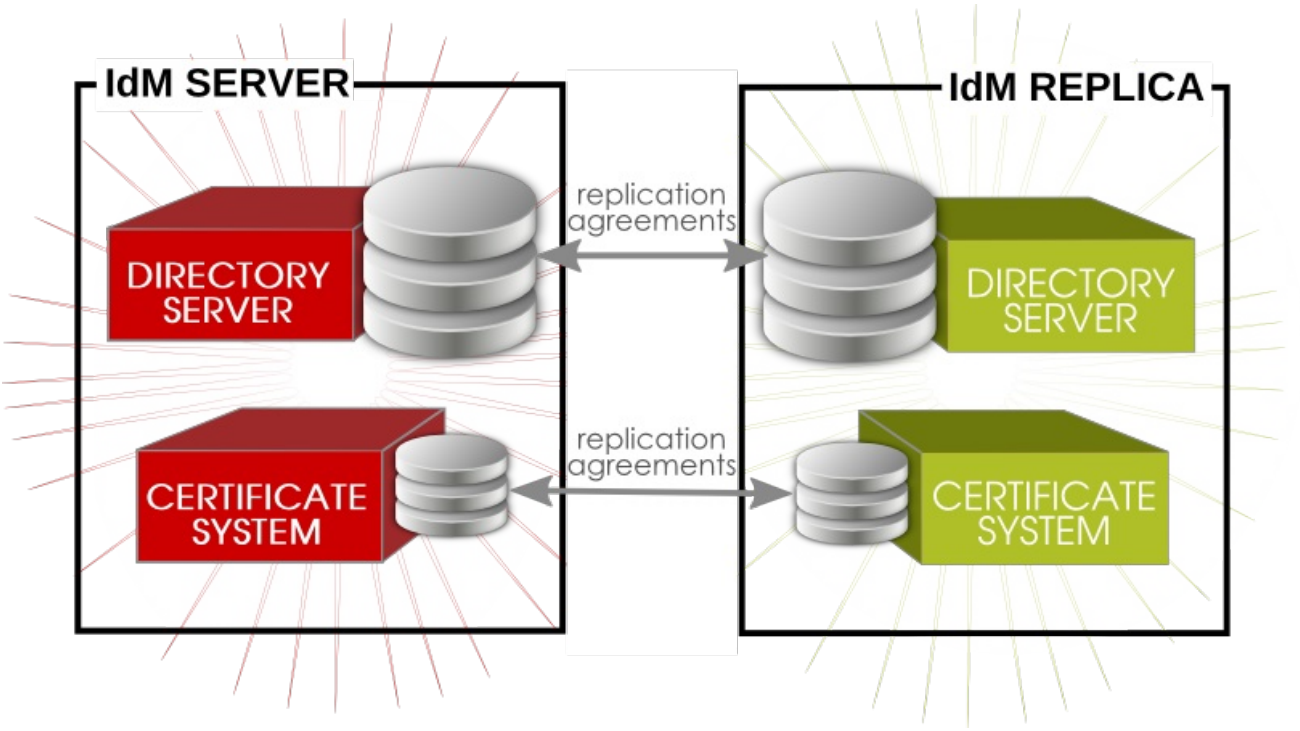
IdM サーバー (およびレプリカ) トポロジーには、柔軟性が十分にあります。たとえば、サーバー A は CA サービスおよび DNS サービスと共にインストールできますが、レプリカ A はサーバー A の設定を基にすることはできますが、DNS や CA サービスをホストできません。レプリカ B は、CA サービスや DNS サービスを使用せずにドメインに追加できます。今後はいつでも、CA または DNS サービスを作成して、レプリカ A またはレプリカ B 上に設定できます。

サーバーとレプリカはいずれも基盤の LDAP ディレクトリーを使用して、ユーザーとホストエントリー、設定データ、ポリシー設定、キータブ、証明書、およびキーを保存します。サーバーおよびレプリカは、**マルチマスターのレプリカ合意** によりデータを相互に伝播します。レプリカ合意は、すべての LDAP バックエンドおよび Dogtag Certificate System が使用する LDAP サブツリーに対して設定されます。サーバーとレプリカはいずれもレプリケーショントポロジーではマスター (ピア) です。

IdM ドメイン内のサーバーはすべて LDAP ピアサーバーであるため、レプリケーショントポロジーは

389 Directory Server ドメインのトポロジー制限に準拠する必要があります。つまり、IdM ドメインに 20 台を超えるピアサーバーを存在させることができません。サーバー/レプリケーショントポロジーのプランニングの詳細は、「サーバー/レプリカトポロジーの計画」を参照してください。

図1.2 サーバーおよびレプリカの対話



ヒント

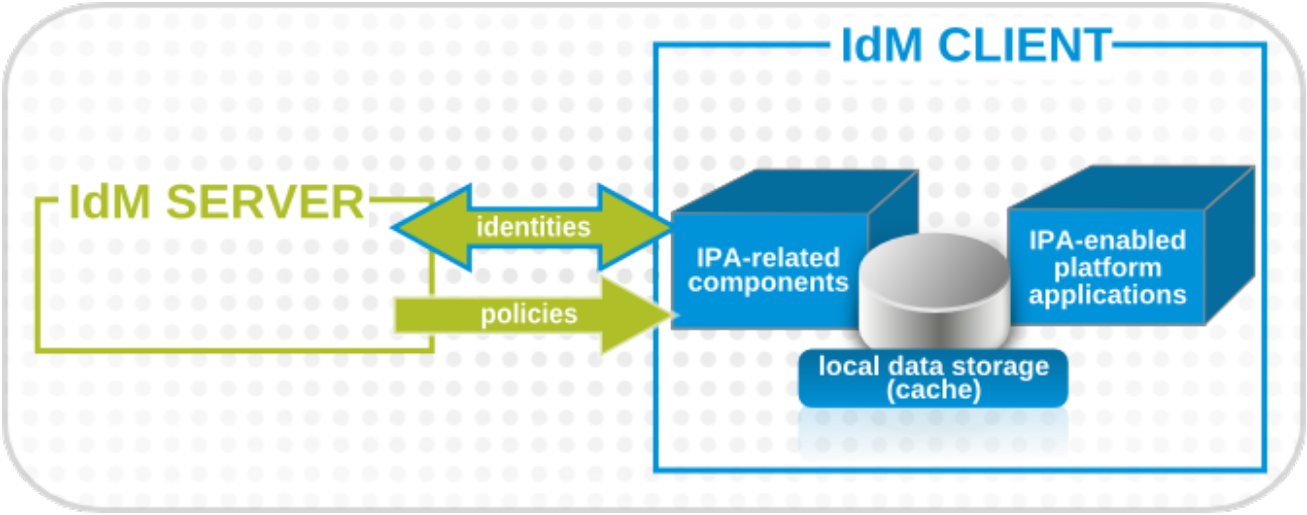
レプリケーショントポロジーは基本的に IdM サーバーのクラウドを作成します。サーバードメインの利点の 1 つに、DNS の SRV レコードを使用した自動負荷分散が挙げられます。SRV レコードは、サーバーやレプリカの問い合わせの優先順位を設定し、加重を使用して同じ優先順位のサーバー/レプリカ間で負荷が分散されます。サーバーおよびレプリカの DNS エントリーを編集して負荷分散を変更できます。この点については、例 17.9「SRV レコード」および「IdM サーバーおよびレプリカの負荷分散の変更」で説明されています。

1.3.2. IdM クライアントの概要

クライアントとは単に、Kerberos および DNS サービス、NTP 設定、および証明書サービスを使用して、IdM ドメイン内で動作するように設定されたマシンのことを指します。クライアントにはデーモンが必要なく、製品をインストールしておく必要がない点が重要な違いです。IdM クライアントにはシステム設定のみが必要で、この設定で、IdM サービスを使用するように指示します。

Red Hat Enterprise Linux システムでは、SSSD (System Security Services Daemon) などの、IdM が使用できるプラットフォームツールが多数あります。IdM サービスと連携する基盤のプラットフォームが使用されている場合には、プラットフォームアプリケーションで IdM が有効になります。特定の PAM モジュールや IdM コマンドラインユーティリティーなどの他のツールは、マシンにインストールされる IdM 固有のパッケージとして Identity Management に同梱されます。これは、Identity Management で使用されるプラットフォームコンポーネントではなく、IdM コンポーネントです。

図1.3 サーバーおよびクライアントの対話



IdM は、クライアントでローカルストレージ (キャッシュ) を使用して、以下のような方法でパフォーマンスを向上します。

- マシンがオフライン時に IdM 情報を保存します。
- クライアントが中央サーバーにアクセスできない場合は、通常のタイムアウト期間が過ぎた後も情報をアクティブな状態で保持します。このキャッシュは、マシンをリブートした後も永続されます。
- サーバーを確認する前にローカルで情報をチェックして、要求のラウンドトリップタイムを短縮します。

情報は、**種類** に応じて LDB データベース (LDAP と同様) またはローカルファイルシステム (XML ファイル) のいずれかに保存されます。

- ユーザー、マシン、およびグループに関する ID 情報は、LDAP ディレクトリーと同じ構文を使用する LDB データベースに保存されます。この ID 情報は、元は IdM サーバーの 389 Directory Server インスタンスに保存されていました。この情報は頻繁に変更、参照されるため、より最新の情報を迅速に呼び出すことが重要ですが、これは、クライアント上の LDB データベースとサーバーの Directory Server を使用することで可能になります。
- ポリシー情報は ID 情報よりも静的で、SELinux または sudo の設定を追加できます。これらのポリシーはサーバー上でグローバルに設定され、クライアントに伝播されます。クライアントでは、 ポリシー情報は XML ファイルのファイルシステムに保存されるので、どのサービスを管理する場合でも、ダウンロードしてネイティブファイルに変換できます。

IdM サーバーの特定のサービスセットは、IdM クライアントのサービスとモジュールのサブセットと対話します。クライアントとは、IdM ドメインからキータブまたは証明書を取得できるマシン (**ホスト**) です。

図1.4 IdM サービス間の対話

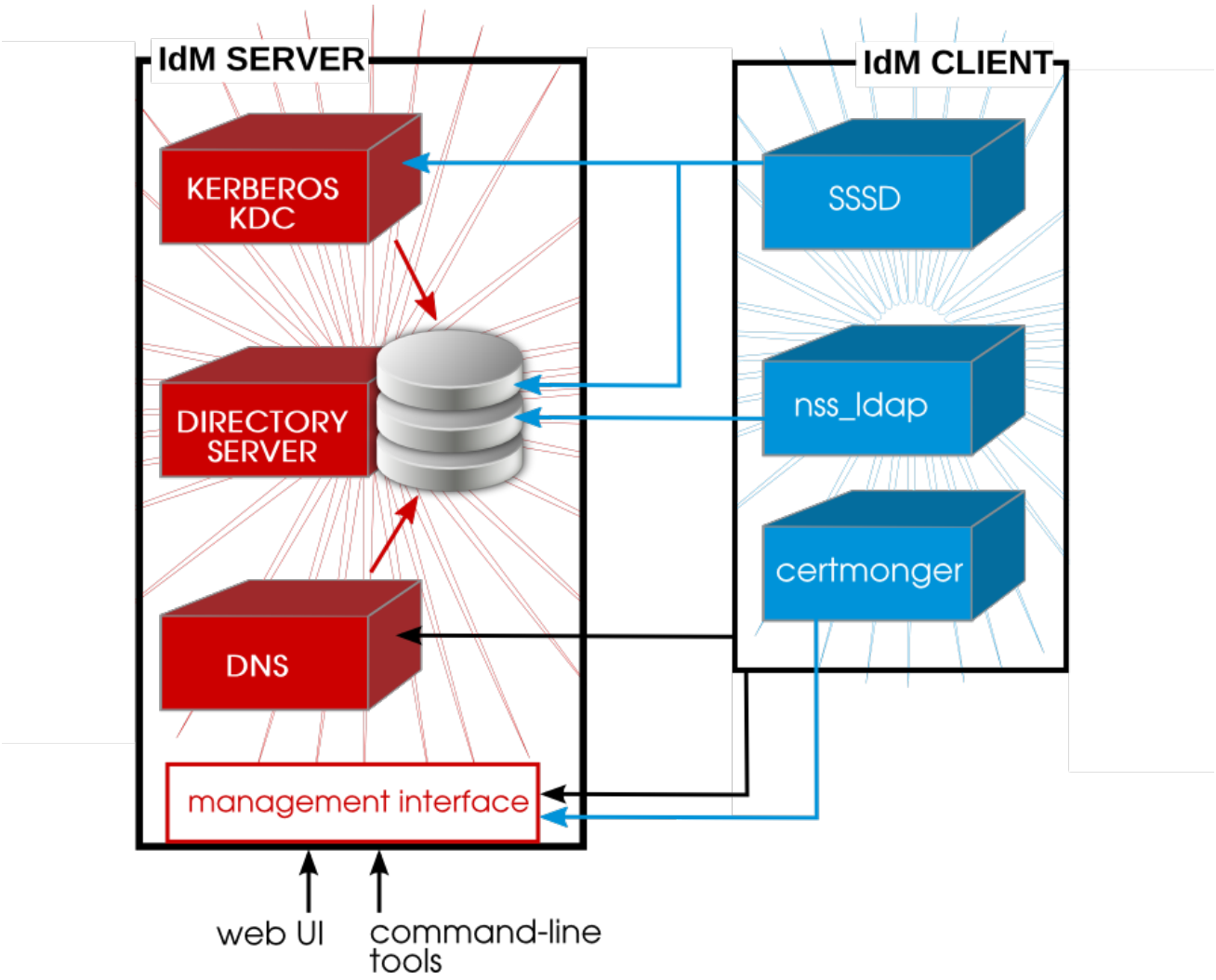


図1.4「IdM サービス間の対話」では、Red Hat Enterprise Linux がネイティブのデーモンを 2 つ使用して IdM サーバーを操作しています。

- SSSD では、マシンのユーザー認証ができ、ホストベースのアクセス制御ルールを有効にします。
- **certmonger** は、クライアント上の証明書を監視、更新します。仮想マシンなどのシステム上のサービス向けに新規の証明書をリクエストできます。

Red Hat Enterprise Linux クライアントがドメインに追加される (**登録される**) と、SSSD と **certmonger** は IdM サーバーに接続するように設定され、必要な Kerberos キータブとホストの証明書が作成されます。(ホスト証明書は、Web サーバーなどの他のサービスで使用される可能性はありますが、IdM では直接使用されません。)

パート I. IDENTITY MANAGEMENT のインストール: サーバーおよびサービス

第2章 インストールの前提条件

IdM をインストールする前に、インストール環境が適切に設定されていることを確認してください。また、レルム名や特定のユーザー名およびパスワードなど、インストールおよび設定手順中に、特定の情報を指定する必要があります。本セクションでは、指定する必要のある情報について説明します。

2.1. サポート対象のサーバープラットフォーム

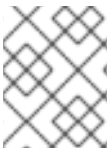
IdM 3.0 へのサポートがあるのは、以下のプラットフォームです。

- Red Hat Enterprise Linux 6 i386
- Red Hat Enterprise Linux 6 x86_64

2.2. ハードウェア推奨事項

証明書のシンプルなホストエントリーと同様に、基本的なユーザーエントリーのサイズは、約 1 KB です。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。すべてのデプロイメントは、ユーザーおよびグループ数や、保存データの種類により異なりますが、以下のように、使用する RAM 容量を判断する一般的な方法があります。

- 10,000 ユーザーおよび 100 グループの場合は、最低 2 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 グループの場合は、最低 16GB の RAM と 4GB のスワップ領域を割り当てます。



注記

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。

IdM サーバーが使用する基本的な Directory Server インスタンスは、パフォーマンスの向上を図るため調整可能です。チューニング情報は、『Directory Server』のドキュメントを参照してください。

2.3. ソフトウェア要件

IdM サーバーに依存するパッケージの大半は、IdM パッケージのインストール時に依存関係としてインストールされます。ただし、IdM パッケージのインストール前に必要なパッケージが複数あります。

- Kerberos 1.10インストールされていない場合は、依存関係としてインストールされます。
- DNS の bind および bind-dyndb-ldap パッケージ。すでにインストールされていない場合には bind パッケージは依存関係としてインストールされますが、bind-dyndb-ldap パッケージは先に明示的にインストールする必要があります。先にインストールされていない場合には、DNS サポートありの IdM サーバーを設定しようとすると失敗します。

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) および **TLSv1.1** に設定します。

```
NSSProtocol TLSv1.0,TLSv1.1
```

2. **httpd** サービスを再起動します。

```
# service httpd restart
```

2.4. システムの要件

ホストシステムに関する一定の前提条件を基に作成された設定スクリプトを使用して、IdM サーバーは設定されています。システムがこの前提条件を満たさないと、サーバーの設定に失敗する可能性があります。

2.4.1. DNS レコード

IdM サーバーおよびレプリカ (サーバーの複製) の療法を設定するには、適切な正引きおよび逆引きの DNS 設定が重要です。DNS は、サーバー間のデータの複製、SSL 証明書でのサーバーの特定、Kerberos チケットなどで使用されます。そのため、サーバーは正引きおよび逆引き両方の DNS 設定で解決できる必要があります。

ホストの DNS 設定は、**ifconfig** と **dig** を使用して簡単に判断できます。

1. ホスト名を取得します。

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

2. IP アドレスを取得します。この例では、**196.2.3.4** の IP アドレスが返されました。

```
[root@server !]# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 52:54:01:4C:E1:2C
          inet addr:196.2.3.4  Bcast:196.9.8.7  Mask:255.255.255.255
          inet6 addr: 2620:52:0:102f:5054:1ff:fe4c:e12c/64 Scope:Global
   inet6 addr: fe80::5054:1ff:fe4c:e12c/64 Scope:Link
...
```

3. **dig** を使用して、ホスト名のクエリーと、返される IP アドレスのチェックを行い、正引き DNS が適切に設定されていることを確認します。この例では、想定される IP アドレスは **196.2.3.4** です。

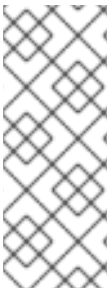
```
[root@server ~]# dig server.example.com
; <<>> DiG 9.8.2rc1-RedHat-9.8.2-0.17.rc1.el6 <<>> server.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 56680
```

```
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 7, ADDITIONAL: 12

;; QUESTION SECTION:
;server.example.com. IN A

;; ANSWER SECTION:
server.example.com. 2946 IN A 196.2.3.4
```

4. **-t ptr** を指定した **dig** を使用して、アドレスの PTR レコード (逆引きレコード) にクエリーを実行し、 逆引き DNS 設定を確認します。これは、**.in-addr.arpa.** が追加された、逆順の IP アドレスです。これにより、ホスト名が解決されます (この例では **server.example.com.**)。

```
[root@server ~]# dig -t ptr 4.3.2.196.in-addr.arpa.
; <<>> DiG 9.8.2rc1-RedHat-9.8.2-0.17.rc1.el6 <<>> -t ptr 241.40.16.10.in-addr.arpa
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 57899
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 7, ADDITIONAL: 10

;; QUESTION SECTION:
;4.3.2.196.in-addr.arpa. IN PTR

;; ANSWER SECTION:
4.3.2.196.in-addr.arpa. 21600 IN PTR server.example.com.
```

DNS レコードは、IdM 証明書で使用されるホスト名を解決する必要があります。



注記

IdM サーバーが独自の DNS サーバーをホストするように設定されている場合には、IdM DNS サービスは全 DNS クエリーを処理します。IdM DNS レコードが優先され、既存の DNS 設定は無視されます。

ドメイン内の全システムは、IdM 管理の DNS サーバーを使用するように設定する必要があります。

2.4.2. ホスト名および IP アドレスの要件

DNS が IdM サーバー内にあるか、外部にあるかに拘らず、サーバーホストには DNS を適切に設定する必要があります。

- ホスト名は完全修飾ドメイン名である必要があります。例: **ipaserver.example.com**

重要

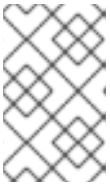
これは、数字、アルファベット文字、およびハイフン (-) のみが使用された有効な DNS 名でなければなりません。ホスト名にアンダースコアなどの他の文字が含まれていると、DNS が正常に機能しなくなります。

- ホスト名はすべて小文字である必要があります。
- サーバーの A レコードを設定し、パブリック IP アドレスを解決する必要があります。

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。**hostname** コマンドの出力は、**localhost** または **localhost6** であってはいけません。

Adn PTR レコードは、サーバーと一致する必要はありません。

- サーバーのホスト名および IP アドレスは独自の **/etc/hosts** ファイルに指定する必要があります。IdM サーバーの完全修飾ドメイン名は、**hosts** ファイルで、エイリアスの **前** にリストする必要があります。



注記

ファイルが正しく設定されていない場合には、IdM コマンドラインツールが正しく機能しなくなり、IdM の Web インターフェースが IdM サーバーに接続できない可能性があります。

また、ホスト名を localhost エントリーに追加することはできません。

たとえば、以下の例では、ホストの IPv4 および IPv6 の localhost エントリーが (正確に) 表示され、最初のエントリーで IdM サーバーの IP アドレスとホスト名がその後に続いています。

```
127.0.0.1 localhost.localdomain localhost
::1  localhost6.localdomain6 localhost6
192.168.1.1 ipaserver.example.com ipaserver
```

- IdM サーバーの管理用に別の DNS ドメインを割り当てることを推奨します。別の DNS ドメインを割り当てることは必須ではありませんが (この IdM ドメインに他のドメインのクライアントを引き続き登録可能)、DNS の管理を行うにあたり便利です。

2.4.3. ディレクトリーサーバー

ホストマシンにインストールされている 389 Directory Server のインスタンスは使用しないでください。

2.4.4. システムファイル

サーバーのスクリプトは、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。システムは、DNS や Kerberos などのサービスのカスタム設定のない、クリーンな状態にしてから、IdM サーバーの設定を行ってください。

2.4.5. システムポート

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、表2.1「IdM ポート」に記載のこれらのポートを解放して利用できるようにする必要があります。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。これらのポートが利用可能かどうかを確認するには、**iptables** ユーティリティーで、利用可能なポートを表示するか、**nc**、**telnet**、または **nmap** ユーティリティーを使用して、ポートに接続するか、ポートのスキャンを実行します。

ポートを解放するには、以下を実行します。

```
[root@server ~]# iptables -A INPUT -p tcp --dport 389 -j ACCEPT
```

iptables(8) man ページには、システムでポートを解放して閉じる方法が詳述されています。

表2.1 IdM ポート

サービス	ポート	タイプ
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
Dogtag Certificate System - LDAP	7389	TCP

2.4.6. NTP

Network Time Protocol (NTP) は、ネットワーク上にあるシステムの時間を同期します。NTP サーバーは、ネットワーククロックの同期を一元管理します。デフォルトでは、Identity Management はドメインで使用される NTP サーバーをインストールして、IdM ドメイン内の他の Identity Management サーバー、レプリカ、システムおよびサービスのクロックを同期します。

正しく機能させるには、一部のドメインタスク (例: Kerberos チケットのメンテナンスおよびトポロジーにあるサーバーとレプリカ間のデータレプリケーション) に対して、NTP サーバーを実行する必要があります。**IdM サーバーが NTP サーバーをホストする必要はありませんが、強く推奨されます。これはデフォルト設定になります。**

サーバーが仮想マシンにインストールされている場合は、そのサーバーで **NTP サーバーを実行しない** でください。IdM の NTP を無効にするには、IdM サーバーの設定時に **--no-ntp** オプションを使用して、NTP サーバーがインストールされないようにします。

2.4.7. NSCD

IdM デプロイメントで **nscd** の使用を回避するか、**制限することを強く推奨** します。**nscd** サービスは、サーバーでの負荷を減らしたり、クライアントの応答性を改善したりするのに非常に便利ですが、システムでも SSSD を使用する場合には独自にキャッシュの操作を行うので、問題が生じる可能性があります。

nscd は、**getent** など、nsswitch でクエリーを実行する全サービスの認証およびアイデンティティー情報をキャッシュします。**nscd** は、ポジティブキャッシュとネガティブキャッシュの両方を実行するので、特定の IdM ユーザーが存在しないと要求が判断した場合には、ネガティブな応答としてキャッシュします。キャッシュに保存されている値は、サーバーでの変更の有無に拘らず、キャッシュの有効期限が切れるまで保持されます。このようなキャッシュを作成すると、新規ユーザーとメンバーシップが表示されない可能性があり、削除されたユーザーおよびメンバーシップが依然として表示される可能性があります。

SSSD キャッシュとの競合やユーザーのロックアウトを回避するには、**nscd** を完全に使用しないようにします。または、**/etc/nscd.conf** ファイルの time-to-live (TTL) のキャッシュの値をリセットして、キャッシュ時間を短縮します。

```
positive-time-to-live   group          3600
negative-time-to-live   group          60
positive-time-to-live   hosts          3600
negative-time-to-live   hosts          20
```

2.4.8. ネットワーク

Red Hat Enterprise Linux ではデフォルトのネットワークサービスとして、NetworkManager が使用されます。ただし、NetworkManager は、IdM および KDC で問題を引き起こす可能性があります。**network サービスを使用して IdM 環境でのネットワーク要件を管理し、NetworkManager サービスを無効にすることを強く推奨します。**

1. シングルユーザーモードでマシンを起動します。
2. 起動リストで NetworkManager サービスを無効にし、NetworkManager サービスを停止します。

```
[root@server ~]# chkconfig NetworkManager off; service NetworkManager stop
```

3. **NetworkManagerDispatcher** がインストールされている場合には、NetworkManagerDispatcher が停止され、無効になっていることを確認します。

```
[root@server ~]# chkconfig NetworkManagerDispatcher off; service
NetworkManagerDispatcher stop
```

4. 次に、**ネットワーク** サービスが適切に起動されていることを確認します。

```
[root@server ~]# chkconfig network on; service network start
```

5. また、静的ネットワークが正しく設定されていることを確認してください。
6. システムを再起動します。

第3章 IDM サーバーのインストール

IdM ドメインは、 IdM **サーバー** (基本的にはドメインコントローラー) によって定義、管理されています。ドメインには、負荷分散とフォールトトレランス向けに、ドメインコントローラーが複数存在する場合があります。これらの追加サーバーは、マスター IdM サーバーの **レプリカ** と呼ばれます。

IdM サーバーおよびレプリカはいずれも、Red Hat Enterprise Linux システムでのみ動作します。サーバーおよびレプリカの両方に、必要なパッケージをインストールする必要があります。その後に、必要な全サービスを構成する設定スクリプトを使用して IdM サーバーまたはレプリカ自体を設定します。

3.1. IDM サーバーパッケージのインストール

IdM サーバーのみをインストールするには、**ipa-server** パッケージだけが必要です。IdM サーバーが DNS サーバーも管理する場合は、DNS の設定に追加パッケージが 2 つ必要になります。

これらのパッケージはすべて、次の **yum** コマンドを使用してインストールできます。

```
[root@server ~]# yum install ipa-server bind bind-dyndb-ldap
```

ipa-server をインストールすると、IdM ツールと合わせて、LDAP サービスの 389-ds-base や Keroberos サービスの krb5-server などの依存関係が多数インストールされます。

パッケージのインストール後に、**ipa-server-install** コマンドを使用してサーバーインスタンスを作成する必要があります。新しいサーバーインスタンスの設定オプションは、「ipa-server-install の概要」に記載されています。

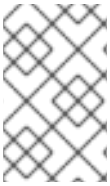
3.2. IPA-SERVER-INSTALL の概要

IdM サーバーインスタンスは、**ipa-server-install** スクリプトを実行して作成されます。このスクリプトは、IdM インスタンスが使用する DNS や Kerberos などのサービスのユーザー定義の設定を指定できます。また、管理者の入力を最小限に抑えられるように時点定義済みの値を指定することもできます。

IdM の設定スクリプトは、IdM ドメインに必要な全サービスの設定などを含めてサーバーインスタンスを作成します。

- ネットワークタイムデーモン(ntpd)
- 389 Directory Server インスタンス
- Kerberos キー配布センター (KDC)
- Apache (httpd)
- 更新された SELinux ターゲットポリシー
- Active Directory WinSync プラグイン
- 認証局
- **任意。** ドメインネームサービス (DNS) サーバー

IdM 設定プロセスは、管理者が指定する情報が一部の必須の情報だけというように、最小限に抑えることができます。それ以外の多くの IdM サービスは、ユーザー定義設定で非常に具体的に指定されています。この設定は、**ipa-server-install** スクリプトで引数を使用して指定します。



注記

「システムポート」と「Identity Management ファイルおよびログ」で定義されているように、IdM が使用するポート番号とディレクトリーの場所はすべて自動的に定義されます。これらのポートおよびディレクトリーは変更またはカスタマイズ **できません**。

必要情報を入力するようにプロンプトが表示されるようにオプションを指定せずに **ipa-server-install** は実行できますが、このコマンドには、複数の引数を指定して、設定プロセスを簡単にスクリプト化したり、対話インストールで要求されない追加情報を指定したりできます。

表3.1「ipa-server-install オプション」には、**ipa-server-install** で使用される共通の引数が記載されています。オプションの完全なリストは **ipa-server-install** の man ページにあります。**ipa-server-install** オプションは、必要に応じて異なるサービスをインストールし、設定するために、特定のデプロイメント環境向けカスタマイズできるほど、汎用性が高くなっています。

表3.1 ipa-server-install オプション

引数	説明
-a **ipa_admin_password**	IdM 管理者のパスワードこれは、admin ユーザーが Kerberos レルムに対して認証する場合に使用されます。
--hostname=**hostname**	IdM サーバーマシンの完全修飾ドメイン名。  **重要** これは、数字、アルファベット文字、およびハイフン (-) のみが使用された有効な DNS 名でなければなりません。ホスト名にアンダースコアなどの他の文字が含まれていると、DNS が正常に機能しなくなります。 さらに、ホスト名がすべて小文字である必要があります。大文字は使用できません。
-n **domain_name**	IdM ドメインに使用する LDAP サーバードメインの名前。これは、通常 IdM サーバーのホスト名に基づいています。
-p **directory_manager_password**	スーパーユーザーのパスワード (LDAP サービスの **cn=Directory Manager**)
-P **kerberos_master_password**	KDC 管理者のパスワード。値が指定されていない場合に無作為に生成されます。
-r **realm_name**	IdM ドメイン用に作成する Kerberos のレルム名。
--subject=**subject_DN**	発行した証明書のサブジェクト DN にベース要素を設定します。デフォルト設定は **O=realm** です。

引数	説明
--forwarder=**forwarder**	DNS サービスで使用する DNS フォワーダーを指定します。複数のフォワーダーを指定するには、このオプションを複数回使用します。
--no-forwarders	フォワーダーではなく DNS サービスを使用するルートサーバーを使用します。
--no-reverse	DNS ドメインの設定時に、逆引き DNS ゾーンが作成 **されないようにします** 。(すでに逆引き DNS ゾーンが設定されている場合は、既存の逆引き DNS ゾーンが使用されます。) このオプションを使用しない場合には、デフォルト値は true になるので、インストールスクリプトで逆引き DNS を設定することを前提としています。
--setup-dns	IdM ドメイン内に DNS サービスを設定するように、インストールスクリプトに指示します。統合 DNS サービスの使用は任意であるため、インストールスクリプトでこのオプションが指定されていない場合には、DNS は設定されません。
--idmax=**number**	IdM サーバーで割り当て可能な ID の最大値を設定します。デフォルト値は ID 開始値 + 199999 です。
--idstart=**number**	IdM サーバーで割り当て可能な ID の最小値 (開始値) を設定します。デフォルト値は無作為に選択されます。
--ip-address	サーバーの IP アドレスを指定します。このオプションは **ipa-server-install** に追加すると、ローカルインターフェースに関連付けられた IP アドレスだけを許可します。

IdM サーバーのインストール方法は、ネットワーク環境、組織内のセキュリティー要件、および必要なトポロジーによって異なります。以下の例は、サーバーのインストール時に使用する一般的なオプションを示しています。これらの例は合わせて使用することができます。同じサーバー呼び出しで CA オプション、DNS オプション、および IdM 設定オプションは問題なく使用できます。単に各設定エリアに必要な内容を明確にするために、上記のオプションは個別に呼び出されます。

3.3. 例: 対話的および無人でのスクリプトの実行

3.3.1. 基本的な対話インストール

ipa-server-install スクリプトを実行するだけで、IdM サーバーを設定できます。これにより、スクリプトが対話的に起動し、サーバーの設定に必要な情報の入力を求めるプロンプトが表示されます。ただし、DNS や CA などの詳細な設定はされません。

1. **ipa-server-install** スクリプトを実行します。

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

2. ホスト名を入力します。ホスト名は逆引き DNS を使用して自動的に決定されます。

```
Server host name [ipaserver.example.com]:
```

3. ドメイン名を入力します。ドメイン名は、ホスト名に基づいて自動的に決定されます。

```
Please confirm the domain name [example.com]:
```

4. 新しい Kerberos レルム名を入力します。Kerberos レルム名は、通常ドメイン名に基づいています。

```
Please provide a realm name [EXAMPLE.COM]:
```

5. Directory Server のスーパーユーザー (**cn=Directory Manager**) のパスワードを入力します。このパスワードには、強度の要件があります。たとえば、パスワードの最小長は 8 文字となっています。

```
Directory Manager password:
Password (confirm):
```

6. 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。このユーザーはマシン上に作成されます。

```
IPA admin password:
Password (confirm):
```

7. 次に、スクリプトにより、ホスト名、IP アドレス、およびドメイン名がもう一度出力されます。情報が正しいことを確認します。

```
The IPA Master Server will be configured with
Hostname:    ipaserver.example.com
IP address:  192.168.1.1
Domain name: example.com
Realm name: EXAMPLE.COM
Continue to configure the system with these values? [no]: yes
```

8. その後、スクリプトで、IdM に関連付けられたサービスをすべて設定します。その際、タスク数と進捗バーが表示されます。

```
Configuring NTP daemon (ntpd)
[1/4]: stopping ntpd
...
Done configuring NTP daemon (ntpd).
Configuring directory server (dirsrv): Estimated time 1 minute
[1/38]: creating directory server user
....
Configuring certificate server (pki-tomcatd): Estimated time 3 minutes 30 seconds
[1/20]: creating certificate server user
...
Done configuring certificate server (pki-tomcatd).
```

```
Configuring Kerberos KDC (krb5kdc): Estimated time 30 seconds
[1/10]: adding sasl mappings to the directory
...
Done configuring Kerberos KDC (krb5kdc).
Configuring kadmin
[1/2]: starting kadmin
[2/2]: configuring kadmin to start on boot
Done configuring kadmin.
Configuring ipa_memcached
[1/2]: starting ipa_memcached
[2/2]: configuring ipa_memcached to start on boot
Done configuring ipa_memcached.
Configuring ipa-otpd
[1/2]: starting ipa-otpd
[2/2]: configuring ipa-otpd to start on boot
Done configuring ipa-otpd.
Configuring the web interface (httpd): Estimated time 1 minute
[1/15]: disabling mod_ssl in httpd
...
Done configuring the web interface (httpd).
Applying LDAP updates
Restarting the directory server
Restarting the KDC
Sample zone file for bind has been created in /tmp/sample.zone.pUfcGp.db
Restarting the web server

Setup complete
```

9. **SSH** サービスを再起動して、Kerberos プリンシパルを取得し、ネームサーバースイッチ (NSS) 設定ファイルを更新します。

```
[root@server ~]# service sshd restart
```

10. admin ユーザーの認証情報を使用して Kerberos レルムに認証を行い、ユーザーが適切に設定され、Kerberos レルムにアクセスできることを確認します。

```
[root@server ~]# kinit admin
Password for admin@EXAMPLE.COM:
```

11. **ipa user-find** のようなコマンドを実行して IdM 設定をテストします。たとえば、以下のようになります。

```
[root@server ~]# 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
----------------------------
```

3.3.2. 無人 (非対話型) インストール

「基本的な対話インストール」に記載されえいるように、IdM サーバーの設定に必要なのはわずかな情報だけです。設定スクリプトを使用すると対話モードでこの情報をプロンプトで求めることができますが、設定コマンドを使用してこの情報を指定する無人自動設定も可能です。

- IdM 管理ユーザーおよび Directory Server のスーパーユーザー (Directory Manager) のパスワード
- サーバーのホスト名
- Kerberos レルム名
- DNS ドメイン名

ipa-server-install に **-U** を指定して上記の情報を渡すことで、ユーザーの操作を必要とせずに強制的に実行できるようになります。

例3.1 非対話式の基本的なインストール

```
[root@server ~]# ipa-server-install -a secret12 --hostname=ipaserver.example.com -r
EXAMPLE.COM -p secret12 -n example.com -U
```

次に、スクリプトにより、指定された値が出力されます。

```
To accept the default shown in brackets, press the Enter key.

The IPA Master Server will be configured with
Hostname:    ipaserver.example.com
IP address:  192.168.1.1
Domain name: example.com
```

サーバー名は、英数字、ハイフン (-) のみで構成される、有効な DNS 名でなければなりません。ホスト名にアンダースコアなどの他の文字が含まれていると、DNS が正常に機能しなくなります。さらに、ホスト名がすべて小文字である必要があります。大文字は使用できません。

次に、「基本的な対話インストール」にあるように、スクリプトが実行され、IdM サービスごとに設定が進められます。

3.4. 例: 異なる CA 設定を使用したインストール

Identity Management では、統合された認証局 (CA) を使用して、ドメイン内のユーザーおよびホストが使用する証明書および Keytab を作成します。Identity Management Web UI の LDAP サーバーや Apache サーバーなどの内部ドメインサービスでも、サーバー間でセキュアな接続を確立するにはサーバー証明書が必要になります。

大抵の場合、Dogtag Certificate System CA は、IdM サーバーでインストールされます。この Dogtag Certificate System CA は **CA 署名証明書** を使用して、IdM ドメイン内で作成されたすべてのサーバー証明書とユーザー証明書を作成して署名します。CA 証明書自体は、発行元の CA で署名する必要があ

ります。発行元の CA を使用して、Dogtag Certificate System を CA 署名証明書に署名する方法は 2 つあります。

- Dogtag Certificate System は、**独自** の証明書に署名できます。つまり、Dogtag Certificate System インスタンスは **ルート CA** であることを意味します。ルート CA より上位の CA はないので、ルート CA cna で独自の証明書ポリシーを設定できます。

  これがデフォルト設定になります。

- Dogtag Certificate System CA は、外部でホストされる CA (例: Verisign) で署名できます。この場合には、外部 CA がルート CA になり、設定された Dogtag Certificate System CA はルート CA の **下位** の証明局になります。つまり、IdM ドメイン内で発行された証明書は、有効期間などの属性に関してルート CA によって設定された制限が適用される可能性があります。

  外部 CA を参照する場合も、引き続き Dogtag Certificate System インスタンスを使用してすべての IdM ドメイン証明書証明書を発行します。唯一の相違点は、初期のドメイン CA 証明書が別の CA によって発行される点です。

他に、CA なしのインストールというオプションがあります。こちらのオプションでは、IdM ドメイン内で使用されているすべての証明書を手動で作成してアップロードし、更新する必要があります。インフラストラクチャー内の他の制限により、さらにメンテナンス負荷がかかる環境もありますが、通常、ほとんどのデプロイメントでは統合 Dogtag Certificate System インスタンス (および **certmonger**) を使用して IdM ドメイン証明書を管理します。

重要

CA 設定は、ドメインの作成後に変更したり、別の設定に移行したりできません。インストールプロセスの開始前に CA 要件を考慮する必要があります。

3.4.1. 内部ルート CA を使用したインストール

デフォルト設定では、独自のルート CA 証明書に署名する Dogtag Certificate System をインストールします。**ipa-server-install** コマンドの実行時に追加のパラメーターや設定手順は必要ありません。

```
[root@server ~]# ipa-server-install
... &< ...

The IPA Master Server will be configured with:
Hostname:      server.example.com
IP address:    10.1.1.1
Domain name:   example.com
Realm name:    EXAMPLE.COM

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

The following operations may take some minutes to complete.
Please wait until the prompt is returned.

... &< ...

Configuring directory server for the CA (pkids): Estimated time 30 seconds
  [1/3]: creating directory server user
  [2/3]: creating directory server instance
  [3/3]: restarting directory server
Done configuring directory server for the CA (pkids).
```

```
Configuring certificate server (pki-cad): Estimated time 3 minutes 30 seconds
  [1/21]: creating certificate server user
...
Done configuring certificate server (pki-cad).

... &< ...
```

3.4.2. 外部 CA を使用したインストール

IdM サーバーは、外部 CA 発行の証明書を使用できます。この外部 CA は、企業 CA や、Verisign や Thawte などのサードパーティー CA を利用できます。通常の設定プロセスと同様に、外部 CA は引き続き IdM サーバーの Dogtag Certificate System インスタンスを使用して、クライアント証明書とレプリカ証明書をすべて発行し、初期 CA 証明書は単に別の CA により発行されるだけです。

外部 CA を使用する場合は、生成された証明書要求を外部 CA に送信し、CA 証明書を読み込み、サーバー証明書を発行する手順 2 つを追加で実行して設定を完了する必要があります。

重要

Identity Management サーバー用に生成された CA 署名証明書は、有効な CA 証明書である必要があります。これには、**Basic Constraint** オプションを **CA=TRUE** に設定するか、証明書に署名できるように、署名証明書に鍵用途エクステンションを設定する必要があります。

重要

CA 設定は、ドメインの作成後に変更したり、別の設定に移行したりできません。インストールプロセスの開始前に CA 要件を考慮する必要があります。

例3.2 外部 CA の使用

1. **--external-ca** オプションを使用して **ipa-server-install** スクリプトを実行します。

   ```
   [root@server ~]# ipa-server-install -a secret12 -r EXAMPLE.COM -P password -p
   secret12 -n ipaserver.example.com --external-ca
   ```

2. このスクリプトは、通常通りに NTP サービスおよび Directory Server サービスを設定し、

3. CA の設定を完了して証明書署名要求 (CSR) が置かれている場所 (**/root/ipa.csr**) に関する情報を返します。この要求は外部 CA に送信する必要があります。

   ```
   Configuring certificate server: Estimated time 6 minutes
     [1/4]: creating certificate server user
     [2/4]: creating pki-ca instance
     [3/4]: restarting certificate server
     [4/4]: configuring certificate server instance
   The next step is to get /root/ipa.csr signed by your CA and re-run ipa-server-install.
   ```

4. CA に要求を送信します。このプロセスは、サービスごとに異なります。

   証明書の適切な拡張を要求する必要がある場合があります。Identity Management サーバー用に生成された CA 署名証明書は、有効な CA 証明書である必要があります。これには、基本制約を **CA=true** に設定するか、証明書に署名できるように、署名証明書に鍵用途エクス

テンションを設定する必要があります。

5. 発行した証明書と、発行元 CA の CA 証明書チェーンを取得します。プロセスは証明書サービスによって異なりますが、通常はWeb ページか通知メールにダウンロードリンクがあり、管理者は、必要な証明書すべてをダウンロードできます。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。

6. 証明書および CA チェーンファイルの場所と名前を指定して **ipa-server-install** をもう一度実行します。たとえば、以下のようになります。

```
[root@server ~]# ipa-server-install --external_cert_file=/tmp/servercert20110601.p12 --external_ca_file=/tmp/cacert.p12
```

7. 「基本的な対話インストール」にあるように、設定プロセスを完了し、すべてが想定通りに機能していることを確認します。

3.4.3. CA なしでのインストール

非常にまれなケースでは、Identity Management サーバーで証明書サービスをインストールすることができない場合があります。このような場合には、**証明書を作成して、個別にインストールしている限り**、統合された証明書システムインスタンスなしで Identity Management をインストールできます。

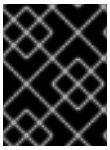
インストールには、3 つの証明書が必要です。

- LDAP サーバー証明書
- Apache サーバー証明書
- LDAP サーバー証明書

この証明書は、インストールプロセスの開始 **前** に、サードパーティーの認証局から要求する必要があります。

Dogtag Certificate System システムインスタンスが統合されていない場合に、証明書の管理方法には重要な制限事項があります。

- 証明書の追跡に **certmonger** が使用されないので、有効期限の警告はありません。
- Identity Management で証明書を更新する方法はありません。
- 証明書管理ツール (**ipa cert-***) は、証明書の表示や管理には使用できません。
- ホスト証明書とサービス証明書はすべて、手動で要求、生成、アップロードする必要があります。これは、**ipa host-add** などのホスト管理ツールが機能する仕組みにも影響します。
- 証明書がエントリーから削除されても、自動的に取り消されません。



重要

CA 設定は、ドメインの作成後に変更したり、別の設定に移行したりできません。インストールプロセスの開始前に CA 要件を考慮する必要があります。

例3.3 CA を使用しない Identity Management のインストール

CA を使用せずにインストールする場合には、必須オプションが 5 つあり、必要な証明書を設定プロセスに直接渡す必要があります。

- **LDAP サーバー証明書**
  - --dirsrv_pkcs12 (LDAP サーバー証明書の PKCS#12 証明書ファイルを指定)
  - --dirsrv_pin (PKCS#12 ファイルにアクセスするパスワードを指定)
- **Apache サーバーの証明書**
  - --http_pkcs12 (Apache サーバー証明書の PKCS#12 証明書ファイルを指定)
  - --http_pin (PKCS#12 ファイルにアクセスするパスワードを指定)
- **ルート CA 証明書** (Apache および LDAP サーバーの証明書をドメイン全体で信頼できるようにする)

```
[root@server ~]# ipa-server-install --http_pkcs12 /tmp-http-server.p12 --http_pin secret1 --
dirsrv_pkcs12 /tmp/ldap-server.p12 --dirsrv_pin secret2 ...
```

3.5. 例: IDM ドメイン内での DNS サービスの設定

IdM は、独自の DNS を管理したり、既存の DNS (デフォルト) を使用したりするように設定できます。設定スクリプトを実行するだけでは DNS は設定されません。DNS の設定には **--setup-dns** オプションが必要です。

警告

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。

IdM ドメインで IdM ホストの DNS サーバーを使用しない場合には、細心の注意を払い、テスト済みで、機能する DNS サービスを利用可能であることを確認します。A および PTR レコードを適切に設定していることが重要です。

基本設定と同様に、DNS 設定では、必要な情報の入力をプロンプトで求めるか、自動または無人セットアッププロセスを許可するスクリプトを使用して DNS 情報を指定できます。

3.5.1. DNS に関する注意事項

- DNS 名の設定時にはワイルドカードを使用できません。明示的な DNS ドメイン名のみがサポートされます。
- **--setup-dns** オプションを指定しても、**rndc** サービスは設定されません。このサービスは、IdM サーバーの設定後に手動で設定する必要があります。

3.5.2. 統合 DNS を使用したインストール

例3.4 対話型の DNS 設定

1. **--setup-dns** オプションを使用して **ipa-server-install** スクリプトを実行します。

```
[root@server ~]# ipa-server-install -a secret12 -r EXAMPLE.COM -P password -p
secret12 -n ipaserver.example.com --setup-dns
```

2. スクリプトを使用すると通常通りにホスト名およびドメイン名を設定します。

3. 次にスクリプトにより、DNS フォワーダー設定のプロンプトが表示されます。フォワーダーを使用する場合は、yes と入力して DNS サーバーの一覧を指定します。IdM で独自の DNS サービスを管理する場合は、no と入力します。

```
Do you want to configure DNS forwarders? [yes]: no
No DNS forwarders configured
```

4. このスクリプトは、NTP、Directory Server、Certificate System、Kerberos、および Apache のサービスを設定します。

5. 設定の完了前に、逆引き DNS サービスを設定するかどうかをスクリプトによりプロンプトが表示されます。yes を選択した場合は、**named** サービスが設定されます。

```
Do you want to configure the reverse zone? [yes]: yes
Configuring DNS (named)
[1/11]: adding DNS container
[2/11]: setting up our zone
[3/11]: setting up reverse zone
[4/11]: setting up our own record
[5/11]: setting up records for other masters
[6/11]: setting up CA record
[7/11]: setting up kerberos principal
[8/11]: setting up named.conf
[9/11]: restarting named
[10/11]: configuring named to start on boot
[11/11]: changing resolv.conf to point to ourselves
Done configuring DNS (named).
==============================================================================
======
Setup complete
```

6. **ipa-dns-install** コマンド (**--setup-dns** オプションの指定時にインストールスクリプトで実行) では、システムの **rndc** サービスは自動的に設定されません。このサービスは、DNS を IdM に設定した後に手動で設定する必要があります。

   a. **rndc** 設定ファイルとキーを作成します。

```
[root@server ~]# /usr/sbin/rndc-confgen -a
[root@server ~]# /sbin/restorecon /etc/rndc.key
```

   これには、キーの作成時にユーザーが入力してエントロピーを作成する必要がある場合があります。

   b. **rndc** キーファイルの所有者と権限を変更します。

```
[root@server ~]# chown root:named /etc/rndc.key
[root@server ~]# chmod 0640 /etc/rndc.key
```

7. 「基本的な対話インストール」にあるように、すべてが想定通りに機能していることを確認します。

DNS を IdM で使用する場合は、使用する DNS フォワーダーの情報、逆引き DNS を使用するかどうかの情報の 2 つが必要になります。非対話的なセットアップを実行する場合は、**--forwarder** または **--no-forwarders** オプションおよび **--no-reverse** オプションを使用してこの情報を渡すことができます。

例3.5 非対話的な DNS の設定

IdM サーバーに DNS サーバーとドメインを設定するには、**--setup-dns** オプションを使用します。追加のフォワーダーを設定するには、**--forwarder** オプションを使用します。複数のフォワーダーを指定する場合には、複数の **--forwarder** の呼び出しを使用します。

```
[root@server ~]# ipa-server-install ... --setup-dns --forwarder=1.2.3.0 --forwarder=1.2.255.0
```

一部のフォワーダー情報が必要です。IdM DNS サービスで外部フォワーダーを使用しない場合は、この **--no-forwarders** オプションを使用してルートサーバーのみを使用することを指定します。

このスクリプトでは、逆引き DNS は DNS があることを前提に設定されているので、逆引きDNSを**有効化** するオプションを使用する必要はありません。逆引き DNS を無効にするには、**--no-reverse** オプションを使用します。逆引き DNS ゾーンがすでに設定されている場合は、この **--no-reverse** オプションを使用すると既存の逆引き DNS ゾーンが使用されます。

```
[root@server ~]# ipa-server-install ... --setup-dns --no-reverse
```

ipa-dns-install コマンド (**--setup-dns** オプションの指定時にインストールスクリプトで実行) では、システムの **rndc** サービスは自動的に設定されません。このサービスは、DNS を IdM に設定した後に手動で設定する必要があります。

1. **rndc** 設定ファイルとキーを作成します。

   ```
   [root@server ~]# /usr/sbin/rndc-confgen -a
   [root@server ~]# /sbin/restorecon /etc/rndc.key
   ```

   これには、キーの作成時にユーザーが入力してエントロピーを作成する必要がある場合があります。

2. **rndc** キーファイルの所有者と権限を変更します。

   ```
   [root@server ~]# chown root:named /etc/rndc.key
   [root@server ~]# chmod 0640 /etc/rndc.key
   ```

第4章 IDM レプリカの設定

レプリカは基本的には既存の Identity Management サーバーのクローンで、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は主に、既存の必要なサーバー設定をコピーし、その情報に基づいてレプリカをインストールするという 2 つの部分で構成されます。

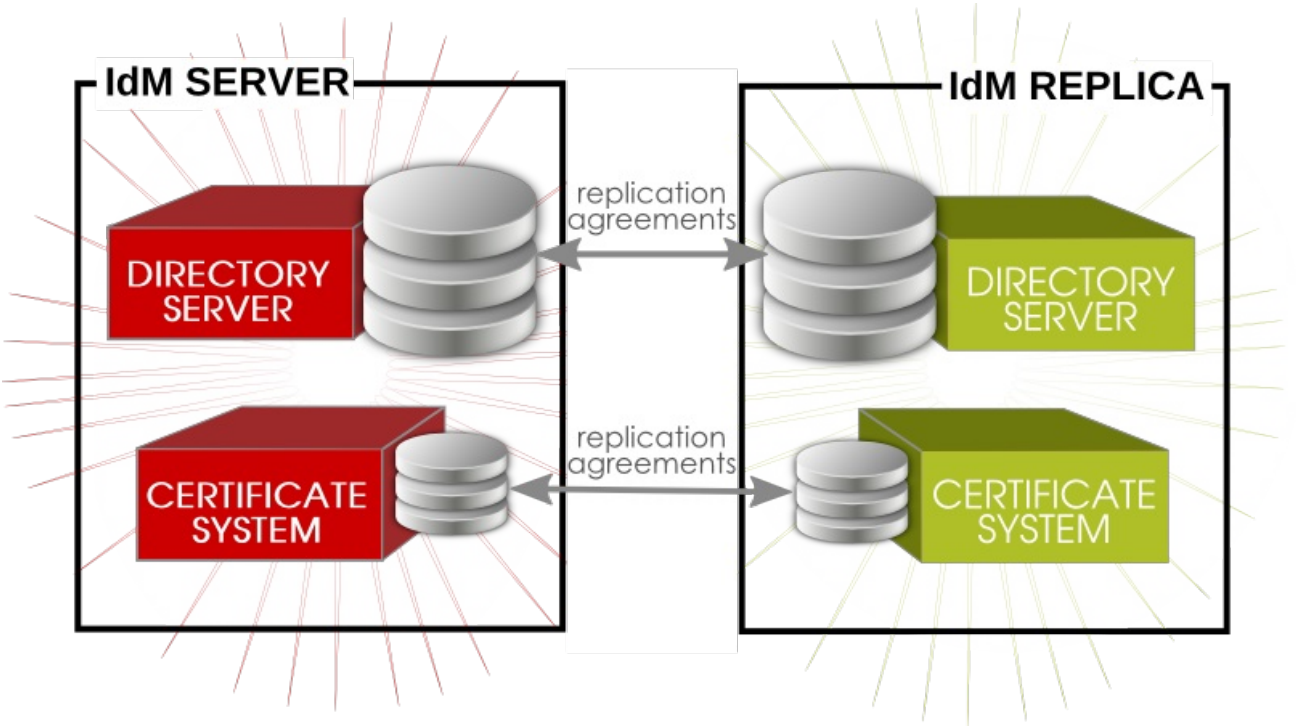
4.1. サーバー/レプリカトポロジーの計画

IdM ドメインには、以下の 3 種のマシンタイプがあります。

- サーバー。ドメインの所属メンバーが使用する全サービスを管理します。
- レプリカ。レプリカは基本的にはサーバーの複製です (一旦複製されるとサーバーと全く同じです)。
- クライアント。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。

レプリカは、特定の IdM サーバーのクローンです。サーバーとレプリカは、ユーザー、マシン、証明書、および設定されたポリシーの内部情報が同じです。これらのデータは、**レプリケーション** と呼ばれるプロセスで、サーバーからレプリカにコピーされます。IdM サーバーが使用する 2 つの Directory Server インスタンス (IdM サーバーが使用する Directory Server インスタンスと、証明書情報を保存する Dogtag Certificate System が使用する Directory Server インスタンス) は、IdM レプリカにより使用される対応のコンシューマー Directory Server インスタンスに複製されます。異なる Directory Server インスタンスは、**レプリカ合意** を使用して相互を認識します。初期レプリカ合意は、レプリカの作成時にマスターサーバーとレプリカとの間で作成されます。他のサーバーまたはレプリカに追加で合意を作成するには、**ipa-replica-manage** コマンドを使用します。

図4.1 サーバーとレプリカの合意



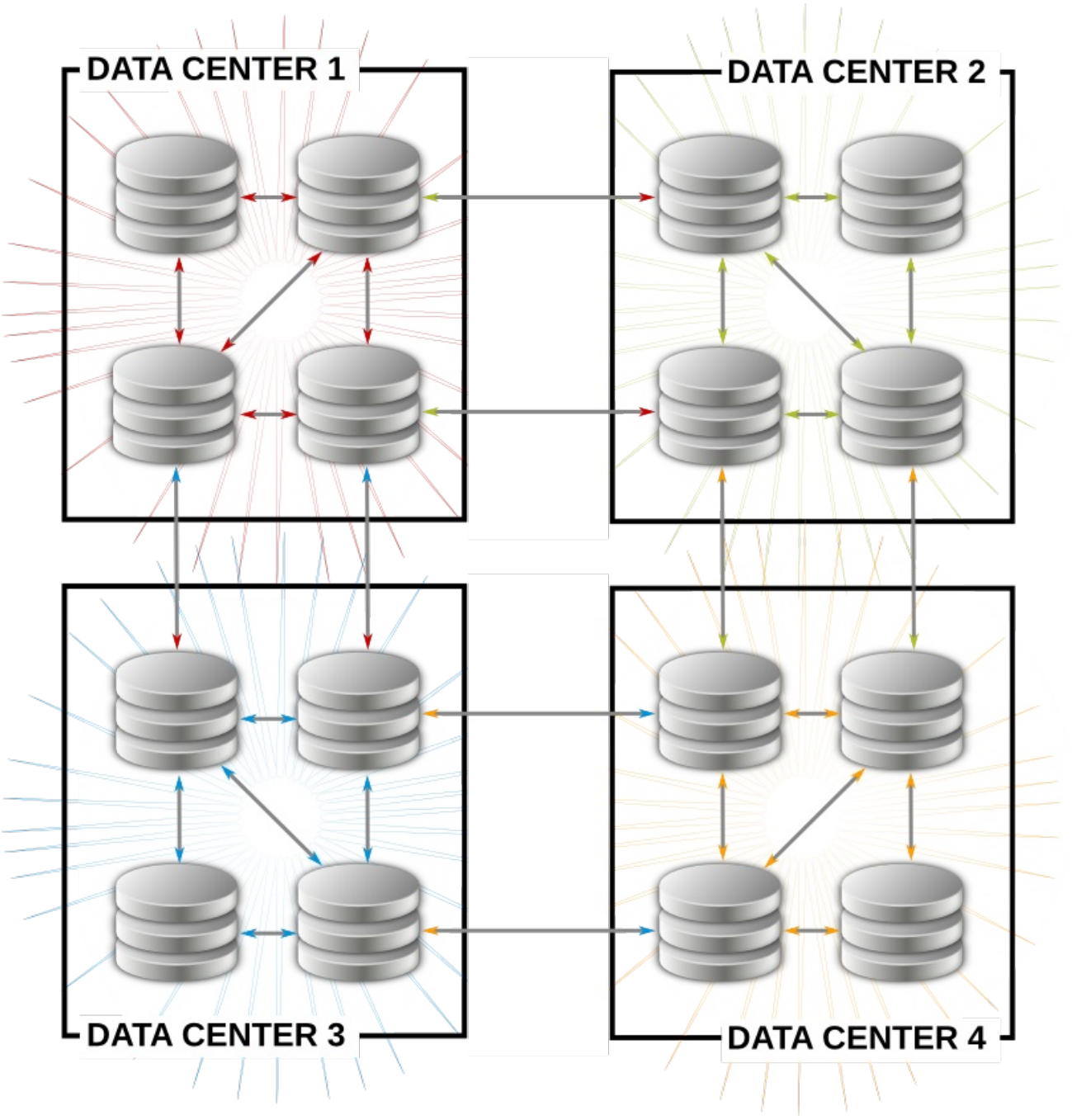
レプリカは、 インストール後はサーバーと機能的に同じになります。

マルチマスターレプリケーションには、ガイドラインがあり、サーバー/レプリカトポロジー全体に制限を指定します。

- 1 つのサーバー/レプリカには、4 つ以上のレプリカ合意を設定できません。
- 1 つの Identity Management ドメインには 20 台以上のサーバーとレプリカを使用すべきではありません。
- すべてのサーバー/レプリカには、別のサーバーに障害が発生した場合に、孤立したサーバーまたはレプリカがないようにするため、最低でも 2 つのレプリカ合意が必要です。

最も耐障害性のあるトポロジーの 1 つとして、サーバー/レプリカのセル設定の作成が挙げられます。この設定では、少数のサーバーがセルにあり、すべてのサーバーには相互にサーバー合意が指定されており (厳密なセル)、そのセルの **外** では、サーバーごとに別のサーバーとレプリカ合意が指定されていて、そのセルと、ドメイン全体にある他のすべてのセルとを疎結合します。

図4.2 トポロジーの例



これを簡単に実現するための推奨の方法がいくつかあります。

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

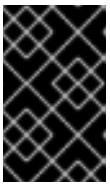
4.2. レプリカサーバーのインストールの前提条件

レプリカは、IdM サーバーと機能的に同じであるため、インストール要件、インストールパッケージは同じです。ただし、レプリカも既存サーバーの **複製** であるため、元のサーバー設定をミラーリングする必要があります。

- そのマシンが「2章 *インストールの前提条件*」に記載の前提条件すべてを満たすようにしてください。
- **レプリカとマスターサーバーは、同じバージョンの IdM を実行している必要があります。**

  レプリカは基本的に、既存のサーバー設定を基にしたサーバーの複製です。そのため、サーバーとレプリカ (サーバーの複製) は、設定をサーバーからレプリカに適切にコピーできるように、同じバージョンの Identity Management を実行している必要があります。

  マスターサーバーが Red Hat Enterprise Linux 6 で IdM バージョン 3.0 を稼働している場合は、レプリカも Red Hat Enterprise Linux 6 で稼働し、IdM 3.0 パッケージを使用する必要があります。

  重要

  マスターと違うバージョンを使用するレプリカの作成は **サポート対象外です**。389 Directory Server インスタンスの設定時に、別のバージョンを使用するレプリカを作成しようとすると失敗します。

- レプリカをインストールするには、レプリカの設定プロセス時に 表2.1「IdM ポート」に記載されているポートに加え、**ポート 22** も解放する必要があります。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。

  既存の Dogtag Certificate System または Red Hat Certificate System インスタンスがレプリカマシンに存在する場合には、レプリカの設定中および設定後に **ポート 7389** を解放しておく必要があります。このポートは、マスター IdM サーバーがレプリカと通信するのに使用されます。

  注記

  ipa-replica-install スクリプトには、必要なポートのステータスを検証する **ipa-replica-conncheck** ユーティリティーが含まれます。トラブルシューティングの目的で、**ipa-replica-conncheck** を個別に実行することもできます。ユーティリティーの使用方法は、ipa-replica-conncheck(1) の man ページを参照してください。

- レプリカはサーバーと同じ CA 設定を使用し、同じルート CA を指定する必要があります。たとえば、サーバーが (Dogtag Certificate System を使用した) 独自のルート CA である場合には、このサーバーはレプリカのルート CA である必要があります。サーバーが外部 CA を使用して証明書を発行した場合には、レプリカでも同じ外部 CA を使用する必要があります。

4.3. レプリカパッケージのインストール

レプリカは (既存サーバーの設定に基づく) IdM サーバーであるため、IdM サーバーパッケージ (**ipa-server**) からインストールされます。レプリカが DNS サービスもホストする場合は、**bind** および **bind-dyndb-ldap** パッケージを含めます。

```
[root@server ~]# yum install ipa-server bind bind-dyndb-ldap
```

重要

ipa-server-install スクリプトを **実行しないでください**。

4.4. レプリカの作成

1. マスターサーバーで、**レプリカ情報ファイル** を作成します。このファイルには、マスターサーバーから取得したレルムや設定情報が含まれており、情報はレプリカサーバーの設定に使用します。

   マスター IdM サーバー で **ipa-replica-prepare** ユーティリティーを実行します。このユーティリティーには、**レプリカ** マシンの完全修飾ドメイン名が必要です。

   --ip-address オプションを使用すると、DNS へのレプリカの A および PTR レコードなど、レプリカの DNS エントリーが自動的に作成されます。

   重要

   IdM サーバーが統合 DNS で設定されている場合にのみ **--ip-address** オプションを渡します。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。

   注記

   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。

```
[root@server ~]# ipa-replica-prepare ipareplica.example.com --ip-address 192.168.1.2

Directory Manager (existing master) password:
Preparing replica for ipareplica.example.com from ipaserver.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-ipareplica.example.com.gpg
Adding DNS records for ipareplica.example.com
Using reverse zone 1.168.192.in-addr.arpa.
The ipa-replica-prepare command was successful
```

これは、数字、アルファベット文字、およびハイフン (-) のみが使用された有効な DNS 名でなければなりません。ホスト名にアンダースコアなどの他の文字が含まれていると、DNS が正常に機能しなくなります。さらに、ホスト名がすべて小文字である必要があります。大文字は使用できません。

各レプリカ情報ファイルは、GPG 暗号化ファイルとして **/var/lib/ipa/** ディレクトリーに作成されます。各ファイルには、**replica-info-ipareplica.example.com.gpg** など、レプリカサーバー向けの名前が付けられます。

注記

レプリカ情報ファイルを使用して、複数のレプリカを作成できません。このファイルを使用できるのは、対象として作成された特定のレプリカとマシンだけです。

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 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. レプリカ情報ファイルは、レプリカサーバーにコピーします。

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

3. レプリカサーバーで、レプリカのインストールスクリプトを実行し、このレプリカ情報ファイルを参照します。サーバーのインストールスクリプトのように、DNS を設定する方法は他にもあります。さらに、レプリカの CA を設定するオプションがあります。CA はサーバー用にデフォルトでインストールされますが、レプリカでは任意です。

   DNS フォワーダーの情報が必要です。フォワーダーごとに **--forwarder** オプションを使用して、設定した DNS フォワーダーの一覧を指定するか、**--no-forwarders** オプションを指定してフォワーダーの設定をスキップできます。

   たとえば、以下のようになります。

```
[root@ipareplica ~]# ipa-replica-install --setup-ca --setup-dns --no-forwarders
/var/lib/ipa/replica-info-ipareplica.example.com.gpg

Directory Manager (existing master) password:

Warning: Hostname (ipareplica.example.com) not found in DNS
Run connection check to master
Check connection from replica to remote master 'ipareplica. example.com':
   Directory Service: Unsecure port (389): OK
   Directory Service: Secure port (636): OK
   Kerberos KDC: TCP (88): OK
```

```
    Kerberos Kpasswd: TCP (464): OK
    HTTP Server: Unsecure port (80): OK
    HTTP Server: Secure port (443): OK

The following list of ports use UDP protocol and would need to be
checked manually:
    Kerberos KDC: UDP (88): SKIPPED
    Kerberos Kpasswd: UDP (464): SKIPPED

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@EXAMPLE.COM password:

Execute check on remote master
admin@example.com's password:
Check connection from master to remote replica 'ipareplica. example.com':
    Directory Service: Unsecure port (389): OK
    Directory Service: Secure port (636): OK
    Kerberos KDC: TCP (88): OK
    Kerberos KDC: UDP (88): OK
    Kerberos Kpasswd: TCP (464): OK
    Kerberos Kpasswd: UDP (464): OK
    HTTP Server: Unsecure port (80): OK
    HTTP Server: Secure port (443): OK

Connection from master to replica is OK.

Connection check OK
```

レプリカのインストールスクリプトは、テストを実行し、インストールされているレプリカファイルが現在のホスト名と一致することを確認します。一致しない場合には、このスクリプトで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。これは、ホスト名が一致しなくても問題とならないマルチホームマシンで発生する可能性があります。

レプリカのインストールスクリプトの他のオプションについては、ipa-replica-install(1) man ページに一覧表示されます。

注記

ipa-replica-install で使用できるオプションの 1 つに **--ip-address** オプションがあります。**ipa-replica-install** に追加すると、このオプションは、ローカルインターフェースに関連付けられた IP アドレスだけを許可します。

4. プロンプトが表示されたら、Directory Manager のパスワードを入力します。次に、スクリプトはレプリカ情報ファイルの情報に基づいて Directory Server インスタンスを設定し、複製プロセスを開始し、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。このプロセスは、**初期化** と呼ばれます。

5. IdM クライアントが新しいサーバーを検出できるように、適切な DNS エントリーが作成されていることを確認します。必須のドメインサービスには、DNS エントリーが必要です。

    - _ldap._tcp
    - _kerberos._tcp

- _kerberos._udp
- _kerberos-master._tcp
- _kerberos-master._udp
- _ntp._udp

DNS が有効な状態で最初のサーバーを作成した場合には、適切な DNS エントリーでレプリカが作成されます。以下に例を示します。

```
[root@ipareplica ~]# DOMAIN=example.com
[root@ipareplica ~]# NAMESERVER=ipareplica
[root@ipareplica ~]# for i in _ldap._tcp _kerberos._tcp _kerberos._udp _kerberos-
master._tcp _kerberos-master._udp _ntp._udp; do echo ""; dig @${NAMESERVER}
${i}.${DOMAIN} srv +nocmd +noquestion +nocomments +nostats +noaa +noadditional
+noauthority; done | egrep -v "^;" | egrep _

_ldap._tcp.example.com. 86400   IN      SRV     0 100 389 ipaserver1.example.com.
_ldap._tcp.example.com. 86400   IN      SRV     0 100 389 ipaserver2.example.com.
_kerberos._tcp.example.com. 86400 IN    SRV     0 100 88  ipaserver1.example.com.
...8<...
```

DNS を有効にせずに最初の IdM サーバーを作成した場合には、サービスの TCP および UDP エントリー両方など、各 DNS エントリーは手作業で追加してください。以下に例を示します。

```
[root@ipareplica ~]# kinit admin
[root@ipareplica ~]# ipa dnsrecord-add example.com _ldap._tcp --srv-rec="0 100 389
ipareplica.example.com."
```

6. **任意。** レプリカの DNS サービスを設定します。マスターサーバーが DNS を使用している場合でも、レプリカの DNS サービスは、設定スクリプトでは設定されません。

   ipa-dns-install コマンドを使用して手動で DNS をインストールし、**ipa dnsrecord-add** コマンドで必要な DNS レコードを追加します。たとえば、以下のようになります。

   ```
   [root@ipareplica ~]# ipa-dns-install

   [root@ipareplica ~]# ipa dnsrecord-add example.com @ --ns-rec ipareplica.example.com.
   ```

重要

最後のピリオド (.) を含めてレプリカの完全修飾ドメイン名を使用します。このピリオドを含めない場合には、BIND はホスト名をドメインの相対値として扱います。

4.5. 他のレプリカ作成オプション

レプリカのコア設定の多くは、レルム名やディレクトリー設定など、作成元のサーバー設定と同じです。ただし、設定は同じでなければなりませんが、レプリカでサーバーと同じサービスを管理する必要はありません。これは、主要なサービス (DNS および CA) およびマイナーサービス (NTP および OpenSSH) が該当します。

異なる設定は、**ipa-replica-prepare** コマンドまたは **ipa-replica-install** コマンドで定義できます。

4.5.1. 各種 DNS 設定

DNS は、**ipa-replica-prepare** コマンドを使用して、レプリカ固有の DNS 設定 (IP アドレスと逆引きゾーン) を設定できます。たとえば、以下のようになります。

```
[root@server ~]# ipa-replica-prepare ipareplica.example.com --ip-address=192.68.0.0 --no-reverse
```

サーバーでどの DNS サービスもホストしない場合には、レプリカを設定して、Identity Management ドメインの DNS サービスをホストすることができます。サーバーをインストールする場合と同様に、**--setup-dns** オプションを使用して、正引きゾーンと逆引きゾーンを設定します。たとえば、フォワーダーなしで、既存の逆引きゾーンを使用して、レプリカの DNS サービスを設定するには以下を行います。

```
[root@server ~]# ipa-replica-install  ipareplica.example.com --setup-dns --no-forwarders --no-reverse
--no-host-dns ...
```

DNS オプションは **ipa-replica-prepare** および **ipa-replica-install** の man ページで説明されています。

4.5.2. 各種 CA 設定

レプリカの CA 設定は、サーバーの CA 設定をコピーする必要があります。サーバーが統合 Dogtag Certificate System インスタンスで設定されている場合には (ルート CA か、外部 CA の下位にある認証局であるかに拘らず)、レプリカではサーバー CA に従属する独自の統合 CA を作成するか、**または** CA を全く指定せずに、すべての要求をサーバーの CA に転送できます。

レプリカに独自の CA がある場合は、この **--setup-ca** オプションを使用します。残りの設定は、サーバーの設定から取得します。

```
[root@ipareplica ~]# ipa-replica-install ipareplica.example.com --setup-ca ...
```

ただし、サーバーが CA なしでインストールされている場合は、新規レプリカインスタンスの証明書を要求する機能など、証明書の操作の転送先がありません。サーバーと同様に、レプリカのインストール前に、レプリカの全証明書の要求および取得を行ってから、証明書をインストールコマンドで送信します。唯一の例外はルート CA 証明書で、これはレプリカ設定の一部としてサーバーから取得します。

```
[root@ipareplica ~]# ipa-replica-install ipareplica.example.com --dirsrv_pkcs12=/tmp/dirsrv-cert.p12 --
dirsrv_pin=secret1 --http_pkcs12=/tmp/http-cert.p12 --http_pin=secret2 ...
```

4.5.3. さまざまなサービス

デフォルトでサーバーおよりレプリカ両方にインストールされるサポート対象のサービスが 3 つあります (NTP、OpenSS クライアントおよび OpenSSH サーバー)。これらすべてまたは一部をレプリカで無効にすることができます。以下に例を示します。

```
[root@server ~]# ipa-replica-install ... --no-ntp --no-ssh --no-sshd ...
```

第5章 IDM クライアントとしてのシステムの設定

クライアント は、Identity Management ドメインに所属するシステムです。多くの場合、クライアントには Red Hat Enterprise Linux システム (IdM には Red Hat Enterprise Linux クライアントを非常にシンプルに設定する特別なツールがあります) を使用しますが、他のオペレーティングシステムを使用するマシンも IdM ドメインに追加できます。

IdM クライアントの重要な仕組みの 1 つとして、システムがドメインの一部であるかどうかを判断できるのは、システム設定 **のみ** である点が挙げられます。(この設定には Kerberos ドメイン、DNS ドメインに所属する設定や、適切な認証および証明書の設定が含まれます。)

注記

IdM では、クライアントがドメインに参加するために、クライアント上でエージェントやデーモンを実行する必要はありません。ただし、最適な管理オプション、セキュリティー、およびパフォーマンスを実現するには、クライアントで System Security Services Daemon (SSSD) を実行する必要があります。

SSSD の詳細は、SSSD プロジェクトページ にある 『デプロイメントガイド』の「SSSD」の章 を参照してください。

この章では、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

クライアントは、少なくとも 1 つの IdM サーバーがインストールされていないと設定できません。

5.1. クライアント設定

Red Hat Enterprise Linux システムでのクライアントの設定がクライアント設定スクリプトを使用する場合でも、手動で行った場合でも、マシンを IdM クライアントに指定する一般的な設定プロセスはほぼ同じですが、プラットフォームにより若干の違いがあります。

- IdM CA の CA 証明書を取得します。
- 別の Kerberos 設定を作成して、指定した認証情報をテストします。

  この設定により、IdM クライアントを IdM ドメインに参加させるのに必要な IdM XML-RPC サーバーへの Kerberos 接続が可能になります。この Kerberos 設定は最終的に破棄されます。

  Kerberos 設定では、レルムおよびドメイン情報、デフォルトのチケット属性を指定します。デフォルトでは、オペレーティングシステムから管理インターフェースへの接続を容易に行い、管理操作の監査ができるように転送可能なチケットが設定されています。たとえば、Red Hat Enterprise Linux システムの Kerberos 設定は以下のようになります。

```
[libdefaults]
default_realm = EXAMPLE.COM
dns_lookup_realm = false
dns_lookup_kdc = false
rdns = false
forwardable = yes
ticket_lifetime = 24h
```

```
[realms]
EXAMPLE.COM = {
    kdc = server.example.com:88
    admin_server = server.example.com:749
    }
[domain_realm]
.example.com = EXAMPLE.COM
example.com = EXAMPLE.COM
```

- **ipa-join** コマンドを実行し、実際の参加させます。
- ホストサービスのサービスプリンシパルを取得して、**/etc/krb5.keytab** にインストールします。例: **host/ipa.example.com@EXAMPLE.COM**
- certmonger を有効にし、SSL サーバー証明書を取得し、**/etc/pki/nssdb** に証明書をインストールします。
- nscd デーモンを無効にします。
- NSS および PAM 設定ファイルなど、SSSD または LDAP/KRB5 を設定します。
- OpenSSH サーバーおよびクライアントを設定し、ホストが DNS SSHFP レコードを作成できるようにします。
- NTP の設定

5.2. システムポート

IdM はサービスとの通信に多くのポートを使用します。IdM クライアントには、IdM サーバーと同じポート (ポート 7389 以外) が必要です。通常のデプロイメントでは大抵の場合、ポート 7389 を開放して利用可能な状態にする必要はありません。

IdM で必要なポートの一覧とそのポートを利用できる状態にする方法については、「システムポート」を参照してください。

5.3. IDM クライアントとしての LINUX システムの設定

Red Hat Enterprise Linux クライアントのクライアント設定プロセスを開始する前に、準備する要素が 2 つあります。

- Kerberos ID (管理ユーザー) を利用できるようにするか、クライアントマシンの登録プロセスを開始する前に、ワンタイムパスワードを使用してクライアントマシンをサーバーのクライアントマシンに手動で追加し、クライアントマシンを Kerberos ドメインに接続する手段を設定する必要があります。
- DNS レコードにサービスを提供するネットワーク上に Active Directory サーバーがある場合には、Active Directory の DNS レコードが原因で、クライアントによる IdM サーバーアドレスの自動検出ができなくなる可能性があります。**ipa-client-install** スクリプトでは、IdM に追加されたレコードの代わりに Active Directory の DNS レコードを取得します。

  この場合は、IdM サーバーアドレスを **ipa-client-install** スクリプトに直接指定する必要があります。

5.3.1. クライアントのインストール (完全な例)

1. クライアントパッケージをインストールします。このパッケージは、システムをクライアントとして簡単に設定でき、SSSD のインストールや設定も行います。

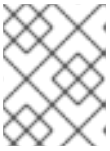
   通常のユーザーシステムの場合には、**ipa-client** パッケージのみが必要です。

   ```
   [root@client ~]# yum install ipa-client
   ```

   管理者マシンには、**ipa-admintools** パッケージも必要です。

   ```
   [root@client ~]# yum install ipa-client ipa-admintools
   ```

2. IdM サーバーを DNS サーバーとして設定し、クライアントと同じドメインに配置した場合には、クライアントの **/etc/resolv.conf** ファイルにあるネームサーバー一覧の最初のエントリーとして、サーバーの IP アドレスを追加します。

   ヒント

   ドメイン内のマシンがすべて IdM クライアントである場合は、IdM サーバーのアドレスを DHCP 設定に追加します。

3. クライアントの設定コマンドを実行します。

   ```
   [root@client ~]# ipa-client-install --enable-dns-updates
   ```

   --enable-dns-updates オプションを使用すると、クライアントマシンの IP アドレスに DNS が更新されます。このオプションは、IdM サーバーが統合 DNS でインストールされている場合か、ネットワーク上の DNS サーバーで GSS-TSIG プロトコルを使用して DNS エントリーを更新できる場合にのみ、使用するようにしてください。

   ipa-client-install のオプションは、**ipa-client-install** の man ページに一覧表示されています。

4. プロンプトが表示されたら、IdM DNS ドメインのドメイン名を入力します。

   ```
   DNS discovery failed to determine your DNS domain
   Please provide the domain name of your IPA server (ex: example.com): example.com
   ```

5. プロンプトが表示されたら、IdM サーバーの完全修飾ドメイン名を入力します。または、クライアントのインストールスクリプトに **--server** オプションを使用して、IdM サーバーの完全修飾ドメイン名を指定します。

   ```
   DNS discovery failed to find the IPA Server
   Please provide your IPA server name (ex: ipa.example.com): server.example.com
   ```

   重要

   これは、数字、アルファベット文字、およびハイフン (-) のみが使用された有効な DNS 名でなければなりません。ホスト名にアンダースコアなどの他の文字が含まれていると、DNS が正常に機能しなくなります。

6. 次にクライアントのスクリプトは、Kerberos ID を入力するようにプロンプトを表示し、その ID を使用して問い合わせを行い、Kerberos レルムに参加します。これらの認証情報を指定すると、クライアントは IdM Kerberos ドメインに参加して設定を完了できます。

```
Continue to configure the system with these values? [no]: y
User authorized to enroll computers: admin
Synchronizing time with KDC...
Password for admin@EXAMPLE.COM:
Successfully retrieved CA cert
Subject: CN=Certificate Authority,O=EXAMPLE.COM
Issuer: CN=Certificate Authority,O=EXAMPLE.COM
Valid From: Tue Aug 13 09:29:07 2013 UTC
Valid Until: Sat Aug 13 09:29:07 2033 UTC
Enrolled in IPA realm EXAMPLE.COM
Created /etc/ipa/default.conf
New SSSD config will be created
Configured /etc/sssd/sssd.conf
Configured /etc/krb5.conf for IPA realm EXAMPLE.COM
Failed to update DNS records.
Adding SSH public key from /etc/ssh/ssh_host_rsa_key.pub
Adding SSH public key from /etc/ssh/ssh_host_dsa_key.pub
Could not update DNS SSHFP records.
SSSD enabled
Configured /etc/openldap/ldap.conf
NTP enabled
Configured /etc/ssh/ssh_config
Configured /etc/ssh/sshd_config
Client configuration complete.
```

7. クライアントが IdM ドメインに正常に接続でき、基本的なタスクを実行できることをテストします。たとえば、IdM ツールを使用して、ユーザーおよびグループ情報を取得できることを確認します。

```
[jsmith@client ~]$ id
[jsmith@client ~]$ getent passwd admin
[jsmith@client ~]$ getent group admins
```

8. NFS サーバーがすでに設定されている場合は、クライアントシステムに NFS を設定して Kerberos と連携させます。

   NFS サーバーは、ドメイン内に設定しておく必要があります。詳細は、「自動マウントの設定」を参照してください。

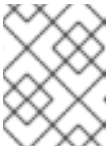
ヒント

NFS の設定時に発生する可能性のあるエラーをトラブルシューティングできるように、**/etc/sysconfig/nfs** ファイルでデバッグ情報を有効にします。

```
RPCGSSDARGS="-vvv"
RPCSVCGSSDARGS="-vvv"
```

   a. IdM サーバーで、NFS クライアントの NFS サービスプリンシパルを追加します。

```
[root@client ~]# kinit admin
[root@client ~]# ipa service-add nfs/ipaclient.example.com@EXAMPLE
```



注記

これは、**ipa** コマンドを使用できるように、ipa-admintools パッケージがインストールされているマシンから実行する必要があります。

b. IdM サーバーで、NFS サービスプリンシパルの keytab を取得します。

```
[root@client ~]# ipa-getkeytab -s server.example.com -p
nfs/ipaclient.example.com@EXAMPLE -k /tmp/krb5.keytab
```

c. IdM サーバーから IdM クライアントに keytab をコピーします。たとえば、以下のようになります。

```
[root@client ~]# scp /tmp/krb5.keytab root@client.example.com:/etc/krb5.keytab
```

d. NFS サーバーで **/etc/exports** ファイルを設定します。

```
/ipashare       gss/krb5p(rw,no_root_squash,subtree_check,fsid=0)
```

e. マウントポイントを作成します。

```
[root@client ~]# mkdir /mnt/ipashare
```

f. クライアントで NFS 共有をマウントします。**-o sec** 設定は、NFS サーバーの **/etc/exports** ファイルで使用する設定と同じものを使用します。

```
[root@client ~]# mount -v -t nfs4 -o sec=krb5p nfs.example.com:/ /mnt/ipashare
```

5.3.2. その他のクライアントインストールオプションの例

ipa-client-install コマンドには、インフラストラクチャーの要件に応じて、さまざまな方法でのクライアントシステムの設定に使用できる設定オプションが複数あります。

例5.1 DNS 更新の有効化

DHCP 設定によっては、クライアントの IP アドレスは、一定の規則をもとに変更できます。IP アドレスを変更すると、IdM サーバーの DNS レコードと、実際に使用中の IP アドレスとの間に差異が発生して、IdM 内で設定されたポリシーやクライアントとサービス間の通信に影響が及ぶ可能性があります。

--enable-dns-updates オプションでは、クライアントの IP アドレスが変更されるたびに DNS エントリーを更新する System Security Services Daemon (SSSD) を設定します。

```
[root@client ~]# ipa-client-install --enable-dns-updates
```

例5.2 ドメイン情報の指定

クライアントインストールコマンドだけを実行すると、スクリプトで、登録する IdM サーバーの名前、DNS ドメイン名、Kerberos レルムおよびプリンシパルなど、必要な IdM ドメイン名が求められます。

上記の基本情報はすべて、インストールコマンドで指定できます (このインストールコマンドは自動インストールに便利です)。

- **--domain**: DNS ドメイン名 (DNS サービスをホストするように IdM サーバーが設定されている場合のみ)
- **--server**: 登録する IdM サーバー (トポロジー内の任意のサーバーまたはレプリカ)

  これは、数字、アルファベット文字、およびハイフン (-) のみが使用された有効な DNS 名でなければなりません。ホスト名にアンダースコアなどの他の文字が含まれていると、DNS が正常に機能しなくなります。
- **--realm**: Kerbero レルム名。オプションで Kerberos プリンシパル名には **-p** を使用します。

```
[root@client ~]# ipa-client-install --domain EXAMPLE.COM --server server.example.com --realm EXAMPLE -p host/server.example.com
```

例5.3 特定の IdM サーバーの設定

IdM サーバートポロジーには、複数のサーバーおよびレプリカが存在する可能性があります。更新やユーザー情報の取得に、クライアントがサーバーに接続する必要がある場合には、(デフォルトでは) サービスを使用してドメイン内をスキャンして利用可能なサーバーとレプリカを検出します。つまり、検出スキャンの結果に応じて、クライアントが実際に接続する先のサーバーは無作為に決定されることになります。

クライアント更新に使用する IdM ドメイン内に、特定のサーバーを設定できます。何らかの理由で、そのサーバーへの接続に失敗した場合には、クライアントはドメイン内の別のサーバーを検出してフェイルオーバーできます。

優先サーバーは、**--fixed-primary** オプションで設定します。

```
[root@client ~]# ipa-client-install --fixed-primary server.example.com
```

例5.4 システム認証ツールの無効化

Red Hat Enterprise Linux は、**authconfig** ツールを使用してローカルシステムの認証クライアントおよびオプションを設定して更新します。Identity Management は System Security Services Daemon (SSSD) を使用して IdM サーバー設定を保存し、IdM ドメイン内で設定したポリシー情報、ユーザー、パスワード、およびグループを取得します。

authconfig および SSSD を使用してユーザー、グループなどの IdM クライアント設定を管理することを強く推奨します。

管理者がシステム認証設定の動的な変更を無効にする状況があります。このような場合は、IdM が **authconfig** または SSSD を更新しないように無効にできます。

--noac オプションは、**authconfig** での変更を防ぎます。**--no-sssd** オプションでは、IdM が SSSD を使用できないようにします。

```
[root@client ~]# ipa-client-install --noac --no-sssd
```

関連のオプションとして **--preserve-sssd** があります。このオプションでは、クライアントが SSSD 設定ファイルを変更して IdM ドメインを設定できますが、以前の SSSD 設定を保存します。

例5.5 パスワードキャッシングの無効化

SSSD の主な機能の 1 つとして **パスワードキャッシュがあります**。通常、システムが外部のパスワードストアを使用する場合は、そのパスワードストアにアクセスできなくなると認証に失敗します。ただし、SSSD では認証の試行に成功するとパスワードをキャッシュし、そのパスワードをローカルに保存することができます。これにより、IdM サーバーにアクセスできなくても、ユーザーはドメインサービス (以前はアクセスしたサービス) にログインしてアクセスできるようになります。

セキュリティーレベルの高い環境では、不正アクセスされないように、パスワードのキャッシュを防止する必要がある場合があります。このような場合には、**--no-krb5-offline-passwords** オプションを使用して、パスワードが SSSD でキャッシュできないようにします。

```
[root@client ~]# ipa-client-install --no-krb5-offline-passwords
```

5.4. LINUX クライアントの手動設定

ipa-client-install コマンドは、Kerberos、SSSD、PAM、NSS などのサービスを自動的に設定します。ただし、何らかの理由で **ipa-client-install** コマンドをシステムで使用できない場合は、IdM クライアントエントリーとサービスを手動で設定できます。

5.4.1. IdM クライアントの設定 (全手順)

1. SSSD がインストールされていない場合はインストールしてください。
2. **任意。** ホストから管理タスクを実行できるように IdM ツールをインストールします。

   ```
   [root@client ~]# yum install ipa-admintools
   ```

3. **IdM サーバーで、** クライアントのホストエントリーを作成します。

   ```
   [jsmith@client ~]$ kinit admin
   [jsmith@client ~]$ ipa host-add --force --ip-address=192.168.166.31 ipaclient.example.com
   ```

   ホストを手動で作成する方法は、「ホストエントリーを追加する他の例」を参照してください。

4. **IdM サーバーで、** クライアントの Keytab を作成します。

   a. IdM 管理者としてログインします。

      ```
      [jsmith@client ~]$ kinit admin
      ```

   b. サーバーが管理するクライアントホストを設定します。

      ```
      [jsmith@client ~]$ ipa host-add-managedby --hosts=server.example.com
      ipaclient.example.com
      ```

c. クライアントの keytab を生成します。

```
[jsmith@client ~]$ ipa-getkeytab -s server.example.com -p host/ipaclient.example.com -k
/tmp/ipaclient.keytab
```

5. Keytab をクライアントマシンにコピーし、名前を **/etc/krb5.keytab** に変更します。

ヒント

既存の **/etc/krb5.keytab** を保存する必要がある場合には、**ktutil** を使用してこの 2 つのファイルを統合できます。

6. **/etc/krb5.keytab** ファイルのユーザーパーミッションを正しく設定します。

```
[root@client ~]# chown root:root /etc/krb5.keytab
[root@client ~]# chmod 0600 /etc/krb5.keytab
```

7. **/etc/krb5.keytab** ファイルの SELinux コンテキストを設定します。

```
[root@client ~]# chcon system_u:object_r:krb5_keytab_t:s0 /etc/krb5.keytab
```

8. **/etc/sssd/sssd.conf** ファイルを編集して、SSSD が IdM ドメインを参照するように設定します。

```
[root@client ~]# touch /etc/sssd/sssd.conf
[root@client ~]# vim /etc/sssd/sssd.conf

[sssd]
config_file_version = 2
services = nss, pam

domains = example.com
[nss]

[pam]

[domain/example.com]
cache_credentials = True
krb5_store_password_if_offline = True
ipa_domain = example.com
id_provider = ipa
auth_provider = ipa
access_provider = ipa
ipa_hostname = ipaclient.example.com
chpass_provider = ipa
ipa_server = server.example.com
ldap_tls_cacert = /etc/ipa/ca.crt
```

9. パスワード、グループ、ユーザー、および netgroups に SSSD を使用するように NSS を設定します。

```
[root@client ~]# vim /etc/nsswitch.conf
```

```
...
passwd:     files sss
shadow:     files sss
group:      files sss
...
netgroup:   files sss
...
```

10. **/etc/krb5.conf** ファイルで、IdM KDC を参照するように設定します。

```
[logging]
 default = FILE:/var/log/krb5libs.log
 kdc = FILE:/var/log/krb5kdc.log
 admin_server = FILE:/var/log/kadmind.log

[libdefaults]
 default_realm = EXAMPLE.COM
 dns_lookup_realm = false
 dns_lookup_kdc = false
 rdns = false
 ticket_lifetime = 24h
 forwardable = yes
 allow_weak_crypto = true

[realms]
 EXAMPLE.COM = {
  kdc = server.example.com:88
  admin_server = server.example.com:749
  default_domain = example.com
}

[domain_realm]
 .example.com = EXAMPLE.COM
 example.com = EXAMPLE.COM
```

11. **pam_sss.so** モジュールを使用するように、**/etc/pam.d** 設定を更新します。

- **/etc/pam.d/fingerprint-auth** の場合:

```
...
account     [default=bad success=ok user_unknown=ignore] pam_sss.so
...
session     optional      pam_sss.so
```

- **/etc/pam.d/system-auth** の場合:

```
...
auth        sufficient    pam_sss.so use_first_pass
...
account     [default=bad success=ok user_unknown=ignore] pam_sss.so
...
password    sufficient    pam_sss.so use_authtok
...
session     optional      pam_sss.so
```

- **/etc/pam.d/password-auth** の場合:

```
...
auth        sufficient    pam_sss.so use_first_pass
...
account     [default=bad success=ok user_unknown=ignore] pam_sss.so
...
password    sufficient    pam_sss.so use_authtok
...
session     optional      pam_sss.so
```

- Enrollment_with_Separation_of_Duties **/etc/pam.d/smartcard-auth** の場合:

```
...
account     [default=bad success=ok user_unknown=ignore] pam_sss.so
...
session     optional      pam_sss.so
```

12. IdM サーバーの CA 証明書をインストールします。

    a. サーバーから証明書を取得します。

    ```
    [root@ipaclient ~]# wget -O /etc/ipa/ca.crt http://ipa.example.com/ipa/config/ca.crt
    ```

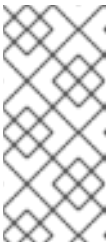
    b. システムの NSS データベースに証明書をインストールします。

    ```
    [root@ipaclient ~]# certutil -A -d /etc/pki/nssdb -n "IPA CA" -t CT,C,C -a -i /etc/ipa/ca.crt
    ```

13. IdM にホストのホスト証明書を設定します。

    a. **certmonger** が実行されていることを確認します。

    ```
    [root@ipaclient ~]# service certmonger start
    ```

    ヒント

    certmonger サービスがデフォルトで起動するように、**chkconfig** を設定します。

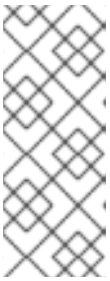
    ```
    [root@ipaclient ~]# chkconfig certmonger on
    ```

    b. **certmonger** を使用し、**ipa-getcert** コマンドで証明書を作成し、管理します。オプションの詳細は、「certmonger で証明書の要求」を参照してください。

    ```
    [root@ipaclient ~]# ipa-getcert request -d /etc/pki/nssdb -n Server-Cert -K
    HOST/ipaclient.example.com -N 'CN=ipaclient.example.com,O=EXAMPLE.COM'
    ```

    クライアントに管理ツールがインストールされていない場合は、IdM サーバーで証明書を生成し、ホストにコピーして、**certutil** を使用してインストールできます。

14. Kerberos と連携するように NFS を設定します。



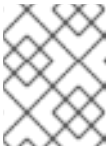
ヒント

NFS の設定時に発生する可能性のあるエラーをトラブルシューティングできるように、**/etc/sysconfig/nfs** ファイルでデバッグ情報を有効にします。

```
RPCGSSDARGS="-vvv"
RPCSVCGSSDARGS="-vvv"
```

a. IdM サーバーで、NFS クライアントの NFS サービスプリンシパルを追加します。

```
[root@ipaclient ~]# ipa service-add nfs/ipaclient.example.com@EXAMPLE
```



注記

これは、**ipa** コマンドを使用できるように、ipa-admintools パッケージがインストールされているマシンから実行する必要があります。

b. IdM サーバーで、NFS サービスプリンシパルの keytab を取得します。

```
[root@ipaclient ~]# ipa-getkeytab -s server.example.com -p
nfs/ipaclient.example.com@EXAMPLE -k /tmp/krb5.keytab
```

注記

Linux の NFS 実装バージョンによっては、暗号化タイプのサポートが限定されます。Red Hat Enterprise Linux 6 よりも前のバージョンで NFS サーバーをホストしている場合は、サーバーおよびすべてのクライアントの両方で、任意の nfs/<FQDN> サービスキータブをサーバーと全クライアント両方で設定するように、**-e des-cbc-crc** オプションを指定して **ipa-getkeytab** コマンドを実行します。これにより、KDC で DES キーのみが生成されるように指示します。

DES キーを使用する場合、この暗号化タイプに依存するクライアントおよびサーバーではすべて、**/etc/krb5.conf** ファイルの **[libdefaults]** セクションで **allow_weak_crypto** オプションを有効にする必要があります。これらの設定変更を行わない場合には、NFS クライアントとサーバーは相互に認証できず、NFS ファイルシステムのマウントに失敗する可能性があります。クライアントの **rpc.gssd** とサーバーの **rpc.svcgssd** デーモンは、DES の暗号化タイプが許可されていないことを示すエラーをログに記録する場合があります。

c. IdM サーバーから NFS サーバーにキータブをコピーします。たとえば、IdM サーバーと NFS サーバーが異なるマシンにある場合は、以下を実行します。

```
[root@ipaclient ~]# scp /tmp/krb5.keytab root@nfs.example.com:/etc/krb5.keytab
```

d. IdM サーバーから IdM クライアントにキータブをコピーします。たとえば、以下のようになります。

```
[root@ipaclient ~]# scp /tmp/krb5.keytab root@client.example.com:/etc/krb5.keytab
```

e. NFS サーバーで **/etc/exports** ファイルを設定します。

```
/ipashare       gss/krb5p(rw,no_root_squash,subtree_check,fsid=0)
```

f. クライアントで NFS 共有をマウントします。

- 共有は必ず、**nfs_server:/ /mountpoint** と指定します。
- **-o sec** 設定は、NFS サーバーの **/etc/exports** ファイルで使用する設定と同じものを使用します。

```
[root@client ~]# mount -v -t nfs4 -o sec=krb5p nfs.example.com:/ /mnt/ipashare
```

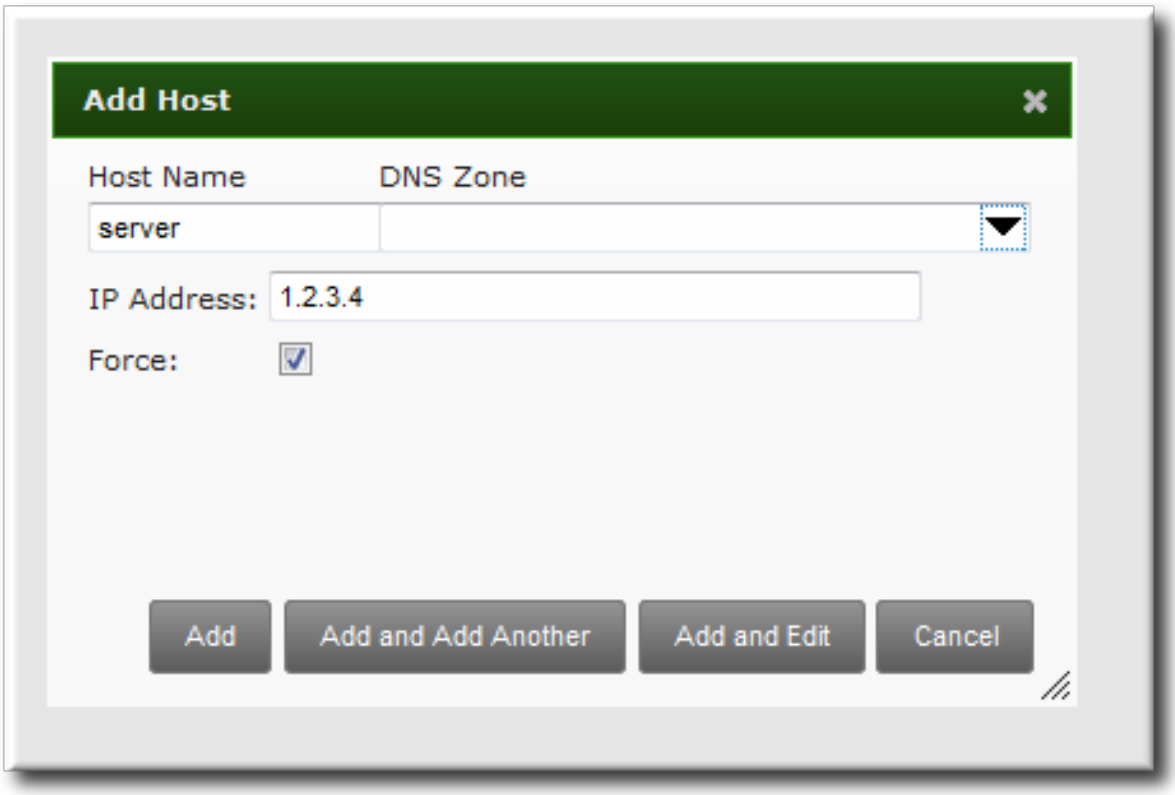
5.4.2. ホストエントリーを追加する他の例

「IdM クライアントの設定 (全手順)」では、IdM クライアントを手動で設定する全手順を説明します。上記の手順の 1 つとして、ホストエントリーの作成が含まれます。ホストの作成の方法及びオプションは複数あります。

5.4.2.1. 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. ホスト一覧の上部にある **Add** をクリックします。

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。



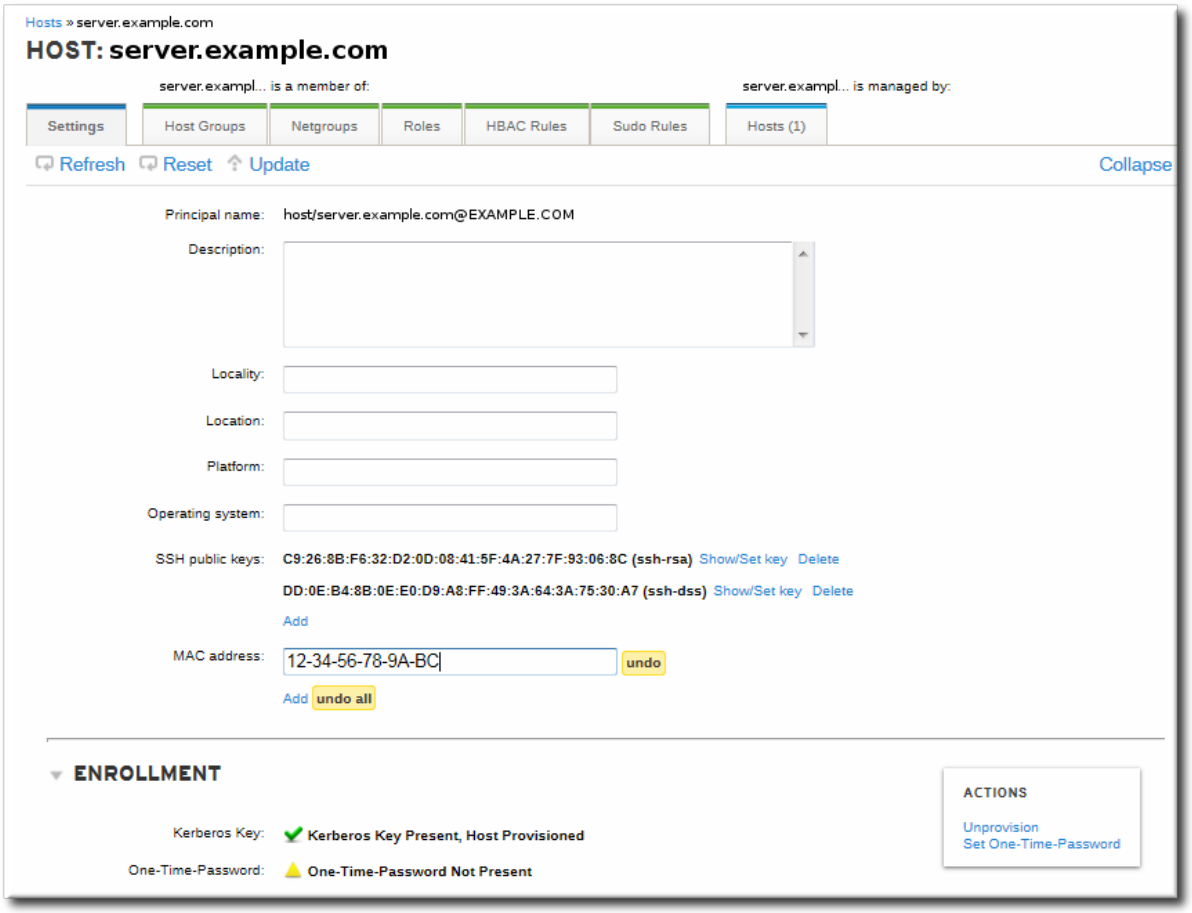
「正引き DNS ゾーンの追加」で説明されているように、DNS ゾーンは IdM で作成可能です。IdM サーバーが DNS サーバーを管理しない場合は、通常のテキストフィールドなど、メニューエリアでゾーンを手動で入力できます。

注記

ホスト名を解決できない場合でも、**Force** チェックボックスを選択して、ホストの DNS レコードを追加します。

これは、DHCP を使用して、静的な IP アドレスがないホストに役に立ちます。これにより、IdM DNS サービスにプレースホルダーエントリーが作成されます。DNS サービスが動的にレコードを更新すると、ホストの現行の IP アドレスが削除され、DNS レコードが更新されます。

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。



5.4.2.2. 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** の全オプション一覧は、**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5.6 静的 IP アドレスを持つ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5.7 DHCP で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
```

5.5. キックスタートでの LINUX クライアントの設定

キックスタートで登録すると、プロビジョニング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。

これには、パスワードを事前定義して、IdM サーバーにホストを予め作成しておく必要があります。このパスワードを使用して、認証を行い、登録操作を完了できます。

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

   ipa-client-install スクリプトが (対話的に) 通常通りに実行されると、IdM ドメインにアクセスするための認証情報の入力が求められます。ただし、スクリプトが自動的に実行される場合には、既存の IdM ユーザーを使用せずに IdM ドメインにアクセスする方法が必要になります。これは、スクリプトでホストプリンシパルを設定し、IdM ドメインへのアクセス用の Kerberos パスワード (ホストアカウントに設定) を使用して実行します。

   以下に例を示します。

   ```
   [jsmith@server ~]$ ipa host-add kickstart-server.example.com --password=secret
   ```

   パスワードは、最初の認証試行後に有効期限が切れます。登録が完了すると、ホストはキータブを使用して認証されます。

2. 他のインストールと共に ipa-client パッケージも追加します。

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

3. 登録前に SSH 鍵が生成されるようにするインストール後の指示を作成し、**ipa-client-install** スクリプトを実行して IdM ドメインサービスへのアクセスおよび設定に必要なすべての情報を渡し、事前設定されたパスワードを指定します。この **--unattended** オプションを使用して、スクリプトが非対話的に実行されるように指示します。

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

   # Generate SSH keys to ensure that ipa-client-install uploads them to the IdM server
   /usr/bin/ssh-keygen -q -t rsa -f /etc/ssh/ssh_host_rsa_key -C '' -N ''
   chmod 600 /etc/ssh/ssh_host_rsa_key
   chmod 644 /etc/ssh/ssh_host_rsa_key.pub
   ```

```
/sbin/restorecon /etc/ssh/ssh_host_rsa_key.pub

/usr/bin/ssh-keygen -q -t rsa1 -f /etc/ssh/ssh_host_key -C '' -N ''
chmod 600 /etc/ssh/ssh_host_key
chmod 644 /etc/ssh/ssh_host_key.pub
/sbin/restorecon /etc/ssh/ssh_host_key.pub

/usr/bin/ssh-keygen -q -t dsa -f /etc/ssh/ssh_host_dsa_key -C '' -N ''
chmod 600 /etc/ssh/ssh_host_dsa_key
chmod 644 /etc/ssh/ssh_host_dsa_key.pub
/sbin/restorecon /etc/ssh/ssh_host_dsa_key.pub

# Get the hostname to set as the host principal
/bin/hostname > /tmp/hostname.txt

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

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。

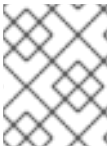
4. キックスタートスクリプトを実行します。

5.6. TWO-ADMINISTRATOR 登録の実行

IdM ドメインでマシンをクライアントとして登録する場合は、2 つのプロセスがあります。クライアントにホストエントリーが作成されて (389 Directory Server インスタンスに格納されて) から、クライアントをプロビジョニングするキータブが作成されます。

プロセスは、いずれも **ipa-client-install** コマンドで自動的に実行されます。この手順は個別に実行することもできます。これにより、管理者は、クライアントを実際に構成する前にマシンと IdM サーバー設定を準備できます。これにより、一括デプロイメントなど、より柔軟な設定シナリオが可能になります。

手動登録を実行すると、ホストエントリーが個別に作成され、クライアントスクリプトの実行時に登録が完了し、必要なキータブが作成されます。



注記

パスワードを設定する方法は 2 つあります。ご自身でパスワードを設定するか、IdM が無作為に生成できます。

グループの管理者によるホストエントリーの **作成** が禁止されている場合があるので、単に **ipa-client-install** コマンドを実行してホストを作成できない場合があります。ただし、管理者にはホストエントリーの作成 **後** にコマンドを実行する権限がある場合があります。このような場合には、管理者はホストエントリーを手動で作成し、2 番目の管理者が **ipa-client-install** コマンドを実行して登録を完了できます。

1. 管理者は、「ホストエントリーを追加する他の例」の説明に従って、ホストエントリーを作成します。

2. 2 つ目の管理者は、「IdM クライアントとしての Linux システムの設定」 の説明のように IdM クライアントパッケージをマシンにインストールします。

3. 2 つ目の管理者が設定スクリプトを実行すると、**ipa-client-install** コマンドで Kerberos パスワードとユーザー名 (プリンシパル) を指定する必要があります。たとえば、以下のようになります。

```
$ ipa-client-install -w secret -p admin2
```

4. キータブは、クライアントマシンが IdM ドメインに接続できないように、サーバーで生成されてクライアントマシンにプロビジョニングされます。このキータブは、所有者が **root:root**、パーミッションが 0600 として保存します。

5.7. クライアントマシンの手動による設定解除

マシンを IdM ドメインから削除して別のドメインに移動するか、または仮想マシンをコピーする必要がある場合があります。IdM の再設定が必要な各種状況が複数あります。最も簡単な解決策として、クライアントをアンインストールしてから最初から設定することです。クライアントをインストールする場合のように **--updatedns** オプションを使用して、ドメイン DNS 設定を自動的に更新します。

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

クライアントを直接アンインストールできない場合は、クライアントシステムから IdM 設定を手動で削除できます。

警告

マシンの登録解除後は、元に戻すことはできません。マシンをもう一度登録し直すことしかできません。

1. クライアントで、メインのキータブから以前のホスト名を削除します。これは、レルムのプリンシパルをすべて削除するか、特定のプリンシパルを削除して実行できます。たとえば、すべてのプリンシパルを削除するには、以下を実行します。

```
[jsmith@client ~]$ ipa-rmkeytab -k /etc/krb5.keytab -r EXAMPLE.COM
```

特定のプリンシパルを削除するには、以下を実行します。

```
[jsmith@client ~]$ ipa-rmkeytab -k /etc/krb5.keytab -p
host/server.example.com@EXAMPLE.COM
```

2. クライアントシステムで、全証明書の **certmonger** の追跡を無効にします。各証明書は、個別に追跡から削除する必要があります。

まず、追跡する全証明書を一覧表示し、各証明書のデータベースとニックネームを取り出します。証明書の数は、ホストに設定されたサービスにより異なります。

```
[jsmith@client ~]$ ipa-getcert list
```

次に、それぞれの追跡を無効にします。以下に例を示します。

```
[jsmith@client ~]$ ipa-getcert stop-tracking -n "Server-Cert" -d /etc/httpd/alias
```

3. IdM サーバーで、IdM DNS ドメインから以前ホストを削除します。これはオプションですが、システムに関連付けられた以前の IdM エントリーを消去して後で正しく登録できるようにします。

```
[jsmith@server ~]$ kinit admin
[jsmith@server ~]$ ipa host-del server.example.com
```

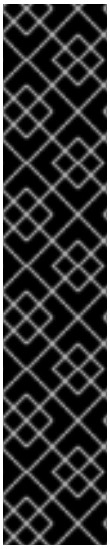
4. 別の場所に移動した仮想マシンなど、新しい IdM ドメインにシステムを追加し直す必要がある場合には、クライアントシステムで **ipa-join** コマンドを使用してシステムを再度参加させることができます。

```
[jsmith@client ~]$ ipa-join
```

第6章 IDENTITY MANAGEMENT のアップグレード

Identity Management は通常、システムが新規リリースにアップグレードされるたびに更新されます。アップグレードは透過的であるため、ユーザーや管理者の介入は必要ありません。

6.1. アップグレードの注意事項



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) および **TLSv1.1** に設定します。

```
NSSProtocol TLSv1.0,TLSv1.1
```

2. **httpd** サービスを再起動します。

```
# service httpd restart
```

- 更新プロセスでは、全スキーマおよび LDAP 設定、Apache 設定、およびその他のサービス設定が自動的に更新され、IdM 関連のサービスがすべて再起動されます。
- レプリカの作成時には、ベースとしたマスターと同じバージョンを使用する必要があります。つまり、サーバーのアップグレードプロセス時に、レプリカを以前の Identity Management バージョンで作成しないようにしてください。アップグレードプロセスが完了するまで待ってから、新しいレプリカを作成します。
- スキーマが変更されると、サーバー間で複製されます。したがって、マスターサーバー 1 台が更新されると、パッケージがまだ更新されていない場合でも、全サーバーおよびレプリカのスキーマが更新されます。これにより、新しいスキーマを使用する新規エントリーを、IdM ドメイン内にある他の全サーバーでそのまま複製できます。

  LDAP のアップグレード操作は、**/var/log/ipaupgrade-log** のアップグレードログに記録されます。LDAP エラーが発生した場合は、上記のログに記録されます。エラーが解決されると、updater スクリプトを実行して LDAP 更新プロセスを手動で開始できます。

```
[root@server ~]# ipa-ldap-updater --upgrade
```

- クライアントには、新しいパッケージをインストールする必要はありません。ドメインでのクライアント登録には、Red Hat Enterprise Linux システムの設定に使用するクライアントパッケージによる影響はありません。
- クライアントパッケージを更新すると、バグ修正を含む **certmonger** など、他の依存関係が更新される可能性がありますが、IdM ドメインでクライアントの機能や動作を維持するためには必要ありません。

6.2. パッケージのアップグレード

IdM サーバーパッケージは、システムパッケージの更新時に更新されます。

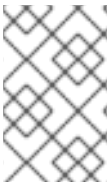
```
[root@ipaserver ~]# yum update
```

Identity Management 機能を提供する SSSD などの関連サービスの更新を自動的にプルするので、IdM サーバーパッケージを更新するのが最も簡単な方法です。

特に IdM サーバーパッケージをアップグレードするには、マスターサーバーで **yum** を実行します。

```
[root@ipaserver ~]# yum update ipa-server
```

更新プロセスで全変更を適用するには、数分かかる可能性があります。



注記

すべてのサーバーとレプリカを同じタイミングで更新する必要はありません。IdM サーバーは相互に連携し、データを正しく複製します。以前の IdM サーバーには、新機能が含まれていないだけです。

6.3. チケット委譲のブラウザー設定の削除 (6.2 からのアップグレード)

Kerberos 認証の設定の一環として、プリンシパルには **ticket granting ticket** (TGT) が割り当てられます。プリンシパルが Kerberos ドメイン内のサービスまたはアプリケーションに接続を試行するたびに、サービスはアクティブな TGT があるかを確認し、そのプリンシパルがサービスにアクセスするために TGT から独自のサービス固有のチケットを要求します。

以前の Identity Management バージョンでは、IdM Web UI (およびその他の Kerberos 対応 Web アプリケーション) へのアクセスに使用する Web ブラウザーを設定するには、TGT 委譲を IdM サーバーに転送する必要がありました。これには、**delegation-uris** パラメーターを Firefox の **about:config** 設定に追加する必要がありました。

```
network.negotiate-auth.delegation-uris .example.com
```

Red Hat Enterprise Linux 6.3 では、Identity Management はユーザー向けの Kerberos サービスをプロキシー (S4U2Proxy) に使用するため、この追加の委譲手順は必要ありません。

既存の設定済みブラウザーの更新

Identity Management の Web UI を使用するように設定されているブラウザーでは、 **delegation-uris** 設定は、**ipa-server-3.0.0** または **ipa-client-3.0.0** にアップグレードしてから消去できます。

delegation-uris 設定の変更後に、ブラウザーを再起動する必要はありません。

新規ブラウザー設定用の configure.jar の更新

ブラウザーの設定は **configure.jar** ファイルに定義されます。この JAR ファイルはサーバーのインストール時に生成され、IdM の更新時に他のファイルと一緒に更新されません。IdM サーバーをアップグレードしても、設定済みのブラウザーには不必要な **delegation-uris** パラメーターが設定されたままになります。ただし、**configure.jar** ファイルは更新できます。

configure.jar の **preferences.html** ファイルは、**delegation-uris** パラメーターを設定します。更新した **preferences.html** ファイルは **configure.jar** に追加してから、**configure.jar** を再署名し、IdM サーバーにデプロイし直すことができます。

注記

最初の IdM サーバーの **configure.jar** ファイルだけを更新します。これは、署名証明書が唯一含まれるマスターサーバーです。次に、更新したファイルを他のサーバーおよびレプリカに伝播します。

1. 最初の IdM マスターサーバー (最初のインスタンス) でパッケージを更新します。これにより、**configure.jar** ファイルを含む 3.0 UI パッケージが作成されます。
2. 既存の **configure.jar** ファイルをバックアップします。

```
[root@ipaserver ~]# mv /usr/share/ipa/html/configure.jar /usr/share/ipa/html/configure.jar.old
```

3. 一時作業ディレクトリーを作成します。

```
[root@ipaserver ~]# mkdir /tmp/sign
```

4. 更新した **preferences.html** ファイルを作業ディレクトリーにコピーします。

```
[root@ipaserver ~]# cp /usr/share/ipa/html/preferences.html /tmp/sign
```

5. **signtool** コマンド (NSS ユーティリティーの 1 つ) を使用して新しい **preferences.html** ファイルを追加し、**configure.jar** ファイルを再署名します。

```
[root@ipaserver ~]# signtool -d /etc/httpd/alias -k Signing-Cert -Z
/usr/share/ipa/html/configure.jar -e ".html" -p `cat /etc/httpd/alias/pwdfile.txt` /tmp/sign
```

   -e オプションは、ツールに対して、拡張子が **.html** のファイルのみに署名するように指示します。**-Z** オプションでは、新しい JAR ファイルを作成します。

6. 再生成された **configure.jar** ファイルを、他の全 IdM サーバーおよびレプリカにコピーします。

6.4. IDM サーバーのアップグレード前のテスト (推奨)

実稼働システムをアップグレードする前に、新しいバージョンの Identity Management をテストすると、有益でより安全です。適切なレプリカを作成し、そのシステムでテストすることで、比較的簡単な方法で実行できます。

1. 「*4章IdM レプリカの設定*」で説明されているように、実稼働サーバーのいずれかを基にレプリカを設定します。この例では、これは Test Replica という名前を使用しています。Test Replica が **実稼働** サーバーおよびドメインに正常に接続できることを確認します。
2. 実稼働ドメインに Test Replica が正常に追加されたことを確認したら、**ネットワークから Test Replica の接続を解除します。**
3. 元の IdM サーバーと Test Replica から、Test Replica のレプリカ合意を削除します。
4. Test Replica をネットワークに再接続します。
5. **yum** またはお使いのシステムに適したパッケージの更新ツールを使用して、Test Replica でパッケージをアップグレードします。たとえば、以下のようになります。

```
[root@ipareplica ~]# yum update ipa*
```

6. Kerberos 認証情報の取得、サーバー UI の表示、コマンドの実行など、Test Replica で一般的な内容をテストします。

第7章 IDM サーバーおよびレプリカのアンインストール

IdM サーバーと IdM レプリカの両方をアンインストールするには、**--uninstall** オプションを **ipa-server-install** コマンドに指定します。

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

第8章 IDM サーバーおよびサービスの基本的な管理

Web UI とコマンドラインを使用して Identity Management にアクセスするにはユーザーは必ず、IdM ドメインに対して認証を行います。本章では、Kerberos 認証の処理、Identity Management へのログイン、および一般的な接続の問題のトラブルシューティングを行うための基本的なブラウザーの設定について説明します。

8.1. IDM ドメインの起動と停止

IdM サーバーのインストール時には、Directory Server、認証局、Web サーバー、DNS、NTP、certmonger、および Kerberos (これに限定されない) など、任意の組み合わせでインストールおよび設定できる複数の異なるサービスがあります。

これらのサーバーはすべて、連携して動作します。サービスには依存関係があるため、サービスの起動と停止の順序は重要です。

(LDAP ディレクトリーや Web サーバーなど) 1 つのサービスに変更を加えると、**service** コマンドを使用してサービスを個別に開始および停止できます。ただし、複数のドメインサービス (または IdM サーバー全体) を再起動する必要がある場合は、**ipactl** コマンドを使用すると、適切な順番にサービスが開始および停止されます。

特定の IdM サーバーにどのサービスを設定するかは、IdM サーバーのホスト名を基に 389 Directory Server 設定で定義します。[1] 389 Directory Server サービスは、最初に開始し、最後に停止してください。残りの実行順序は、設定されているサービスにより異なります。

ipactl コマンドで、サービスの起動、停止、再起動が可能です。

```
ipactl start | stop | restart
```

chkconfig コマンドは、システムの再起動時に自動的に起動するサービスを設定します。**ipactl** コマンドを使用すると、**chkconfig** の実行順に個別に設定する必要なく、適切な順序で起動できます。

```
[root@server ~]# chkconfig ipactl on
```

8.2. IDM クライアントツールの概要

IdM は、汎用的に適用される認証ソースおよび共通のポリシーを使用して認識済みのサービス、ホストマシン、ユーザーのドメインを作成します。クライアントマシンと IdM ユーザーの視点からすると、初期設定が済むとドメイン自体は非常に透過的になります。ユーザーはすべて Kerberos を使用してドメインにログインするだけです。

ただし、管理者は継続して、IdM の Kerberos ドメインにプリンシパルを追加するタスク、ドメインの対話と統制するドメインポリシーとサーバー設定を設定するタスクの 2 つのタスクを実行する必要があります。Identity Management には、管理者がドメイン、サービス、および IdM エントリーの管理に使用するコマンドラインおよび Web ベースのインターフェースの両方があります。

コマンドラインツールを使用するのがドメイン管理の最も一般的な方法です。Identity Management には、管理者が利用できる幅広いスクリプトとコマンドのセットがあります。ドメインのエントリー管理機能は、1 つのスクリプト (**ipa**) で実行されます 。このスクリプトは、関連付けられたサブコマンドの親または制御スクリプトです。各サブコマンドは、特定のエントリータイプに関連します。

コマンドラインスクリプトには、以下のような複数の利点があります。

- スクリプトを使用すると、手動による介入なしに一貫した方法で管理タスクを自動化して実行できます。
- エントリーには、1回の手順で設定可能な属性 (または任意の属性のサブセット) を追加できます。Web UI では、エントリーを完全に設定するには、最初にエントリーを作成して次にオプション属性を追加するという 2 つの手順が必要になります。
- コマンドラインスクリプトでは、別の属性の追加 (UI では対応していない場合あり) や、スキーマが設定されている場合にはエントリーへのカスタム属性の追加にも対応します。

8.2.1. ipa コマンドの構造

基本的には、**ipa** コマンドは、大きいプラグインコンテナーです。ipa は多くのサブコマンドをサポートします。これらのサブコマンドは、実際には Identity Management で特定のオブジェクトタイプを管理するプラグインです。

最初のタイプのサブコマンドは、オブジェクトタイプ (user、sudo、group、host、dns など) を識別し、2 つ目はそのオブジェクトで実行される操作を特定します。

```
ipa objectType-operation objectName --option=value
```

たとえば、ユーザーの追加は、**user-add** サブコマンドを使用します。

```
ipa user-add entryName options
```

関連するサブコマンドは **plug-in モジュール** にグループ化されます。**dnszone-add** および **dnsrecord-add** のような DNS エントリーの管理コマンドはすべて、 **dns** モジュールまたは **topic** に属します。特定のエリアを管理する全情報、サポート対象の全コマンド、それぞれの例は、対象のトピックのヘルプを表示すると確認できます。

```
ipa help topic
```

ヒント

利用可能なトピックをすべて表示するには以下を実行します。

```
ipa help topics
```

トピックやコマンドのエリアではすべて、エントリーの管理方法には一貫したパターンがあります。

8.2.1.1. ipa でのエントリーの追加、編集、および削除

新しいエントリーは ***-add** コマンドを使用して追加します。たとえば、以下のようになります。

```
$ ipa user-add jsmith
```

add の操作では、コマンドで通常、必要な設定属性を入力するようにプロンプトを表示し、コマンドラインオプションとして、または **--set/addattr** オプション (「--setattr、--addattr、および --delattr を使用したエントリー属性の管理」) を使用してその内容を渡します。

```
$ ipa user-add
First name: John
```

```
Last name: Smith
User login [jsmith]: jsmith
--------------------
Added user "jsmith"
--------------------
...
```

同様に、エントリーは通常 ***-mod** コマンドを使用して編集します。編集後には、新規または編集した属性がオプションとして一覧表示されます。

```
$ ipa user-mod jsmith --title="Editor III"
```

最後に、***-del** コマンドおよびエントリー名を使用してエントリーを削除できます。

```
$ ipa user-del jsmith
```

8.2.1.2. ipa でのエントリーの検索および表示

-find** コマンドおよび任意の検索条件を使用して、全タイプのエントリーを検索できます。検索条件は、完全に一致する文字列または検索属性値のサブ文字列のいずれかで指定します。たとえば、**smith** の文字列に完全一致するもの (Smith の ***sn の値など) と、**jsmith** のユーザー名や **Smithson** といった長い名前などの値の一部に一致するものを検索します。

```
ipa user-find smith
```

検索はすべて自動的に文字列の部分検索を行うので、ワイルドカードを指定する必要はありません。

検索条件がないと、対象のタイプの全エントリーが表示されます。

検索 (***-find** コマンド) 時には、返されるエントリー数 (サイズ制限) および検索時間 (時間制限) など、サーバー設定の一部に制限が課されます。詳細は、「IdM 検索制限の設定」を参照してください。サーバー設定では、検索時のサイズや時間制限に関するグローバル初期設定を指定するものもあります。これらの制限は常に Web UI で適用されますが、***-find** コマンドに **--sizelimit** および **--timelimit** オプションを指定して上書きできます。たとえば、デフォルトの時間制限が 60 秒で検索にかかる時間が長くなる場合に、時間制限を 120 秒に増やすことができます。

```
[jsmith@ipaserver ~]$ ipa user-find smith --timelimit=120
```

エントリータイプの属性すべてを検索できるわけではありません。特定の属性サブセットは検索用に事前定義され、インデックス化されています。(このリストはユーザーおよびグループに対して設定可能ですが、他のタイプのエントリーには対応していません)。

エントリーが返されると、そのエントリーとともに特定のデフォルト属性のみが表示されます。エントリーに現在設定されている属性をすべて返すには、**--all** オプションを使用します。

特定のエントリーを表示するには、***-show** コマンドとエントリー名を使用します。検索と同様に、**--all** オプションが使用されない限り、エントリーとともに属性のサブセットのみが表示されます。

8.2.1.3. ipa でのグループおよびコンテナーへのメンバーの追加

グループメンバーは、単にエントリーを変更する以外に、別のコマンドを使用して追加、削除します。メンバーコマンドでは基本的に、さまざまな IdM エントリーの間で関係が作成されます。従来の group-member ロールではこれは明確ですが、エントリーが別のエントリーに関連付けられているポリシーエントリー (SELinux ポリシーや sudo ポリシーなど) にも該当します。

最も一般的に、メンバーエントリーの追加のコマンド形式は、***-add-member** ですが、このコマンドで ***-add-user** など、エントリータイプを指定できます。

同様に、メンバーのエントリーは、 (エントリー自体を削除するのではなく) ***-remove-member** または ***-remove-type** コマンドを使用して削除します。

8.2.2. ipa コマンドの位置要素

通常、**ipa** サブコマンドには、修正するエントリー名 (**オブジェクト**) と、サブコマンドで利用可能なオプションの要素が 2 つだけ含まれます。

```
ipa command entryName --options=values
```

ただし、エントリーによっては、エントリー名自体だけでなく、エントリーの **親** も指定する必要があります。たとえば、**automount** コマンドなどが例として挙げられます。automount (自動マウント) では、新しい鍵またはマップが作成されるたびに場所を含める必要があります。

親エントリー名を最初に、次に子エントリー名を指定します。たとえば、automount (自動マウント) の場合は、最初に場所を、次にマップまたはキーエントリー名を指定します。

```
ipa command parentEntryName childEntryName --childOptions=childValues
```

8.2.3. --setattr、--addattr、および --delattr を使用したエントリー属性の管理

Identity Management の全 ID および設定は、標準の Attribute-Value Assertion (AVA) を使用して LDAP エントリーとして格納されます。エントリーが UI または CLI 経由で作成されたかどうかに拘らず、エントリータイプのデフォルトおよびカスタムオブジェクトクラスによって、必要な特定の属性と利用可能な属性があります。

最も一般的な属性では、**ipa** コマンドは、コマンドライン引数を使用して値を設定します。たとえば、ユーザーにメール属性を追加するには、**--mail** 引数を使用し、DNS ゾーンの動的更新を有効にするには、zone コマンドに **--allow-dynupdate** オプションを使用します。また、自動マウントのマッピングのマップキーを指定するには **--key** オプションを使用します。

ただし、コマンドライン (または UI) オプションのない属性でもエントリーの設定が可能です。基盤となる LDAP スキーマには、許容可能な属性が多数あり、特にユーザーエントリーについては非常にリッチであることが理由の一部となっています。また、Identity Management ではユーザーおよびグループのスキーマ拡張が可能で、これらのカスタムスキーマ要素は必ずしも UI またはコマンドラインツールに反映されているとは限りません。

--setattr および **--addattr** オプションを使用して、サポート対象の属性をエントリーに追加または編集できます。

重要

追加する属性の値は、変更コマンドや **--setattr** または **--addattr** オプションでは検証されません。

いずれのオプションも、形式は以下のとおりです。

```
--setattr=attribute=value
```

--setattr オプションは、指定された属性に値を 1 つ設定し、既存の値は複数値の属性であっても上書きされます。

--addattr オプションは、属性に新しい値を追加します。複数値の属性の場合は、既存の値を維持しながら新しい値を追加します。

--setattr オプションと **--addattr** は、同じコマンド呼び出しで複数回使用できます。たとえば、以下のようになります。

```
$ ipa user-mod jsmith --addattr=mail=johnnys@me.com --addattr=mail=jsmith@example.com --
setattr=description="backup IT manager for the east coast branch"
```

同様に、属性または特定の属性値は、**--delattr** オプションを使用してエントリーから削除できます。値が 1 つだけの属性の場合には、属性は削除されます。値が複数ある属性の場合は、指定された値のみが削除されます。以下に例を示します。

```
$ ipa user-mod jsmith --delattr=mail=johnnys@me.com
```

注記

属性の追加または編集してから最後に、属性の削除が評価されます。1 回の変更操作で、同じ属性を追加して削除した場合は、何も操作はされません。

```
$ ipa user-mod jsmith --addattr=mail=johnnys@me.com --
delattr=mail=johnnys@me.com
```

8.2.4. IdM ツールでの特殊文字の使用

IdM コマンドラインツールは、シェルの他のユーティリティーとして実行されます。コマンドに引用符括弧 (> および <)、アンパサンド(&)、アスタリスク(*)、およびパイプ (|) など、特殊文字がある場合には、これらの特殊文字をエスケープする必要があります。エスケープしていない場合には、シェルがエスケープされていない文字を正しく解析できないため、コマンドの実行に失敗します。

8.2.5. 実行前の IdM ドメインへのログイン

IdM コマンド (**ipa-server-install** などのインストールコマンドを除く) を実行する前に、ユーザーは最初に Kerberos チケットを取得して IdM ドメインに対して認証する必要があります。これには、**kinit** を使用します。

```
[jsmith@ipaserver ~]$ kinit admin
```

「IdM へのログイン」では、他のログインオプションについて説明しています。

8.3. IDM へのログイン

ユーザーは、Kerberos 認証を使用して、コマンドラインツールや Web UI などの IdM サービスに対して認証されます。そのため、Identity Management にログインするには **kinit** を実行する必要があります。

kinit を実行すると、ユーザーに Kerberos **チケット** が発行されます。すべてのドメインサービスにアクセスするのに一度だけしかログインする必要がないように、このチケットはいずれかの IdM または Kerberos 対応のサービスにより確認されます。ドメインサービスには、IdM の Web UI、マウントした

ファイル共有、wiki、または IdM を ID/認証ストアとして使用するその他のアプリケーションが含まれます。

8.3.1. IdM へのログイン

Identity Management にログインするには、IdM ドメイン内のクライアントで **kinit** を実行する必要があります。

```
$ kinit
```

クライアントが IdM KDC で認証されるように、IdM ドメイン内でクライアントとして設定されたマシンから **kinit** コマンドを実行する必要があります。

kinit を実行するだけで、現在ログイン中のユーザーアカウントとして IdM にログインできます。IdM Kerberos ドメインに対して正常に認証するには、このユーザーアカウントも IdM ユーザーでなければなりません。たとえば、**user** としてマシンにログインしている場合は、以下を実行します。

```
$ kinit
Password for user@EXAMPLE.COM:
```

注記

SSSD または **pam_krb5** が IdM クライアントマシンに設定されている場合には、ユーザーがマシンにログインすると、**sudo** など認証が必要なマシンサービスに使用できるチケットが作成されます。

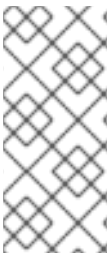
8.3.2. IdM ユーザーがシステムユーザーではない場合のログイン

ユーザーのシステムユーザー名は、IdM のユーザー名とは異なります。IdM ユーザー名を指定するか、アカウントを切り替えるには、**kinit** コマンドを再度実行して、新しいユーザーを指定するだけです。たとえば、以下のようになります。

```
$ kinit userName
Password for userName@EXAMPLE.COM:
```

サーバーの初期設定時に、通常の管理アクティビティーを実行する管理ユーザー **admin** が作成されます。admin ユーザーとして認証するには、**kinit** の実行時に、ユーザー名として admin を使用します。

```
$ kinit admin
```



注記

チケットは、ログインユーザーごとに 1 つだけ保存できます。現在保存されている認証情報は、IdM サービスへのアクセス時に使用される認証情報です。

別のユーザーとして IdM Web UI に接続している場合は、ブラウザーを更新して、新規ユーザーの更新済みの情報を表示します。

8.3.3. 現在ログインしているユーザーの確認

klist コマンドを使用して、サーバーからの ID および ticket granting ticket (TGT)を検証します。

```
$ klist
Ticket cache: FILE:/tmp/krb5cc_500
Default principal: ipaUser@EXAMPLE.COM

Valid starting     Expires            Service principal
11/10/08 15:35:45  11/11/08 15:35:45  krbtgt/EXAMPLE.COM@EXAMPLE.COM

Kerberos 4 ticket cache: /tmp/tkt500
klist: You have no tickets cached
```

認証済みのユーザーしか、IdM サービスにアクセスできないので、認証されたユーザーを特定することが重要です。**kinit** の Kerberos クライアントライブラリーには制限があります。その 1 つとして、 **kinit** を新たに呼び出すと、現在のチケットが上書きされる点が挙げられます。ユーザー A として認証してからユーザー B として認証すると、ユーザー A のチケットが上書きされます。

マシンで複数の認証ユーザーを存在させるには、**KRB5CCNAME** 環境変数を設定します。この変数は、認証情報のキャッシュを異なるシェルに分離します。

8.3.4. ユーザーの Kerberos チケットのキャッシュ

チケットは、ログインユーザーごとに 1 つだけ保存できます。現在保存されている認証情報は、IdM サービスへのアクセス時に使用される認証情報です。

たとえば **admin** として認証し、新規ユーザーの追加、パスワードの設定を行い、そのユーザーとして認証を行おうとすると、管理者のチケットがなくなります。

別のシェルに、認証情報キャッシュを分離するには、**KRB5CCNAME** の特別な環境変数を使用します。

8.4. IDM WEB UI の使用

Web UI を使用するには、IdM Kerberos ドメインでユーザーを認証して、このユーザーには有効な Kerberos チケットが必要です。「IdM へのログイン」を参照してください。通常、Web UI には IdM サーバーまたはクライアントマシンからしかアクセスできないので、ユーザーをローカルで認証する必要があります。回避策は 2 つあり、ドメインを使用しないマシンで Kerberos を設定して Kerberos ドメインに接続するか (「別のシステムでのブラウザーの使用」を参照)、パスワードを使用して UI への認証を行います。

8.4.1. Web UI の概要

Web UI には主に 3 つの機能エリアがあります。各機能エリアは、IdM の主要な機能それぞれ (Identity Management、ポリシー管理、ドメイン設定) に対応します。

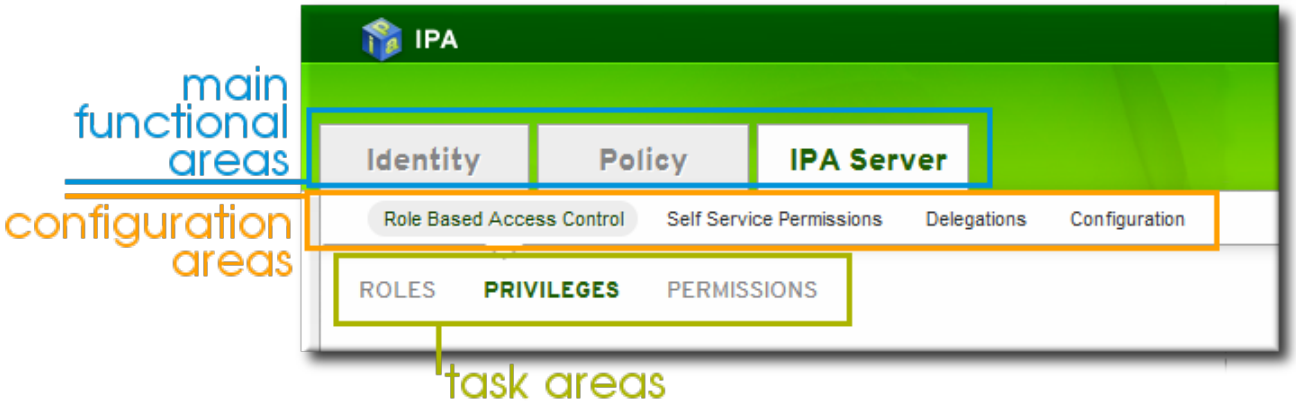
表8.1 タブごとの設定エリア

メインメニュータブ	設定エリア

メインメニュータブ	設定エリア
アイデンティティー	• ユーザーエントリー • ユーザーグループエントリー • ホスト/クライアントエントリー • ホストグループエントリー • netgroups エントリー • ドメインサービスエントリー • DNS (設定されている場合)
ポリシー	• ホストベースのアクセス制御 • Sudo ルール • automount • ユーザーパスワードポリシー • Kerberos チケットポリシー
IdM サーバー (Identity Management 内のアクセス制御)	• ロールベースのアクセス制御 (グループメンバーシップに基づくパーミッション) • 自己権限 • 委譲 (他のユーザーに対するユーザーアクセス制御)

全ページの上部にある **メインメニュー** には、表8.1「タブごとの設定エリア」に記載の機能エリアに対応するタブが 3 つあります。タブを選択すると、各種設定エリアを含むサブメニューがあります。設定エリアによっては複数のエントリーがある場合があります。たとえば、ロールベースのアクセス制御はユーザーロール/グループを定義し、アクセスを付与/拒否 (特権) できるエリア、これらのエリアに付与されるパーミッションを定義します。個別の設定エリアには、主の設定エリアの下に独自のタスクエリアがあります。

図8.1 メインメニュー



8.4.2. IdM Web UI の表示

「ブラウザーの設定」の記載どおりに、ブラウザーを正しく設定して、ユーザーが UI に接続できるように Kerberos 認証をサポートする必要があります。

Web UI を開くには、以下を実行します。

1. 「IdM へのログイン」の記載のように、**kinit** を使用して有効な Kerberos チケットを取得します。
2. IdM の URL を開きます。完全な URL は **https://IPAserver-FQDN/ipa/ui** ですが、このサービスにも **https://IPAserver-FQDN** を開くだけでアクセスできます。たとえば、以下のようになります。

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

8.4.3. ブラウザーの設定

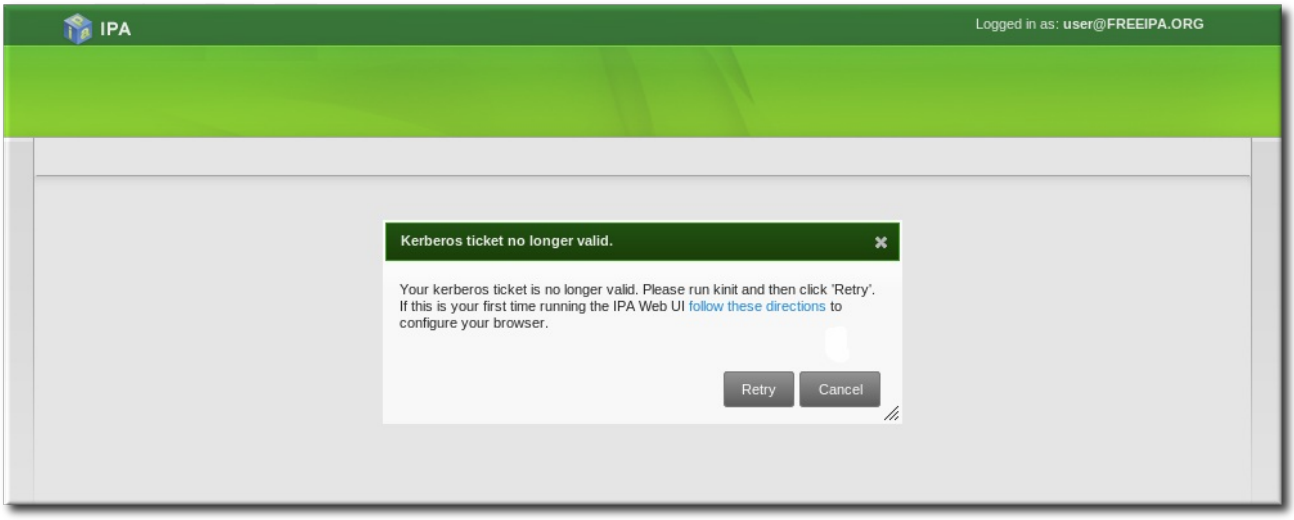
Web UI への接続に対応する Web ブラウザーは Firefox バージョン 17 以降および Google Chrome です。ブラウザーの設定に関する詳細は、以下の適切な項を参照してください。

- 「Firefox の設定」
- 「Chrome の設定」

8.4.3.1. Firefox の設定

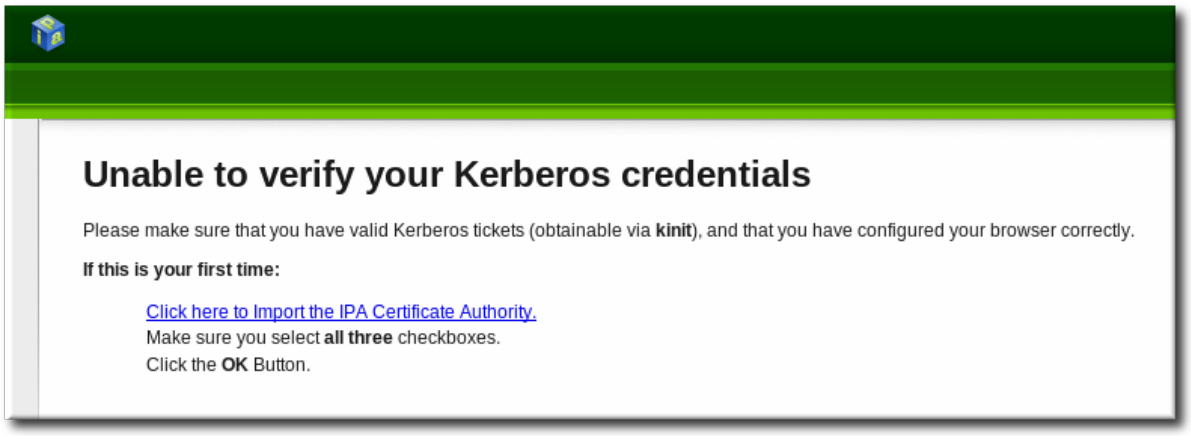
Firefox では、Kerberos 認証情報を使用して IdM UI に対して認証できますが、IdM ドメインを使用するように Kerberos 交渉を設定する必要があります。初回ログイン時に、Firefox が Kerberos 認証をサポートするように設定されていない場合は、エラーメッセージが表示されます。

図8.2 Kerberos 認証エラーメッセージ



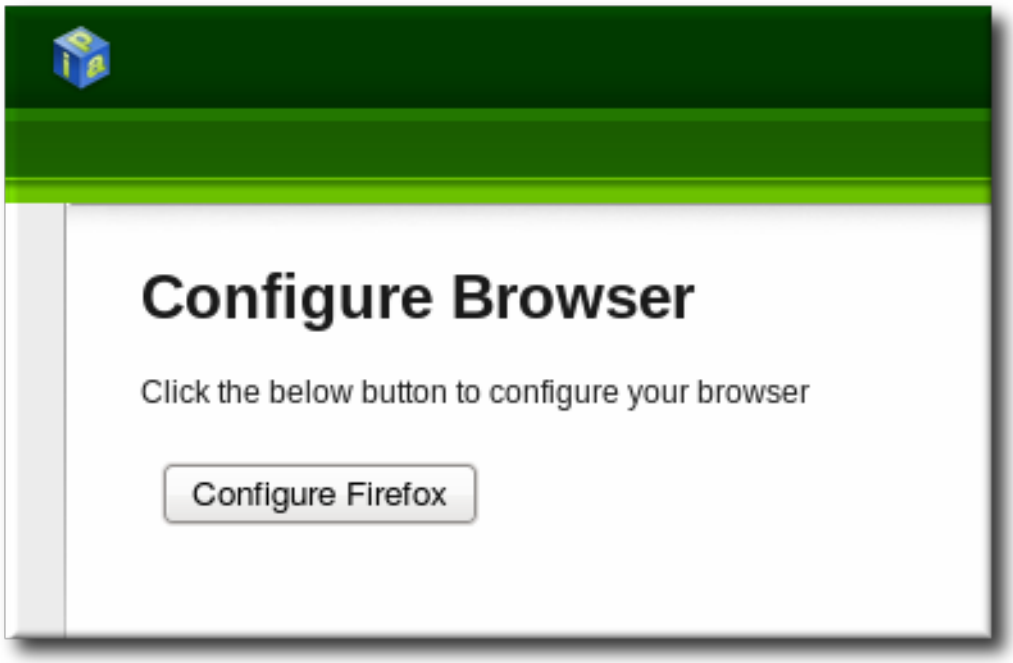
このエラーが表示された場合には、IdM の Web UI で以下の必要な設定を実行してください。

1. **follow these directions** リンクをクリックします。
2. IdM サーバーの CA 証明書のインポートリンクをクリックします。



3. CA 証明書の Web サイトおよびソフトウェア開発者 (最初と最後) トラストの部分を設定します。

4. **Configure Firefox** ボタンをクリックします。クリックすると、Firefox 設定の **negotiate** オプションすべてが入力され、IdM ドメインの設定に使用されます。



プロセスが完了すると、Firefox がシングルサインオンの設定を完了した旨の成功表示のポップアップが表示されます。そこから、IdM Web UI にリダイレクトされます。

この手順は手動で行うこともできます。

1. Firefox を起動します。
2. アドレスバーに **about:config** と入力します。
3. **Search** フィールドに **negotiate** と入力して Kerberos 関連のパラメーターをフィルターします。
4. Red Hat Enterprise Linux で、URI パラメーターのドメイン名を一番前のピリオド (.) も含めて入力し、**gsslib** パラメーターを true に設定します。

   ```
   network.negotiate-auth.trusted-uris  .example.com
   network.negotiate-auth.using-native-gsslib true
   ```

   Windows で、信頼できる URI およびライブラリーパスを設定し、認証用の組み込みの Microsoft Kerberos を無効にします。

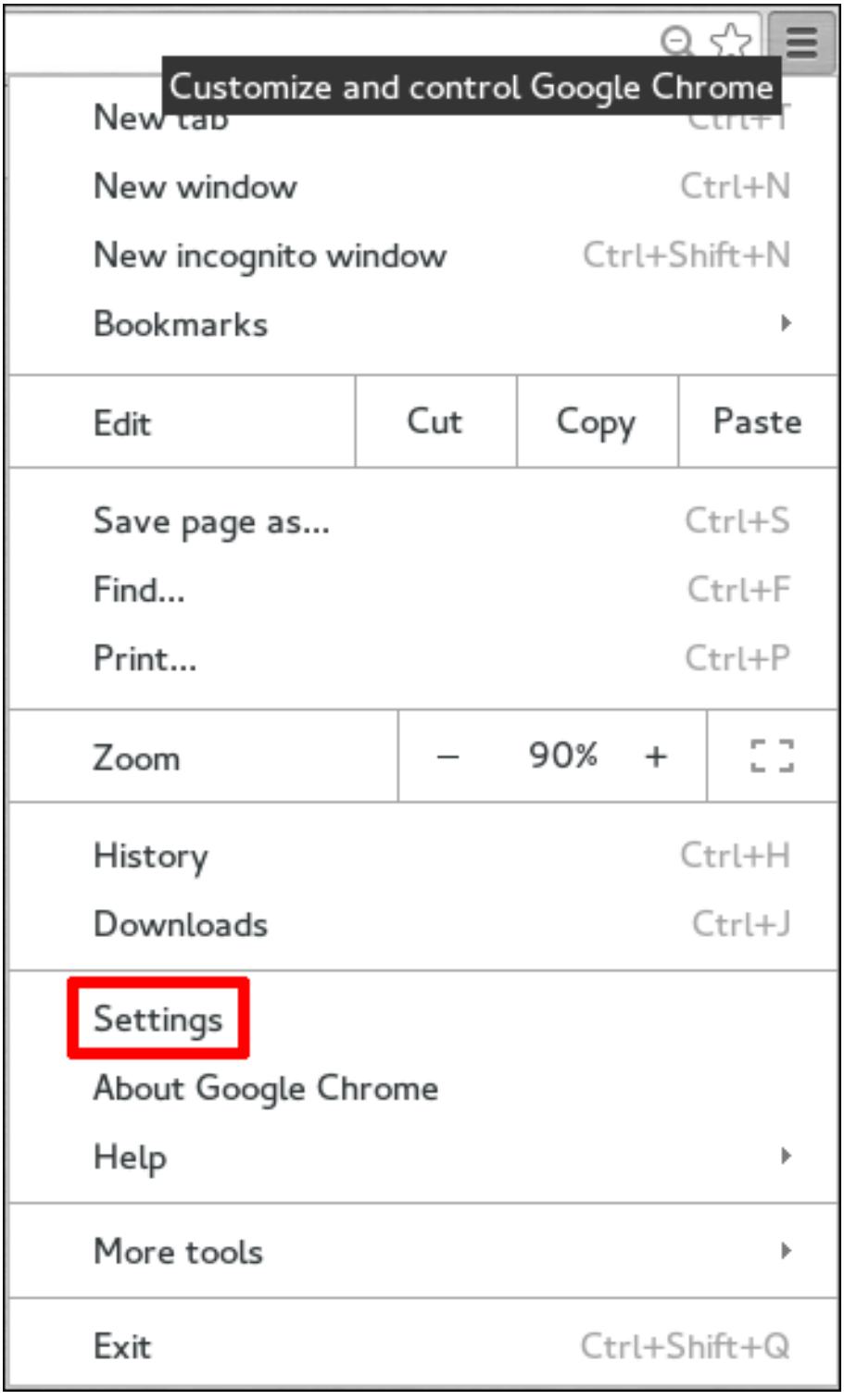
   ```
   network.negotiate-auth.trusted-uris .example.com
   network.auth.use-sspi false
   network.negotiate-auth.gsslib: C:\Program Files\MIT\Kerberos\bin\gssapi32.dll
   ```

   64 ビットシステムでは、ライブラリーは **C:\Program Files(x86)\MIT\Kerberos\bin\gssapi32.dll** に配置されています。
5. **http://ipaserver.example.com** など、IdM サーバーの完全修飾ドメイン名に移動して、Web UI を開きます。Web UI を開き、Kerberos の認証エラーがないことを確認します。
6. 次に IdM サーバーの CA 証明書を **http://ipa.example.com/ipa/config/ca.crt** からダウンロードします。
7. 表示された **Downloading Certificate** ウィンドウで、最初 (**Trust this CA to identify web sites**) と 3 番目 (**Trust this CA to identify software developers**) のチェックボックスを選択します。

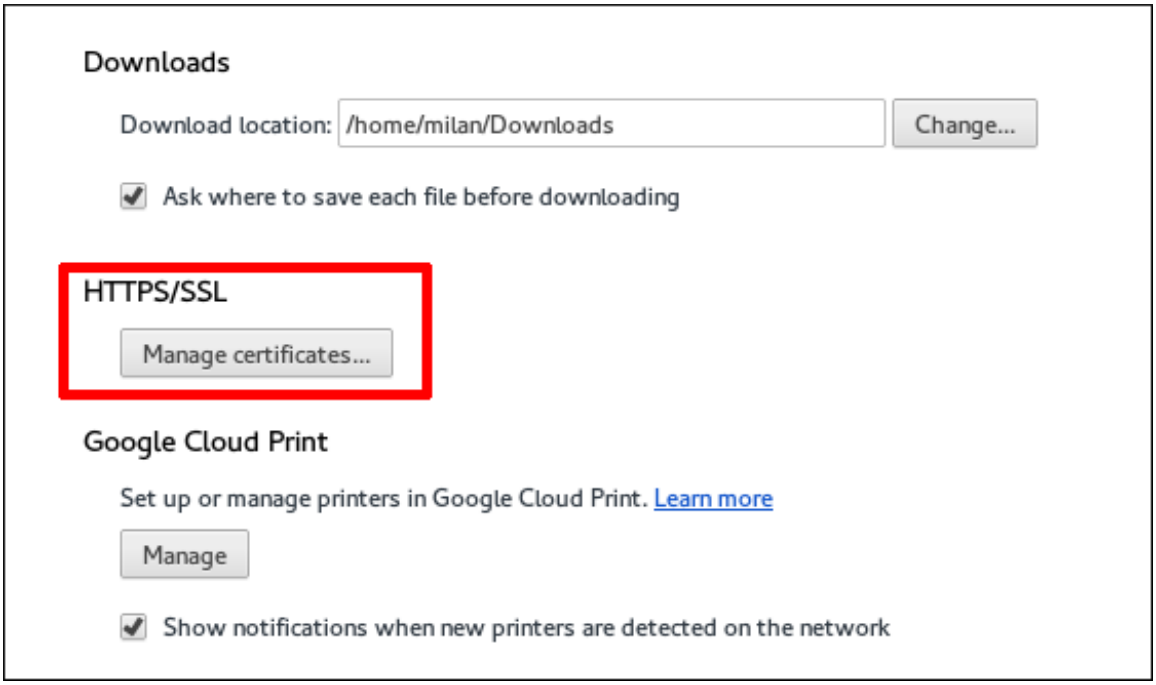
8.4.3.2. Chrome の設定

1. CA 証明書のインポート
   a. **http://*my.ipa.server*/ipa/config/ca.crt** から CA 証明書をダウンロードします。ホストが IdM クライアントでもある場合は、**/etc/ipa/ca.crt** で証明書を確認することができます。

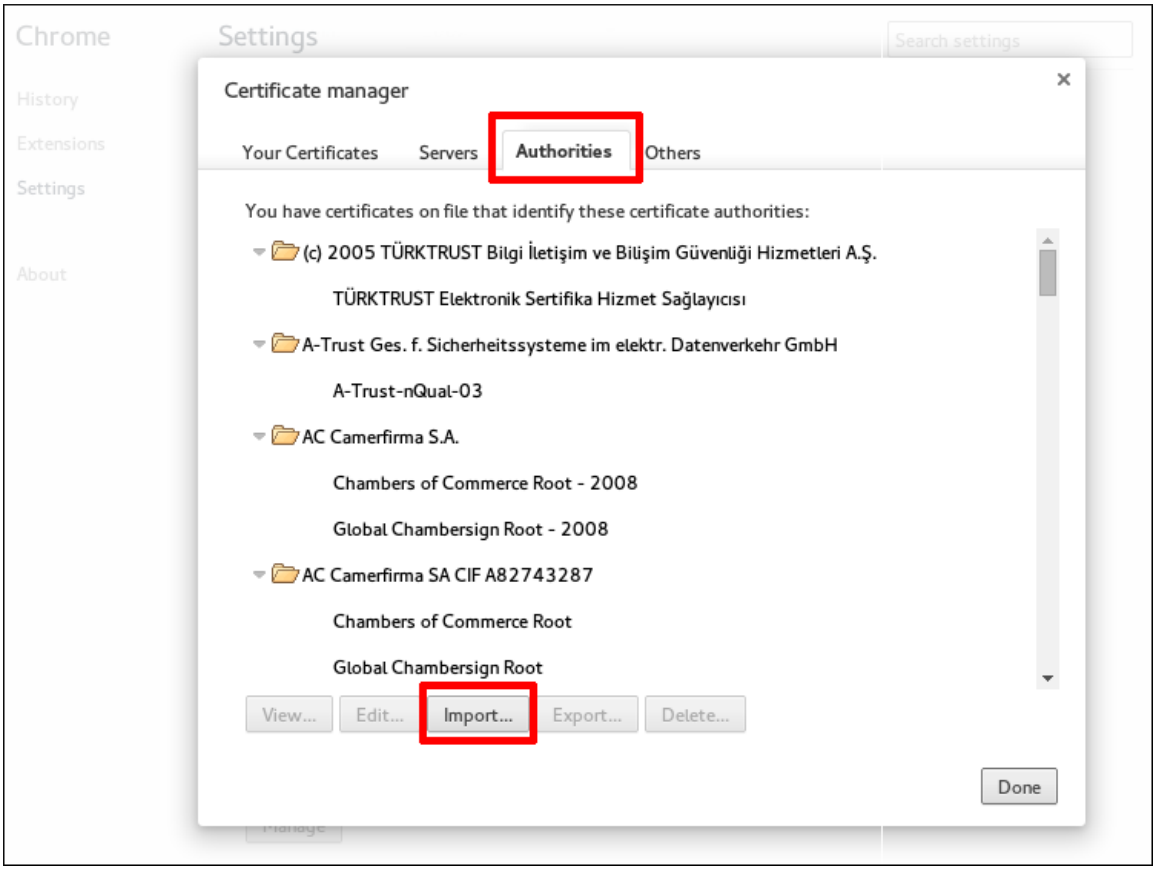
b. デフォルトでは Chrome の右上隅にある **Customize and control Google Chrome** のツールチップのメニューボタンをクリックし、**Settings** をクリックします。



c. **Show advanced settings** をクリックして他のオプションを表示し、**HTTPS/SSL** ヘディングの下にある **Manage certificates** ボタンをクリックします。



d. **Authorities** タブで、下部の **インポート** ボタンをクリックします。



e. 最初の手順でダウンロードした CA 証明書ファイルを選択します。

2. SPNEGO (Simple and Protected GSSAPI Negotiation Mechanism) を有効にして Chrome で Kerberos 認証を使用します。

a. 以下を実行して、必要なディレクトリーを作成していることを確認します。

```
[root@client]# mkdir -p /etc/opt/chrome/policies/managed/
```

b. 書き込み権限をシステム管理者または root に限定して新しい **/etc/opt/chrome/policies/managed/*mydomain.json*** ファイルを作成し、以下の行を追加します。

```
{ "AuthServerWhitelist": "*.example.com" }
```

これには以下を実行します。

```
[root@server]# echo '{ "AuthServerWhitelist": "*.example.com" }' >
/etc/opt/chrome/policies/managed/mydomain.json
```

8.4.4. 別のシステムでのブラウザーの使用

IdM ドメインのメンバーでは **ない** システムから Identity Management の Web UI に接続できます。このような場合には、**kinit** を実行する前に外部 (IdM 以外の) マシンで IdM 固有の Kerberos 設定ファイルを指定できます。その後に、IdM サーバードメインに対して認証が可能になります。

これは特に、インフラストラクチャー全体で複数のレルムや重複ドメインがある場合に役立ちます。

1. IdM サーバーから **/etc/krb5.conf** ファイルをコピーします。

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

警告

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. 「ブラウザーの設定」のように、外部マシンで Firefox を設定します。

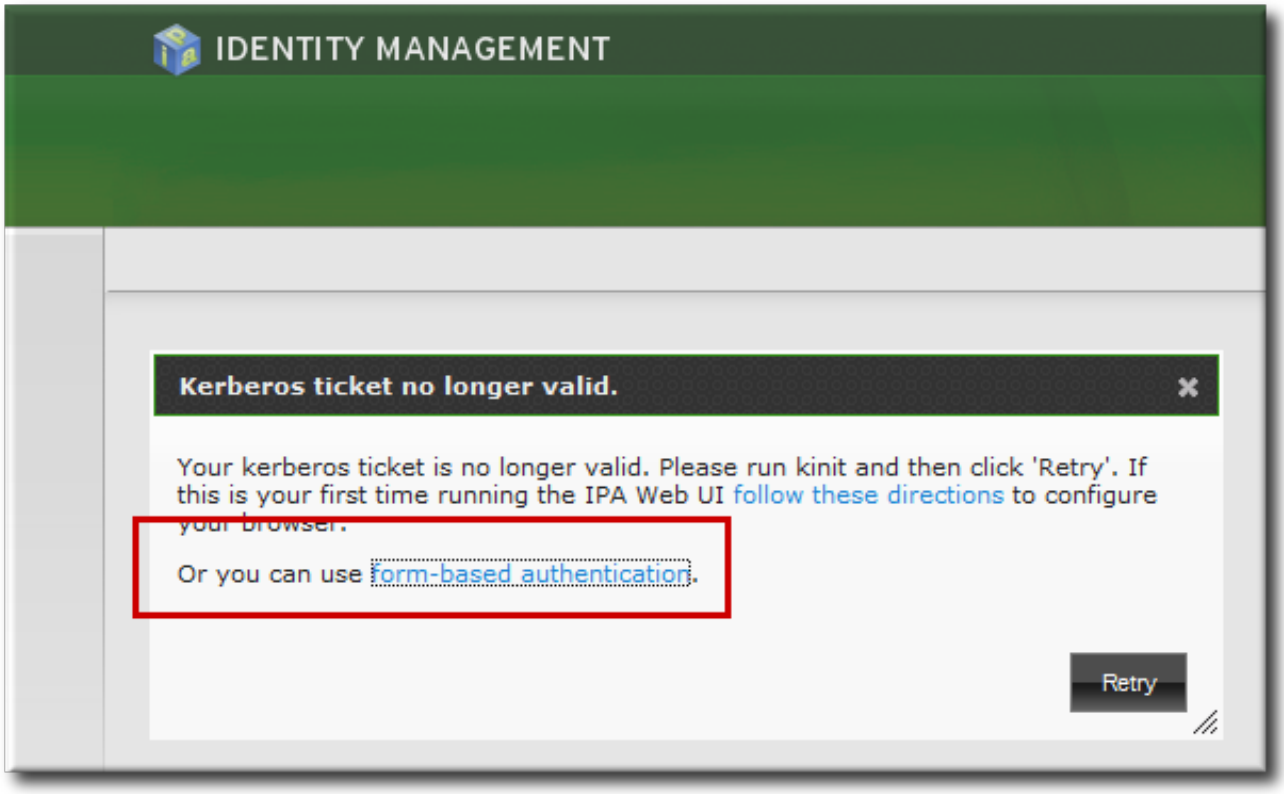
8.4.5. 簡易ユーザー名/パスワード認証情報でのログイン

Kerberos 認証が失敗すると、ブラウザーログインも失敗します。そのため、IdM の Web UI にアクセスができなくなります。UI の簡易認証により、Kerberos サービスに問題がある場合やシステムが IdM ドメイン外にある場合でもログインできるようになります。

Web UI にログインを試行するユーザーの、有効な Kerberos チケットが IdM サーバーで見つからない場合には、エラーメッセージが表示されます。IdM ドメインサービスに対する推奨の接続方法 (UI を含む) は、Kerberos 認証を使用する方法であるため、エラーでは最初に Kerberos 認証情報を更新するか、ブラウザーが Kerberos 認証に対応する設定を行うように指示します。

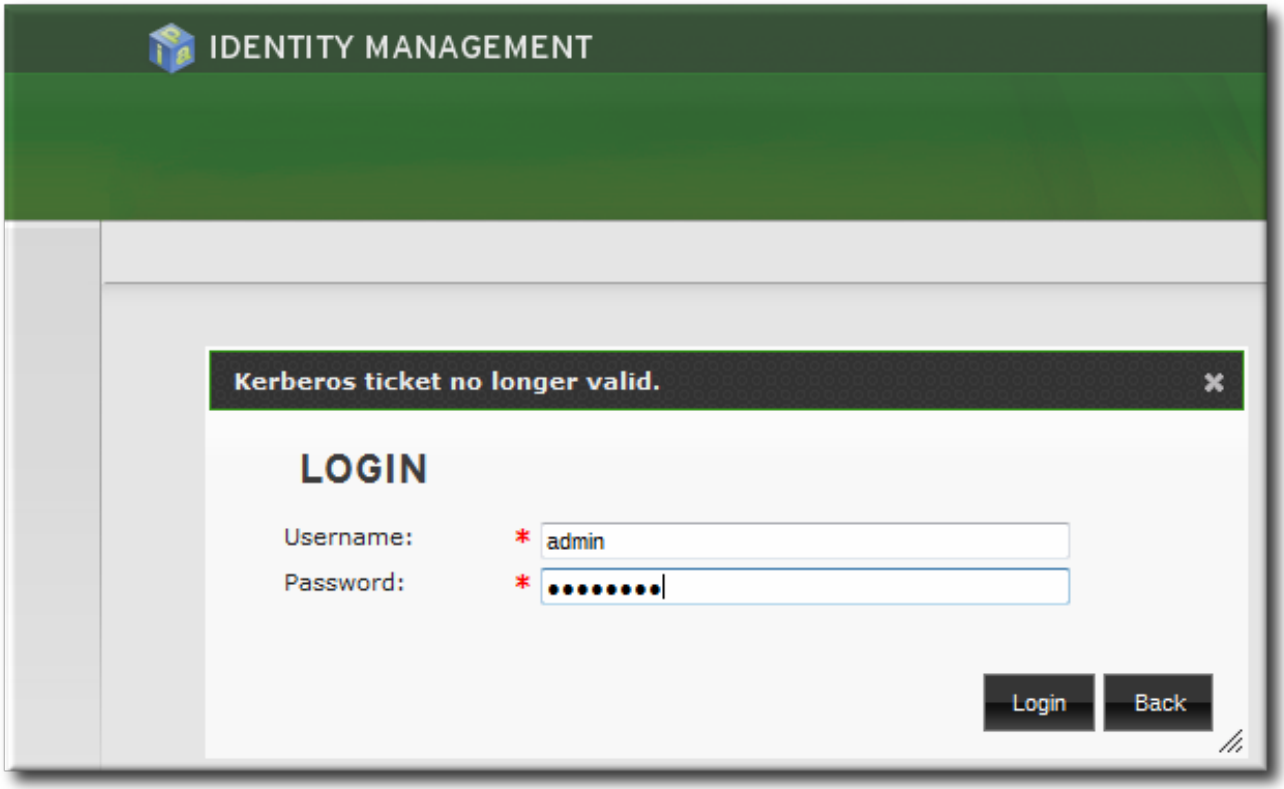
メッセージの 2 番目の部分で、簡易認証の代わりに使用する手段を提示します。**フォームベースの認証** リンクから、ログインページが開きます。

図8.3 IdM フォームベースのログインオプション



次に、設定された IdM ユーザーの UID およびパスワードを指定するだけです。

図8.4 IdM パスワードのプロンプト



8.4.6. プロキシーサーバーでの UI の使用

プロキシーサーバーを使用すると、IdM で追加設定なしに Web UI にアクセスできます。

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、IdM でプロキシーサーバーを使用できるので、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。

8.5. TLS 1.2 環境で実行する IDM サーバーの設定

詳細は、Red Hat ナレッジベースの「 Configuring TLS 1.2 for Identity Management in RHEL 6.9 」を参照してください。

---

[1] ディレクトリールックアップで使用するホスト名は、**/etc/ipa/default.conf** 設定ファイルで制御できます。

第9章 アイデンティティー: ユーザーおよびユーザーグループの管理

Identity Management のユーザーは、Kerberos 認証を使用してドメイン内のサービスおよびサーバーにアクセスできます。本章では、ユーザー、グループ、パスワードポリシー、および他のユーザー設定に関する一般的な管理タスクについて説明します。

9.1. ユーザーホームディレクトリーの設定

IdM ユーザーには、ホームディレクトリーが必要です。ホームディレクトリーが想定される場所にないと、ユーザーはドメインにログインできない可能性があります。システム管理者は IdM 以外でホームディレクトリーを管理できますが、PAM モジュールを使用して、IdM サーバーとクライアントの両方で自動的にホームディレクトリーを作成することもできます。

9.1.1. ホームディレクトリーの概要

IdM は、ユーザー管理の一環として、ユーザーのホームディレクトリーを管理できます。ただし、IdM には、管理対象のホームディレクトリーに対して、特定の定義済みパラメーターがあります。

- 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接頭辞は **/home** です。
- IdM では、ユーザーのログイン時に、ホームディレクトリーは自動的に作成されません。ホームディレクトリーの自動作成には、**pam_oddjob_mkhomedir** モジュールまたは **pam_mkhomedir** モジュールが必要です。このモジュールは、「PAM ホームディレクトリーモジュールの有効化」に記載されているように、クライアントのインストールの一部として、またはインストール後に設定できます。

  IdM のホームディレクトリープロセスでは、まず **pam_oddjob_mkhomedir** モジュールの使用を試みます。このモジュールでは、ホームディレクトリーの作成に必要なユーザー権限やアクセス権限が少なくて済み、SELinux とスムーズに統合できるようにするためです。このモジュールが利用できない場合には、プロセスは **pam_mkhomedir** モジュールにフォールバックします。

  注記

  Red Hat Enterprise Linux 5 クライアントでは、クライアントのインストールスクリプトは **pam_oddjob_mkhomedir** モジュールが利用できる場合でも、**pam_mkhomedir** モジュールを使用します。Red Hat Enterprise Linux 5 で **pam_oddjob_mkhomedir** モジュールを使用するには、PAM 設定を手動で編集します。

- ドメイン内の全マシンが利用できる **/home** を提供する NFS ファイルサーバーを使用し、IdM サーバーに自動マウントすることができます。

  NFS ユーザーへの root アクセス割り当てに関連す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の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、NFS の使用時に問題が発生する可能性があります。Identity Management では NFS の使用に関する一般的なガイドラインがあります。

  - automount を使用して、ユーザーがログインした時のみ、**/home** ツリー全体を読み込むのではなく、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。
  - 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ので、**sudo** または同様のプログラムを使用して IdM サーバーへの限定的なパーミッションを許可し、NFS サーバーにホームディレクトリーを作成できます。

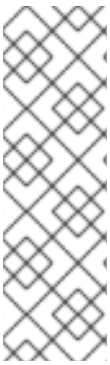
- **pam_oddjob_mkhomedir** モジュールなどのメカニズムを使用して、そのユーザーとしてホームディレクトリーを作成します。

ホームディレクトリーに自動マウントを使用する方法は、「ホームディレクトリーを手動でマウントする手順」を参照してください。

- ホームディレクトリーの作成に適したディレクトリーとメカニズムがない場合は、ログインできない可能性があります。

9.1.2. PAM ホームディレクトリーモジュールの有効化

ユーザーのログイン時にホームディレクトリーを自動的に作成するには、IdM で **pam_oddjob_mkhomedir** モジュールまたは **pam_mkhomedir** モジュールを使用できます。必要なパフォーマンスが少なく、SELinux と適切に連携するので、IdM では **pam_oddjob_mkhomedir** モジュールの使用が優先されます。このモジュールがインストールされていない場合は、**pam_mkhomedir** モジュールにフォールバックされます。



注記

IdM では **pam_oddjob_mkhomedir** モジュールまたは **pam_mkhomedir** モジュールが必要ではありません。これは、共有ストレージが利用できない場合でも、***_mkhomedir** モジュールがホームディレクトリーを作成しようとするためです。このモジュールでホームディレクトリーを作成できない場合は、ユーザーは IdM ドメインにログインできなくなります。

システム管理者は、必要に応じて各クライアントまたはサーバーでこのモジュールをアクティベートする必要があります。

pam_oddjob_mkhomedir (または **pam_mkhomedir**) モジュールを有効にする方法は 2 つあります。

- **--mkhomedir** オプションは **ipa-client-install** コマンドで使用できます。このオプションはクライアントでは可能ですが、サーバーで設定しても利用できません。
- システムの **authconfig** コマンドを使用して、**pam_oddjob_mkhomedir** モジュールを有効にできます。たとえば、以下のようになります。

```
authconfig --enablemkhomedir --update
```

このオプションは、インストール後のサーバーマシンとクライアントマシンの両方に使用できます。

注記

Red Hat Enterprise Linux 5 クライアントでは、クライアントのインストールスクリプトは **pam_oddjob_mkhomedir** モジュールが利用できる場合でも、**pam_mkhomedir** モジュールを使用します。Red Hat Enterprise Linux 5 で **pam_oddjob_mkhomedir** モジュールを使用するには、PAM 設定を手動で編集します。

9.1.3. ホームディレクトリーを手動でマウントする手順

PAM モジュールを使用すると、ユーザーのホームディレクトリーを自動作成できますが、この動作は環境によって適していない場合があります。このような場合に、ホームディレクトリーは、NFS 共有および **automount** を使用して別の場所から IdM サーバーに手動で追加できます。

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。

```
[bjensen@server ~]$ ipa automountlocation-add userdirs
Location: userdirs
```

2. 新しい場所の **auto.direct** ファイルに直接マップを追加します。この例では、マウントポイントは **/share** です。

```
[bjensen@server ~]$ ipa automountkey-add userdirs auto.direct --key=/share --info="-ro,soft,
ipaserver.example.com:/home/share"

Key: /share
Mount information: -ro,soft, ipaserver.example.com:/home/share
```

IdM で自動マウントの使用は、「18章 *ポリシー: 自動マウントの使用*」で詳細に説明されています。

9.2. ユーザーエントリーの管理

9.2.1. ユーザー名の形式

ユーザー名のデフォルトの長さは 32 文字です。

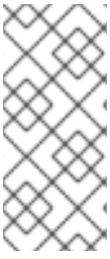
IdM は、以下の正規表現に基づいて、さまざまなユーザー名形式をサポートします。

```
[a-zA-Z0-9_.][a-zA-Z0-9_.-]{0,252}[a-zA-Z0-9_.$-]?
```

ヒント

Samba 3.x マシンがサポートされる場合は、末尾に $ 記号を使用できます。

IdM のユーザー名には、Unix システムでの数字で始まるユーザー名などのシステム制限が適用されます。



注記

ユーザー名の作成時、大文字と小文字は区別されません。つまり、大文字と小文字はどちらでも入力できますが、ユーザー名の保存時には大文字と小文字は無視されます。

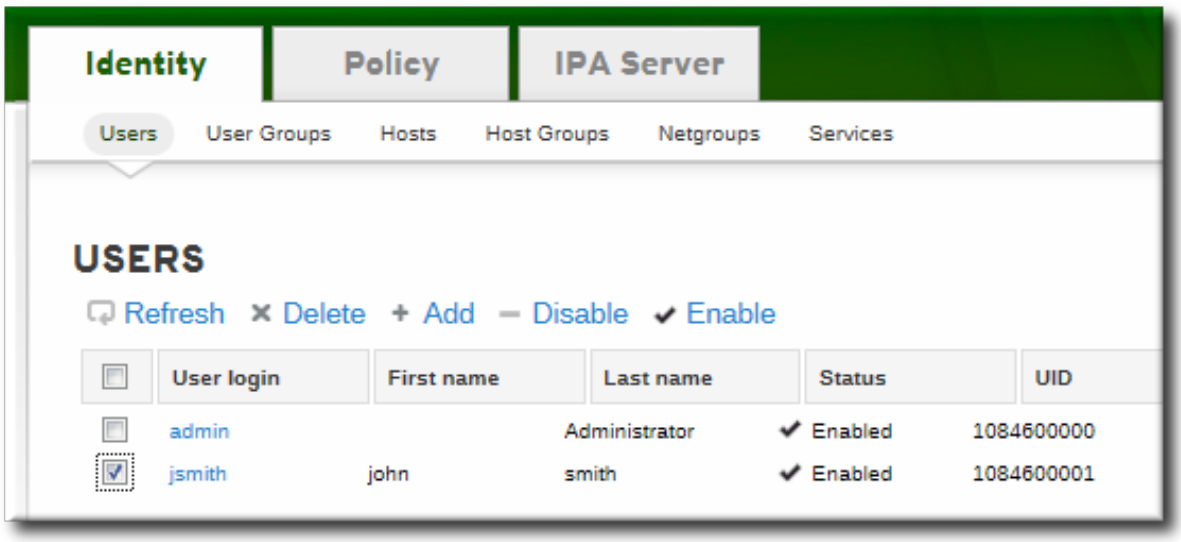
ユーザー名は、大文字と小文字が混在して作成された場合でも、すべての小文字になるように自動的に正規化されます。

9.2.2. ユーザーの追加

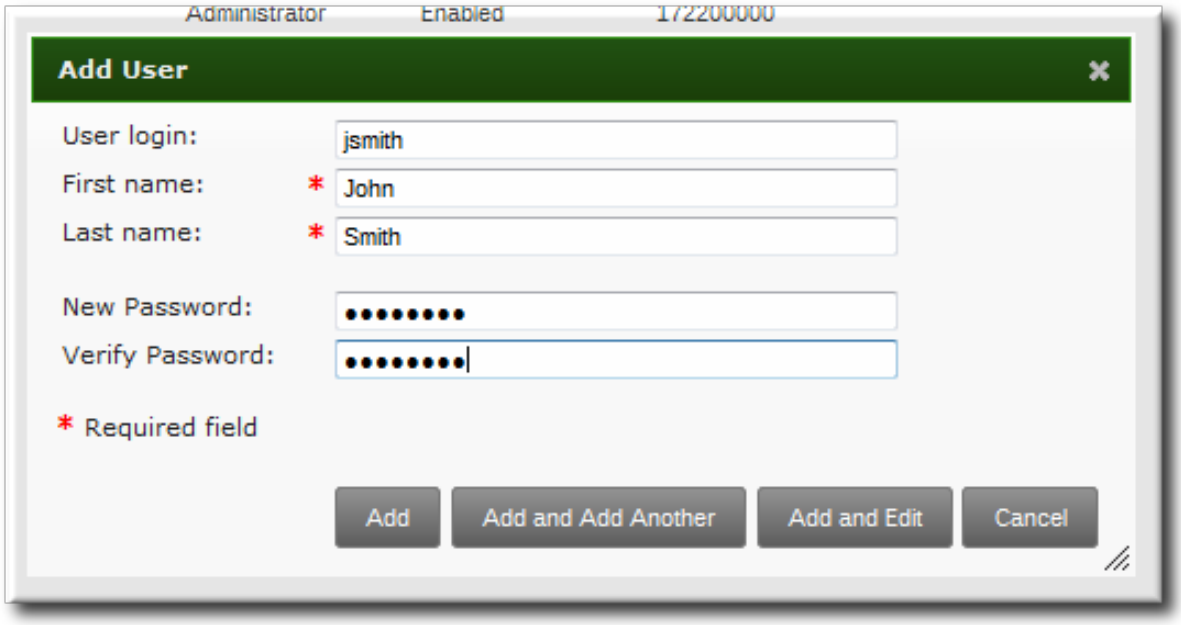
9.2.2.1. Web UI での操作

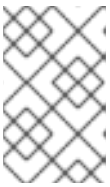
1. **Identity** タブを開き、サブタブの **ユーザー** を選択します。

2. ユーザー一覧上部にある **Add** をクリックします。



3. ユーザーの名と姓を入力します。ユーザーログイン (UID) はユーザーのフルネームに基づいて自動的に生成されますが、**Optional field** リンクをクリックすると手動で設定できます。

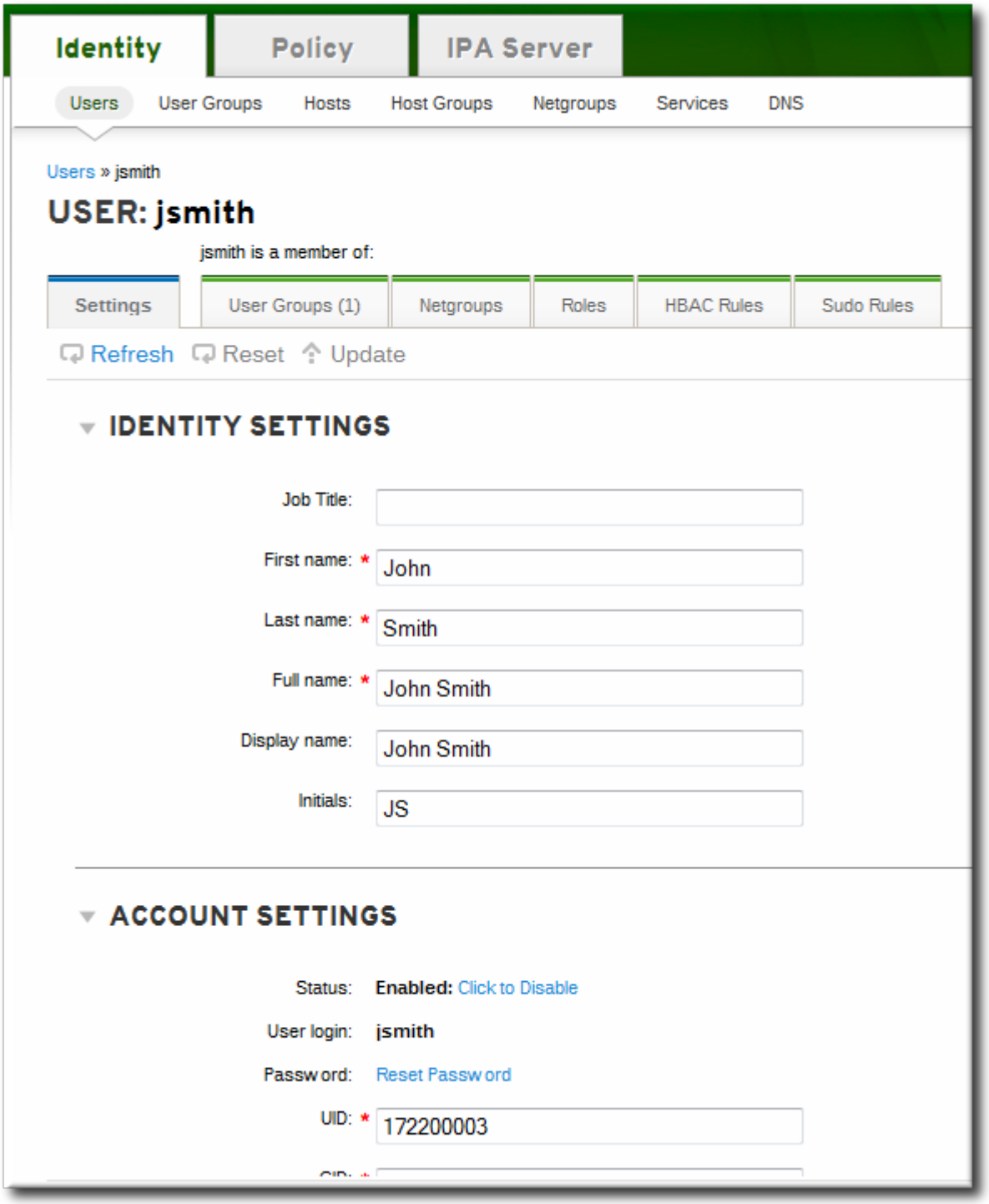




注記

ユーザー名の作成時には大文字と小文字は区別されませんので、大文字、小文字は無視されます。ユーザー名は、大文字と小文字が混在して作成された場合でも、すべての小文字になるように自動的に正規化されます。

4. 「Web UI での操作」にあるように、**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。ユーザーエントリーは、指定のユーザー情報およびユーザーエントリーテンプレートに基づいて、すでに入力されている基本情報で作成されます。



9.2.2.2. コマンドラインでの操作

user-add コマンドで、新しいユーザーエントリーが追加されます。表9.2「デフォルトの Identity Management ユーザー属性」にリストされている属性は、特定の値でエントリーに追加でき、コマンドは引数なしで実行できます。

```
[bjensen@server ~]$ ipa user-add [username] [attributes]
```

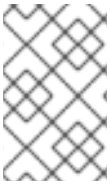
引数を使用しない場合には、コマンドにより、必要なユーザーアカウントの情報を求められ、他の属性には、以下に出力されているデフォルト値が使用されます。以下に例を示します。

```
[bjensen@server ~]$ ipa user-add
First name: John
```

```
Last name: Smith
User login [jsmith]: jsmith
--------------------
Added user "jsmith"
--------------------
User login: jsmith
First name: John
Last name: Smith
Full name: John Smith
Display name: John Smith
Initials: JS
Home directory: /home/jsmith
GECOS: John Smith
Login shell: /bin/sh
Kerberos principal: jsmith@EXAMPLE.COM
Email address: jsmith@example.com
UID: 882600007
GID: 882600007
Password: False
Member of groups: ipausers
Kerberos keys available: False
```

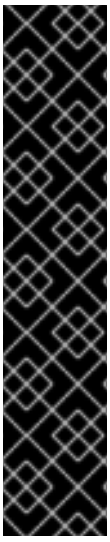
任意のユーザー属性をコマンドで指定できます。コマンドで指定すると、任意の属性の値が設定されるか、デフォルト属性のデフォルト値が上書きされます。

```
[bjensen@server ~]$ ipa user-add jsmith --first=John --last=Smith --manager=bjensen --
email=johnls@example.com --homedir=/home/work/johns --password
```



注記

ユーザー名の作成時には大文字と小文字は区別されませんので、大文字、小文字は無視されます。ユーザー名は、大文字と小文字が混在して作成された場合でも、すべての小文字になるように自動的に正規化されます。



重要

UID または GID の番号を指定せずにユーザーを作成すると、ユーザーアカウントには、サーバーまたはレプリカの範囲で次に利用可能な ID 番号が自動的に割り当てられます。(数値の範囲は「一意の UID および GID 番号の割り当て管理」 で詳述されています。) つまり、UID 番号およびプライベートグループ (設定されている場合) には一意の番号が常に割り当てられることになります。

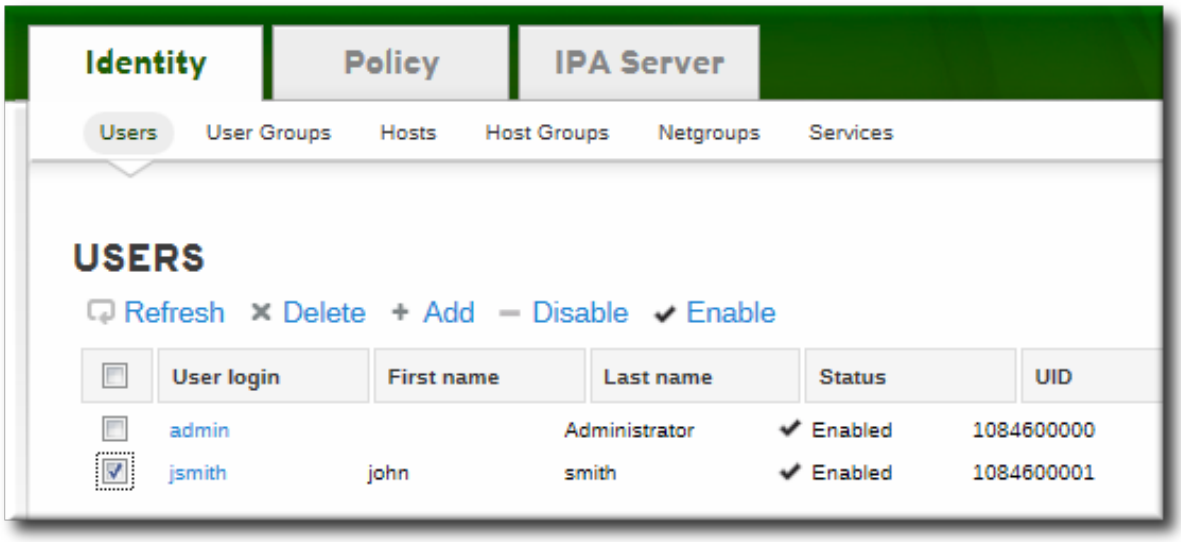
数値がユーザーエントリーに **手動で** 割り当てられると、サーバーでは ***uidNumber*** が一意であるかどうかは検証されません。ID を重複させることができます。POSIX エントリーでは、想定されている動作 (非推奨) です。

2 つのエントリーに同じ ID 番号が割り当てられている場合に、検索では、対象の ID 番号の最初のエントリーだけが返されます。ただし、他の属性の検索時や、 **ipa user-find --all** を使用時には、両方のエントリーが返されます。

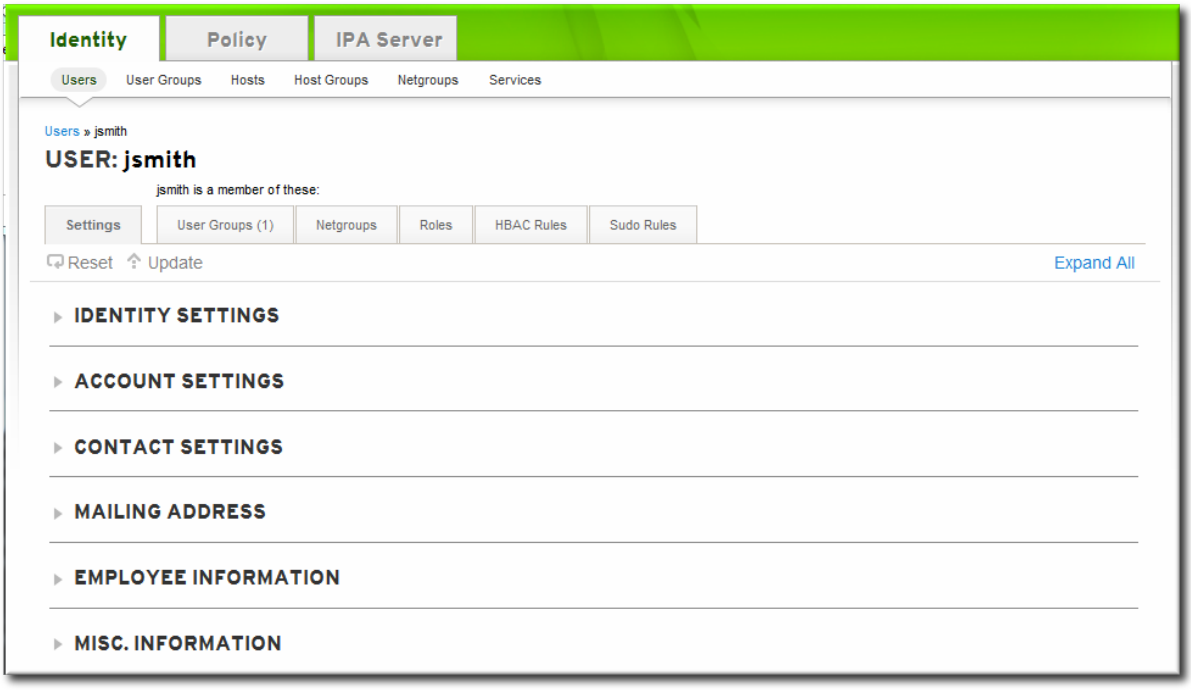
9.2.3. ユーザーの編集

9.2.3.1. Web UI での操作

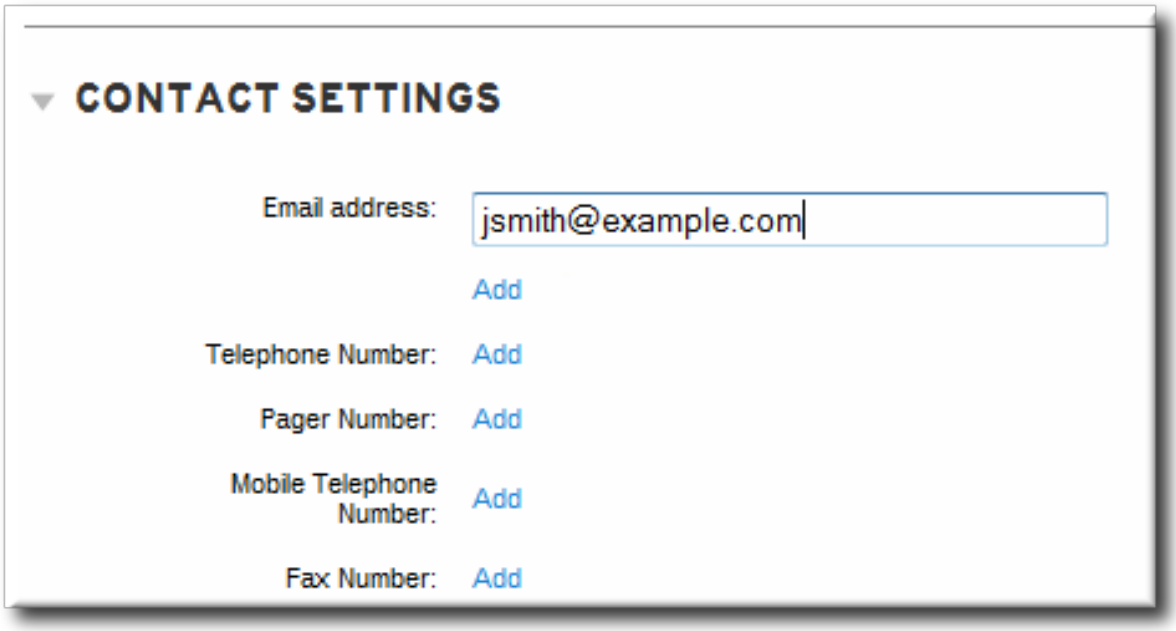
1. **Identity** タブを開き、サブタブの **ユーザー** を選択します。
2. 編集するユーザー名をクリックします。



3. ユーザー用に編集できる属性には、さまざまなタイプがあります。デフォルト属性はすべて、表9.2「デフォルトの Identity Management ユーザー属性」 に記載されています。**Identity Settings** と **Account Settings** エリアの属性のほとんどは、ユーザー情報またはユーザーエントリーのテンプレートを基にデフォルト値が入力されています。



4. フィールドを編集するか、必要に応じて属性別に **Add** リンクをクリックし、エントリーで属性を作成します。



5. 編集が完了したら、ページ上部にある **Update** リンクをクリックします。

9.2.3.2. コマンドラインでの操作

user-mod コマンドでは、属性を追加または変更してユーザーアカウントを編集します。基本的には **user-mod** は (ログイン ID で) ユーザーアカウント、編集する属性、新しい値を指定します。

```
[bjensen@server ~]$ ipa user-mod loginID --attributeName=newValue
```

たとえば、ユーザーの役職を **Editor II** から **Editor III** に変更するには以下を実行します。

```
[bjensen@server ~]$ ipa user-mod jsmith --title="Editor III"
```

Identity Management では複数の値を指定可能な LDAP の属性をもとに、**多値** 属性を使用できます。たとえば、あるユーザーがメールアドレスを 2 つ (仕事用と個人用) 使用している場合には、いずれも mail 属性に格納されます。多値属性は、**--addattr** オプションで管理できます。

mail のように、複数の値を属性に指定できる場合には、単純にコマンドラインの引数を使用することで新しい値に上書きされます。これは、**--setattr** の使用時も同様です。ただし、**--addattr** を使用すると新しい属性が追加されます。多値属性の場合には、既存の値に新しい値が追加されます。

例9.1 複数のメール属性

最初に、職場のメールアカウントでユーザーを作成します。

```
[bjensen@server ~]$ ipa user-add jsmith --first=John --last=Smith --email=johnls@example.com
```

次に、個人メールアカウントを追加します。

```
[bjensen@server ~]$ ipa user-mod jsmith --addattr=mail=johnnys@me.com
```

このユーザーに両方のメールアドレスが表示されます。

```
[bjensen@server ~]$ ipa user-find jsmith --all
--------------
```

```
1 user matched
--------------
  dn: uid=jsmith,cn=users,cn=accounts,dc=example,dc=com
  User login: jsmith
  .....
  Email address: jsmith@example.com, jsmith@new.com
```

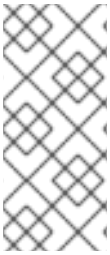
同時に 2 つの値を設定するには、**--addattr** オプションを 2 回使用します。

```
[bjensen@server ~]$ ipa user-add jsmith --first=John --last=Smith --email=johnls@example.com --addattr=mail=johnnys@me.com --addattr=mail=admin@example.com
```

9.2.4. ユーザーの削除

ユーザーアカウントを完全に削除すると、ユーザーエントリーとグループメンバーシップやパスワードなど、そのユーザーの情報をすべて IdM から削除します。システムアカウントやホームディレクトリーなどの外部設定は、作成されたサーバーまたはローカルマシンに存在しますが、IdM 経由ではアクセスできません。

ユーザーアカウントを削除すると元に戻せません。情報を復元することはできず、新しいアカウントを作成する必要があります。



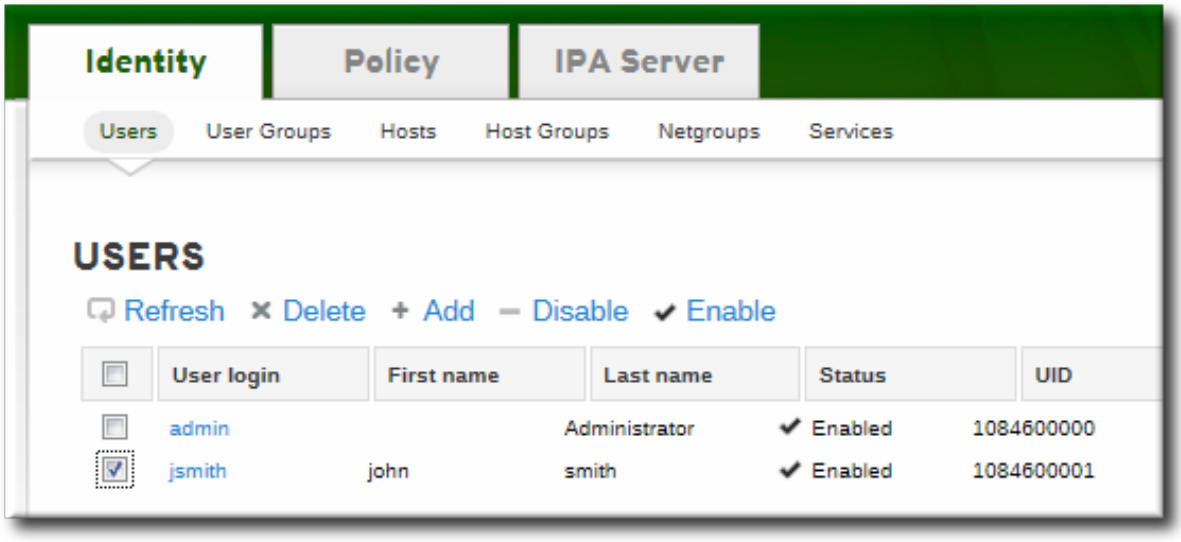
注記

すべての管理ユーザーが削除された場合、Directory Manager アカウントを使用して新しい管理ユーザーを作成する必要があります。

または、グループ管理ロールが割り当てられたユーザーでも、新しい管理ユーザーを追加できます。

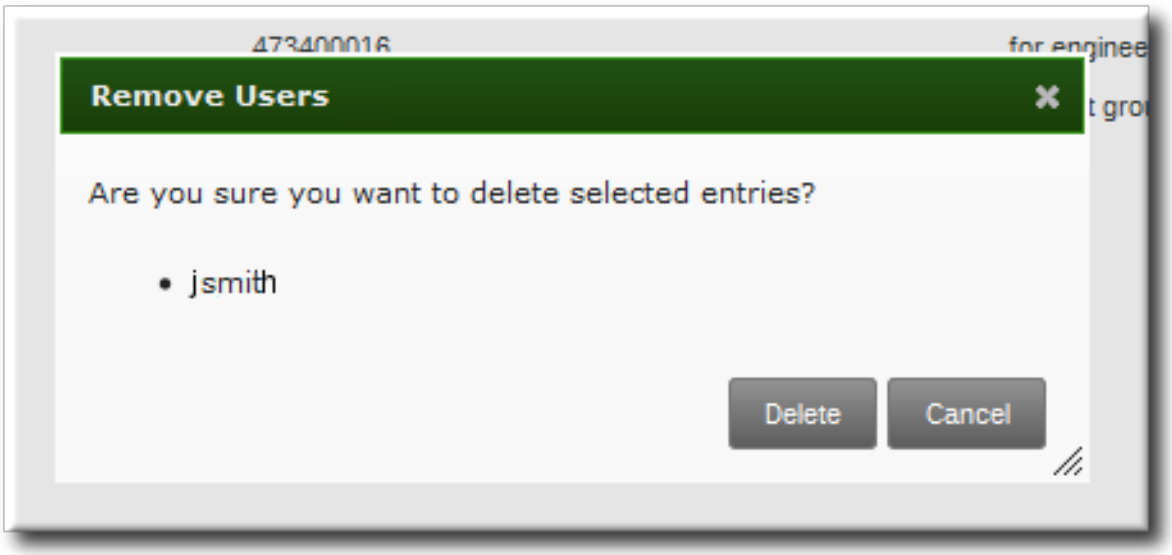
9.2.4.1. Web UI の使用

1. **Identity** タブを開き、サブタブの **ユーザー** を選択します。
2. ユーザー名のチェックボックスを選択して削除します。



3. タスクエリアの上部にある **Delete** リンクをクリックします。

4. プロンプトが表示されたら、削除を確定します。



9.2.4.2. コマンドラインでの操作

ユーザーの削除には **user-del** コマンドでユーザーを削除し、ユーザーログインを削除します。たとえばユーザーを 1 つだけ削除する場合には以下を実行します。

```
[bjensen@server ~]$ ipa user-del jsmith
```

複数のユーザーを削除するには、スペースで区切ってユーザーをリストします。

```
[bjensen@server ~]$ ipa user-del jsmith bjensen mreynolds cdickens
```

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、コマンドの完了時に標準出力 (stdout) に出力されます。**--continue** を使用しない場合には、このコマンドはエラーが発生する **まで**、操作を続行し、(エラーが発生すると) 操作を終了します。

9.3. ユーザーの公開 SSH 鍵の管理

OpenSSH は、**公開鍵と秘密鍵のペア** を使用してユーザーを認証します。ユーザーがネットワークリソースにアクセスを試行するときに、このキーペアを提示します。ユーザーの初回認証時には、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、マシンはユーザーの公開鍵を **authorized_keys** ファイルに保存します。ユーザーがリソースに再度アクセスを試みると、マシンは単に **authorized_keys** ファイルをチェックして、承認済みのユーザーに自動的にアクセスを許可します。

このシステムには、以下の問題があります。

- 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ユーザーキーを許可する必要がありますが、ユーザーまたはキー発行者を適切に検証することが困難であるため、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) がユーザーの 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ユーザーキーを 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、ユーザー SSH 鍵の配布や更新、検証を考慮する必要がありません。

9.3.1. 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、**id_rsa.pub** などのキーファイルでは、キーエントリーは先にタイプで、次にキー自体、その後に追加のコメントまたは識別子で識別されます。たとえば、特定のホスト名に関連付けられた RSA 鍵の場合:

```
"ssh-rsa ABCD1234...== ipaclient.example.com"
```

キーファイルの 3 要素はすべて、ユーザーエントリーにアップロードして表示できます。または、キーだけをアップロードすることもできます。

9.3.2. ユーザー SSH 鍵の Web UI でのアップロード

1. ユーザーキーを生成します。たとえば、以下のように OpenSSH ツールを使用します。

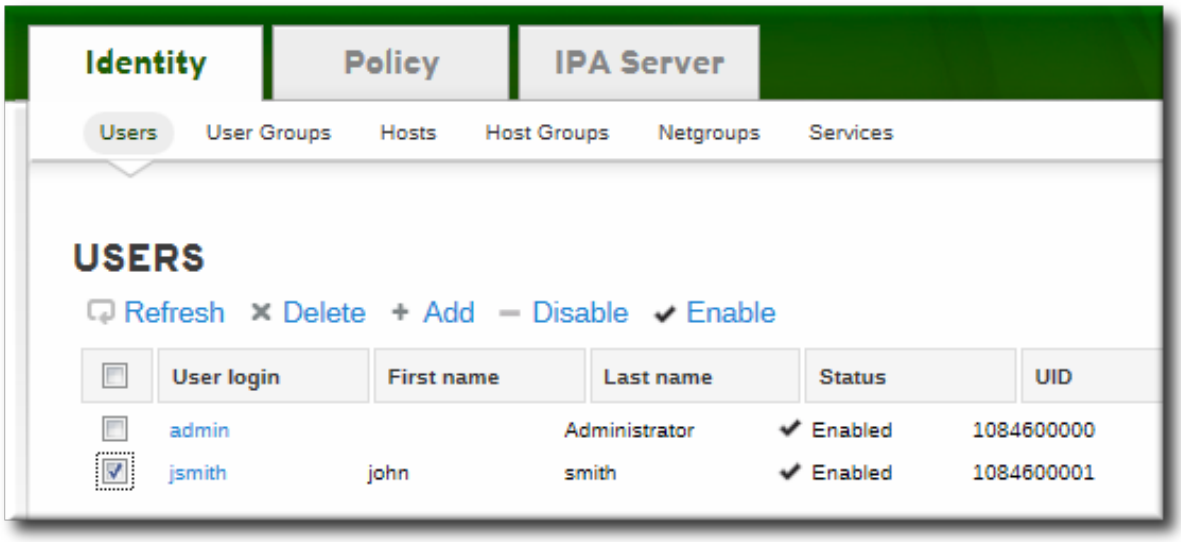
```
[jsmith@server ~]$ ssh-keygen -t rsa -C jsmith@example.com
Generating public/private rsa key pair.
Enter file in which to save the key (/home/jsmith/.ssh/id_rsa):
Created directory '/home/jsmith/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/jsmith/.ssh/id_rsa.
Your public key has been saved in /home/jsmith/.ssh/id_rsa.pub.
The key fingerprint is:
a5:fd:ac:d3:9b:39:29:d0:ab:0e:9a:44:d1:78:9c:f2 jsmith@example.com
The key's randomart image is:
+--[ RSA 2048]----+
|             |
|    + .      |
|   + =   .   |
|    =   +    |
|   . E S..   |
|  .    . .o  |
|   . .  . oo.|
|  . o .  +.+o|
|   o  .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 **type key== comment** 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。

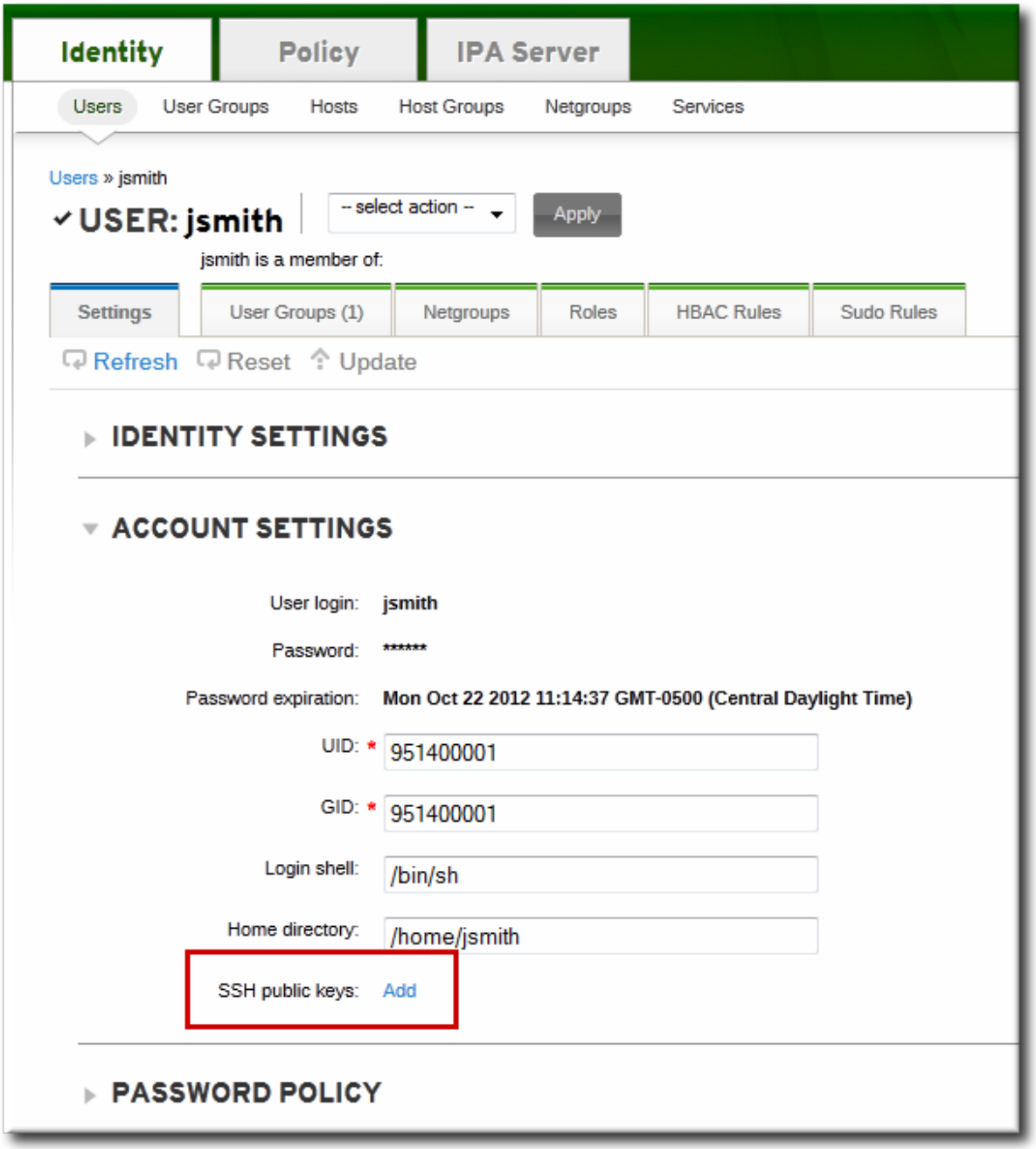
```
[jsmith@server ~]$ cat  /home/jsmith/.ssh/id_rsa.pub

ssh-rsa AAAAB3NzaC1yc2E...tJG1PK2Mq++wQ== jsmith@example.com
```

3. **Identity** タブを開き、サブタブの **ユーザー** を選択します。

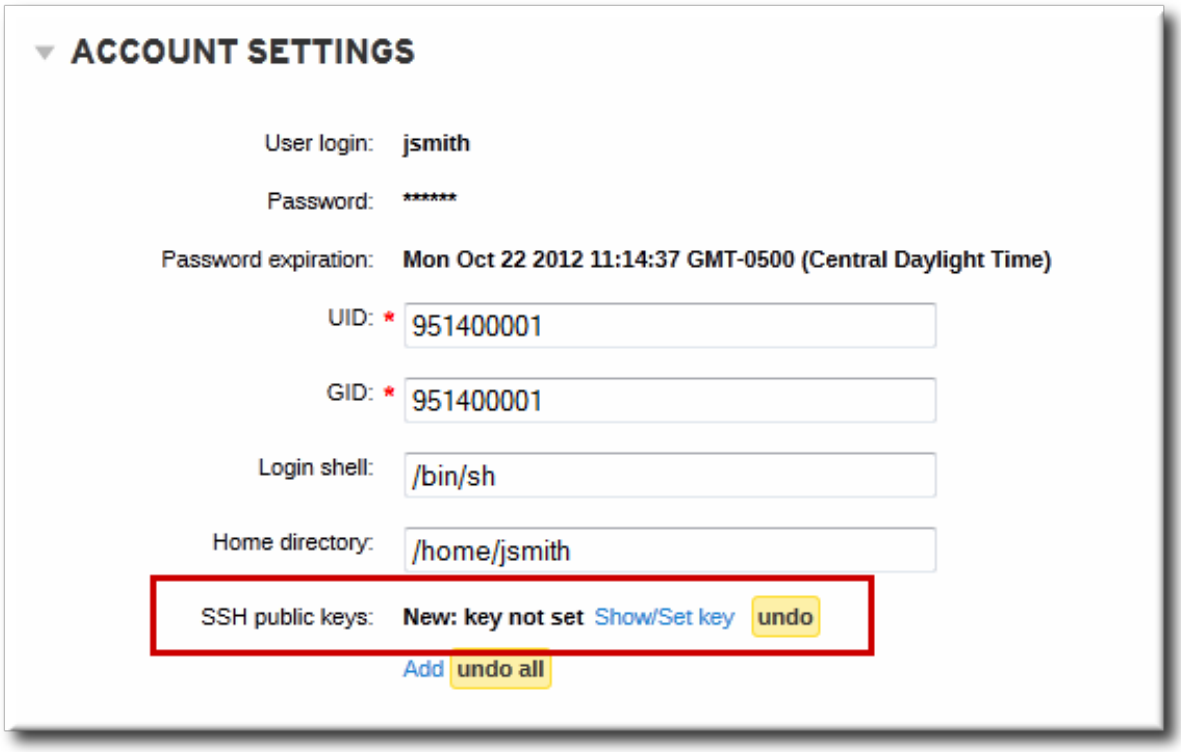
4. 編集するユーザー名をクリックします。



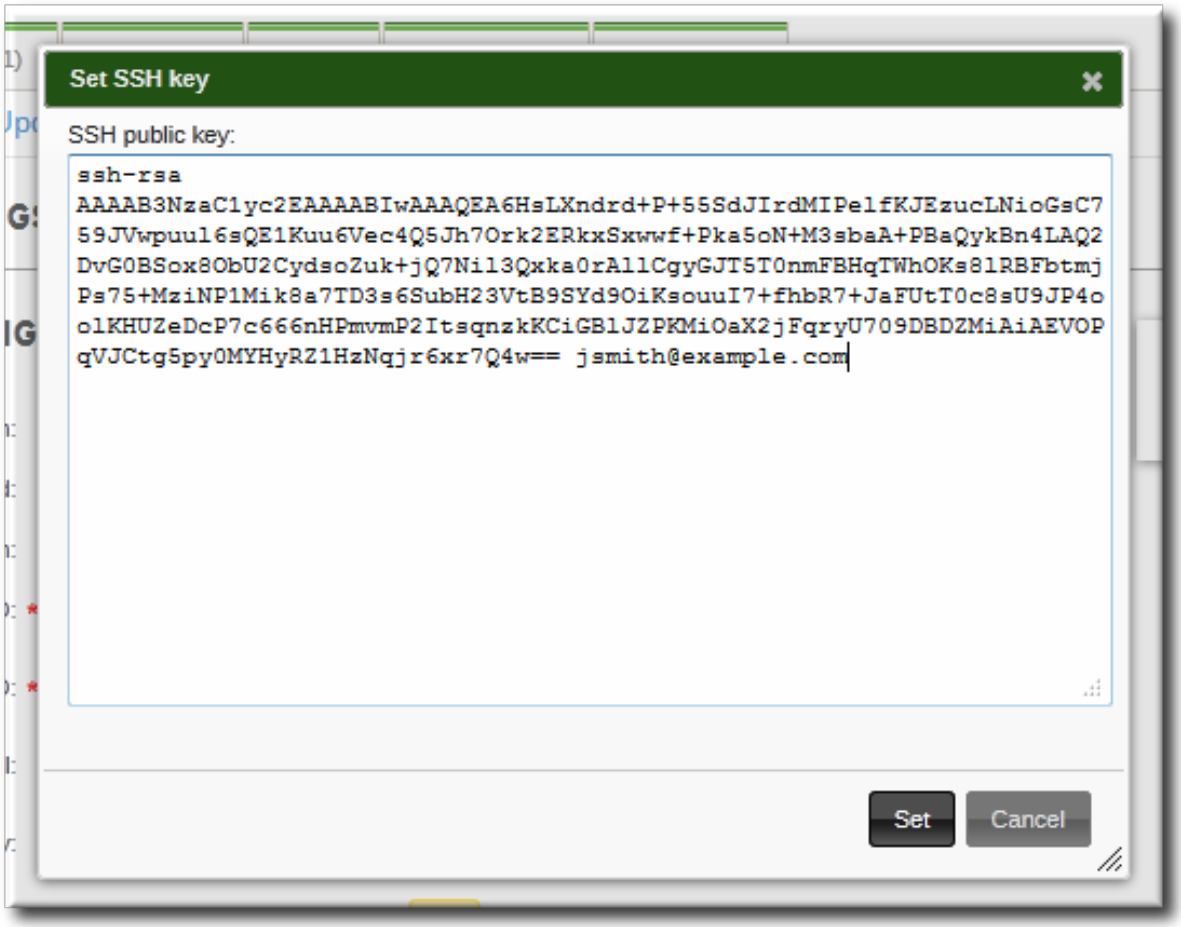
5. **Settings** タブの **Account Settings** エリアで **SSH public keys: Add** リンクをクリックします。



6. **SSH public keys** の横にある **Add** リンクをクリックします。



7. ユーザーの公開鍵に貼り付けて、**Set** ボタンをクリックします。



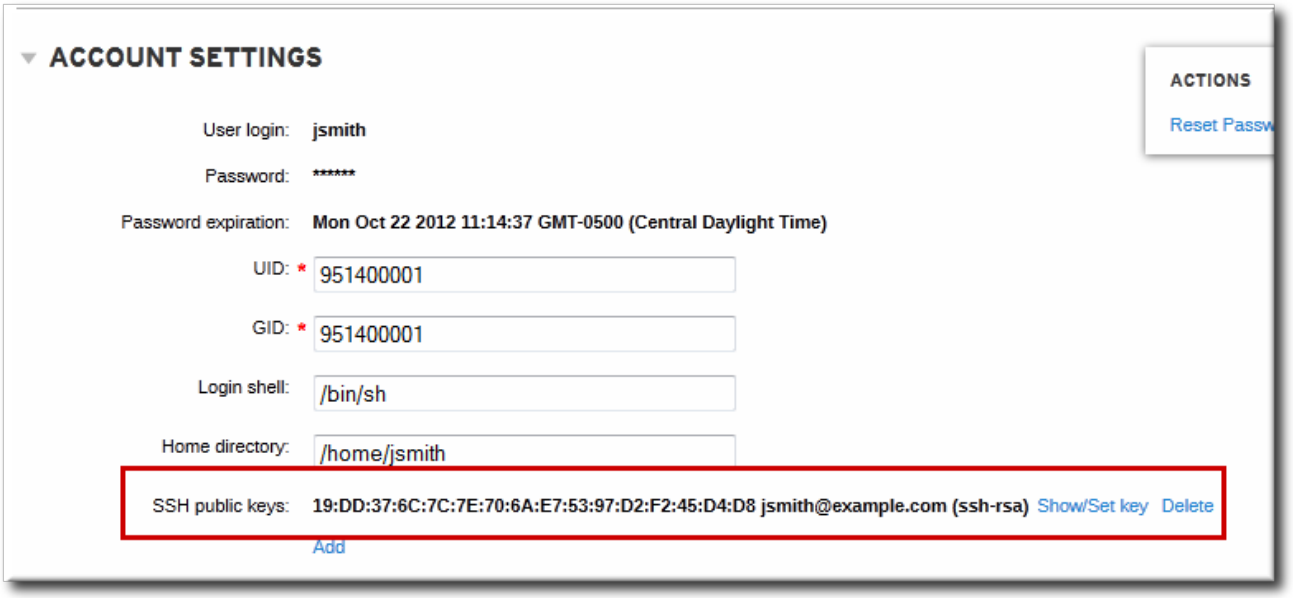
SSH public keys フィールドに **New: key set** と表示されるようになります。**Show/Set キー** のリンクをクリックすると、送信したキーが表示されます。

8. 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。

9. すべてのキーが送信されたら、ユーザーページ上部の **Update** ボタン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].

図9.1 保存された公開鍵



ユーザーキーをアップロードしたら、Identity Management を ID ドメインの 1 つとして使用するように SSSD を設定し、OpenSSH がユーザーキー管理に SSSD を使用するように設定します。これは、『デプロイメントガイド』で説明されています。

9.3.3. コマンドラインでのユーザーの SSH 鍵のアップロード

--sshpubkey オプションは、64 ビットエンコードの公開鍵をユーザーエントリーにアップロードします。たとえば、以下のようになります。

```
[jsmith@server ~]$ ipa user-mod jsmith --sshpubkey="ssh-rsa 12345abcde= ipaclient.example.com"
```

実際のキーでは、キーはこの例よりも長く、通常は末尾が等号 (=) になります。

複数のキーをアップロードするには、**--sshpubkey** オプション 1 つでコンマ区切りのキー一覧を指定します。

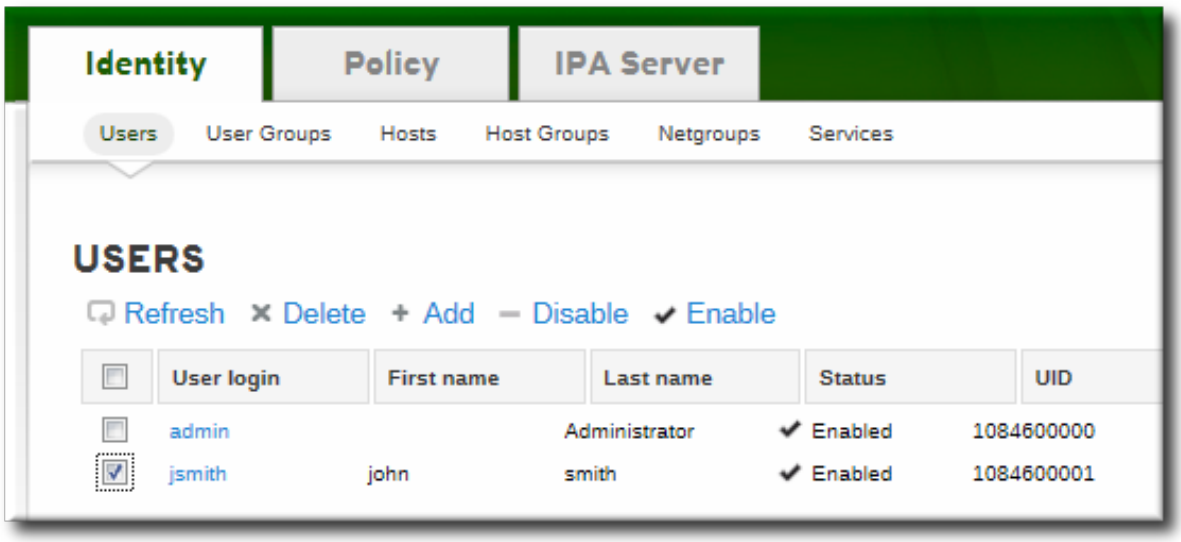
```
--sshpubkey="12345abcde==,key2==,key3=="
```

ユーザーキーをアップロードしたら、Identity Management を ID ドメインの 1 つとして使用するように SSSD を設定し、OpenSSH がユーザーキー管理に SSSD を使用するように設定します。これは、『Red Hat Enterprise Linux デプロイメントガイド』で説明しています。

9.3.4. ユーザーキーの削除

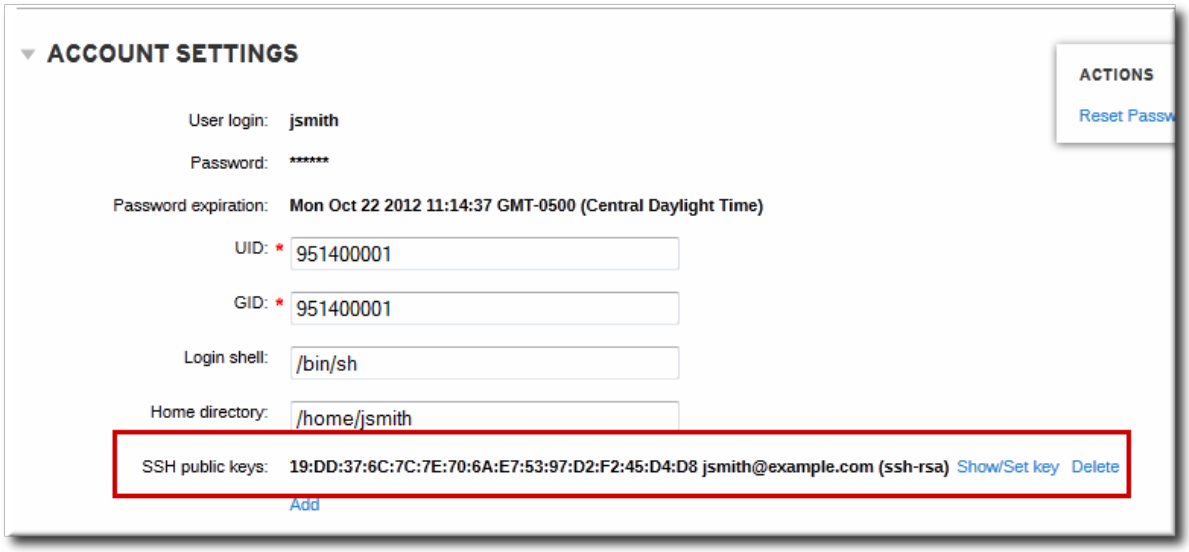
1. **Identity** タブを開き、サブタブの **ユーザー** を選択します。

2. 編集するユーザー名をクリックします。



3. **Settings** タブの **Account Settings** エリアを開きます。

4. 削除するキーのフィンガープリントの横にある **Delete** のリンクをクリックします。



5. 変更を保存するには、ユーザーページの上部にある **Update** リンクをクリックします。

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 user-mod** を実行します。これで、対象ユーザーの公開鍵が **すべて** 削除されます。たとえば、以下のようになります。

```
[jsmith@server ~]$ kinit admin
[jsmith@server ~]$ ipa user-mod --sshpubkey= jsmith
```

9.4. パスワードの変更

パスワードの変更操作には、パスワードポリシー (19章 *ポリシー: パスワードポリシーの定義*) や最小限のアクセス制限を適用できます。

- 管理者権限のない通常ユーザーは、個人パスワードのみを変更でき、すべてのパスワードは IdM パスワードポリシーの制限が適用されます。

こうすることで、最終的なパスワードのセキュリティーを確保しつつ、管理者は初期パスワードを作成するか、簡単にパスワードをリセットできます。管理者がユーザーに送信したパスワードは一時的なものであるため、セキュリティーリスクはほぼありません。

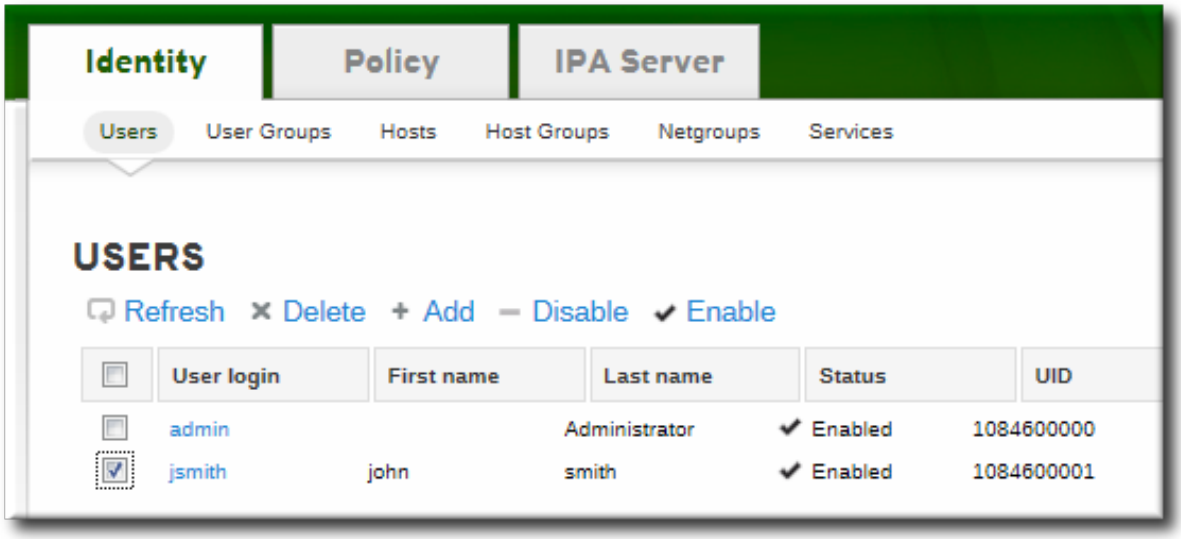
- IdM の管理ユーザーとしてパスワードを変更すると、IdM パスワードポリシーは上書きされますが、パスワードの有効期限はすぐに切れます。そのため、ユーザーは次回のログイン時にパスワードを変更する必要があります。同様に、パスワードの変更権限があるユーザーは、パスワードを変更でき、パスワードポリシーは適用されませんが、別のユーザーは次回のログイン時にパスワードをリセットする必要があります。

- **LDAP ツールを使用して** LDAP Directory Manager ユーザーとしてパスワードを変更すると、IdM パスワードポリシーがすべて上書きされます。

9.4.1. Web UI での操作

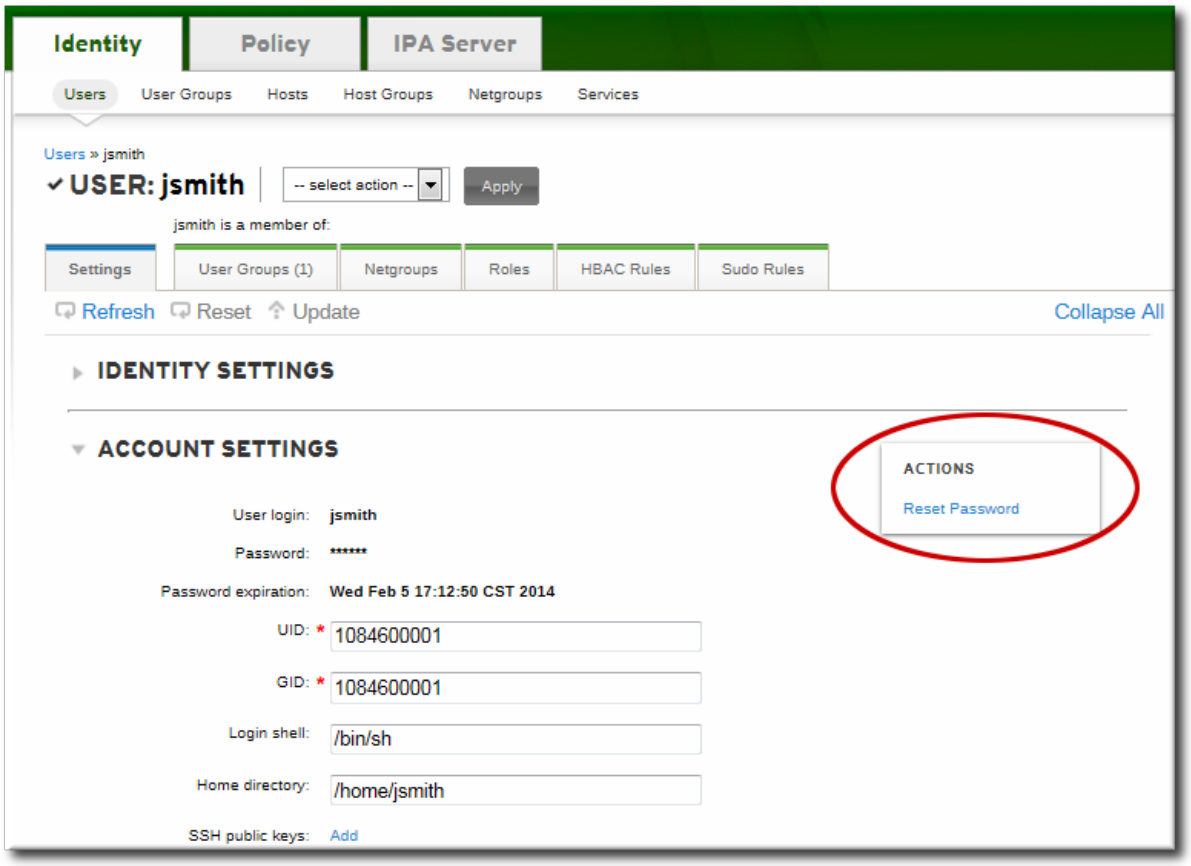
1. **Identity** タブを開き、サブタブの **ユーザー** を選択します。

2. パスワードをリセットするユーザーの名前をクリックします。すべてのユーザーは自分のパスワードを変更できますが、管理者または、権限を委譲されたユーザーのみが、他のユーザーのパスワードを変更できます。

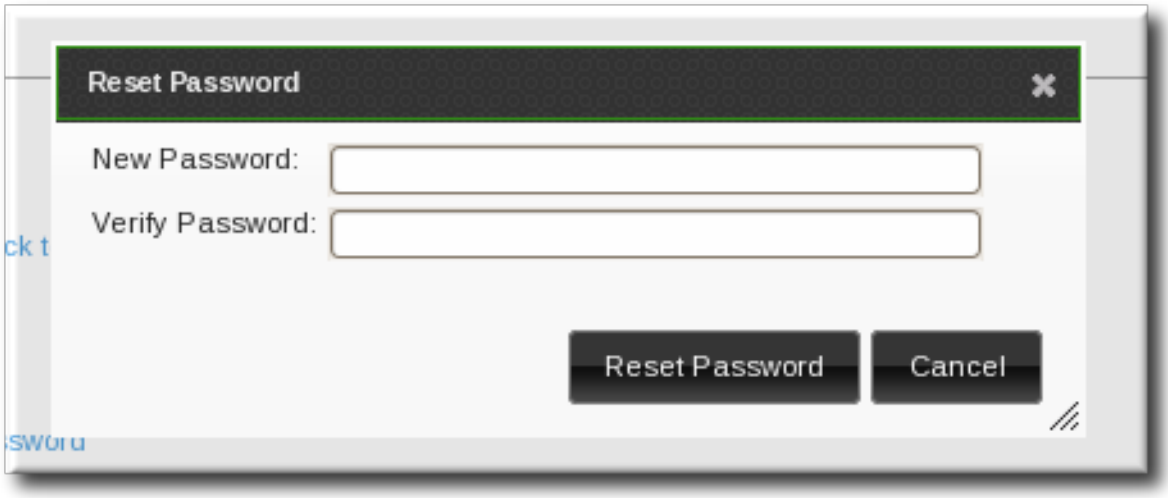


3. **Account Settings** エリアまでスクロールします。

4. **Reset Password** のリンクをクリックします。



5. ポップアップボックスで、新しいパスワードを入力して確認します。



9.4.2. コマンドラインでの操作

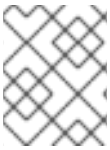
パスワード (ユーザーまたは別のユーザー) の変更は、その他のユーザーアカウントの変更と同様に **user-mod** コマンドを使用します。

```
[bjensen@ipaserver ~]$ kinit admin
[bjensen@ipaserver ~]$ ipa user-mod jsmith --password
```

9.5. ユーザーアカウントの有効化、無効化

ユーザーアカウントは非アクティブにしたり、**無効** にしたりできます。無効にしたユーザーは、IdM ま

たは IdM 関連サービス (Kerberos など) にログインできないため、タスクを実行できません。ただし、このユーザーアカウントはそのまま Identity Management に残り、関連の情報はすべて変更されません。



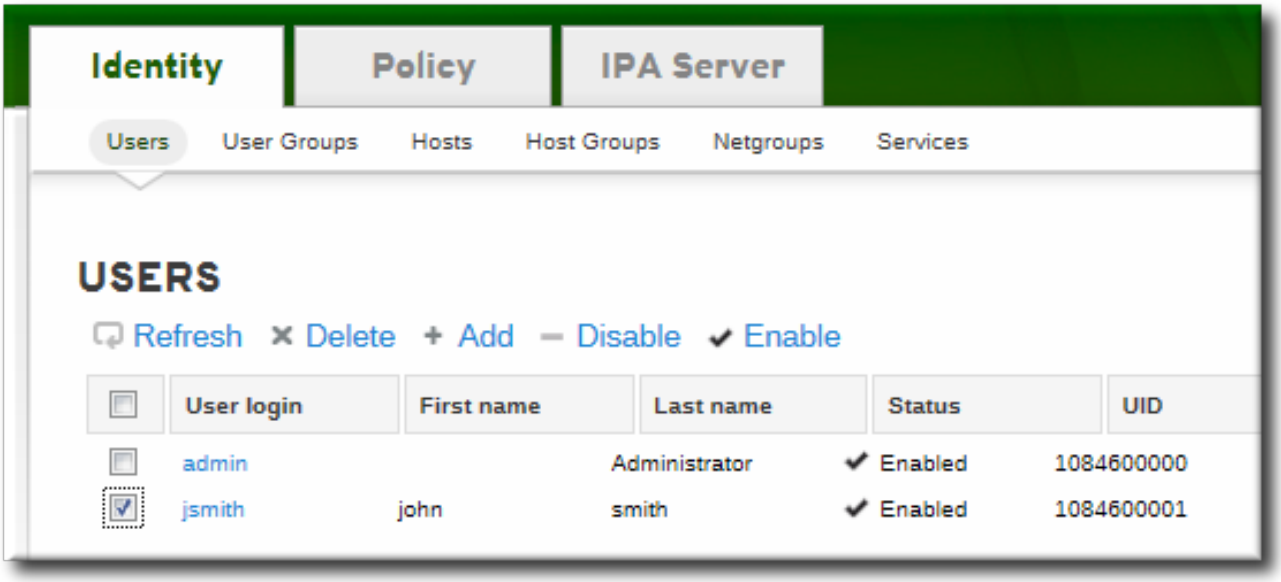
注記

既存の接続は、Kerberos TGT およびその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、ユーザーはチケットを更新できません。

9.5.1. Web UI での操作

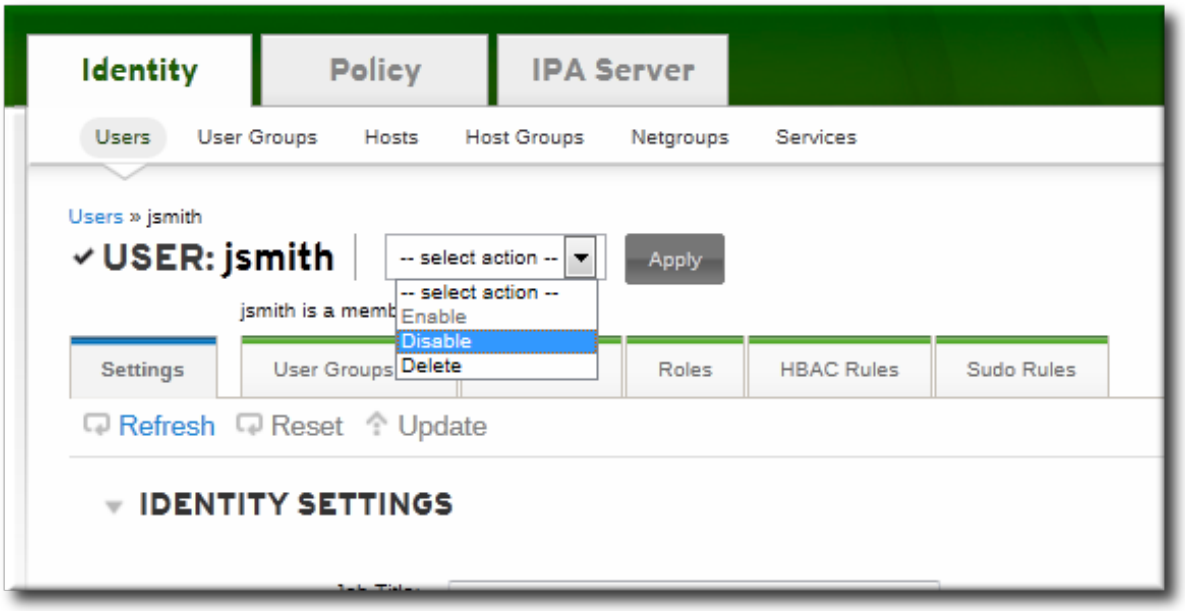
対象のユーザーの横にあるチェックボックスを選択し、リストの上部にある **Disable** リンクをクリックすることで、全ユーザーリストから複数のユーザーを無効にできます。。

図9.2 ユーザー一覧の上部でのオプションの無効化/有効化



ユーザーアカウントは、ユーザーの個別エントリーページからも無効にできます。

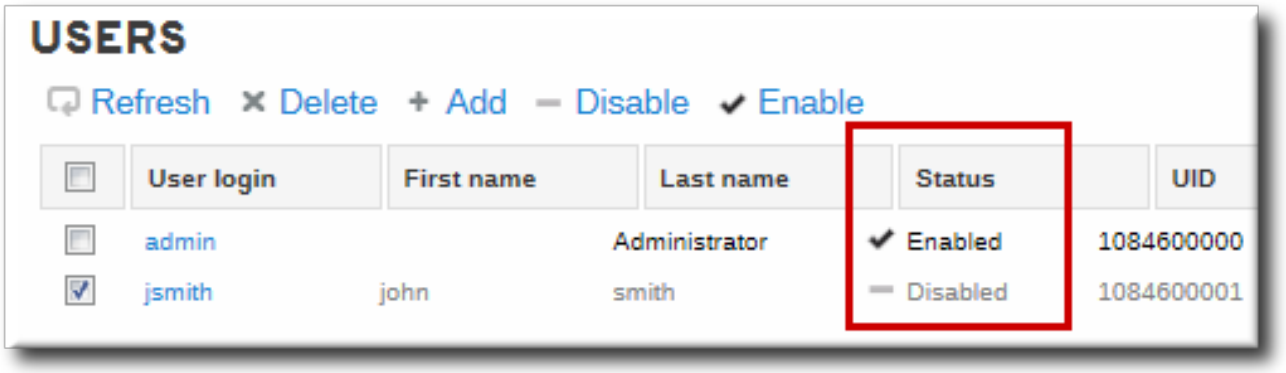
1. **Identity** タブを開き、サブタブの **ユーザー** を選択します。
2. ユーザー名をクリックして、非アクティブまたはアクティブにします。
3. アクションのドロップダウンメニューで、**Disable** を選択します。



4. **Accept** ボタンをクリックします。

ユーザーアカウントが無効になっている場合には、ユーザー一覧のユーザーステータスと、エントリーページのユーザー名の横に、マイナス (-) アイコンで示されます。また、ユーザーの文字は (非アクティブであることを示すため) 黒ではなくグレーになります。

図9.3 ユーザーステータスのアイコンの無効化



9.5.2. コマンドラインでの操作

user-enable および **user-disable** コマンドを使用してユーザーを有効化/無効化します。ユーザーがログインするだけで実行できます。以下に例を示します。

```
[bjensen@server ~]$ ipa user-disable jsmith
```

9.6. ログイン失敗後のユーザーアカウントのロック解除

ユーザーのログイン試行時に一定の回数、誤ってパスワードを入力すると、そのユーザーアカウントはロックされます。アカウントをロックするまでの失敗試行回数およびロックアウトの期間は、パスワードポリシー内で定義します (「アカウントロックアウトポリシーの設定」)。

パスワードポリシーでは、リセット期間を暗黙的に定義して、一定期間後にアカウントのロックが解除されるようにできます。ただし、リセット期間が比較的長い場合や、デプロイメントに強力なセキュリティーチェックをしてからアカウントのロックを解除する必要がある場合など、管理者はアカウントのロックを手作業で解除できます。

user-unlock コマンドを使用して、アカウントのロックを解除します。たとえば、以下のようになります。

```
[bjensen@ipaserver ~]$ kinit admin
[bjensen@ipaserver ~]$ ipa user-unlock jsmith
```

9.7. スマートカード

パスワードの代わりに、スマートカードに基づいた認証を使用できます。ユーザーの認証情報がスマートカードに格納され、特別なソフトウェアやハードウェアを使用して、その情報にアクセスします。この方法で認証するには、ユーザーはリーダーにスマートカードを設置してから、そのスマートカードの PIN コードを提示する必要があります。

Red Hat Enterprise Linux 6 クライアントは、SSSD が実行されており、Red Hat Enterprise Linux 7.3 以降をベースした Identity Management サーバーに登録されている場合は、ローカルのスマートカード認証を使用できます。

9.7.1. Identity Management でのスマートカードおよびスマートカードリーダーのサポート

お使いのスマートカードが coolkey パッケージでサポートされている場合には、このパッケージのインストール後に、必要な PKCS #11 モジュールがすでに中央の **/etc/pki/nssdb/** の NSS データベースに配置されています。

スマートカードに対応していない場合は、以下の手順を実行します。

1. **modutil** ユーティリティーを使用して、必要な PKCS #11 モジュールを手動で追加します。たとえば、以下のようになります。

   ```
   [root@ipaclient ~]# modutil -dbdir /etc/pki/nssdb/ -add "My PKCS#11 module" -libfile
   libmypkcs11.so
   ...
   Module "My PKCS#11 Module" added to database.
   ```

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

2. NSS データベースに、スマートカードで証明書を検証する必要のある認証局 (CA) の証明書をすべて追加します。たとえば、**ca_certificate.pem** ファイルの CA 証明書を NSS データベースに追加するには、次のコマンドを実行します。

   ```
   [root@ipaclient ~]# certutil -A -d /etc/pki/nssdb/ -n 'CA certificate' -t CT,C,C -a -i
   ca_certificate.pem
   ```

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

9.7.2. スマートカードからの証明書のエクスポート

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。

   ```
   [user@ipaclient ~]$ certutil -L -d /etc/pki/nssdb/ -h all
   Certificate Nickname         Trust Attributes
   ```

```
                    SSL,S/MIME,JAR/XPI

my_certificate          CT,C,C
```

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。

3. 証明書を Base64 形式で **user.crt** に抽出するには、前のステップのニックネームを使用します。

```
[user@ipaclient ~]$ 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。

9.7.3. IdM ユーザーのスマートカード証明書の保存

ユーザーのスマートカード証明書を保存するには、Red Hat Enterprise Linux 7 サーバーに証明書を追加します。『『Linux ドメイン ID、認証、およびポリシーガイド』』 の「外部 CA で発行された証明書の管理」を参照してください。

9.7.4. Identity Management クライアントでのスマートカード認証

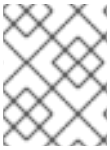
Red Hat Identity Management (IdM) は、以下のスマートカードベースの認証オプション 2 つに対応しています。

ローカル認証

- テキストコンソール
- Gnome Display Manager (GDM) などのグラフィカルコンソール
- **su**, または **sudo** などのローカル認証サービス

ssh でのリモート認証

スマートカードの証明書は、PIN で保護される SSH の秘密鍵と合わせて保存されます。



注記

IdM では、スマートカード認証用に上記のローカル認証サービスと **ssh** のみをサポートします。FTP などの他のサービスには対応していません。

SSSD ベースのスマートカード認証が設定されていると、ユーザーがログインを試行すると、システムはスマートカードの PIN コードの入力を求めます。入力した PIN が正しく、スマーどカードの証明書が有効で、ログインを試行しているユーザーが所有しており、他の設定可能な条件が満たされている場合には、ユーザーの認証に成功します。

9.7.4.1. IdM クライアントでのスマートカード認証の設定

クライアントでスマートカードを使用して認証できるようにするには、次の手順を実行します。

1. スマートカードのサポートを有効にするには、SSSD がパスワード、ワンタイムパスワード (OTP)、またはスマートカードの PIN を要求できるようにします。これには、**/etc/pam.d/password-auth** および **/etc/pam.d/system-auth** の PAM 設定ファイルの **auth** の行を変更します。

a. デフォルトの **/etc/pam.d/password-auth** で以下の行を削除します。

```
auth        required      pam_env.so
auth        sufficient    pam_unix.so nullok try_first_pass
auth        requisite     pam_succeed_if.so uid >= 500 quiet
auth        sufficient    pam_sss.so use_first_pass
auth        required      pam_deny.so
```

以下の行に置き換えます。

```
auth        required      pam_env.so
auth        [default=1 success=ok] pam_localuser.so
auth        [success=done ignore=ignore default=die] pam_unix.so nullok try_first_pass
auth        requisite     pam_succeed_if.so uid >= 500 quiet
auth        sufficient    pam_sss.so forward_pass
auth        required      pam_deny.so
```

b. 同様に、デフォルトの **/etc/pam.d/system-auth** で以下の行を削除し ます。

```
auth        required      pam_env.so
auth        sufficient    pam_unix.so nullok try_first_pass
auth        requisite     pam_succeed_if.so uid >= 500 quiet
auth        sufficient    pam_sss.so use_first_pass
auth        required      pam_deny.so
```

以下の行に置き換えます。

```
auth        required      pam_env.so
auth        [default=1 success=ok] pam_localuser.so
auth        [success=done ignore=ignore default=die] pam_unix.so nullok try_first_pass
auth        requisite     pam_succeed_if.so uid >= 500 quiet
auth        sufficient    pam_sss.so forward_pass
auth        required      pam_deny.so
```

2. **/etc/sssd/sssd.conf** の以下のオプションを **true** に設定します。

```
[pam]
pam_cert_auth=true
```

3. SSSD を再起動します。

```
[root@ipaclient ~]# systemctl restart sssd
```

9.7.4.2. スマートカードを使用した SSH ログイン

スマートカード認証を使用しており、**ssh** でログインする場合には、スマートカードリーダーモジュールに以下のパスも指定する必要があります。たとえば、以下のようになります。

```
$ ssh -I /usr/lib/libmypkcs11.so -l user@example.com host.example.com
Enter PIN for 'Smart Card':
```

9.8. ユーザープライベートグループの管理

Red Hat Enterprise Linux システムでは、ユーザーを作成するたびに、その新規ユーザーが唯一のメンバーとして所属するシークレットユーザーグループが自動的に作成されます。これは、**ユーザープライベートグループ** です。**umask** のデフォルト設定では、グループアクセスの制限はなく、ユーザーアクセスだけに制限を課すので、ユーザープライベートグループを使用すると、ファイルおよびディレクトリーのパーミッションの管理が簡単で安全になります。

IdM ドメインに新しいユーザーが作成されると、Red Hat Enterprise Linux の規則に従って、対応するプライベートグループで作成されます。多くの環境では、これはデフォルトで許容範囲内の動作ですが、プライベートグループを必要としないユーザーまたはユーザータイプがある場合や環境にすでに、NIS グループまたは他のシステムグループに GID が割り当てられている場合があります[3]。

9.8.1. ユーザープライベートグループの表示

ユーザープライベートグループは 1 ユーザーに固有となっており、システムでのみ使用されます。このグループはプライベートであるため、IdM UI では表示されません。ただし、ユーザー作成時のオプションによっては、全ユーザーにプライベートグループが設定されているわけではないので、IdM ユーザードメイン内で設定されたプライベートグループの一覧を取得すると便利です。プライベートグループは、**group-find** コマンドに **--private** オプションを指定して検索および一覧表示できます。たとえば、以下のようになります。

```
[root@server ~]# ipa group-find --private
---------------
1 group matched
---------------
  Group name: jsmith
  Description: User private group for jsmith
  GID: 1084600001
----------------------------
Number of entries returned 1
----------------------------
```

9.8.2. 特定ユーザーのプライベートグループの無効化

--noprivate オプションを使用してユーザーが作成されると、プライベートグループの作成を無効にできます。

プライベートグループなしでユーザーを追加する場合に、Linux システムには新しいユーザー用の GID のユーザーが必要である点を注意してください。ただし、デフォルトのユーザーグループ (**ipausers**) は、POSIX 以外のグループであるため、GID は関連付けられていません。追加操作は失敗しないため、**--gid** オプションを使用して明示的にユーザー GID を設定するか、GID でグループを作成し、 **自動メンバールール** を使用して (「25章 *ポリシー: 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*」で説明) そのグループにユーザーを追加する必要があります。

```
[jsmith@server ~]$ ipa user-add jsmith --first=John --last=Smith --noprivate --gid 10000
```

9.8.3. グローバルでのプライベートグループの無効化

ユーザープライベートグループは、389 Directory Server の管理対象エントリープラグインにより管理されます。このプラグインを無効にして、全新規ユーザーのプライベートグループの作成を実質的に無効にできます。

これは、**ipa-managed-entries** コマンドを使用して行います。

1. **ipa-managed-entries** コマンドを使用して、利用可能な管理エントリープラグイン定義を一覧表示します。デフォルトでは、新規ユーザー (UPG) に 1 つ、ネットグループに 1 つ (NGP) の合計 2 つあります。

```
[root@ipaserver ~]# ipa-managed-entries --list -p DMpassword
Available Managed Entry Definitions:
UPG Definition
NGP Definition
```

2. 任意の管理対象エントリープラグインインスタンスを無効にします。以下に例を示します。

```
[root@ipaserver ~]# ipa-managed-entries -e "UPG Definition" -p DMpassword disable
Disabling Plugin
```

3. 389 Directory Server を再起動して、新しいプラグイン設定を読み込みます。

```
[root@ipaserver ~]# service dirsrv restart
```

管理対象エントリープラグインインスタンスは、**enable** オプションを使用して再度有効にできます。

9.9. 一意の UID および GID 番号の割り当て管理

IdM サーバーは、無作為に UID および GID の値を生成し、同時にレプリカが同じ UID または GID 値を生成しないようにする必要があります。1 つの組織に異なる複数のドメインがある場合は、IdM ドメイン全体で UID および GID の数字が一意になる必要があります。

9.9.1. ID 数値の範囲の概要

UID および GID 番号は **範囲** に分けられます。個別のサーバーとレプリカでそれぞれの数的範囲を維持することで、サーバーまたはレプリカで発行された数字が別のサーバーまたはレプリカで発行された数字と重複する可能性を最小限に抑えられます。範囲は、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、DNA (Dynamic Numeric Assignment) プラグインを使用してサーバーとレプリカの間で更新および共有されます。同じ範囲がユーザー ID (***uidNumber***) およびグループ ID (***gidNumber***) にも使用されます。ユーザーとグループで同じ ID が指定される可能性がありますが、ID は異なる属性に設定されているので、競合はありません。ユーザーとグループの両方に同じ ID 番号を使用することで、管理者はユーザープライベートグループを設定できます。この場合に、各ユーザーに一意のシステムグループが作成され、ユーザーとグループ両方に同じ ID 番号が使用されます。

UID または GID の番号を指定せずに、または対話的にユーザーを作成すると、サーバーまたはレプリカの範囲で次に利用可能な ID 番号で、ユーザーアカウントが作成されます。つまり、UID 番号およびプライベートグループ (設定されている場合) には一意の番号が常に割り当てられることになります。

重要

数値がユーザーエントリーに **手動で** 割り当てられると、サーバーでは ***uidNumber*** が一意であるかどうかは検証されません。ID を重複させることができます。POSIX エントリーでは、想定されている動作 (非推奨) です。グループエントリーの場合も同様で、重複した ***gidNumber*** を手動でエントリーに割り当てることができます。

2 つのエントリーに同じ ID 番号が割り当てられている場合に、検索では、対象の ID 番号の最初のエントリーだけが返されます。ただし、他の属性の検索時や、 **ipa user-find --all** を使用時には、両方のエントリーが返されます。

9.9.2. インストール中の ID 範囲の割り当ての概要

IdM 管理者はまず、サーバーのインストール時に、**--idstart** および **--idmax** オプションを指定して **ipa-server-install** を使用し、範囲を定義できます。これらのオプションは必須ではないので、インストール時に設定スクリプトで範囲を無作為に割り当てることができます。

最初の IdM サーバーのインストール時に範囲が設定されていないと、20 万の ID 範囲がランダムに選択されます。使用可能な範囲は 1 万個あります。この数からランダムな範囲を選択すると、今後 2 つの別の IdM ドメインがマージされた場合でも競合する可能性が低くなります。

IdM サーバーが 1 台の場合には、範囲内で ID が順番にエントリーに割り当てられます。レプリカの場合には、初期サーバーの ID 範囲は分割され、分散されます。

レプリカのインストール時に、無効な範囲を使用して設定されます。また、有効な範囲を要求可能な場所をレプリカに指示するディレクトリーエントリーもあります (これは複数のレプリカの間で共有されます)。レプリカが起動するか、現在の範囲内に利用可能な ID が 100 未満になると、利用可能なサーバーの 1 つに、新たな範囲を割り当てるように問い合わせできます。特別な拡張操作を使用して、範囲を 2 つに分割し、元のサーバーとレプリカのそれぞれで、利用可能な範囲を半分ずつに割り当てます。

9.9.3. 競合する ID 範囲に関する注記

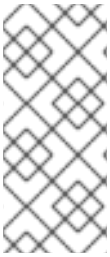
管理者は、**sssd.conf** ファイル内の **min_id** および **max_id** オプションを使用して ID 番号の範囲を定義できます。デフォルトの **min_id** 値は **1** です。ただし、Red Hat は、システム用に予約されている UID と GID 番号との競合を避けるため、この値を **1000** に設定することを推奨しています。

9.9.4. 新しい範囲の追加

ドメイン全体の範囲がゼロに近づいてきた場合には、新しい範囲を手動で選択してマスターサーバーの 1 つに割り当てることができます。その後、すべてのレプリカは必要に応じてマスターから ID 範囲を要求します。

範囲を変更するには、389 Directory Server 設定を編集して DNA プラグインインスタンスを変更します。範囲は ***dnaNextRange*** パラメーターで定義します。以下に例を示します。

```
ldapmodify -x -D "cn=Directory Manager" -W -h server.example.com -p 389
Enter LDAP Password: *******
dn: cn=POSIX IDs,cn=Distributed Numeric Assignment Plugin,cn=plugins,cn=config
changetype: modify
add: dnaNextRange
dnaNextRange: 123400000-123500000
```



注記

このコマンドでは、指定された範囲の値だけを追加し、その範囲内の値が実際に利用できるかどうかは確認しません。このような値を割り当てようとした場合にのみ、このチェックが実行されます。すでに割り当て済みの値をほぼ含む範囲が追加された場合には、システムは、システム全体を循環して、未割り当ての値を検索し、最終的に未割り当ての値が見つからない場合には、失敗します。

9.9.5. 変更された UID および GID 番号の修復

ユーザーを作成すると、ユーザーに、ユーザー ID 番号とグループ ID 番号が自動的に割り当てられます。

ユーザーが IdM システムまたはサービスにログインすると、そのシステム上の SSSD は、関連付けられた UID/GID 番号でそのユーザー名をキャッシュします。次に、UID 番号がユーザーの ID キーとして使用されます。ユーザー名が同じで UID が異なるユーザーがシステムにログインすると、SSSD は名前が競合する 2 つの異なるユーザーとして処理します。

つまり、SSSD は UID 番号の変化を認識しません。SSSD は、異なる UID が指定された既存ユーザーとしてではなく、別の新規ユーザーとして解釈します。既存のユーザーの UID 番号が変更されると、そのユーザーは SSSD 、関連のサービスやドメインにログインできなくなります。また、これは ID 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があり、競合が発生したユーザーは検索されず、これらのアプリケーションにアクセスできなくなります。

重要

Identity Management または SSSD では、UID/GID の変更に対応していません。

何らかの理由で UID/GID 番号が変更された場合は、そのユーザーが再ログインする前に、ユーザーの SSSD キャッシュを消去する必要があります。以下に例を示します。

```
[root@server ~]# sss_cache -u jsmith
```

9.10. ユーザーおよびグループスキーマの管理

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9.1 デフォルトの Identity Management ユーザーオブジェクトクラス

詳細	オブジェクトクラス
IdM オブジェクトクラス	ipaobject
	ipasshuser
人物のオブジェクトクラス	person
	organizationalperson
	inetorgperson
	inetuser
	posixaccount

詳細	オブジェクトクラス
Kerberos のオブジェクトクラス	krbprincipalaux krbticketpolicyaux
Managed エントリー (テンプレート) のオブジェクトクラス	mepOriginEntry

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9.1「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9.2 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
First name	--first	必須
Last name	--last	必須
Full name	--cn	任意
Display name	--displayname	任意
Initials	--initials	デフォルト
Home directory	--homedir	デフォルト
GECOS field	--gecos	デフォルト
Shell	--shell	デフォルト
Kerberos principal	--principal	デフォルト
Email address	--email	任意
Password	--password [b]	任意
User ID number[c]	--uid	デフォルト

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]
Group ID number[c]	--gidnumber	デフォルト
Street address	--street	任意
City	--city	任意
State/Province	--state	任意
Zip code	--postalcode	任意
Telephone number	--phone	任意
Mobile telephone number	--mobile	任意
Pager number	--pager	任意
Fax number	--fax	任意
Organizational unit	--orgunit	任意
Job title	--title	任意
Manager	--manager	任意
Car license	--carlicense	任意
	--noprivate	任意
SSH Keys	--sshpubkey	任意
Additional attributes	--addattr	任意

[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。

[c] UID 番号を指定せずにユーザーを作成すると、ユーザーアカウントには、サーバーまたはレプリカの範囲で次に利用可能な ID 番号が自動的に割り当てられます。(数値の範囲は「一意の UID および GID 番号の割り当て管理」で詳述されています。) つまり、UID 番号およびプライベートグループ (設定されている場合) には一意の番号が常に割り当てられることになります。

数値がユーザーエントリーに **手動で** 割り当てられると、サーバーでは ***uidNumber*** が一意であるかどうかは検証されません。ID を重複させることができます。POSIX エントリーでは、想定されている動作 (非推奨) です。

2 つのエントリーに同じ ID 番号が割り当てられている場合に、検索では、対象の ID 番号の最初のエントリーだけが返されます。ただし、他の属性の検索時や、 **ipa user-find --all** を使用時には、両方のエントリーが返されます。

9.10.1. 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 (「ユーザーおよびグループスキーマの管理」)。

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一覧全体を定義する必要があります。たとえば、企業が従業員の誕生日や就業開始日などの情報を保存するためのカスタムのオブジェクトクラスを作成したとします。管理者は単にカスタムのオブジェクトクラスをリストに追加することはできません。新規オブジェクトクラスに**加えて** 現行のデフォルトのオブジェクトクラス一覧全体を設定する必要があります。設定を更新する際は常に、**既存の**デフォルトのオブジェクトクラスを含める必要があります。これを含めないと現行設定が上書きされ、パフォーマンスに関する重大な問題が発生することになります。

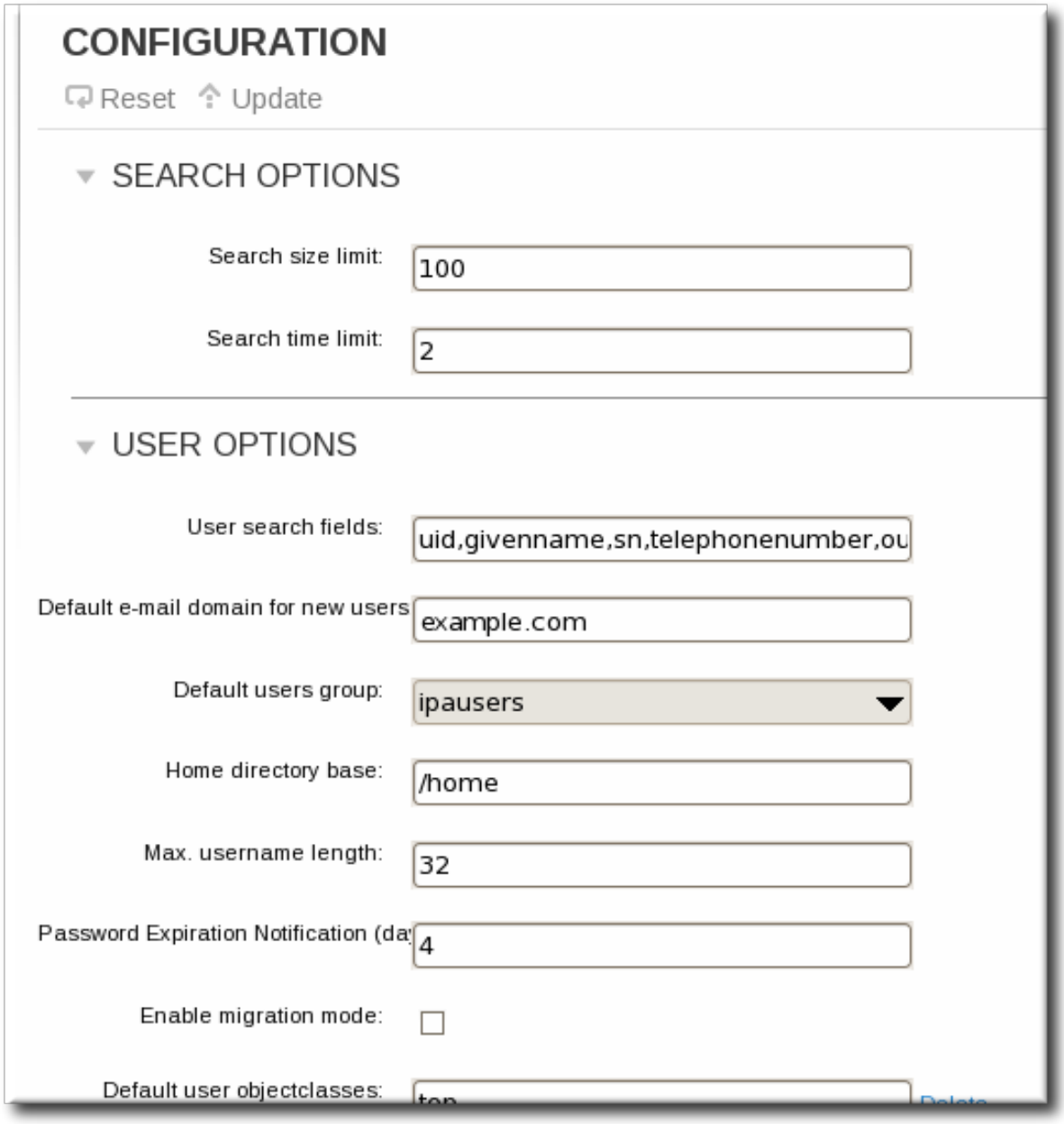
9.10.2. カスタムのオブジェクトクラスを新規ユーザーエントリーに適用する

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。

9.10.2.1. Web UI での操作

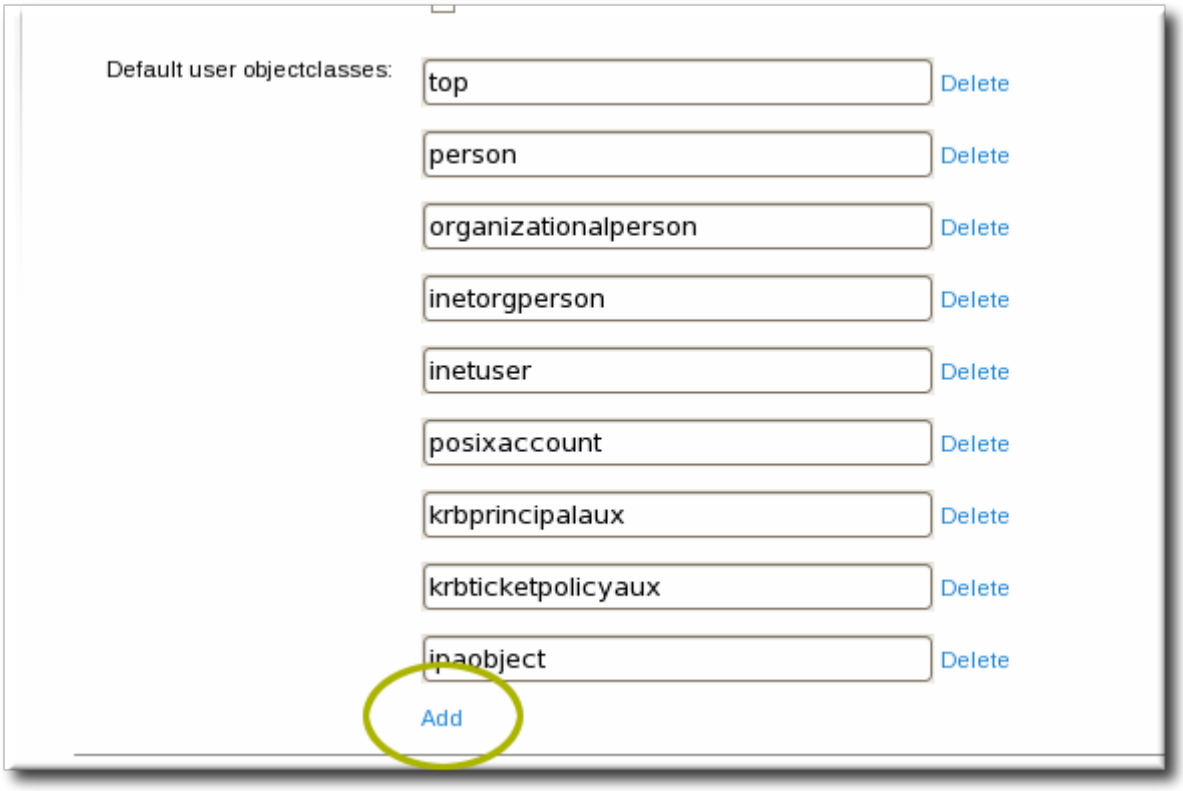
1. カスタムスキーマ要素をすべて、Identity Management が使用する 389 Directory Server インスタンスに追加します。スキーマ要素の追加については、『Directory Server Administrator's Guide』の「スキーマ」の章で説明します。
2. **IPA Server** タブを開きます。
3. **Configuration** サブタブを選択します。
4. **User Options** エリアまでスクロールします。



5. ユーザーエリア下部にある **Add** リンクをクリックして、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。



6. 変更が完了したら、**Configuration** ページ上部の **Update** リンクをクリックします。

9.10.2.2. 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 Administrator's Guide』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一覧に新規オブジェクトクラスを追加します。ユーザーのオブジェクトクラスのオプションは **--userobjectclasses** です。

重要

設定の更新時は、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。

たとえば、以下のようになります。

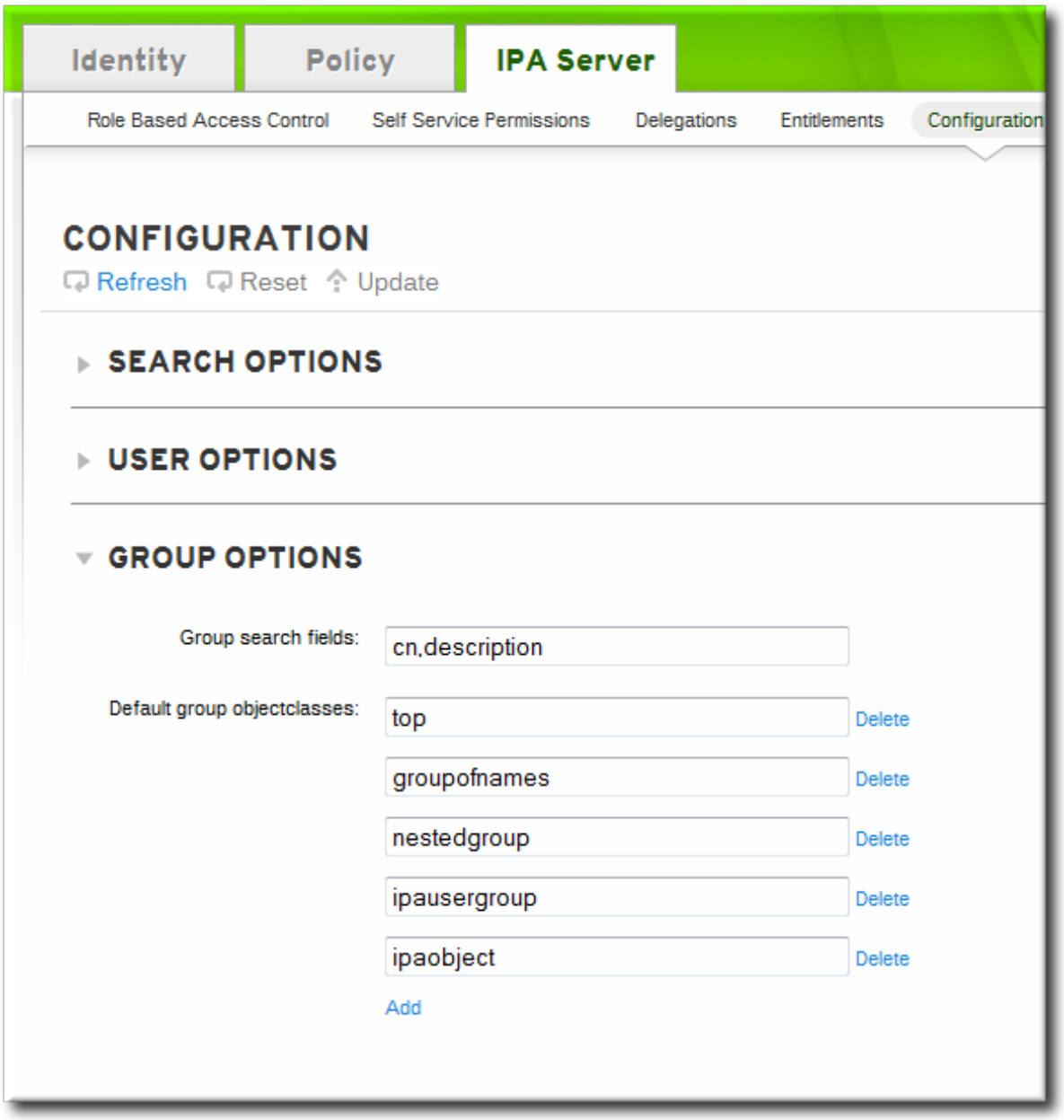
```
[bjensen@server ~]$ ipa config-mod --
userobjectclasses=top,person,organizationalperson,inetorgperson,inetuser,posixaccount,
krbprincipalaux,krbticketpolicyaux,ipaobject,ipasshuser,employeeinfo
```

9.10.3. 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、管理者はグループエントリーに適用する必要のあるカスタム属性を持つカスタムオブジェクトクラスを指定できます。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。

9.10.3.1. 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 Administrator's Guide』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。



5. **Add** リンクをクリックして、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。

6. 変更が完了したら、**Configuration** ページ上部の **Update** リンクをクリックします。

9.10.3.2. 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 Administrator's Guide』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一覧に新規オブジェクトクラスを追加します。グループのオブジェクトクラスのオプションは、**--groupobjectclasses** です。

重要

設定の更新時は、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。

たとえば、以下のようになります。

```
[bjensen@server ~]$ ipa config-mod --
groupobjectclasses=top,groupofnames,nestedgroup,ipausergroup,ipaobject,ipasshuser,empl
oyeegroup
```

9.10.4. 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9.3 デフォルトのユーザーパラメーター

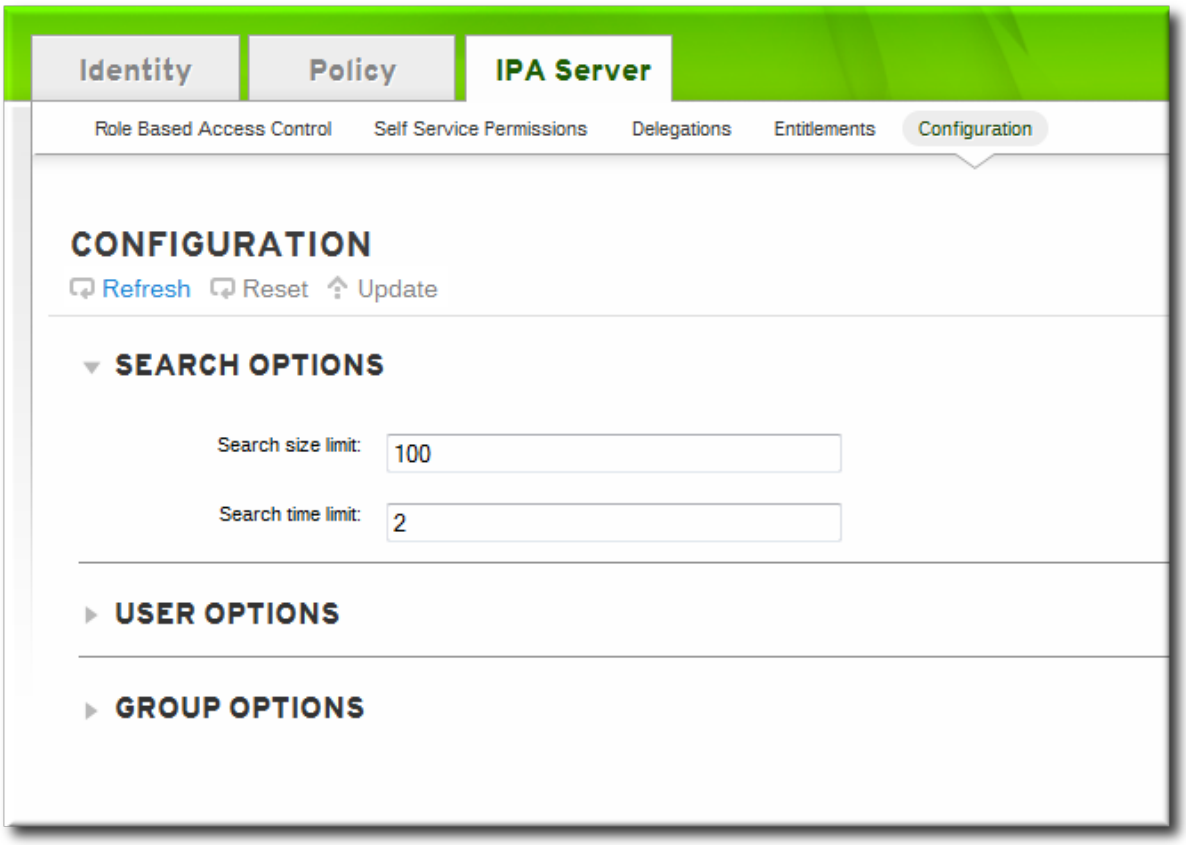
フィールド	コマンドラインオプション	説明
ユーザー名の最大長	--maxusername	ユーザー名の最大長を設定します。デフォルト値は 8 文字です。
Root for home directories	--homedirectory	ユーザーのホームディレクトリーに使用するデフォルトのディレクトリーを設定します。デフォルト値は **/home** です。

フィールド	コマンドラインオプション	説明
Default shell	--defaultshell	ユーザーに使用するデフォルトのシェルを設定します。デフォルト値は **/bin/sh** です。
Default user group	--defaultgroup	新規作成のアカウントを追加するデフォルトグループを設定します。デフォルト値は **ipausers** で、これは IdM サーバーのインストールプロセスで自動的に作成されます。
Default e-mail domain	--emaildomain	新規アカウントに基づいて電子メールアドレスを作成するために使用する電子メールドメインを設定します。デフォルトは IdM サーバードメインです。
Search time limit	--searchtimelimit	サーバー検索結果を返すまでに費やす最長時間を秒単位で設定します。
Search size limit	--searchrecordslimit	返される検索結果の最大数を設定します。
User search fields	--usersearch	検索文字列として使用可能なユーザーエントリー内のフィールドを設定します。記載される属性にはインデックスがその属性のために維持されるので、多く設定しすぎるとサーバーのパフォーマンスに影響が出る場合があります。
Group search fields	--groupsearch	検索文字列として使用可能なグループエントリー内のフィールドを設定します。
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。
Default user object classes	--userobjectclasses	IdM ユーザーアカウントの作成に使用するオブジェクトクラスの一覧を設定します。
Default group object classes	--groupobjectclasses	IdM グループアカウントの作成に使用するオブジェクトクラスの一覧を設定します。

フィールド	コマンドラインオプション	説明
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。

9.10.4.1. 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。



9.10.4.2. 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。

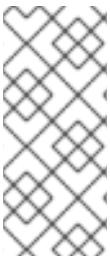
```
[bjensen@server ~]$ kinit admin
[bjensen@server ~]$ ipa config-show --all
dn: cn=ipaConfig,cn=etc,dc=example,dc=com
Maximum username length: 32
```

```
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
```

9.11. ユーザーグループの管理

ユーザーグループは、特にアクセス制御およびパスワードポリシーなどの重要な管理タスクを一元管理する方法の 1 つです。インストール時に、IdM 操作専用のグループが 4 つ作成されます。

- ipausers。全ユーザーが含まれます。
- admins。管理ユーザーが含まれます。初期設定されている **admin** ユーザーはこのグループに属します。
- trusted admins。Active Directory トラストの管理に使用する管理ユーザーが含まれます。
- editors。Web UI で作業するユーザー向けの特別なグループです。このグループは、管理ユーザーの全権限がなくても、他のユーザーのエントリーを **編集** できます。



注記

オペレーティングシステムによっては、システムユーザーに割り当て可能なグループの数が限定される場合があります。たとえば、Solaris システムおよび AIX システムでは、各ユーザーに指定できるグループ数は 16 個までとなっています。これは、ネストされたグループを使用しており、ユーザーが自動的に複数のグループに追加される場合に問題になる可能性があります。

9.11.1. IdM のグループの種類

Identity Management のすべてのグループは基本的に **静的** であるため、グループのメンバーは、手作業で明示的にグループに追加されます。IdM では、グループが他のグループに所属する **ネストされたグループ** を暗黙的に許可します。この場合には、グループに含まれるメンバーはすべて自動的に親グループにも所属します。

自動メンバールールを使用すると、ユーザーエントリーの属性を使用して、ユーザーが属するグループを判断して、新しいユーザーをグループに自動的に追加できます。自動メンバールールは、「25章 *ポリシー: 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*」で説明されてい ます。

IdM のグループの定義方法はシンプルですが、グループにはさまざまな設定オプションがあり、どのようなメンバーを追加できるかを変更できます。

IdM のグループタイプには、メンバーの追加方法ではなく、メンバーが最初に追加された場所をもとにしているものもあります。

- 内部グループ (デフォルト): すべてのメンバーが IdM ドメインに所属します。
- 外部グループ。一部またはすべてのメンバーが IdM ドメイン外の ID ストアに存在します。外部グループには、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のいずれかを指定できます。

もう 1 つの違いは、POSIX 属性でグループが作成されるかどうかです。ほとんどの Linux ユーザーには、さまざまな POSIX 属性が必要ですが、Active Directory または Samba と対話するグループは、POSIX 以外でなければなりません。デフォルトでは、IdM は POSIX 以外のグループを作成しますが、POSIX グループを作成する (**posixgroup** オブジェクトクラスを追加) 明示的なオプションがあります。

グループの作成は簡単なので、作成するグループや、整理する方法を非常に柔軟に決定できます。グループは、部署、物理的な場所などの組織部門や、アクセス制御に関する IdM またはインフラストラクチャーの使用ガイドラインをもとに定義できます。

9.11.2. グループオブジェクトクラス

グループエントリーが作成されると、自動的に特定の LDAP オブジェクトクラスが割り当てられます。(LDAP オブジェクトクラスおよび属性については、『『Directory Server Deployment Guide』』および『『Directory ServerSchema Reference』』を参照してください。) 実際には、グループで重要な属性は名前と説明の 2 つのみです。

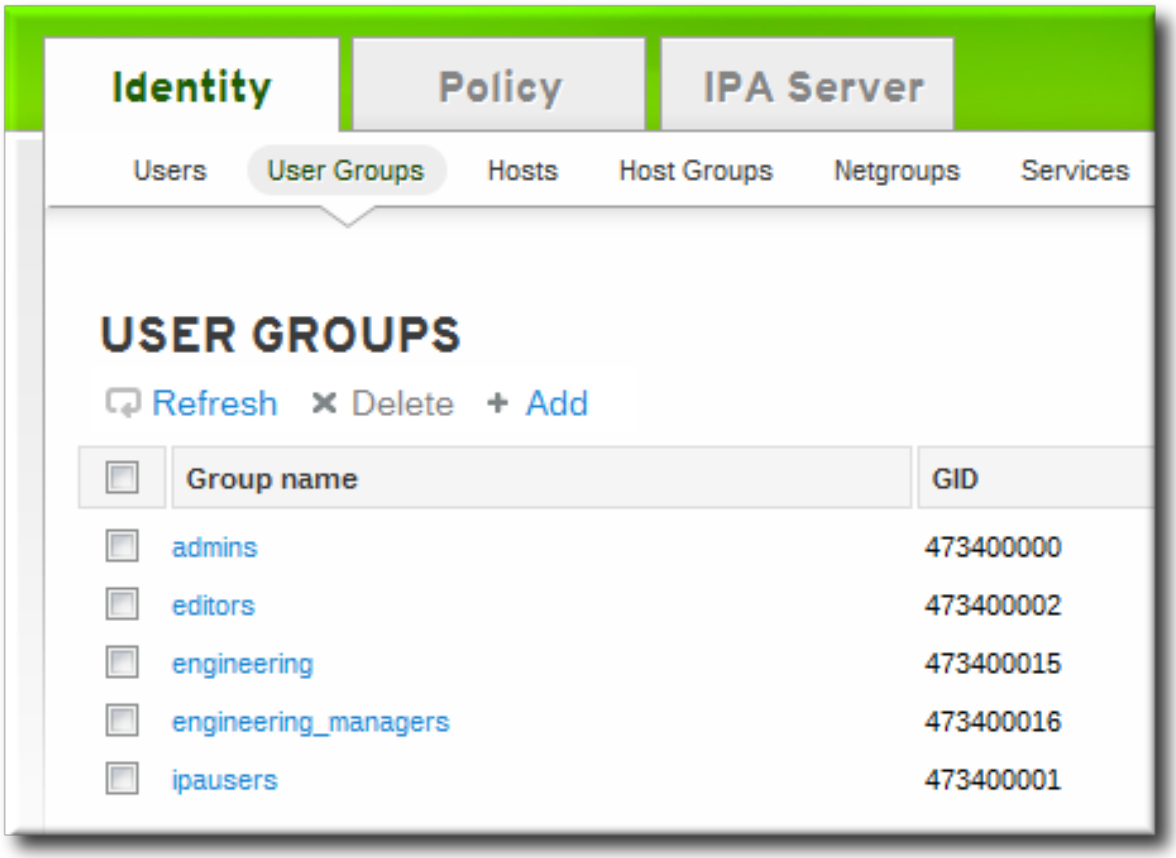
表9.4 デフォルトの Identity Management グループオブジェクトクラス

詳細	オブジェクトクラス
IdM オブジェクトクラス	ipaobject ipausergroup nestedgroup
グループオブジェクトクラス	groupofnames

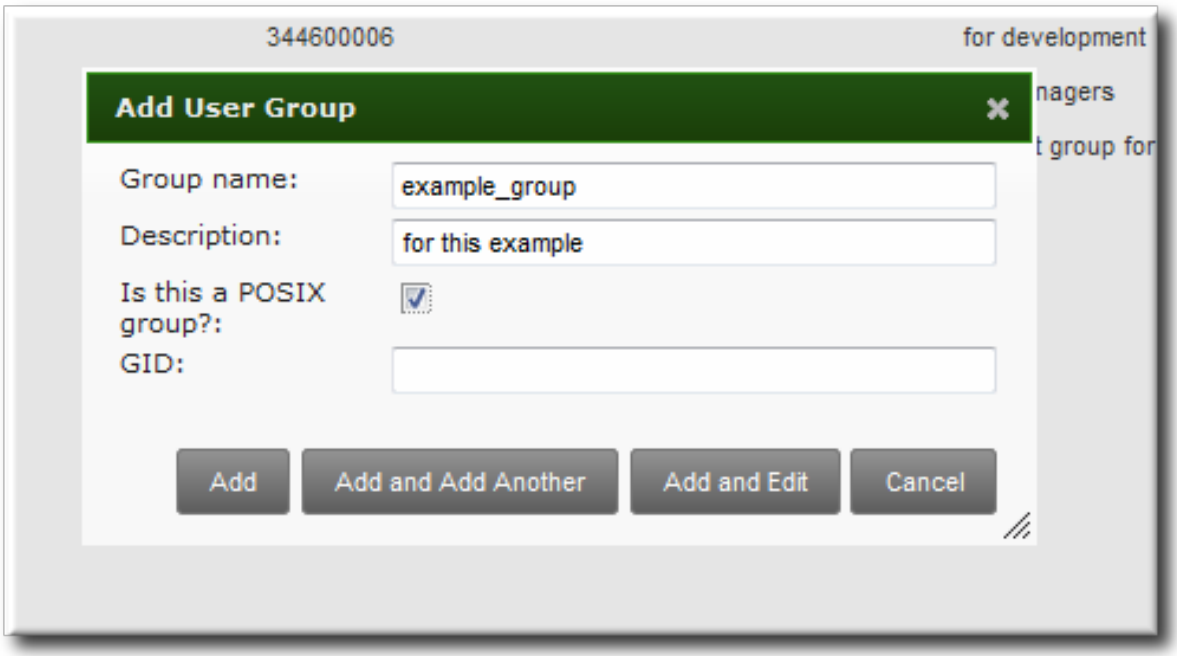
9.11.2.1. ユーザーグループの作成

9.11.2.1.1. Web UI の使用

1. **Identity** タブを開き、サブタブの **User Groups** を選択します。
2. グループ一覧上部にある **Add** をクリックします。



3. グループの全情報を入力します。



- 一意な名前。これは、IdM ドメインのグループに使用される ID で、作成後に変更できません。この名前にはスペースを含めることはできませんが、アンダースコア (_) のような他の区切り文字は使用できます。
- グループの文字での説明。
- グループが POSIX グループかどうか。エントリーに Linux 固有の情報を追加します。デフォルトでは、明示的に設定されない限り、すべてのグループは POSIX グループになります。POSIX 以外のグループを作成して、Windows または Samba との相互運用性を確保で

きます。

- グループの GID 番号 (任意)。すべての POSIX グループには GID 番号が必要ですが、IdM では GID 番号は自動的に割り当てられます。

  競合のリスクがあるため、GID 番号を設定する必要はありません。GID 番号を手動で指定すると、IdM は指定した GID 番号を上書きしません (一意でない場合でも)。

4. **Add and Edit** ボタンをクリックすると、すぐにメンバー選択ページに移動します。

5. 「Web UI (グループページ) の使用」で説明されているようにメンバーを選択します。

9.11.2.1.2. コマンドラインの使用

新規グループは、**group-add** コマンドを使用して作成します。(このコマンドではグループだけが追加され、メンバーは別に追加します。)

グループ名とグループの説明の 2 つの属性が常に必要になります。これらの属性が引数として指定されていない場合には、スクリプトでグループ名と説明を入力するように求められます。

```
[bjensen@server ~]$ ipa group-add groupName --desc="description" [--nonposix]
```

さらに、他にもう 1 つ **--nonposix** という設定オプションがあります。(デフォルトでは、グループはすべて POSIX グループとして作成されます。) Samba などの Windows ユーザーおよびグループおよびプログラムとの相互運用性を確保するため、この **--nonposix** オプションを使用して POSIX 以外のグループを作成できます。このオプションは、スクリプトで **posixGroup** のオブジェクトクラスがエントリーに追加されないように指示します。

たとえば、以下のようになります。

```
[bjensen@server ~]$ ipa group-add examplegroup --desc="for examples" --nonposix

----------------------
Added group "examplegroup"
----------------------
  Group name: examplegroup
  Description: for examples
  GID: 855800010
```

引数を使用しない場合には、このコマンドにより、必要なグループアカウント情報の入力が求められます。

```
[bjensen@server ~]$ ipa group-add
Group name: engineering
Description: for engineers
-------------------------
Added group "engineering"
-------------------------
  Group name: engineering
  Description: for engineers
  GID: 387115842
```

重要

GID 番号を指定せずにグループを作成すると、グループエントリーには、サーバーまたはレプリカ範囲で次に利用可能な ID 番号が割り当てられます。(数値の範囲は「一意の UID および GID 番号の割り当て管理」で詳述されています。) つまり、グループには必ず、一意の GID 番号を割り当てられます。

数値がグループエントリーに **手動で** 割り当てられると、サーバーでは ***gidNumber*** が一意であるかどうかは検証されません。ID を重複させることができます。POSIX エントリーでは、想定されている動作 (非推奨) です。

2 つのエントリーに同じ ID 番号が割り当てられている場合に、検索では、対象の ID 番号の最初のエントリーだけが返されます。ただし、他の属性の検索時や、 **ipa group-find --all** を使用時には、両方のエントリーが返されます。

注記

グループ名は編集できません。グループ名はプライマリーキーであるため、グループ名の変更は、グループを削除して新しいキーを作成する操作と同じです。

9.11.2.2. グループメンバーの追加

9.11.2.2.1. Web UI (グループページ) の使用

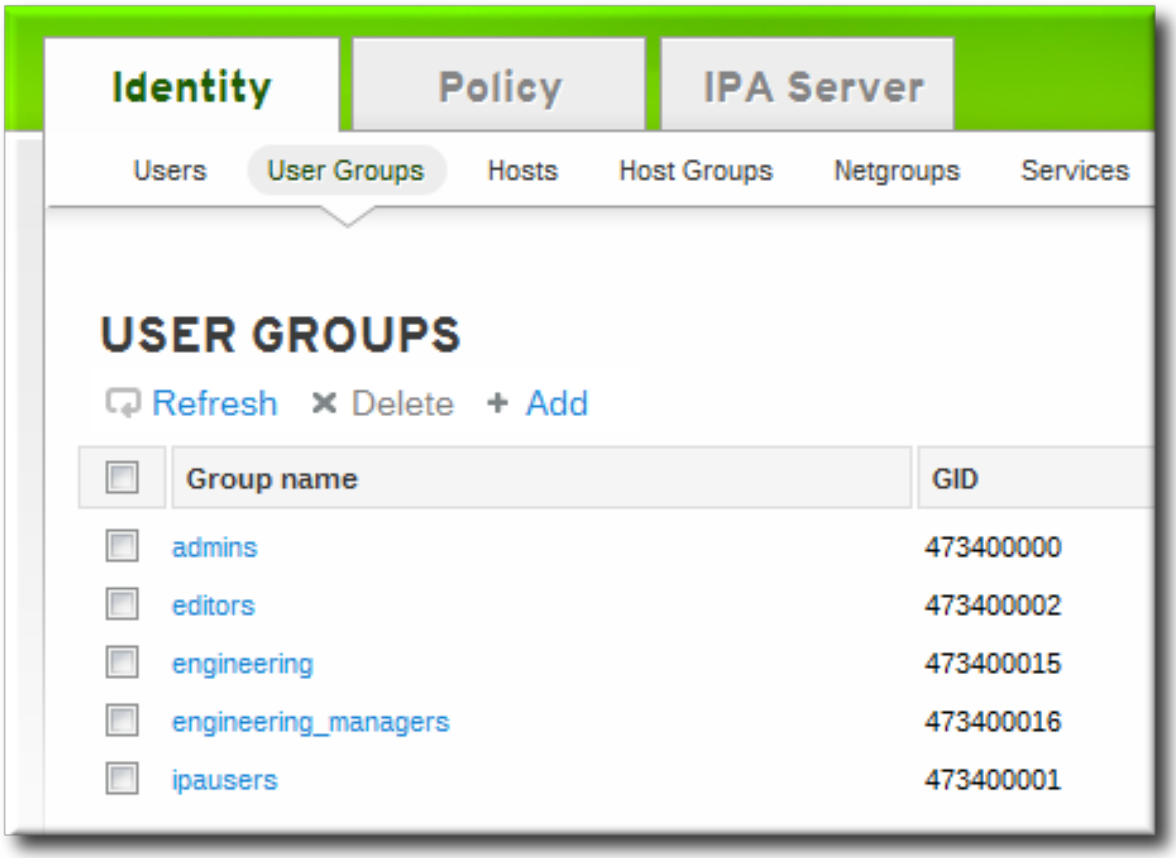
注記

この手順では、グループにユーザーを追加します。ユーザーグループには、他のユーザーグループをメンバーとして追加できます。このようなグループは、**ネスト化されます**。

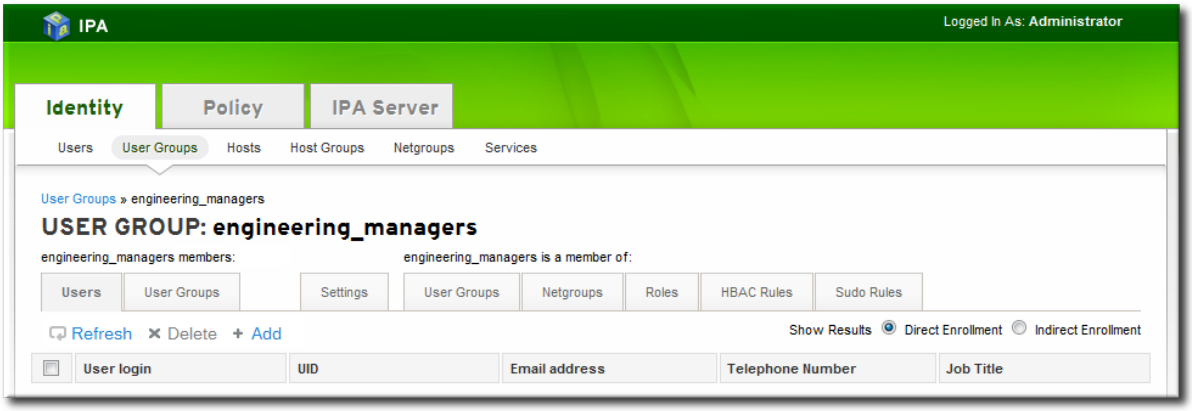
子グループのメンバーが親グループのメンバーとして表示されるまで、最長で数分かかる場合があります。これは特に、ネストされたグループのメンバーが 500 を超える仮想マシンに該当します。

ネスト化されたグループを作成する場合は、**再帰** グループを作成しないようにしてください。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。

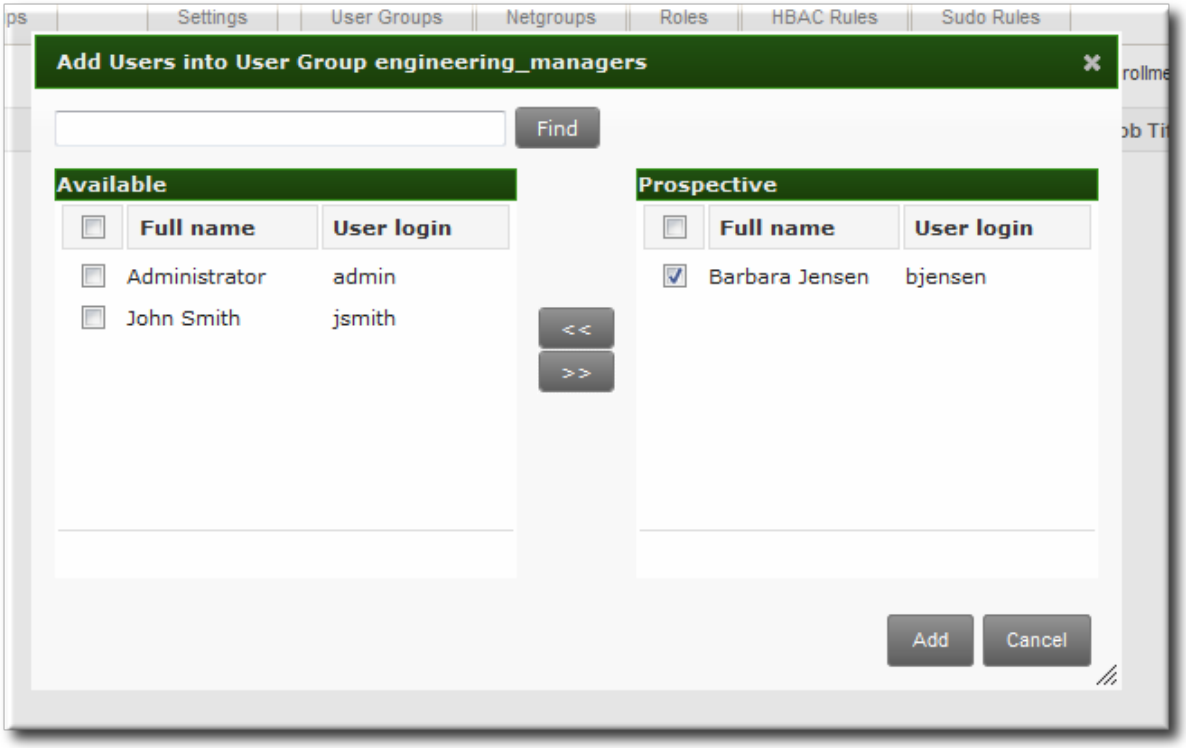
1. **Identity** タブを開き、サブタブの **User Groups** を選択します。
2. メンバーを追加するグループ名をクリックします。



3. タスクエリア上部にある **Add** をクリックします。



4. 追加するユーザーの名前の横にあるチェックボックスをクリックし、右矢印ボタン (**>>**) をクリックし、名前を選択項目のボックスに移動します。



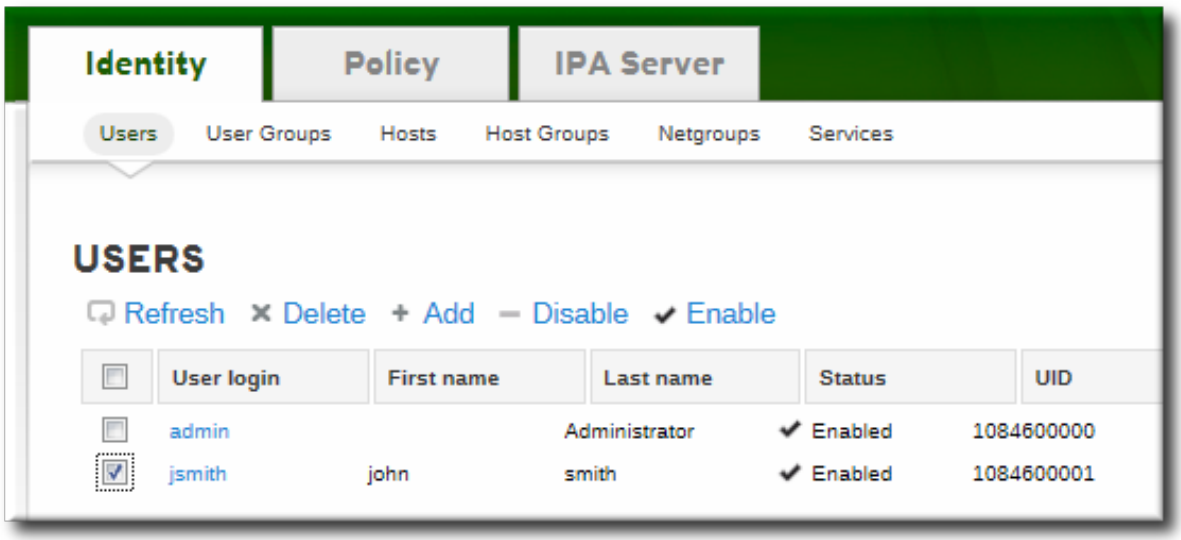
5. **追加** ボタンをクリックします。

グループのメンバーには、ユーザーまたは、他のユーザーグループを指定できます。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、最長で数分かかる場合があります。これは特に、ネストされたグループのメンバーが 500 を超える仮想マシンに該当します。

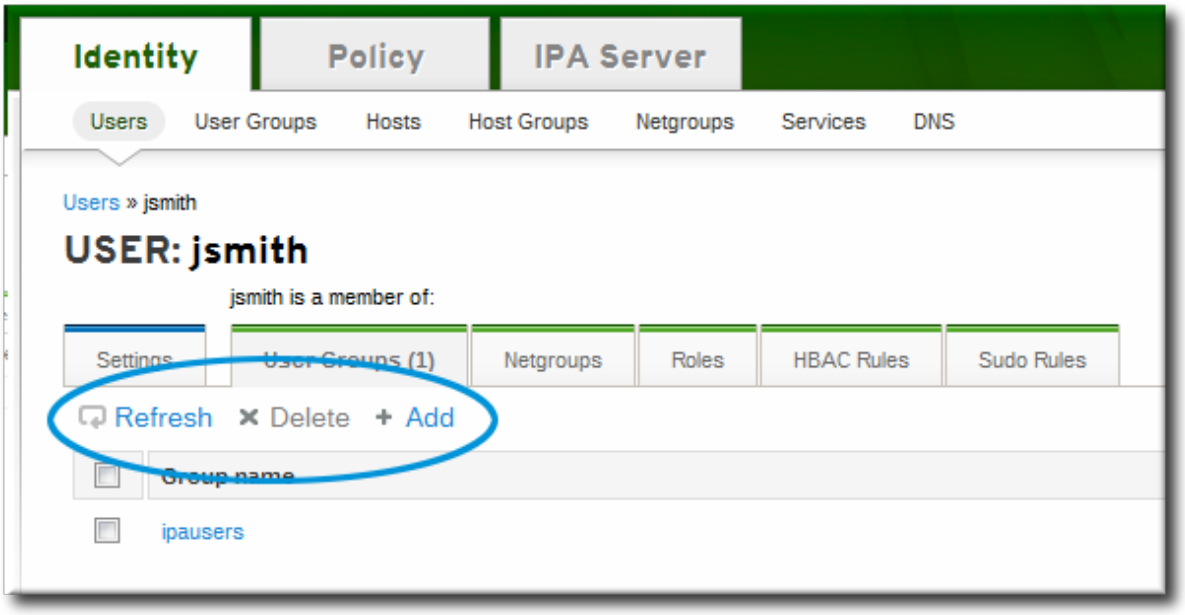
9.11.2.2.2. Web UI (ユーザーページ) の使用

ユーザーは、ユーザーのページからグループに追加することもできます。

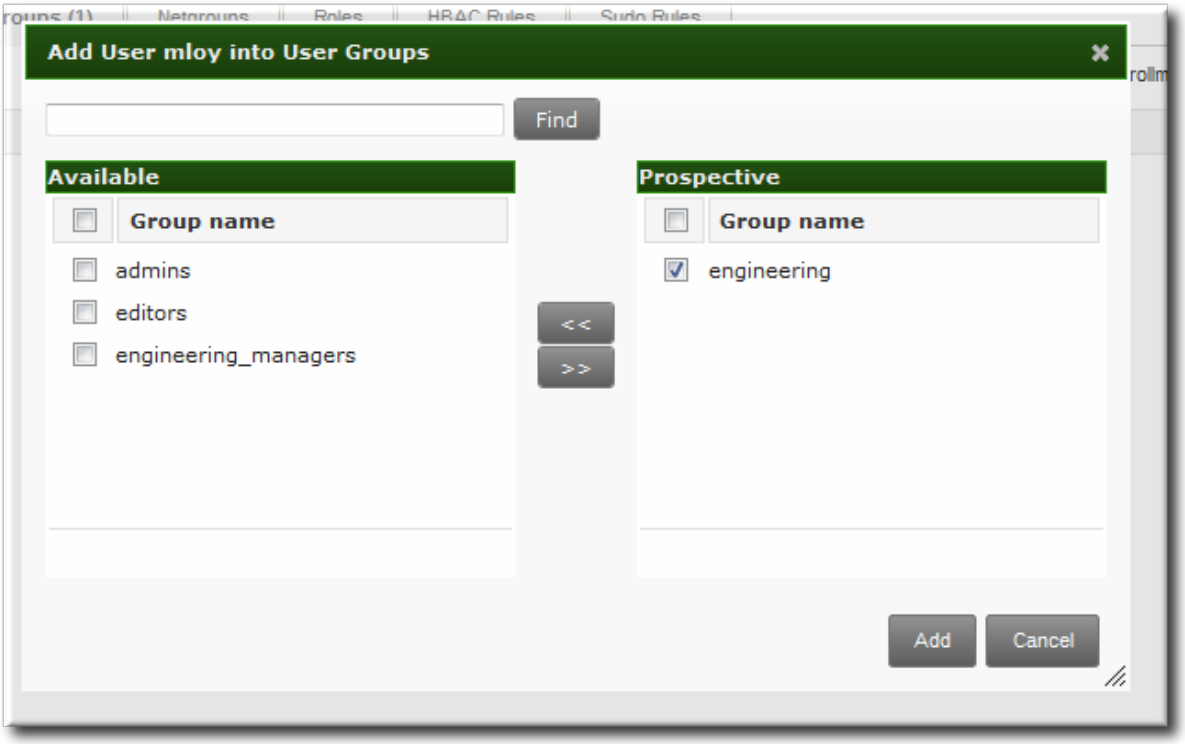
1. **Identity** タブを開き、サブタブの **ユーザー** を選択します。
2. 編集するユーザー名をクリックします。



3. ユーザーエントリーページの **User Groups** タブを開きます。
4. タスクエリア上部にある **Add** をクリックします。



5. 追加するユーザーのグループ名の横にあるチェックボックスをクリックしてから、右矢印ボタン (**>>**) をクリックしグループを選択項目のボックスに移動します。



6. **追加** ボタンをクリックします。

9.11.2.2.3. コマンドラインの使用

group-add-member コマンドを使用して、メンバーをグループに追加します。このコマンドは、両方のユーザーと、他のグループをグループメンバーとして追加できます。

group-add-member コマンドの構文では、グループ名と、追加するユーザーのコンマ区切りリストのみが必要になります。

```
[bjensen@server ~]$ ipa group-add-member groupName [--users=list] [--groups=list]
```

たとえば、以下は、ユーザー 3 つを **engineering** グループに追加します。

```
[bjensen@server ~]$ ipa group-add-member engineering --users=jsmith,bjensen,mreynolds
  Group name: engineering
  Description: for engineers
  GID: 387115842
  Member users: jsmith,bjensen,mreynolds
-------------------------
Number of members added 3
-------------------------
```

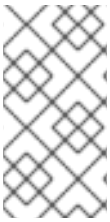
同様に、他のグループをメンバーとして追加して、ネスト化されたグループを作成することもできます。

```
[bjensen@server ~]$ ipa group-add-member engineering --groups=dev,qe1,dev2
  Group name: engineering
  Description: for engineers
  GID: 387115842
  Member groups: dev,qe1,dev2
  -------------------------
  Number of members added 3
  -------------------------
```

ネスト化されたグループを表示すると、メンバーはメンバーとして、メンバーグループのメンバーは間接メンバーとして表示されます。以下に例を示します。

```
[bjensen@server ~]$ ipa group-show examplegroup
  Group name: examplegroup
  Description: for examples
  GID: 93200002
  Member users: jsmith,bjensen,mreynolds
  Member groups: californiausers
  Indirect Member users: sbeckett,acalavicci
```

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、最長で数分かかる場合があります。これは特に、ネストされたグループのメンバーが 500 を超える仮想マシンに該当します。



注記

ネスト化されたグループを作成する場合は、**再帰** グループを作成しないようにしてください。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。

グループメンバーは、**group-remove-member** コマンドを使用して削除します。

```
[bjensen@server ~]$ ipa group-remove-member engineering --users=jsmith

  Group name: engineering
  Description: for engineers
  GID: 855800009
  Member users: bjensen,mreynolds
```

```
---------------------------
Number of members removed 1
---------------------------
```

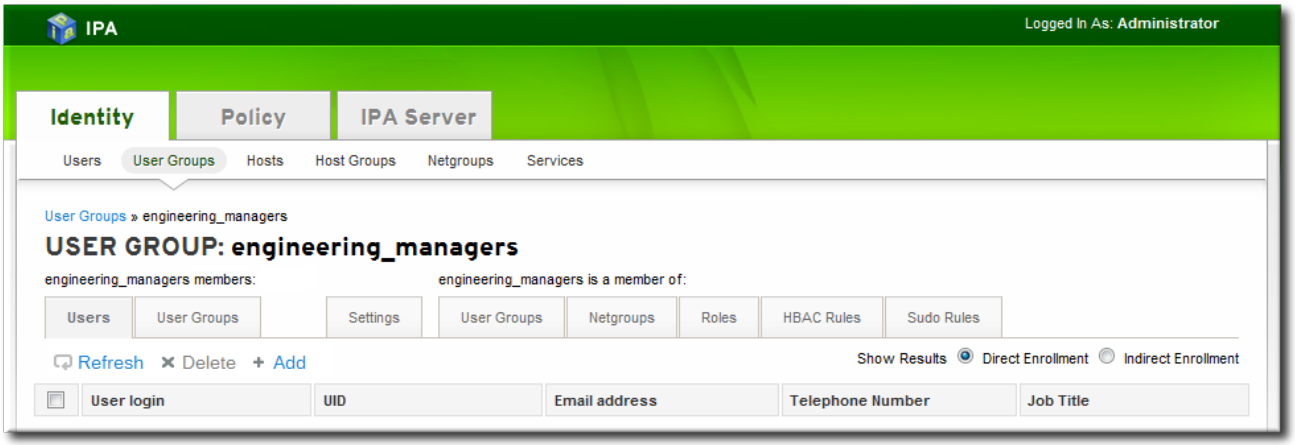
9.11.2.2.4. グループの直接メンバーおよび間接メンバーの表示

ユーザーグループには、他のユーザーグループをメンバーとして追加できます。これは **ネストされたグループ** と呼ばれます。つまり、グループにメンバーが 2 種類含まれます。

- **直接メンバー**: グループに明示的に追加されます。
- **間接メンバー**: 別のユーザーグループのメンバーではあるものの、このユーザーグループがこの対象グループのメンバーであるため、このグループのメンバーとなっています。

IdM の Web UI では、グループの直接メンバーおよび間接メンバーを簡単に表示できます。メンバーリストはメンバータイプでフィルタリングされ、メンバーリストの右上隅にある **Direct** および **Indirect** ラジオボタンを選択して切り替えることができます。

図9.4 グループの直接および間接メンバー



間接メンバーを追跡できるので、メンバーシップを複製せずに、グループメンバーシップを適切に割り当てやすくなります。

9.11.2.3. ユーザーグループの削除

ユーザーグループが削除されると、グループのみが削除されます。グループメンバー (ネスト化されたグループを含む) のユーザーアカウントには影響はありません。また、対象グループに適用されるアクセス制御の委譲も削除されます。

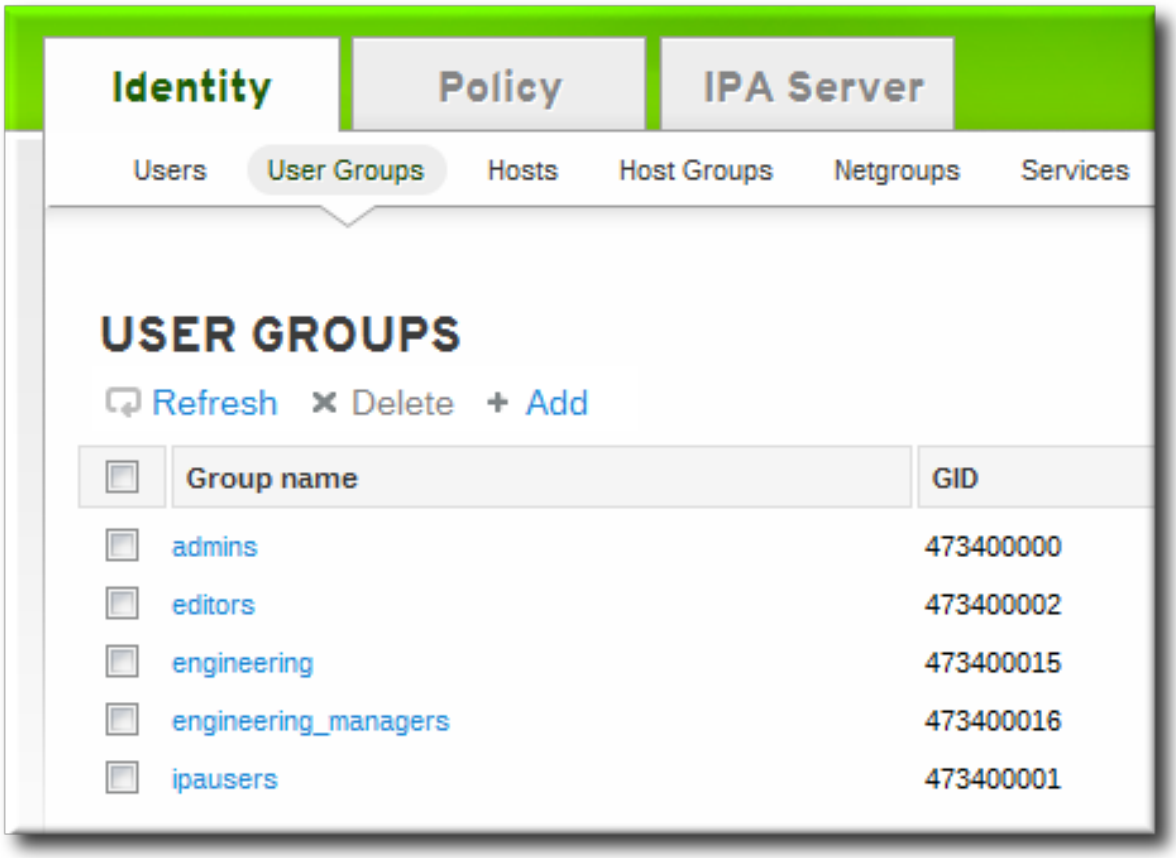
警告

グループすぐに、完全に削除されます。グループ設定 (委譲など) が必要な場合には、別のグループに割り当てるか、新しいグループを作成する必要があります。

9.11.2.3.1. Web UI の使用

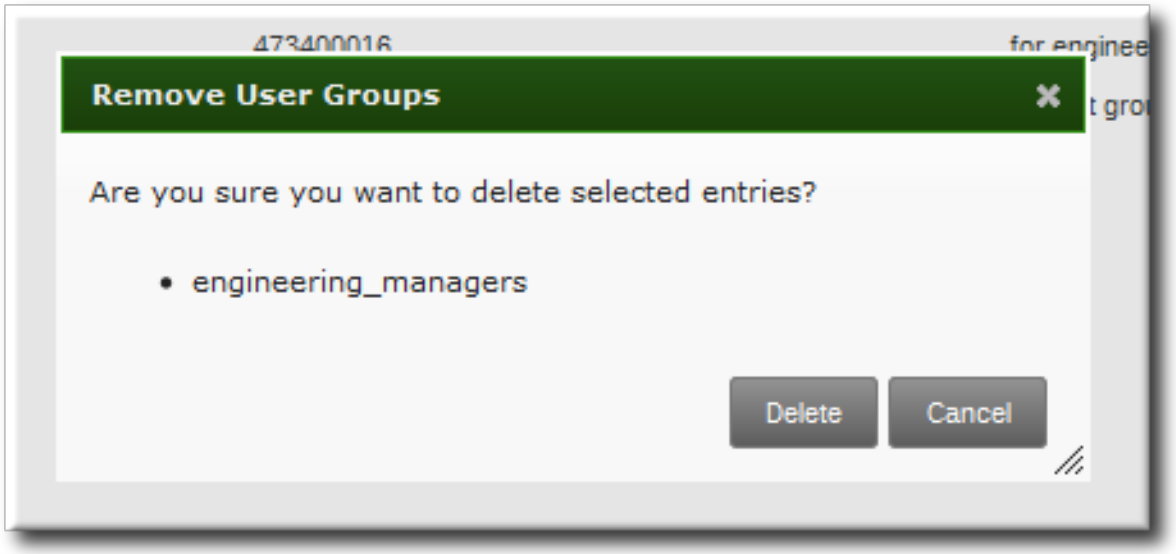
1. **Identity** タブを開き、サブタブの **User Groups** を選択します。

2. 削除するグループ名の横にあるチェックボックスを選択します。



3. タスクエリアの上部にある **Delete** リンクをクリックします。

4. プロンプトが表示されたら、削除を確定します。



9.11.2.3.2. コマンドラインの使用

group-del コマンドは、指定のグループを削除します。たとえば、以下のようになります。

```
[bjensen@server ~]$ ipa group-del examplegroup
```

9.11.3. ユーザーとグループの検索

IdM でのユーザー検索は、単純な文字列 (完全な単語) または部分的な文字列に対して実行できます。「デフォルトのユーザーおよびグループ属性の指定」のように、検索する属性の範囲は、デフォルトの IdM 設定の一部として設定されます。

9.11.3.1. 検索での制限設定

9.11.3.1.1. 検索制限の種類および適用先

検索によっては、多数のエントリーが返される場合があります。すべてのエントリーが返される可能性さえもあります。検索制限では、サーバーが検索に費やす時間と、返されるエントリー数を制限することで、サーバー全体のパフォーマンスが向上します。

検索制限には、検索負荷を低減してサーバーのパフォーマンスを向上させること、返す結果を減らしてユーザビリティーを改善することの 2 つの目的があります。

IdM サーバーでは、検索時にさまざまな制限があります。

- **IdM サーバーの検索制限設定。** これは、IdM サーバー自体の設定で、通常のページを表示するために 全 IdM クライアント、IdM CLI ツール、および IdM Web UI からサーバーに送信されるすべてのリクエストに適用されます。

  デフォルトでは、エントリーの上限は 100 件となっています。

- **IdM サーバーの時間制限設定。** 時間制限は、検索サイズの制限と同様に、IdM サーバーでの検索実行にかける最大時間を設定します。制限に達すると、サーバーは検索を停止し、その時点で返されたすべてのエントリーを返します。

  デフォルトでは、この制限は 2 秒です。

- **ページサイズの制限。** 厳密には検索制限ではありませんが、ページサイズの制限で、ページごとに返されるエントリーの数を制限します。IdM サーバーは、検索のエントリー上限数を返し、次にそれをソートしてページにエントリーを 20 件表示します。ページ結果を使用することで、結果を分かりやすく、見やすくします。

  この数は全検索に対して 20 件までとハードコード化されています。

- **LDAP 検索の制限 (--pkey オプション)。** UI で実行した全検索、**--pkey** オプションを使用した CLI 検索は、IdM サーバー設定に指定されている検索制限を上書きし、基盤の LDAP ディレクトリーに指定されている検索制限を使用します。

  デフォルトでは、エントリーの上限は 2000 件となっています。この値を変更するには、389 Directory Server 設定を編集します。

9.11.3.1.2. IdM 検索制限の設定

検索制限 は、ユーザーまたはグループエントリーのデータベースのクエリー時に返されるレコード数や費やした時間に上限を設定します。検索制限には、時間制限とサイズ (数値) 制限の 2 つのタイプがあります。

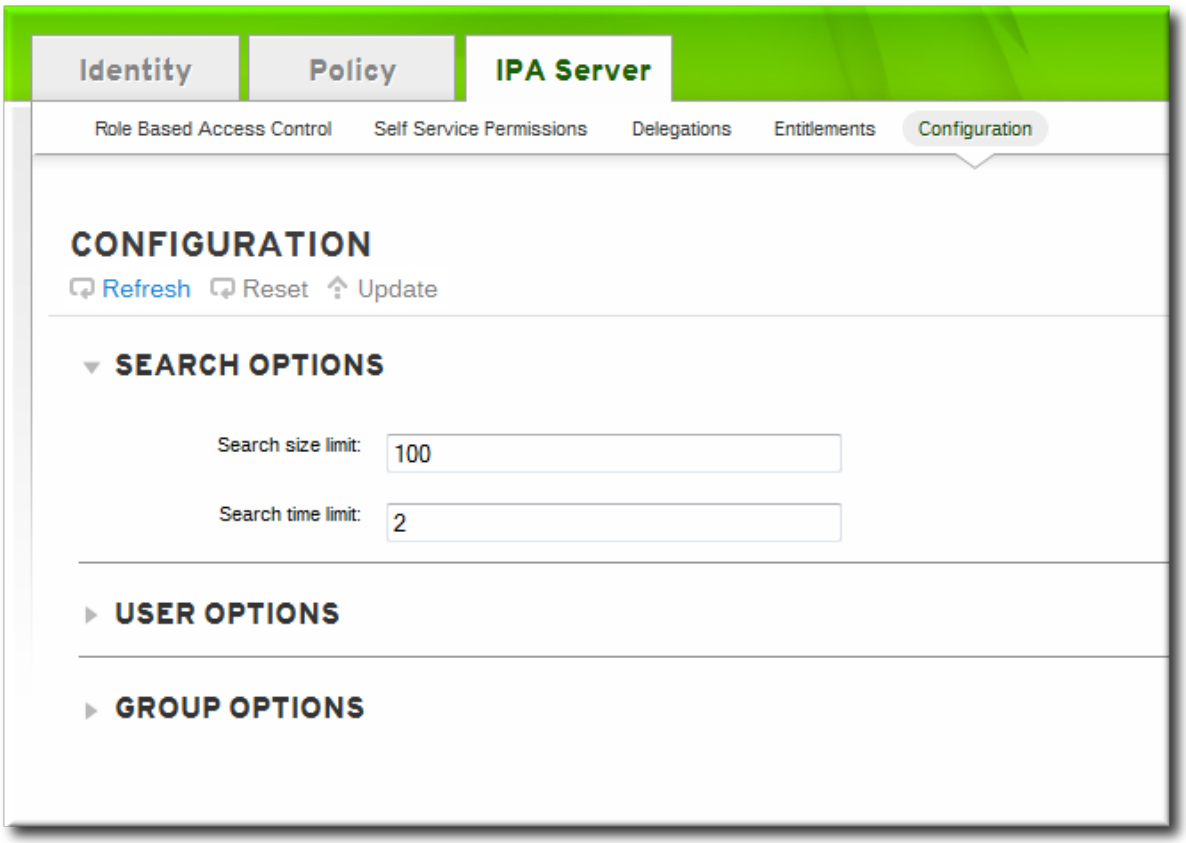
デフォルト設定では、1 回の検索で返されるレコード数は 100 件未満、検査時間は 2 秒となっています。

重要

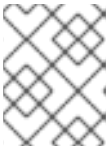
検索のサイズや時間制限を高く設定しすぎると、IdM サーバーのパフォーマンスにマイナスの影響が出る可能性があります。

9.11.3.1.2.1. Web UI の使用

1. **IPA Server** タブを開きます。
2. **Configuration** サブタブを選択します。
3. **Search Options** 領域までスクロールします。



4. 検索制限の設定を変更します。
   - **検索サイズ制限**: 返される検索結果の最大数を設定します。
   - **検索時間制限**: 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。



ヒント

時間制限またはサイズ制限の値を -1 に設定すると、検索に制限がないことを意味します。

5. 変更が完了したら、**Configuration** ページ上部の **Update** リンクをクリックします。

9.11.3.1.2.2. コマンドラインの使用

検索制限は、**config-mod** コマンドを使用して変更できます。

```
[bjensen@server ~]$ ipa config-mod --searchtimelimit=5 --searchrecordslimit=500

  Max. username length: 32
  Home directory base: /home
  Default shell: /bin/sh
  Default users group: ipausers
  Default e-mail domain for new users: example.com
  Search time limit: 5
  Search size limit: 50
  User search fields: uid,givenname,sn,telephonenumber,ou,title
  Group search fields: cn,description
  Enable migration mode: FALSE
  Certificate Subject base: O=EXAMPLE.COM
  Password Expiration Notification (days): 4
```

ヒント

時間制限またはサイズ制限の値を -1 に設定すると、検索に制限がないことを意味します。

9.11.3.1.3. 検索のデフォルトの上書き

サーバー設定では、検索時のサイズや時間制限に関するグローバル初期設定を指定するものもあります。これらの制限は常に Web UI で適用されますが、コマンドラインで ***-find*** コマンドを実行して上書きできます。

--sizelimit および **--timelimit** オプションは、対象コマンドの実行時にそれぞれ指定して、別のサイズ、時間を設定できます。どのような結果が必要かによって、制限を増やしたり減らしたりできます。

たとえば、デフォルトの時間制限が 60 秒で検索にかかる時間が長くなる場合に、時間制限を 120 秒に増やすことができます。

```
[jsmith@ipaserver ~]$ ipa user-find smith --timelimit=120
```

9.11.3.2. 検索属性の設定

ユーザーまたはグループを検索しても、対象属性で該当する属性が自動的にすべて検索されるわけではありません。代わりに、属性の特定のサブセットを検索します。また、そのリストは設定可能です。

ユーザーまたはグループの検索フィールドに属性を追加する場合は、その属性の LDAP ディレクトリーに対応するインデックスがあることを確認してください。検索はインデックスに基づいて実行されます。大半の標準 LDAP 属性にはインデックスがありますが、カスタム属性には、インデックスを作成する必要があります。インデックスの作成については、『Directory Server Administrator's Guide』の「インデックス」 の章で説明されています。

9.11.3.2.1. 検索でチェックされるデフォルトの属性

デフォルトでは、ユーザー検索には 6 つの属性が、グループ検索には 2 つの属性がインデックス化されています。これらについては、表9.5「デフォルトの検索属性」に一覧表示されます。検索属性はすべて、ユーザー/グループ検索の対象となります。

表9.5 デフォルトの検索属性

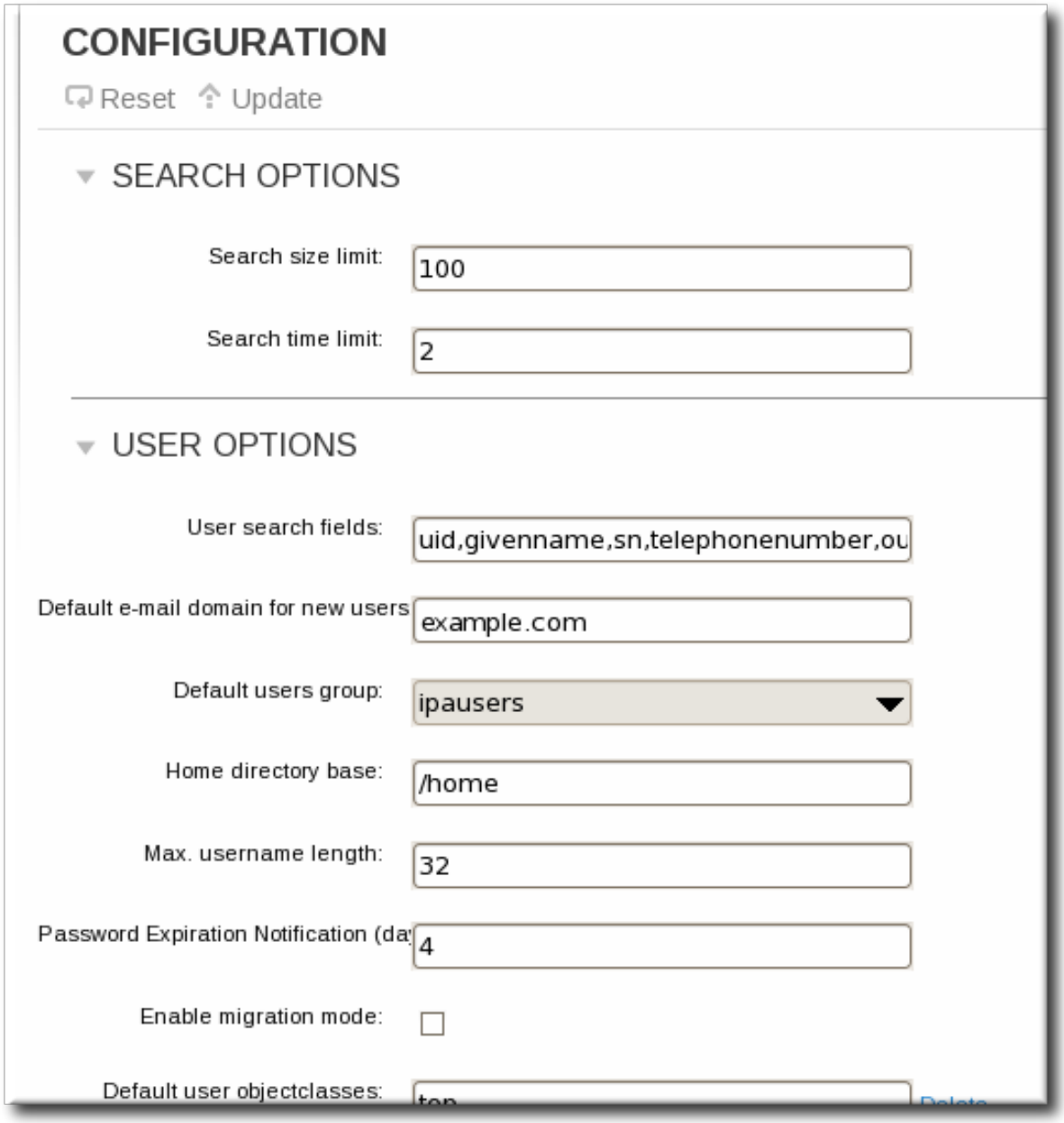
ユーザー検索の属性	
First name	Last name
Login ID	Job title
Organizational unit	Phone number
グループ検索の属性	
Name	詳細

「検索属性の設定」および「グループ検索属性の変更」で説明されているように、ユーザーおよびグループの検索の対象となる属性を変更できます。

9.11.3.2.2. ユーザー検索属性の変更

9.11.3.2.2.1. Web UI での操作

1. **IPA Server** タブを開きます。
2. **Configuration** サブタブを選択します。
3. **User Options** エリアまでスクロールします。

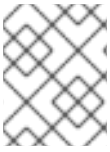


4. 他の検索属性は、**User search fields** フィールドにコンマ区切りの一覧として追加します。
5. 変更が完了したら、**Configuration** ページ上部の **Update** リンクをクリックします。

9.11.3.2.2.2. コマンドラインでの操作

検索属性を変更するには、**--usersearch** オプションを使用してユーザー検索の属性を設定します。

```
[bjensen@server ~]$ ipa config-mod --usersearch=uid,givenname,sn,telephonenumber,ou,title
```



注記

常に検索属性の完全な一覧を指定してください。設定の引数で指定した値は、以前の設定を上書きます。

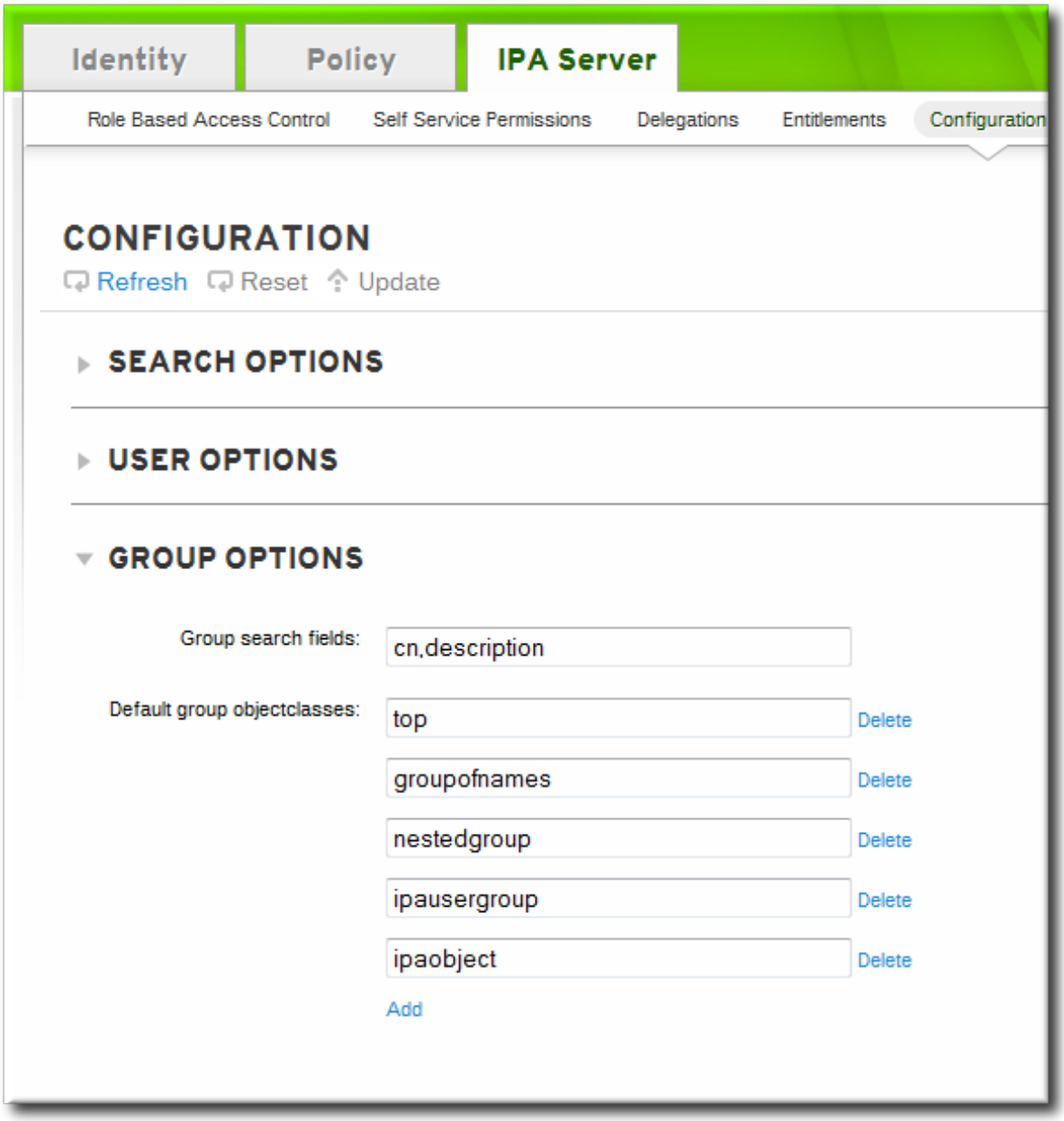
9.11.3.2.3. グループ検索属性の変更

ユーザーまたはグループを検索しても、対象属性で該当する属性が自動的にすべて検索されるわけではありません。代わりに、属性の特定のサブセットを検索します。また、そのリストは設定可能です。

ユーザーまたはグループの検索フィールドに属性を追加する場合は、その属性の LDAP ディレクトリーに対応するインデックスがあることを確認してください。検索はインデックスに基づいて実行されます。大半の標準 LDAP 属性にはインデックスがありますが、カスタム属性には、インデックスを作成する必要があります。インデックスの作成については、『Directory Server Administrator's Guide』の「インデックス」 の章で説明されています。

9.11.3.2.3.1. Web UI での操作

1. **IPA Server** タブを開きます。
2. **Configuration** サブタブを選択します。
3. **Group Options** エリアまでスクロールします。

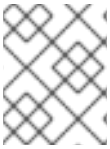


4. 他の検索属性は、**Group search fields** フィールドにコンマ区切りの一覧として追加します。
5. 変更が完了したら、**Configuration** ページ上部の **Update** リンクをクリックします。

9.11.3.2.3.2. コマンドラインでの操作

検索属性を変更するには、**--groupsearch** オプションを使用してグループ検索の属性を設定します。

```
[bjensen@server ~]$ ipa config-mod --groupsearch=cn,description
```



注記

常に検索属性の完全な一覧を指定してください。設定の引数で指定した値は、以前の設定を上書きます。

9.11.3.2.4. 検索結果で返される属性の制限

UI に表示されない属性で検索を実行できます。つまり、指定のフィルターと一致しない検索で、エントリーを返すことができます。特に、検索情報が非常に短くして一致率を上げる場合などによく使用されます。

9.11.3.3. タイプを基にしたグループ検索

グループ定義はシンプルですが、実際には、グループ、メンバーを暗黙的に含むネスト化グループ、POSIX などのメンバー属性をもとにしたグループにエントリーを自動的に割り当てる自動メンバールールを作成することができるので、グループの定義は非常に複雑です。

group-find コマンドには、各種オプションが多数あり、所属メンバーまたは所属していないメンバー、グループ定義の他の属性をもとにグループを検索できます。

たとえば、ユーザープライベートグループは IdM UI には表示されないため、通常の検索では返されません。ただし、この **--private** オプションを使用すると、プライベートグループだけに検索結果を絞り込みます。

```
[root@server ~]# ipa group-find --private
---------------
1 group matched
---------------
  Group name: jsmith
  Description: User private group for jsmith
  GID: 1084600001
----------------------------
Number of entries returned 1
----------------------------
```

グループ検索は、グループに所属するまたは所属しないメンバーをもとに実行可能です。つまり、単一ユーザー、他のグループ、ロールやホストベースのアクセス制御定義など、他の設定エントリーなどを基に検索できます。たとえば、最初の検索では、ユーザー **jsmith** が所属するグループが表示されます。

```
[root@server ~]# ipa group-find --user=jsmith
---------------
1 group matched
---------------
  Group name: ipausers
  Description: Default group for all users
  Member users: jsmith
```

```
----------------------------
Number of entries returned 1
----------------------------
```

他の検索では、**jsmith** が **属さない** 全グループが表示されます。

```
[root@server ~]# ipa group-find --no-user=jsmith
---------------
3 groups matched
---------------
  Group name: admins
  Description: Account administrators group
  GID: 1084600000
  Member users: admin

  Group name: editors
  Description: Limited admins who can edit other users
  GID: 1084600002

  Group name: trust admins
  Description: Trusts administrators group
  Member users: admin
----------------------------
Number of entries returned 3
----------------------------
```

表9.6「一般的なグループ検索オプション」には、便利なグループ検索オプションがに一覧表示されています。

表9.6 一般的なグループ検索オプション

オプション	条件の説明
--private	プライベートグループのみを表示します。
--gid	指定した GID に完全一致するグループのみを表示します。
--group-name	指定の名前または部分的な名前が含まれるグループのみを表示します。
--users、--no-users	メンバーとして指定のユーザーが含まれる (または含まれない) グループのみを表示します。
--in-hbacrules、--not-inhbac-rules	指定のホストベースのアクセス制御ルールに属するグループ (または **--not-in** オプションの場合はルールに属さないグループ) のみを表示します。指定した sudo ルールおよびロールに属するグループを表示する (またはしない) オプションと似ています。

オプション	条件の説明
--in-groups、 --not-in-groups	別のグループに属するグループのみを表示します。指定したグループ (または **--not-in** オプションの場合は属さないグループ) だけを表示します。指定した netgroup に属するグループを表示する (またはしない) オプションと似ています。

[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。

[3] GID/UID 割当範囲の変更に関する情報は、「一意の UID および GID 番号の割り当て管理」を参照してください。

第10章 アイデンティティー: 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 (例: クライアント名の変更) 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

10.1. ホスト、サービス、およびマシン ID と認証

登録プロセスの基本的な役割は、IdM ディレクトリー内でクライアントマシン用の **ホスト** エントリーを作成することです。このホストエントリーは、他のホストとドメイン内のサービスの関係を確立するために使用されます。この関係では、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
- ホストとサービスのプリンシパル
- 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。

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。(「Linux サービスの統合」 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

マシンは、IdM が管理する別のアイデンティティーとして処理されます。IdM サーバーで、ユーザー ID

が 389 Directory Server インスタンスに保存されるのと同様に、クライアントは、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識別します。マシンはユーザーのように、Kerberos または証明書を使って、ドメインに対して認証し、マシンの ID を検証できます。

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 公開キーが作成され、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。詳細は、「ホストの公開 SSH 鍵の管理」および『Red Hat Enterprise Linux デプロイメントガイド』を参照してください。
- 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 20章 *ポリシー: Kerberos ドメインの管理* で説明されています。
- 機械の証明書。この場合には、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。「付録B *certmonger を使った作業*」で説明されているように、クライアントでは、証明書は **certmonger** というサービスが管理します。

10.2. ホストエントリー設定のプロパティー

ホストエントリーには、ホストの物理的な場所や MAC アドレス、鍵および証明書など、システム設定以外の情報を追加できます。

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、情報を追加する必要があります。

表10.1 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
Description	--desc=**description**	ホストの説明。
Locality	--locality=**locality**	ホストの位置情報
Location	--location=**location**	データセンターラックなど、ホストの位置情報

UI フィールド	コマンドラインオプション	説明
Platform	--platform=**string**	ホストのハードウェアまたはアーキテクチャー
Operating system	--os=**string**	ホストのオペレーティングシステムおよびバージョン
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。
Principal name (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のデフォルト値に設定されます。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。
Set One-Time Password	--password=**string**	一括登録で使用可能なホストのパスワードを設定します。
-	--random	一括登録で使用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	これは IP アドレス変更時にホストが DNS エントリーを動的に更新できるかどうかを設定する属性切り替え。

10.3. ホストエントリーの無効化および再有効化

アクティブなホストは、ドメイン内の他のサービスやホスト、ユーザーからアクセス可能です。アクティビティーからホストを削除する必要がある場合もあります。ただし、ホストを削除するとエントリーや関連する設定もすべて完全に削除されてしまいます。

10.3.1. 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。

たとえば、以下のようになります。

```
[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。

10.3.2. ホストの再有効化

ホストを無効にすると、実質的に現行のアクティブなキータブを強制終了します。キータブを削除すると、ホストの設定エントリーを変更せずにホストを IdM ドメインから削除することになります。

ホストを再度有効にするには、**ipa-getkeytab** コマンドを使用するだけです。**-s** オプションは、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。

新規のホストキータブを要求する場合は、以下のようになります。

```
[jsmith@ipaserver ~]$  ipa-getkeytab -s ipaserver.example.com -p host/server.example.com -k
/etc/krb5.keytab -D fqdn=server.example.com,cn=computers,cn=accounts,dc=example,dc=com -w
password
```

アクティブな IdM クライアントまたはサーバーで **ipa-getkeytab** コマンドを実行すると、LDAP 認証情報 (**-D** および **-w**) なしで実行できます。IdM ユーザーは、Kerberos 認証情報を使用してドメインへの認証を行います。無効化されたホストでコマンドを直接実行するには、LDAP 認証情報を指定してから IdM サーバーに認証します。認証情報は、再度有効にするホストまたはサービスに一致する必要があります。

10.4. ホストの公開 SSH 鍵の管理

OpenSSH は、**公開鍵**を使ってホストに対して認証を行います。あるマシンが別のマシンに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。

10.4.1. 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。たとえば、以下のようになります。

```
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。

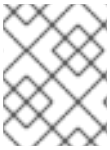
```
ssh-rsa AAA...ZZZ== host.example.com,1.2.3.4
```

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。

10.4.2. 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。

10.4.3. 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。たとえば、以下のようになります。

```
server.example.com,1.2.3.4 ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAQEApvjBvSFSkTU0WQW4eOweeo0DZZ08F9Ud21xlLy6F
OhzwpXFGIyxvXZ52+siHBHbbqGL5+14N7UvElruyslIHx9LYUR/pPKSMXCGyboLy5aTNl5OQ5
EHwrhVnFDIKXkvp45945R7SKYCUtRumm0Iw6wq0XD4o+ILeVbV3wmcB1bXs36ZvC/M6riefn
9PcJmh6vNCvIsbMY6S+FhkWUTTiOXJjUDYRLlwM273FfWhzHK+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:
4f:61:ee:2c:f7:d7:da:41:17:93:de:1d:19:ac:2e:c8 server.example.com
The key's randomart image is:
+--[ RSA 2048]----+
|             .. |
|              .+|
|         o   .* |
|        o . .. *|
|       S + .  o+|
|        E . .. .|
|       . = .  o |
|        o .  ..o|
|           .....|
+-----------------+
```

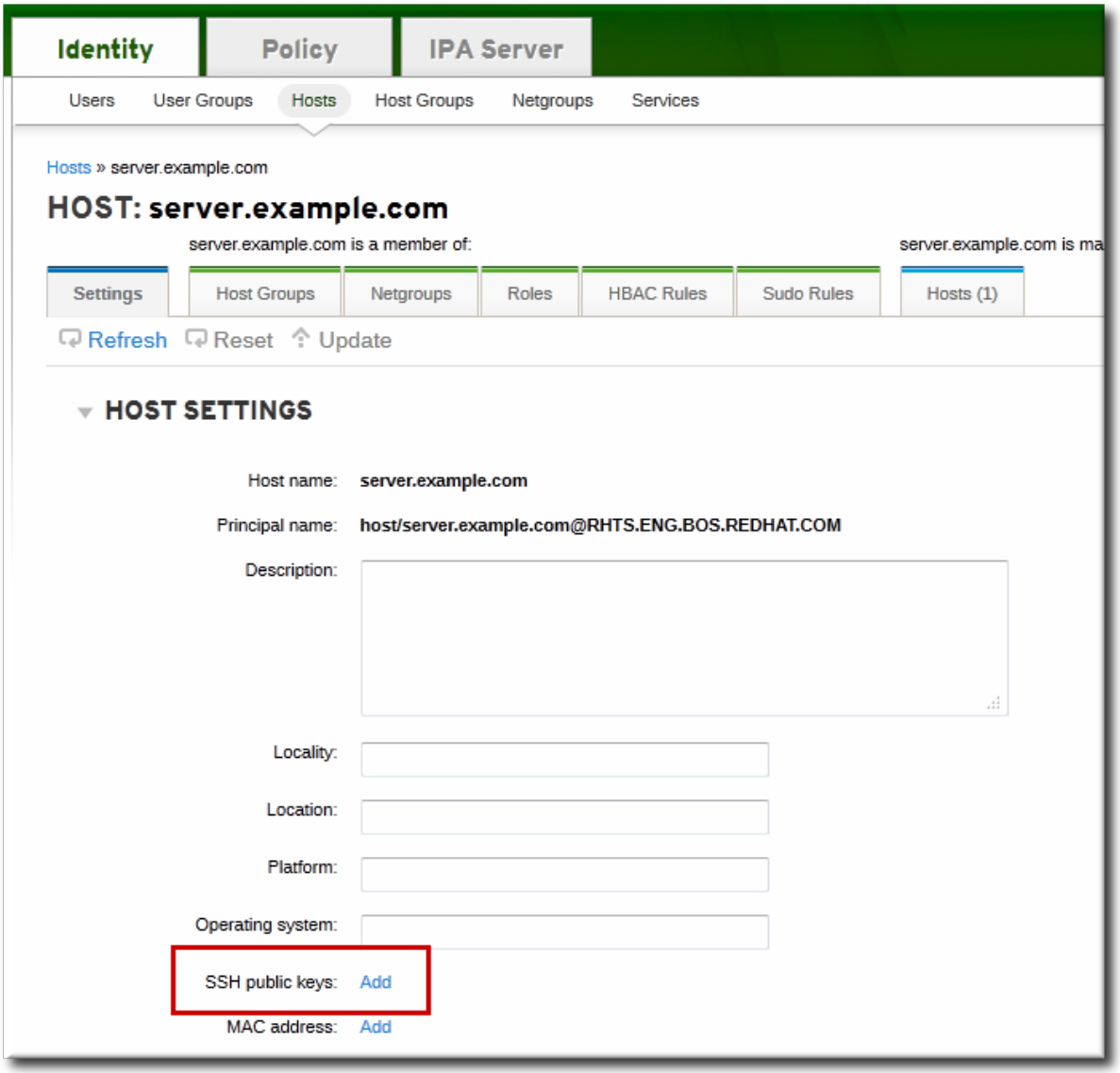
2. 公開鍵をキーファイルからコピーします。完全なキーエントリーは、**hostname,IP type key==** の形式です。**key==** は必須ですが、エントリー全体を保存できます。エントリーの全要素を使用するには、エントリーを再編成して、順番が **type key== [hostname,IP]** になるように設定します。

```
[jsmith@server ~]$ cat /home/jsmith/.ssh/host_keys.pub

ssh-rsa AAAAB3NzaC1yc2E...tJG1PK2Mq++wQ== server.example.com,1.2.3.4
```

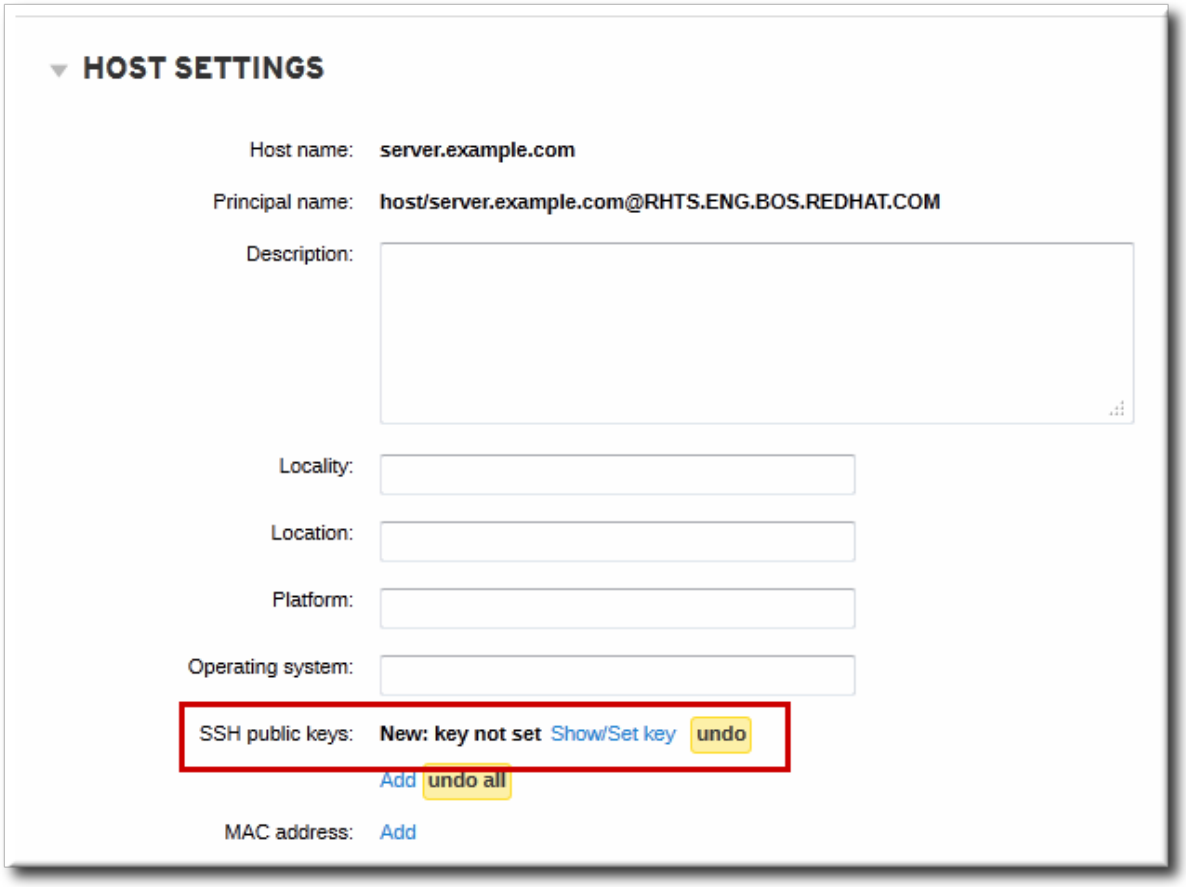
3. **Identity** タブを開き、サブタブの **ホスト** を選択します。

4. 編集するホスト名をクリックします。

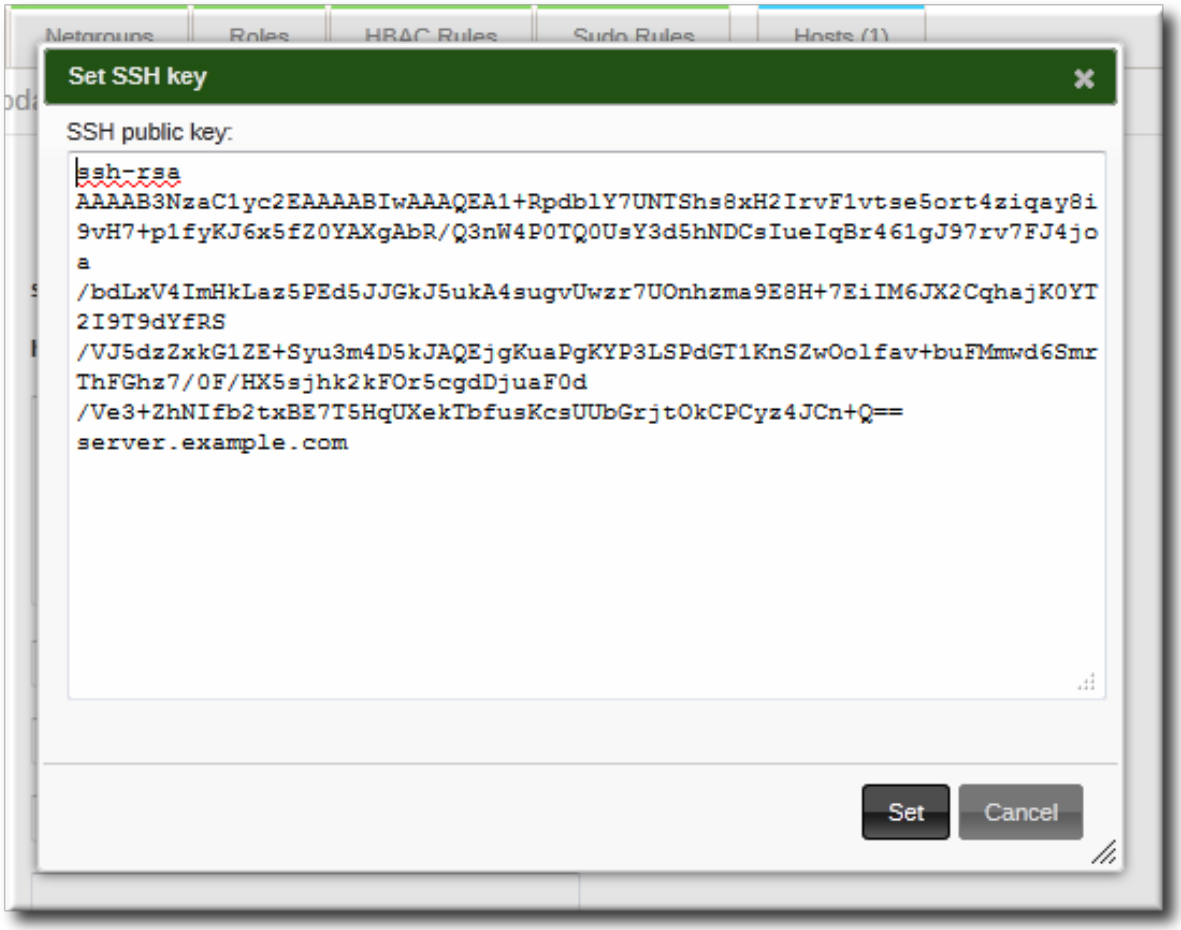
5. **Settings** タブの **Host Settings** エリアで、**SSH public keys: Add** リンクをクリックします。



6. UI で新しいリンク (**New: key not set Show/Set key**) が開きます。**Show/Set key** リンクをクリックします。



7. ホストの公開鍵を貼り付けて、**Set** ボタンをクリックします。



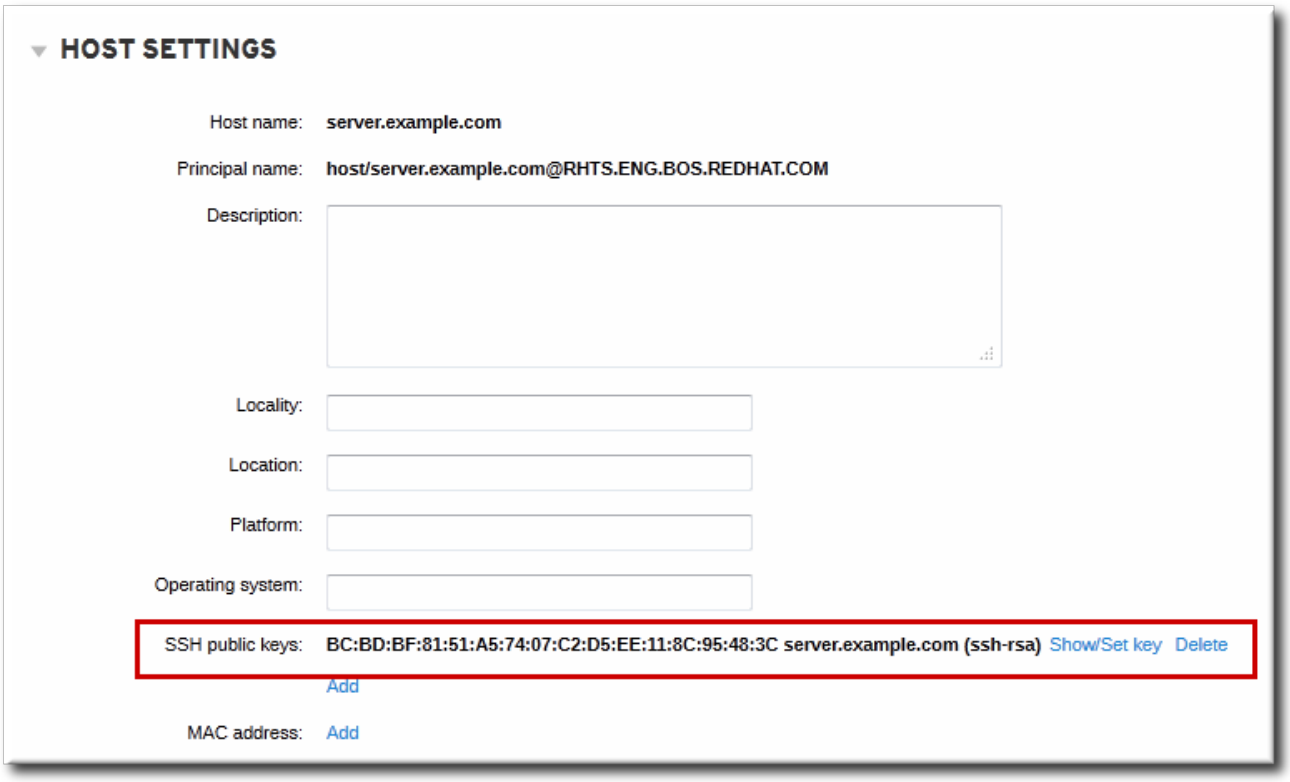
SSH public keys フィールドに **New: key set** と表示されるようになります。**Show/Set キー** のリンクをクリックすると、送信したキーが表示されます。

8. 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。

9. すべてのキーが送信されたら、ホストページ上部の **Update** ボタン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。[4].

図10.1 保存された公開鍵

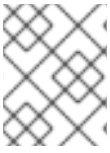


ホストキーをアップロードしたら、Identity Management を ID ドメインの 1 つとして使用するよう SSSD を設定し、OpenSSH がホストキー管理に SSSD ツールを使用するよう設定します。

ホストキーをアップロードしたら、Identity Management を ID ドメインの 1 つとして使用するよう SSSD を設定し、OpenSSH がホストキー管理に SSSD ツールを使用するよう設定します。これは、『Red Hat Enterprise Linux デプロイメントガイド』で説明しています。

10.4.4. 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** を使っ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、64 ビット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** オプションも使ってホストの DNS エントリーも更新します。

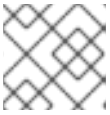
   たとえば、以下のようになります。

   ```
   [jsmith@server ~]$ ipa host-mod --sshpubkey="ssh-rsa 12345abcde==" --updatedns
   host1.example.com
   ```

   実際のキーでは、キーはこの例よりも長く、通常は末尾が等号 (=) になります。

   複数のキーをアップロードするには、**--sshpubkey** オプション 1 つでコンマ区切りのキー一覧を指定します。

   ```
   --sshpubkey="12345abcde==,key2==,key3=="
   ```

   ヒント

   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、Identity Management を ID ドメインの 1 つとして使用するよう SSSD を設定し、OpenSSH がホストキー管理に SSSD ツールを使用するよう設定します。これは、『Red Hat Enterprise Linux デプロイメントガイド』で説明しています。

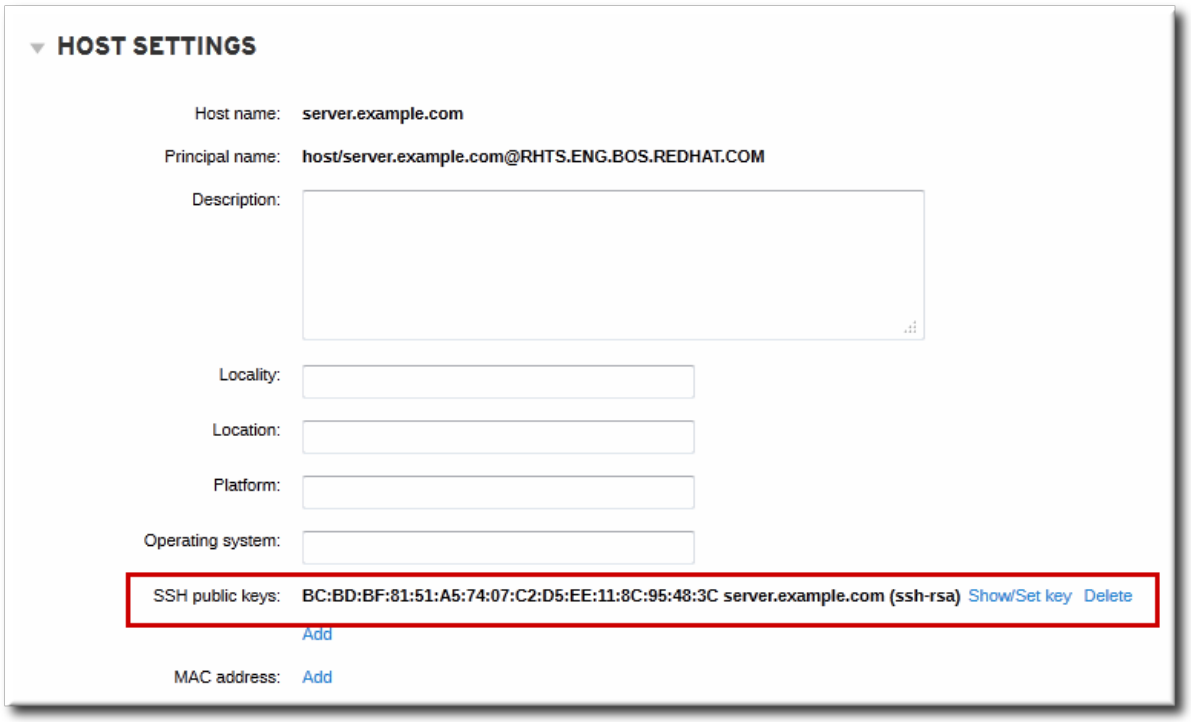
10.4.5. ホストキーの削除

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。

3. **Settings** タブの **Host Settings** エリアを開きます。
4. 削除するキーのフィンガープリントの横にある **Delete** のリンクをクリックします。



5. 変更を保存するには、ホストページの上部にある **Update** リンクをクリックします。

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。たとえば、以下のようになります。

```
[jsmith@server ~]$ kinit admin
[jsmith@server ~]$ ipa host-mod --sshpubkey= --updatedns host1.example.com
```

10.5. 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。たとえば、以下のようになります。

```
[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 ~]# getnt ethers server.example.com
```

10.6. マシンの名前変更および IDM クライアントオプションの再設定

Kerberos と SSL を正しく操作するには、システムのホスト名が重要になります。Kerberos および SSL のセキュリティーメカニズムはいずれもホスト名に依存し、指定されたホスト間で通信が行われるようにします。仮想マシンまたはクラスター化されたサービスを使用するインフラストラクチャーでは一般的に、システムのコピー、移動、名前変更により、名前が変更されたホストが含まれます。

Red Hat Enterprise Linux には、IdM ホストの名前を簡単に変更するシンプルな名前変更コマンドがありません。IdM ドメインのホストの名前を変更するには、IdM のエントリーの削除、クライアントソフトウェアのアンインストール、ホスト名の変更を行い、新しい名前を使用して再登録する必要があります。さらに、ホストの名前変更を行う上で、サービスプリンシパルを再生成する必要があります。

クライアントを再設定するには、以下を実行します。

1. マシンで実行されているサービスを特定します。これらのファイルは、マシンの再登録時に作成し直す必要があります。

```
# ipa service-find server.example.com
```

   各ホストには、サービス一覧に表示されないデフォルトのサービスがあります。このサービスは、「ホストサービス」と呼ばれます。ホストサービスのサービスプリンシパルは、**host/<hostname>** です (例: **host/server.example.com**)。このプリンシパルは **ホストプリンシパル** とも呼ばれます。

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。

```
[root@client ~]# kinit admin
[root@client ~]# ipa hostgroup-find server.example.com
```

3. 証明書が関連付けられているサービスを特定します。**ldapsearch** コマンドを使用して、IdM LDAP データベースのエントリーを直接チェックすることでサービスの特定ができます。

```
[root@client ~]# ldapsearch -x -b "cn=accounts,dc=example,dc=com" "(&
(objectclass=ipaservice)(userCertificate=*))" krbPrincipalName -D "cn=directory manager" -w
secret -h ipaserver.example.com -p 389
```

4. (ホストプリンシパルに加えて) サービスプリンシパルの場合は、**server.example.com** にある対応のキータブの場所を判断します。サービスごとにキータブの場所が異なりますが、IdM にはこの情報は保存されません。

   クライアントシステム上の各サービスには **ldap/server.example.com@EXAMPLE.COM** など、**service_name/hostname@REALM** の形式で Kerberos プリンシパルが含まれています。

5. 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。

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

6. **/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
```

7. IdM サーバーで、IdM 管理者として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 ~]# kinit admin
[root@server ~]# ipa host-del server.example.com
```

この時点で、ホストは IdM から完全に削除されました。

8. マシンの名前を変更します。
9. IdM でシステムを再登録します。

```
[root@client ~]# ipa-client-install
```

再登録することで、**/etc/krb5.keytab** に、新規ホスト名でホストプリンシパルが生成されます。

10. IdM サーバーで、全サービスに対して新しいキータブを追加します。

```
[root@server ~]# ipa service-add serviceName/new-hostname
```

11. サービスの証明書を生成するには、**certmonger** または IdM の管理ツールを使用します。
12. 該当するホストグループにホストを再度追加します。

10.7. ホストグループの管理

ホストグループは、重要な管理タスク (特にアクセス制御) を一元管理する方法の 1 つです。

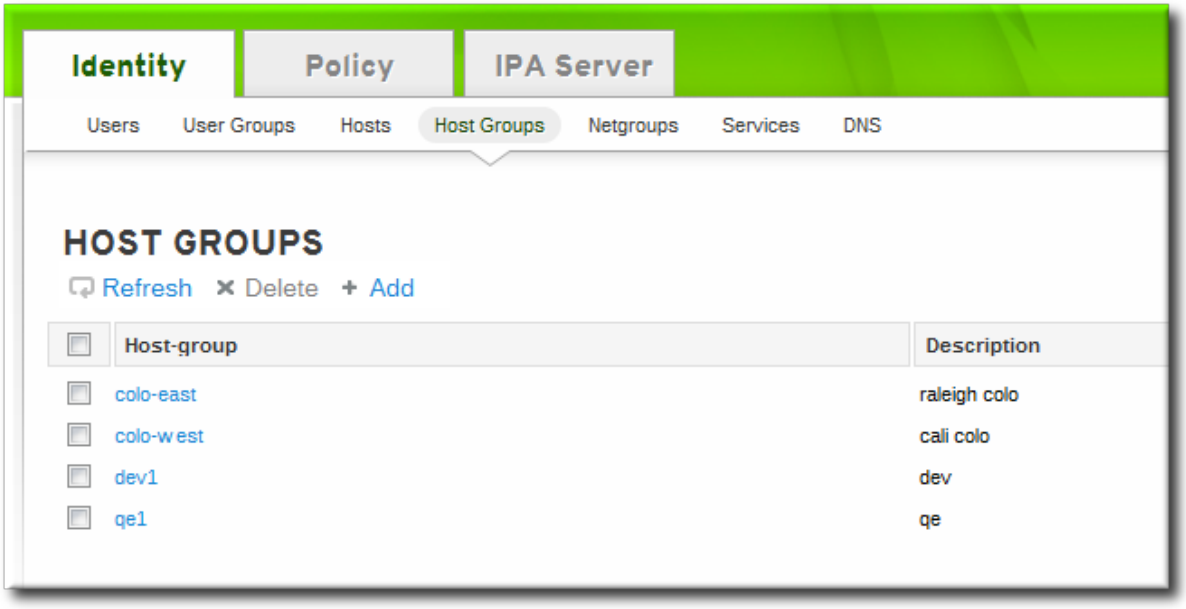
Identity Management のすべてのグループは基本的に **静的** であるため、グループのメンバーは、手作業で明示的にグループに追加されます。IdM では、グループが他のグループに所属する **ネストされたグループ** を暗黙的に許可します。この場合には、グループに含まれるメンバーはすべて自動的に親グループにも所属します。

グループの作成は簡単なので、作成するグループや、整理する方法を非常に柔軟に決定できます。グループは、部署、物理的な場所などの組織部門や、アクセス制御に関する IdM またはインフラストラクチャーの使用ガイドラインをもとに定義できます。

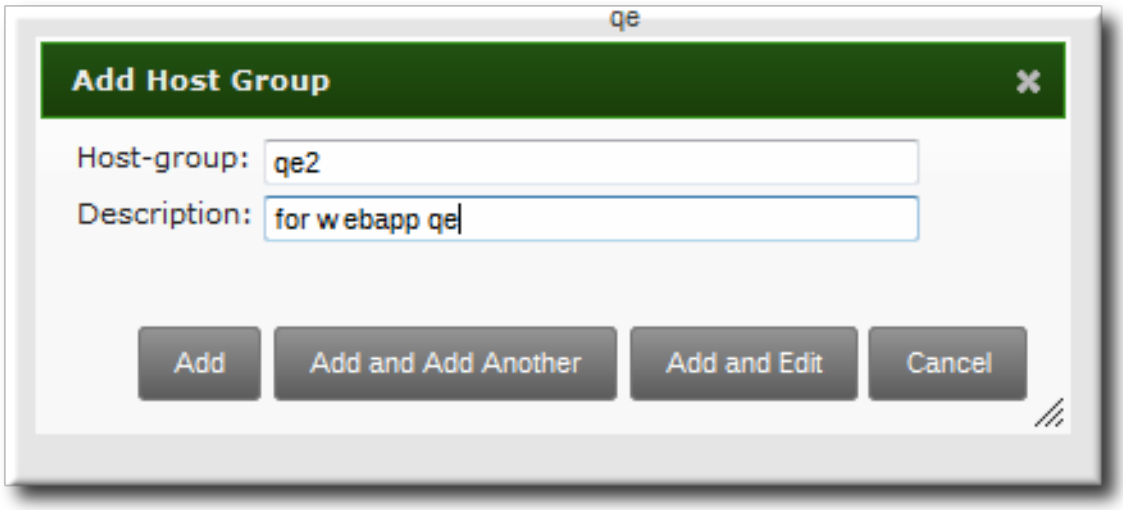
10.7.1. ホストグループの作成

10.7.1.1. Web UI でホストグループの作成

1. **Identity** タブを開き、サブタブの **Host Groups** を選択します。
2. グループ一覧上部にある **Add** をクリックします。



3. グループの名前と説明を入力します。



4. **Add and Edit** ボタンをクリックすると、すぐにメンバー選択ページに移動します。

5. 「Web UI でホストグループメンバーの追加」で説明されているようにメンバーを選択します。

10.7.1.2. コマンドラインでのホストグループの作成

新規グループは、**hostgroup-add** コマンドを使用して作成します。(このコマンドではグループだけが追加され、メンバーは別に追加します。)

グループ名とグループの説明の 2 つの属性が常に必要になります。これらの属性が引数として指定されていない場合には、スクリプトでグループ名と説明を入力するように求められます。

```
$ ipa hostgroup-add groupName --desc="description"
```

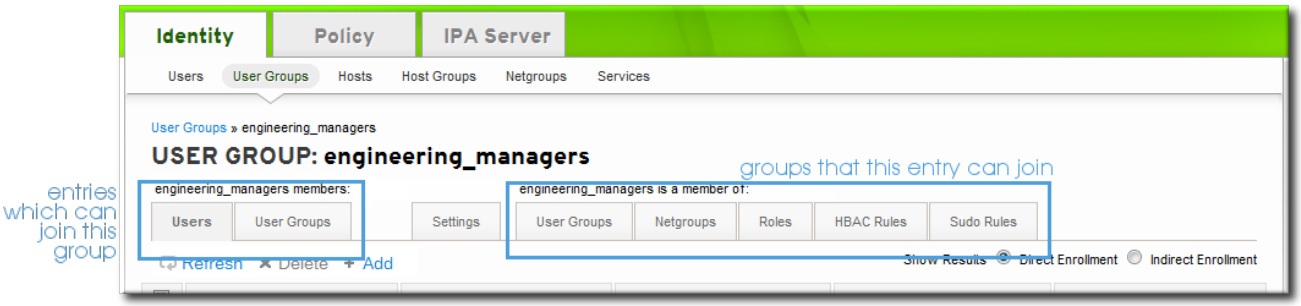
10.7.2. ホストグループメンバーの追加

10.7.2.1. グループメンバーの表示および変更

グループ設定を使用してメンバーをグループに追加できます。グループ所属可能な全メンバータイプのタブがあり、管理者は一致する全エントリーを選択してメンバーとして追加します。

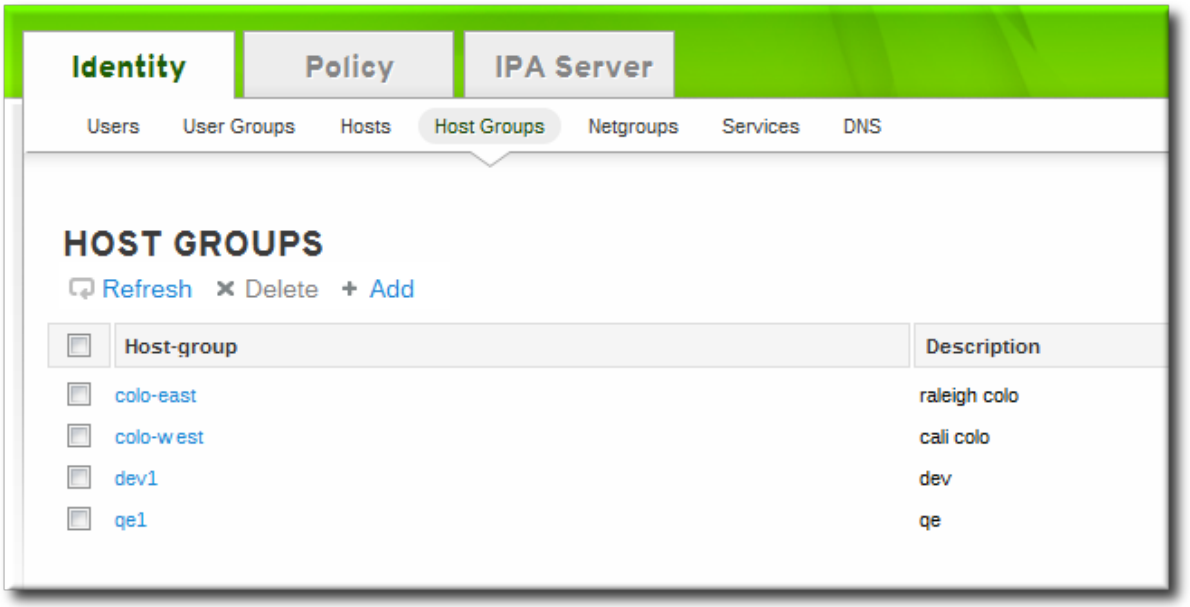
ただし、独自の設定でエンティティーをグループに追加することもできます。各エントリーにはタブの一覧があり、参加可能なグループタイプが表示されます。対象のタイプの全グループ一覧が表示され、エンティティーを同時に複数のグループに追加できます。

図10.2 メンバー

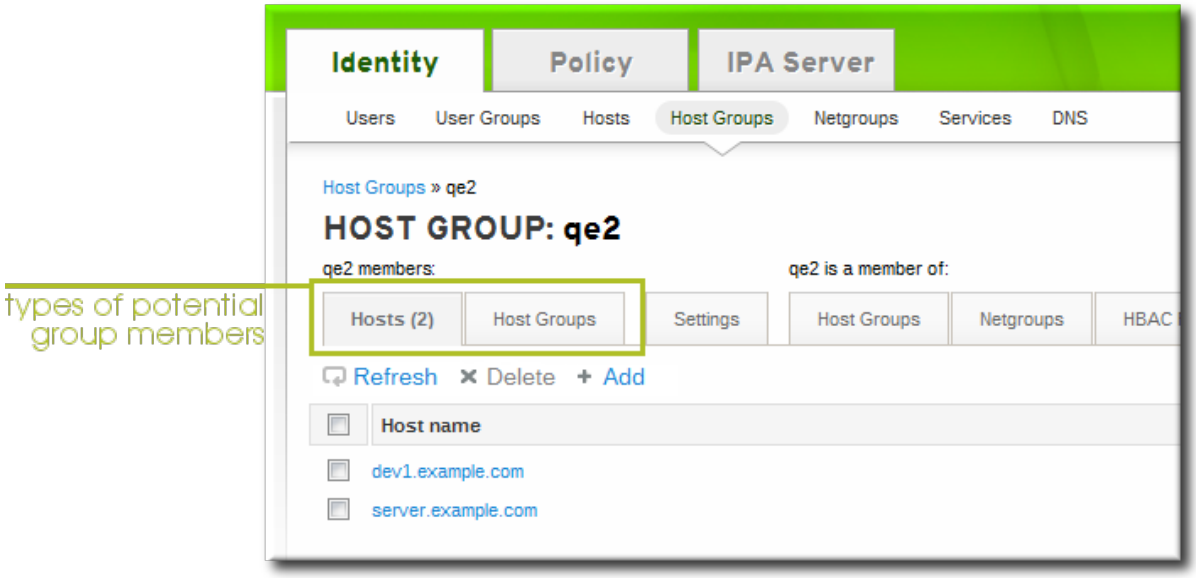


10.7.2.2. Web UI でホストグループメンバーの追加

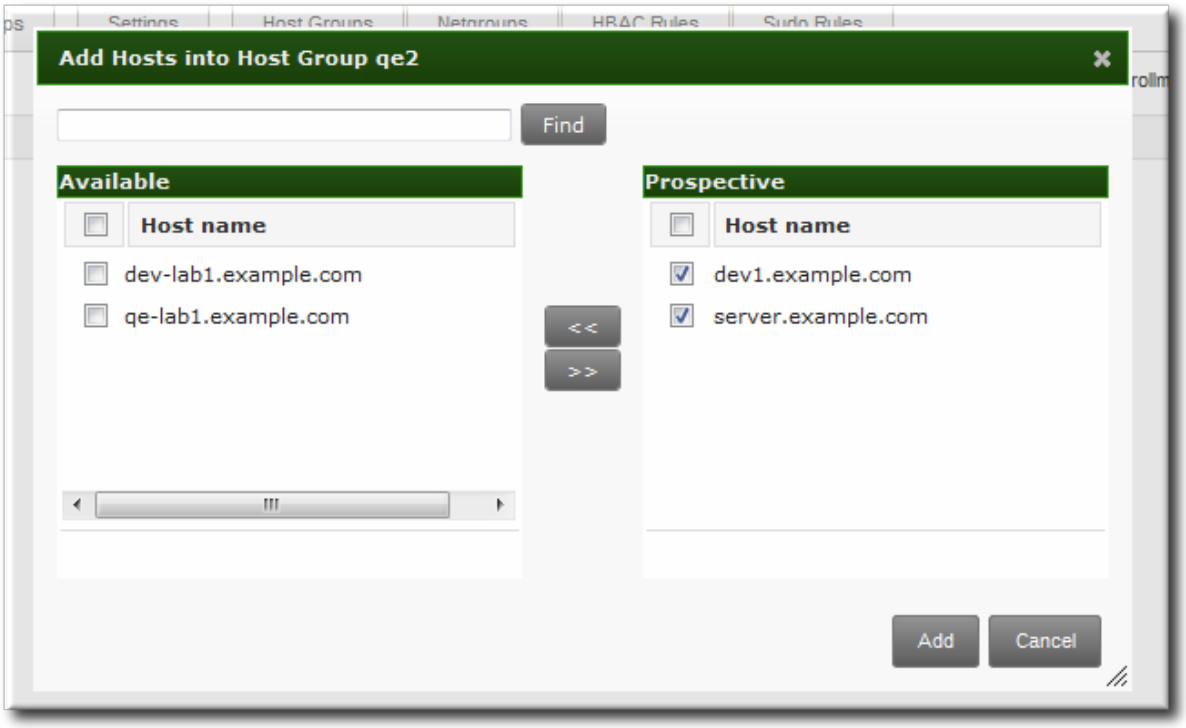
1. **Identity** タブを開き、サブタブの **Host Groups** を選択します。
2. メンバーを追加するグループ名をクリックします。



3. タスクエリア上部にある **Add** をクリックします。



4. 追加するホストの名前の横にあるチェックボックスをクリックし、右矢印ボタン (**>>**) をクリックし、ホストを選択項目のボックスに移動します。



5. **追加** ボタンをクリックします。

10.7.2.3. コマンドラインでのホストグループメンバーの追加

メンバーは、**hostgroup-add-member** コマンドを使用して、ホストグループに追加します。このコマンドは、両方のホストと、他のグループをグループメンバーとして追加できます。

hostgroup-add-member コマンドの構文では、追加するグループ名のコンマ区切りの一覧のみが必要になります。

```
$ ipa hostgroup-add-member groupName [--hosts=list] [--hostgroups=list]
```

たとえば、以下は、ホスト 3 つを **caligroup** グループに追加します。

```
$ ipa hostgroup-add-member caligroup --
hosts=ipaserver.example.com,client1.example.com,client2.example.com
  Group name: caligroup
  Description: for machines in california
  GID: 387115842
  Member hosts: ipaserver.example.com,client1.example.com,client2.example.com
-------------------------
Number of members added 3
-------------------------
```

同様に、他のグループをメンバーとして追加して、ネスト化されたグループを作成することもできます。

```
$ ipa hostgroup-add-member caligroup --groups=mountainview,sandiego
  Group name: caligroup
  Description: for machines in california
  GID: 387115842
  Member groups: mountainview,sandiego
  -------------------------
  Number of members added 2
  -------------------------
```

[4] 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11章 アイデンティティー: 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。(「Linux サービスの統合」 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

11.1. 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。

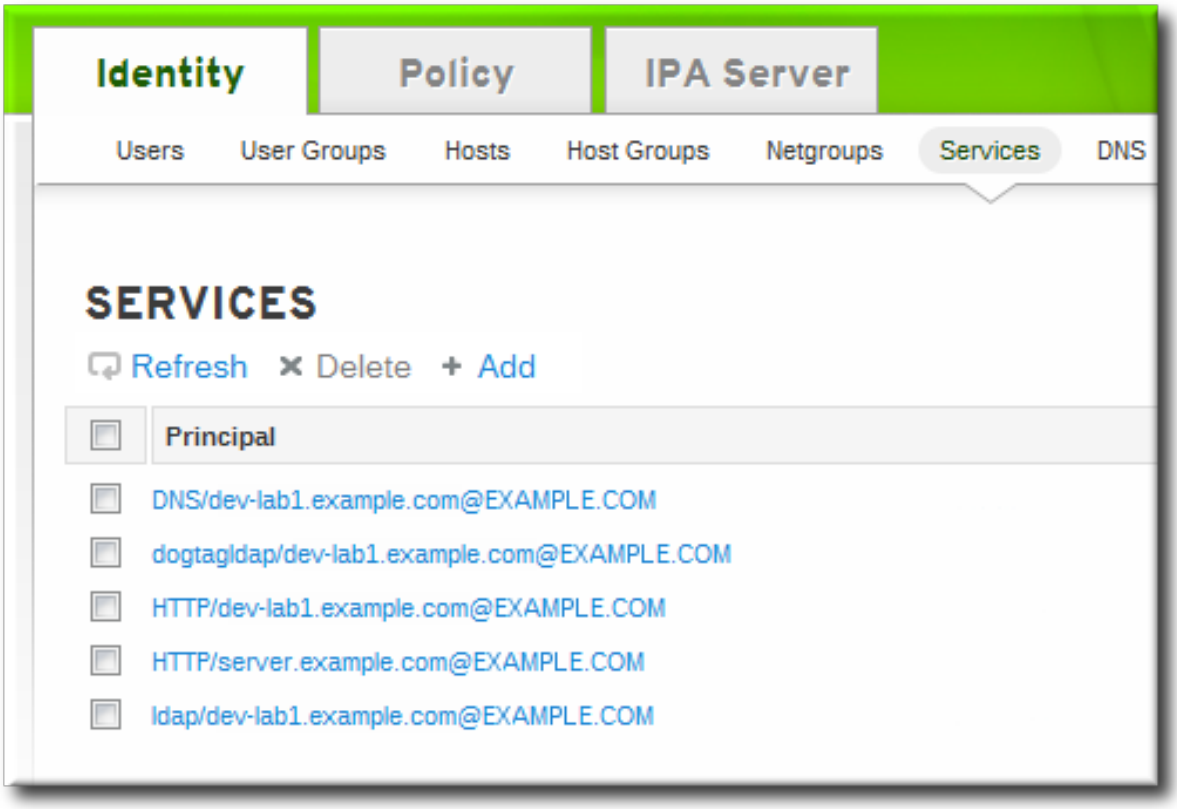
注記

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。

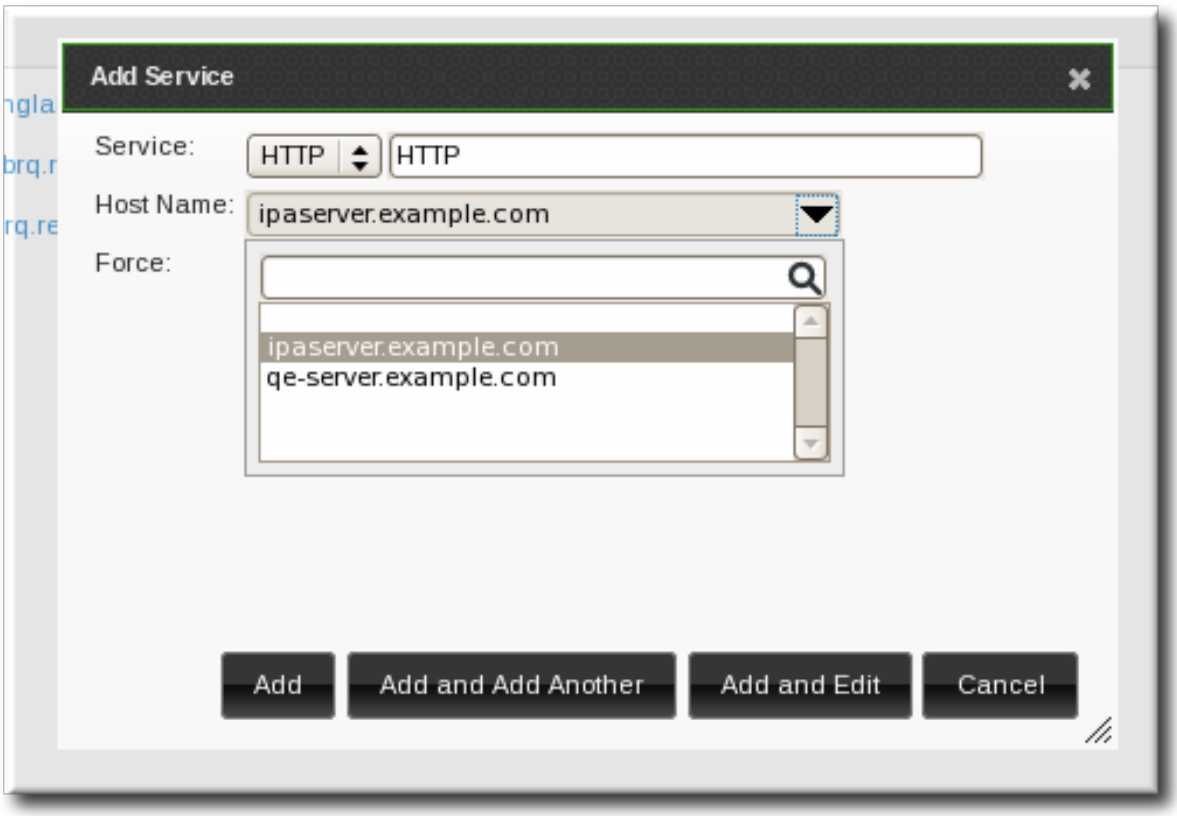
11.1.1. Web UI でのサービスとキータブの追加

1. **Identity** タブを開き、**Services** サブタブを選択します。
2. サービス一覧の上部にある **Add** リンクをクリックします。



3. ドロップダウンメニューからサービスタイプを選択し、名前を付けます。

4. サービスが実行される IdM ホストのホスト名を選択します。ホスト名を使用して、完全なサービスプリンシパル名を構成します。



5. **Add** ボタンをクリックして、新しいサービスプリンシパルを保存します。

6. **ipa-getkeytab** コマンドを使用し、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、コンマ区切りの暗号化タイプの一覧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。

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。

11.1.2. 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
```

たとえば、以下のようになります。

```
$ 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一覧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。

警告

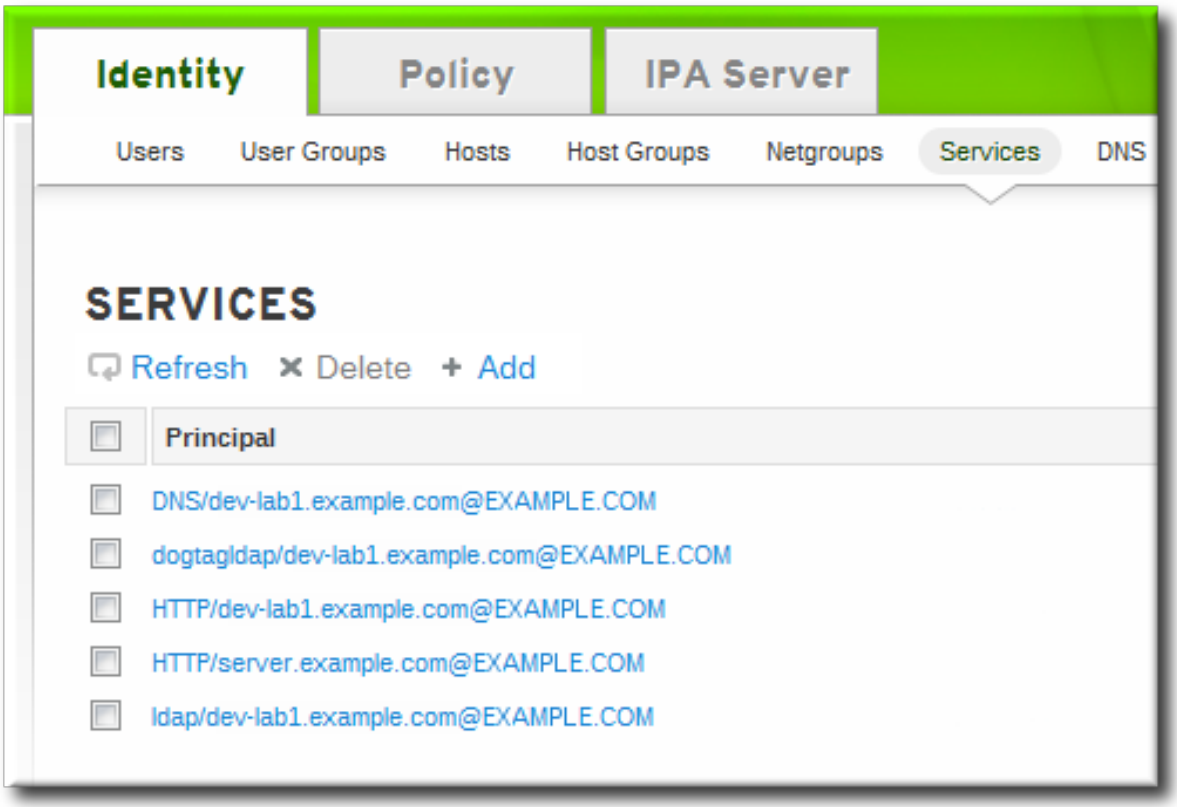
ipa-getkeytab コマンドは、指定されたプリンシパルのシークレットをリセットします。つまり、そのプリンシパルの他のキータブすべてが無効になります。

11.2. サービスおよびサービスの証明書の追加

サービスではキータブを使用できますが、サービスによってアクセスするのに証明書が必要な場合があります。このような場合には、サービスはサービスエントリーに含めるように、サービスを追加 (または変更) できます。

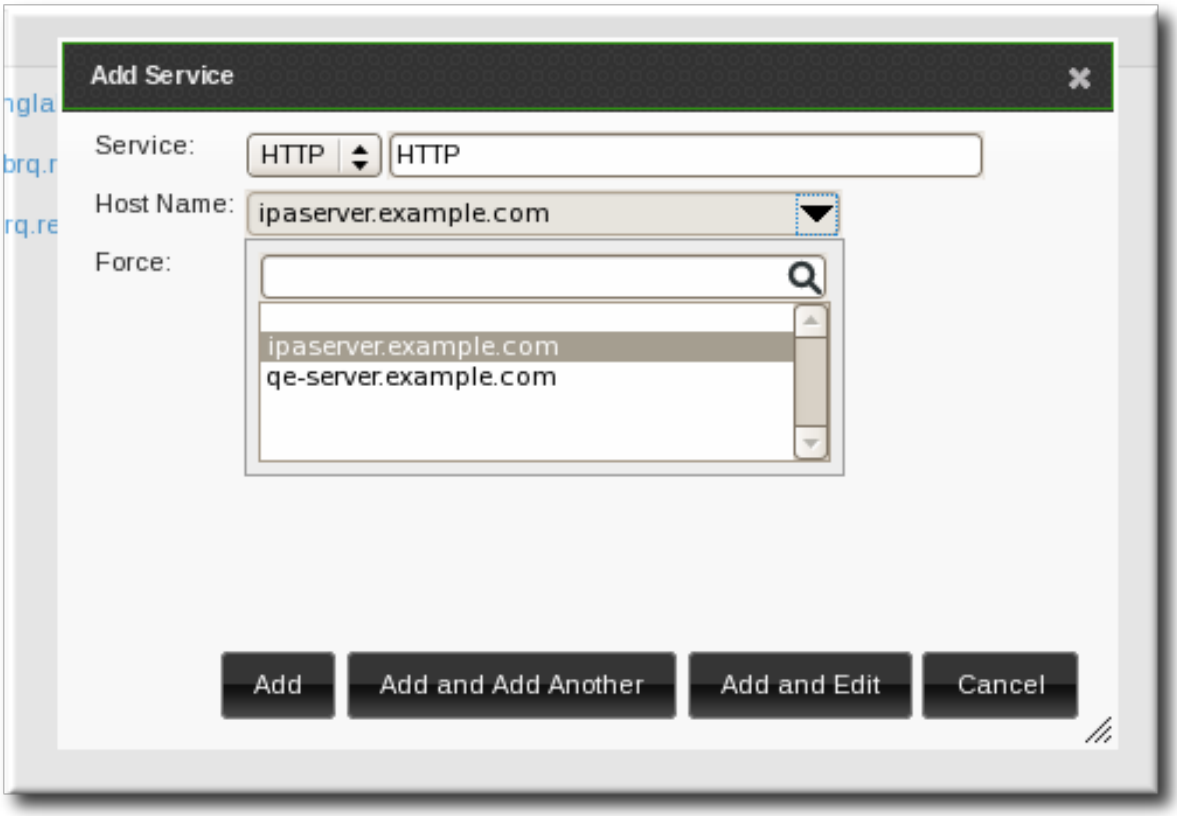
11.2.1. Web UI でのサービスおよび証明書の追加

1. **Identity** タブを開き、**Services** サブタブを選択します。
2. サービス一覧の上部にある **Add** リンクをクリックします。



3. ドロップダウンメニューからサービスタイプを選択し、名前を付けます。

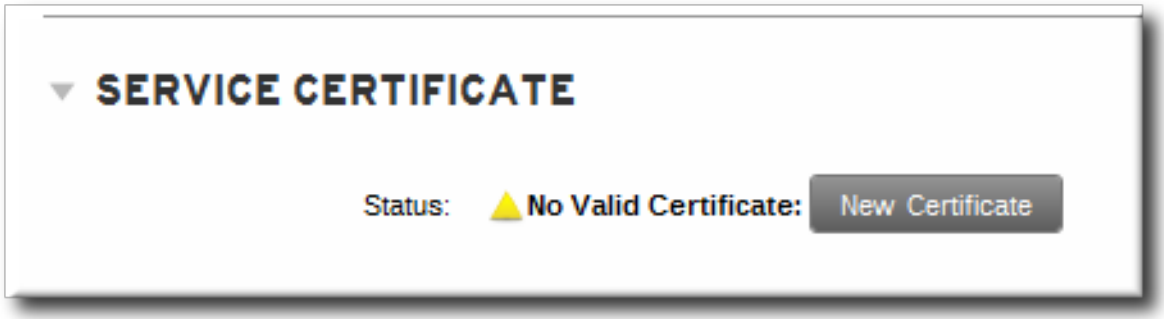
4. サービスが実行される IdM ホストのホスト名を選択します。ホスト名を使用して、完全なサービスプリンシパル名を構成します。



5. **Add and Edit** ボタンをクリックして、サービスエントリーページに直接移動します。

6. ページの下部にある **Service Certificate** セクションまでスクロールします。

7. **New Certificate** ボタンをクリックしてサービス証明書を作成します。



11.2.2. コマンドラインでのサービスおよび証明書の追加

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。

```
[jsmith@ipaserver ~]$ kinit admin
[jsmith@ipaserver ~]$ ipa service-add serviceName/hostname
```

以下に例を示します。

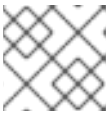
-

```
$ 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. サービスの証明書を作成します。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 cert-request --principal=HTTP/web.example.com example.csr
   ```

   ヒント

   --add オプションを使用して、証明書の要求時にサービスを自動作成します。

   または、**getcert** コマンドを使用し、**certmonger** で証明書を作成して管理します。オプションの詳細は、「certmonger で証明書の要求」を参照してください。

   ```
   $ ipa-getcert request -d /etc/httpd/alias -n Server-Cert -K HTTP/client1.example.com -N
   'CN=client1.example.com,O=EXAMPLE.COM'
   ```

11.3. NSS データベースでの証明書の保存

サービスが証明書を使用する場合には、証明書およびキーは NSS データベースに格納できます (このデータベースはサービス自体や Identity Management で使用できます)。

1. NSS データベースを作成します。

   ```
   $ certutil -N -d /path/to/database/dir
   ```

2. NSS ツール (**certutil**) を使用して証明書を要求します。

   ```
   $ certutil -R -s "CN=client1.example.com,O=EXAMPLE.COM" -d /path/to/database/dir -a >
   example.csr
   ```

IdM ドメインが CA に証明書システムを使用している場合は、サブジェクト名の CN のみが使用されます。自己署名 CA の場合には、サブジェクトを、設定済みの証明書のサブジェクトベースと一致させる必要があります。IdM サーバーは、この値とは異なるサブジェクトベースを使用する要求を拒否します。

11.4. クラスタサービスの設定

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。

11.5. 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。

11.6. サービスエントリーの無効化および再有効化

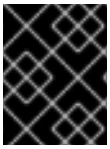
アクティブなサービスは、ドメイン内の他のサービスやホスト、ユーザーからアクセス可能です。アクティビティーからホストやサービスを削除する必要がある場合もあります。ただし、サービスやホストを削除するとエントリーやすべての関連する設定も永続的に削除されてしまいます。

11.6.1. 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
$ 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。

11.6.2. 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
```

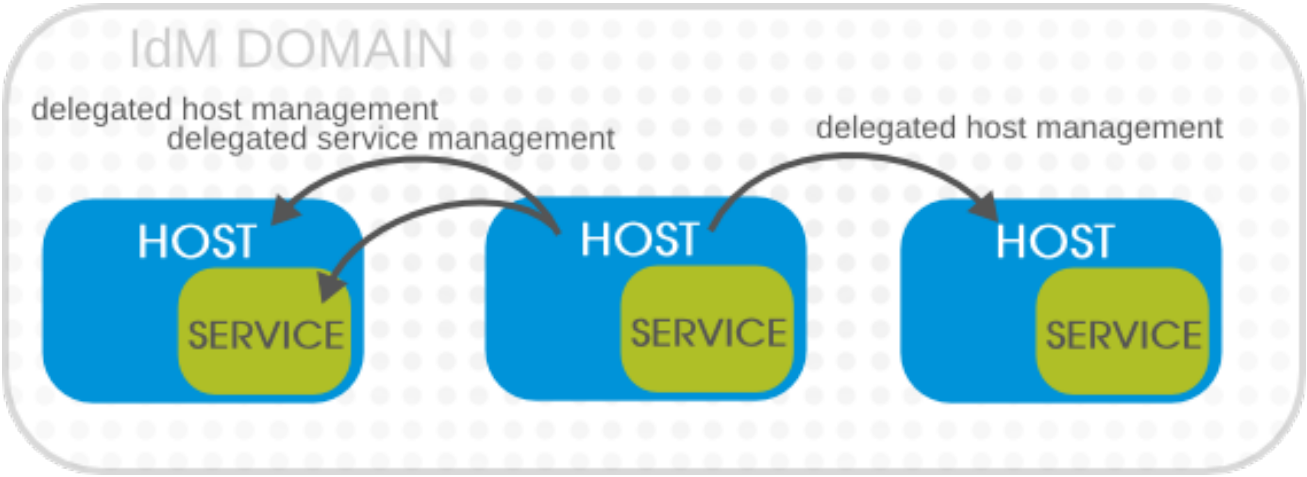
アクティブな IdM クライアントまたはサーバーで **ipa-getkeytab** コマンドを実行すると、LDAP 認証情報 (**-D** および **-w**) なしで実行できます。IdM ユーザーは、Kerberos 認証情報を使用してドメインへの認証を行います。無効化されたホストでコマンドを直接実行するには、LDAP 認証情報を指定して IdM サーバーへの認証を行います。認証情報は、再度有効にするホストまたはサービスに一致する必要があります。

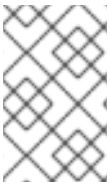
第12章 アイデンティティー: ホストおよびサービスへのアクセス委譲

「サーバーとクライアント間の関係」で説明されているように、IdM ドメイン内で **管理する** ということは、キータブおよび、別のサービスまたはホストの証明書を取得できる必要があります。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12.1 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。

12.1. サービス管理の委譲

service-add-host コマンドを使用してサービスの制御をホストに委譲します。サービスの委譲は、プリンシパルの指定と、ホストの制御指定 (コンマ区切りの一覧) の 2 つの部分で構成されます。

```
# ipa service-add-host principal --hosts=hostnames
```

以下に例を示します。

```
# 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ってサービスを管理できます。

```
# kinit -kt /etc/krb5.keytab host/`hostname`
# ipa-getkeytab -s `hostname` -k /tmp/test.keytab -p http/web.example.com
Keytab successfully retrieved and stored in: /tmp/test.keytab
```

このサービスのチケットを作成するには、委譲された認証局がホストで証明書要求を作成し、**cert-request** コマンドでサービスエントリーを作成して認証情報を読み込みます。

```
# ipa cert-request --add --principal=http/web.example.com web.csr
  Certificate: MIICETCCAXqgA...[snip]
  Subject: CN=web.example.com,O=EXAMPLE.COM
  Issuer: CN=EXAMPLE.COM Certificate Authority
  Not Before: Tue Feb 08 18:51:51 2011 UTC
  Not After: Mon Feb 08 18:51:51 2016 UTC
  Fingerprint (MD5): c1:46:8b:29:51:a6:4c:11:cd:81:cb:9d:7c:5e:84:d5
  Fingerprint (SHA1):
  01:43:bc:fa:b9:d8:30:35:ee:b6:54:dd:a4:e7:d2:11:b1:9d:bc:38
  Serial number: 1005
```

12.2. ホスト管理の委譲

他のホストへの権限の委譲は **host-add-managedby** コマンドを使用します。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。

   ```
   # 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。

   ```
   # ipa host-add-managedby client2.example.com --hosts=client1.example.com
   ```

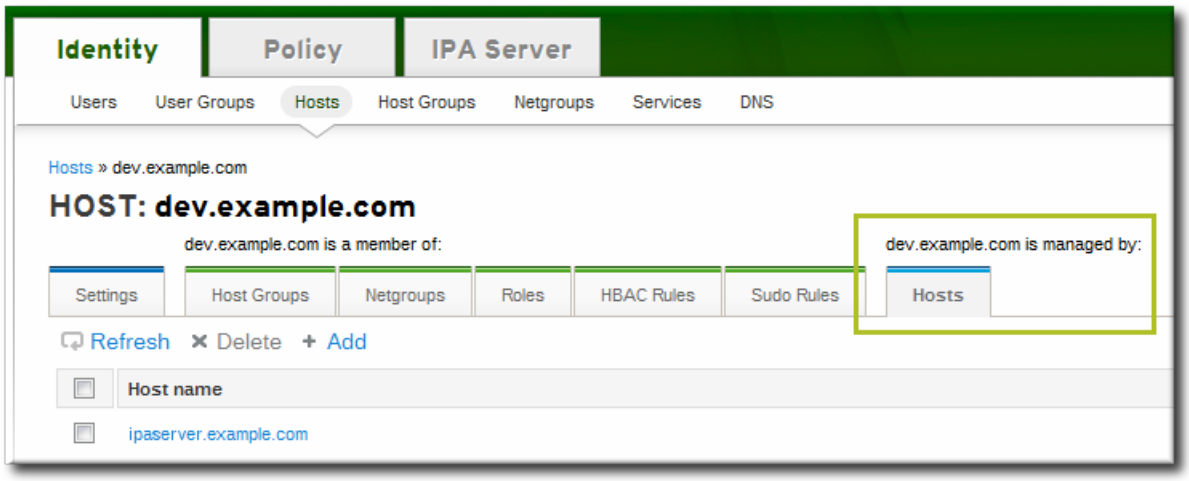
3. ホストの **client1** としてチケットを取得してから、**client2** のキータブを取得します。

   ```
   # kinit -kt /etc/krb5.keytab host/`hostname`
   # ipa-getkeytab -s `hostname` -k /tmp/client2.keytab -p host/client2.example.com
   Keytab successfully retrieved and stored in: /tmp/client2.keytab
   ```

12.3. WEB UI を使ったホストまたはサービス管理の委譲

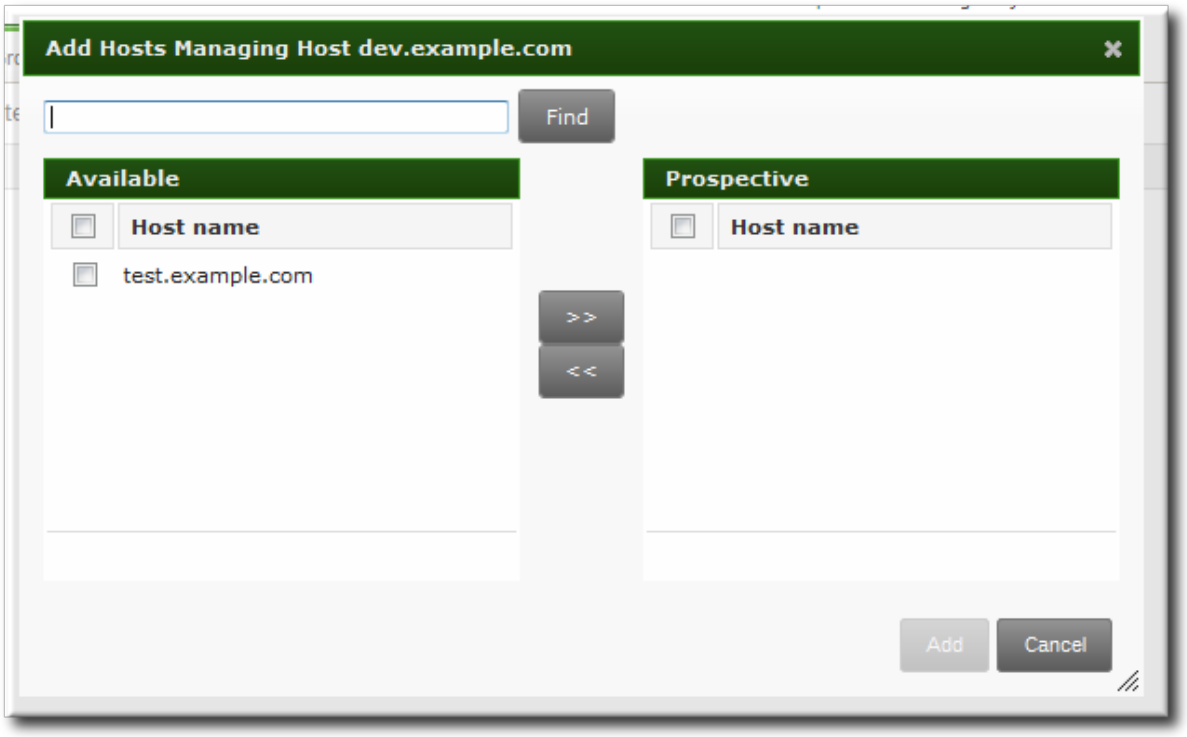
各ホストおよびサービスエントリーには、どのホストがホストやサービスの管理を委譲されているかを示す説明タブがあります。

1. **Identity** タブを開き、**Hosts** または **Services** サブタブを選択します。
2. **委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。



4. 一覧上部にある **Add** をクリックします。

5. ホスト/サービスの管理を委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。



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。

12.4. 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。

たとえば、ホストにアクセスするには、次のコマンドを実行します。

```
# 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。

```
# kinit -kt /etc/httpd/conf/krb5.keytab http/ipa.example.com@EXAMPLE.COM
```

第13章 アイデンティティー: NIS ドメインおよびネットグループとの統合

ネットワーク情報サービス (NIS) は、Unix ネットワーク上の ID および認証を管理する最も一般的な方法の 1 つです。使いやすくシンプルではありますが、セキュリティーリスクがあり、柔軟性に欠けるため、NIS ドメインの管理しにくくなる可能性があります。

ID 管理では、netgroup およびその他の NIS データを IdM ドメインに統合する方法を提供します。IdM ドメインには、NIS 設定に比べ、IdM の強力なセキュリティー構造が組み込まれています。または、管理者が単に、ユーザーとホストの ID を NIS ドメインから IdM ドメインに移行することもできます。

13.1. NIS および IDENTITY MANAGEMENT の概要

ネットワーク情報サービス (NIS) は、ユーザーおよびパスワード、ホストと IP アドレス、POSIX グループなどの認証およびアイデンティティー情報を一元管理します。これは、ID と認証ルックアップだけに焦点を当てていたため、**イエローページ** (略称 YP) と呼ばれていました。

NIS にはホスト認証のメカニズムがなく、ネットワークに、パスワードハッシュなど、暗号化されていない情報を流すので、NIS は最新のネットワーク環境の多くで安全性が低いとみなされます。管理者には NIS は人気がありませんが、システムクライアントの多くで活発に使用されています。上記のような安全性の低さを回避する方法があります。その方法として、NIS を他のプロトコルと統合して、セキュリティーを強化します。

Identity Management では、NIS オブジェクトは基礎となる LDAP ディレクトリーを使用して IdM に統合されます。LDAP サービス (RFC 2307 で定義) は、NIS オブジェクトのサポートを提供します。Identity Management はこの NIS オブジェクトをカスタマイズして、他のドメイン ID との統合が改善されるようにします。NIS オブジェクトは LDAP サービス内に作成され、**nss_ldap** や SSSD などのモジュールが、暗号化された LDAP 接続を使用してオブジェクトを取得します。

NIS エンティティーは **netgroup** に保存されます。netgroup では、標準の UNIX グループでサポートされない、ネスト化 (グループ内のグループ) が可能です。また、netgroup には、Unix グループにないホストをグループ化する方法をあります。

NIS グループは、ユーザーとホストを大規模なドメインのメンバーとして定義することで機能します。netgroup は、3 つの情報 (ホスト、ユーザー、ドメイン) を設定します。これは **トリプル** と呼ばれます。

```
host,user,domain
```

netgroup トリプルは、ユーザーまたはホストをドメインに関連付けますが、ユーザーとホスト間での関連付けはありません。したがって、通常トリプルでは、明確性および管理性を向上するためにホストまたはユーザーを定義します。

```
host.example.com,,nisdomain.example.com
-,jsmith,nisdomain.example.com
```

NIS は netgroup データを配信するだけではありません。ユーザーとパスワード、グループ、ネットワークデータ、およびホストに関する情報も保管します。Identity Management は NIS リスナーを使用してパスワード、グループ、および netgroups を IdM エントリーにマッピングできます。

IdM LDAP エントリーでは、netgroup のユーザーは単一のユーザーまたはグループで、いずれも ***memberUser*** パラメーターで識別されます。同様に、ホストは単一ホストまたはホストグループのいずれかになります。これらは ***memberHost*** 属性で識別されます。

```
dn: ipaUniqueID=d4453480-cc53-11dd-ad8b-0800200c9a66,cn=ng,cn=accounts,...
objectclass: top
objectclass: ipaAssociation
objectclass: ipaNISNetgroup
ipaUniqueID: d4453480-cc53-11dd-ad8b-0800200c9a66
cn: netgroup1
memberHost: fqdn=host1.example.com,cn=computers,cn=accounts,...
memberHost: cn=VirtGuests,cn=hostgroups,cn=accounts,...
memberUser: cn=jsmith,cn=users,cn=accounts,...
memberUser: cn=bjensen,cn=users,cn=accounts,...
memberUser: cn=Engineering,cn=groups,cn=accounts,...
nisDomainName: nisdomain.example.com
```

Identity Management では、これらの netgroup エントリーは **netgroup-*** コマンドを使用して処理され、基本的な LDAP エントリーが表示されます。

```
# ipa netgroup-show netgroup1
Netgroup name: netgroup1
Description: my netgroup
NIS domain name: nisdomain
Member User: jsmith
Member User: bjensen
Member User: Engineering
Member Host: host1.example.com
Member Host: VirtGuests
```

Identity Management は、クライアントが NIS netgroup にアクセスしようとすると、LDAP エントリーを従来の NIS マップに変換して NIS プロトコル経由でクライアントに送信するか (NIS プラグインを使用)、または RFC 2307 または RFC 2307bis に準拠する LDAP 形式に変換します。

13.2. IDENTITY MANAGEMENT の NIS ポートの設定

IdM サーバーは、サーバーの起動時に無作為に選択したポートで NIS サービスにバインドします。対象のポート割り当てをポートマッパーに送信し、NIS クライアントが IdM サーバーへの問い合わせ時に使用するポートを認識できるようにします。

管理者は、NIS クライアントのファイアウォールを開放する必要がある場合や、事前にポート番号を把握し、その番号を同じままにする必要のあるサービスが他にある場合があります。この場合には、管理者は使用するポートを指定できます。

注記

NIS プラグイン設定には、1024 未満の利用可能なポート番号を使用できます。

NIS 設定は、Identity Management 内の Directory Server インスタンスの NIS プラグインにあります。ポートを指定するには、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. プラグイン設定を編集し、ポート番号を引数として追加します。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 secret

dn: cn=NIS Server,cn=plugins,cn=config
changetype: modify
add: nsslapd-pluginarg0
nsslapd-pluginarg0: 514

modifying entry "cn=NIS Server,cn=plugins,cn=config"
```

3. Directory Server を再起動して、新しいプラグインの設定を読み込みます。

```
[root@ipaserver ~]# service dirsrv restart
```

13.3. NETGROUPS の作成

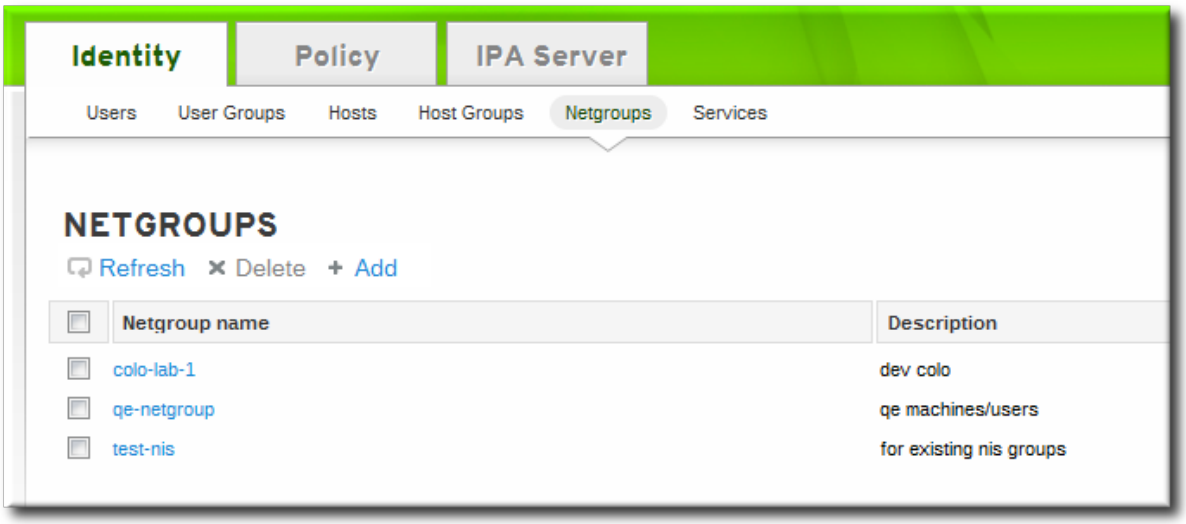
Identity Management のすべての netgroup は基本的に **静的** であるため、グループのメンバーは、手作業で明示的にグループに追加されます。IdM では、グループが他のグループに所属する **ネストされたグループ** を暗黙的に許可します。この場合には、グループに含まれるメンバーはすべて自動的に親グループにも所属します。

netgroup は、グループ自体の作成、そのグループへのメンバーの追加の 2 つの手順で追加できます。

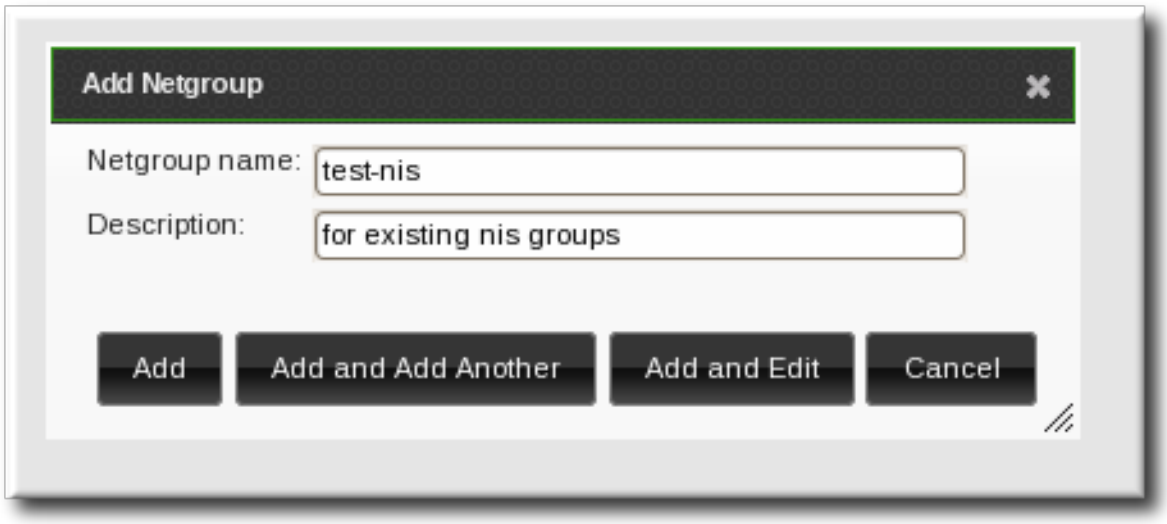
13.3.1. Netgroup の追加

13.3.1.1. Web UI の使用

1. **Identity** タブを開き、**Netgroups** サブタブを選択します。
2. netgroups 一覧の上部にある **Add** をクリックします。



3. netgroup に一意の名前と説明の両方を入力します。名前と説明の両方が必要です。



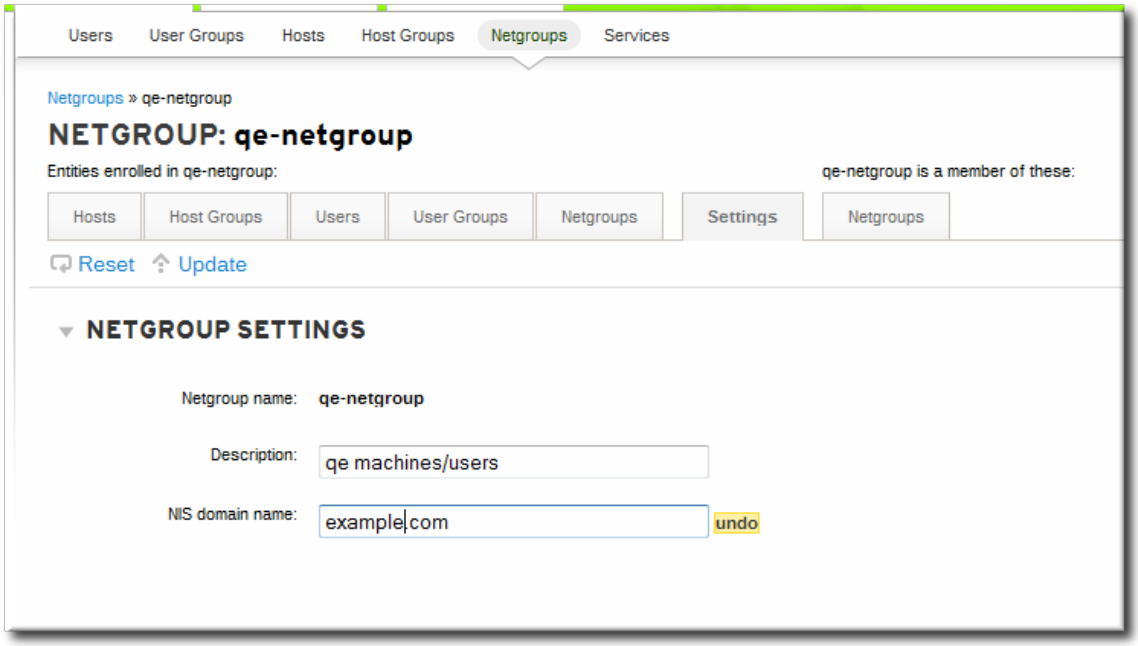
グループ名は、IdM ドメインの netgroup に使用する IDで、作成後に変更できません。この名前にはスペースを含めることはできませんが、アンダースコア (_) のような他の区切り文字は使用できます。

4. **Add and Edit** ボタンをクリックすると、すぐに netgroup の編集ページに移動します。

5. 任意で、netgroup の NIS ドメインを設定します。デフォルトではこれは IdM ドメインですが、変更できます。

   a. **Settings** タブをクリックします。

   b. **NIS domain name** フィールドで別の NIS ドメイン名を入力します。



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 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 の使用」にあるように、メンバーを追加します。

13.3.1.2. コマンドラインの使用

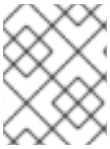
新しい netgroups は **netgroup-add** コマンドを使用して追加されます。これによりグループのみが追加

され、メンバーは別に追加されます。グループ名とグループの説明の 2 つの属性が常に必要になります。これらの属性が引数として指定されていない場合には、スクリプトでグループ名と説明を入力するように求められます。また、グループに使用する NIS ドメイン名を設定するオプションもあります。デフォルトは IdM ドメインですが、ネットワーク設定に応じて、別の設定を指定できます。

```
$ ipa netgroup-add --desc="description"  [--nisdomain=domainName]  groupName
```

以下に例を示します。

```
# ipa netgroup-add --desc="my new netgroup" example-netgroup
# ipa netgroup-add-member --hosts=ipa.example.com example-netgroup
# ypcat -d example.com -h ipa.example.com netgroup
(ipa.example.com,-,example.com)
```



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**。

13.3.2. Netgroup メンバーの追加

注記

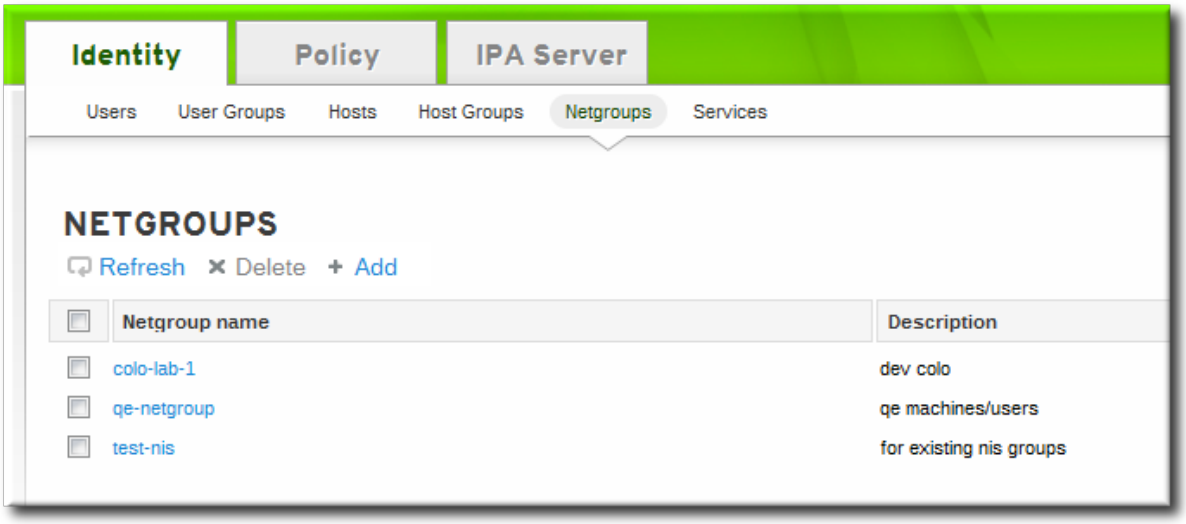
netgroup は、ユーザーグループ、ホストグループ、およびその他の netgroup をメンバーとして追加できます。このようなグループは、**ネスト化されます**。

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、最長で数分かかる場合があります。これは特に、ネストされたグループのメンバーが 500 を超える仮想マシンに該当します。

ネスト化されたグループを作成する場合は、**再帰** グループを作成しないようにしてください。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。

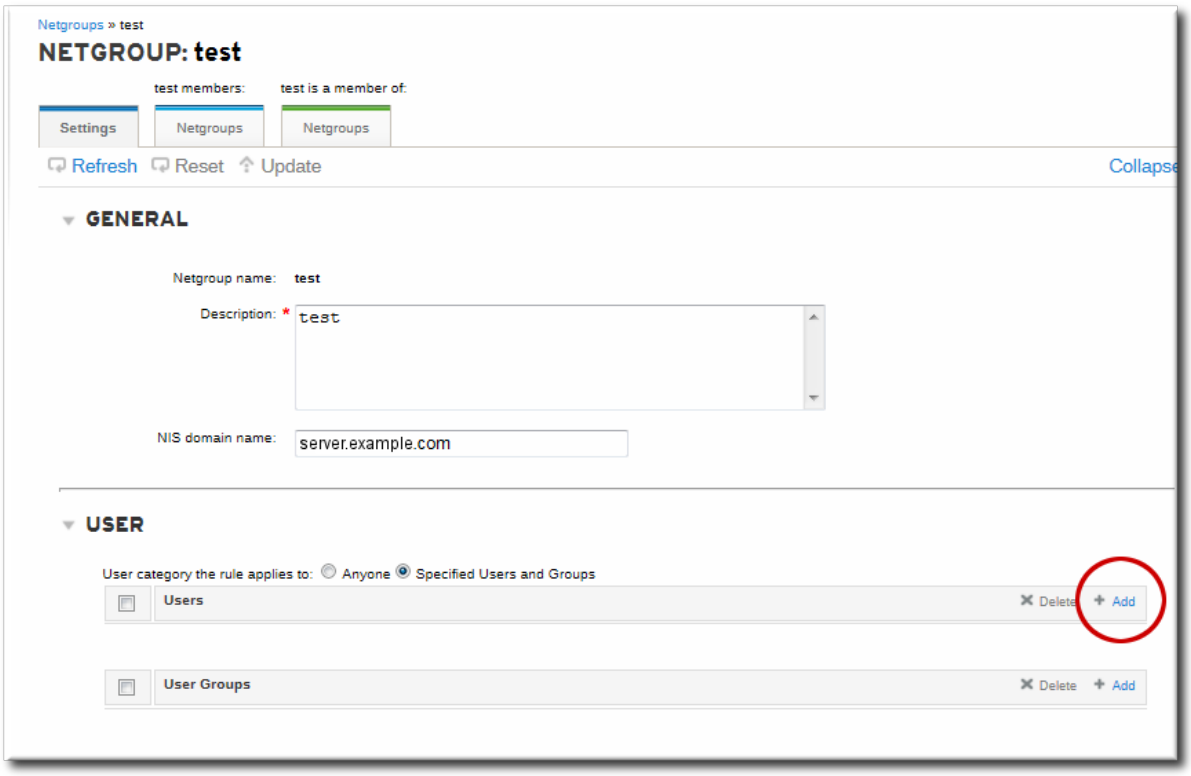
13.3.2.1. Web UI の使用

1. **Identity** タブを開き、**Netgroups** サブタブを選択します。
2. メンバーを追加する netgroup 名をクリックします。

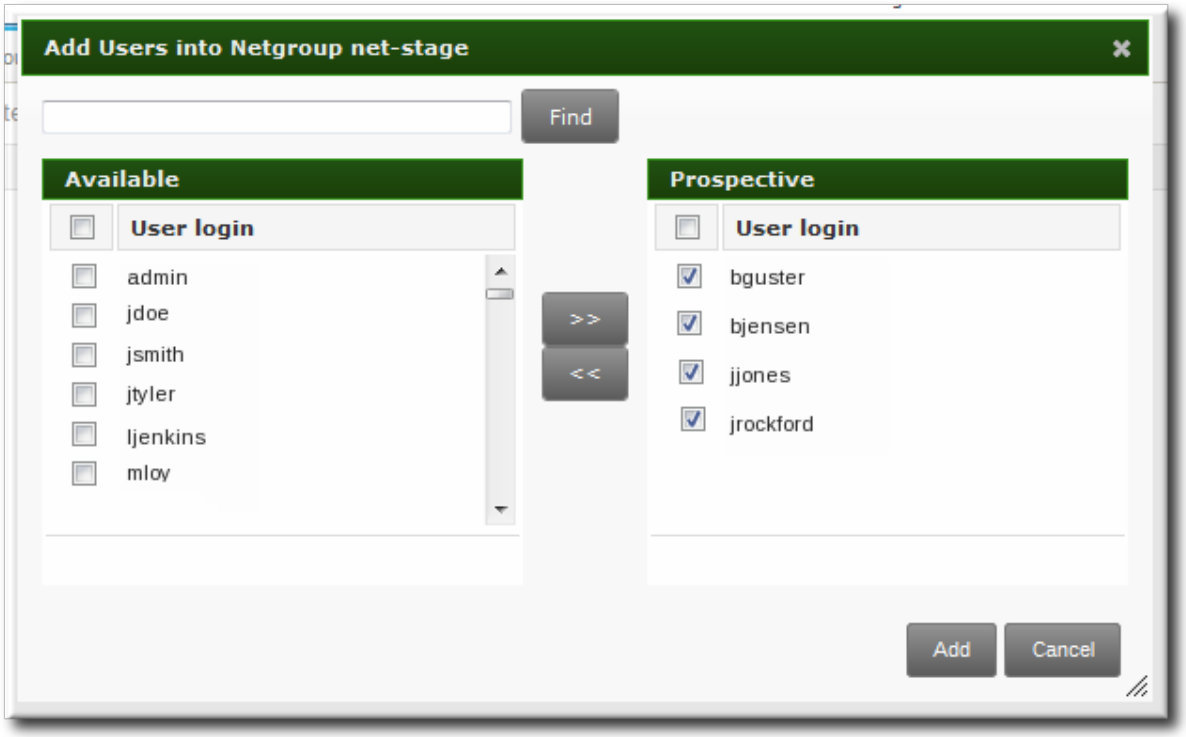


3. 追加する netgroup メンバータイプのタブを選択します。netgroup は、ユーザー、ユーザーグループ、ホスト、ホストグループ、およびその他の netgroup をメンバーとして指定できます。

4. タスクエリア上部にある **Add** をクリックします。



5. 追加するユーザーの名前の横にあるチェックボックスをクリックし、右矢印ボタン (**>>**) をクリックし、名前を選択項目のボックスに移動します。



6. **追加** ボタンをクリックします。

13.3.2.2. コマンドラインの使用

グループが設定されたら、**netgroup-add-member** コマンドを使用して netgroup メンバーの追加を開始します。ユーザー、グループ、ホスト、ホストグループ、その他の netgroup はすべて netgroup エントリーに追加できます。編集する NIS グループのエントリー名は通常、コマンドの最後にあります。

```
# ipa netgroup-add-member --users=users --groups=groups --hosts=hosts --hostgroups=hostGroups
--netgroups=netgroups  groupName
```

複数のメンバーを設定するには、オプションを指定してコンマ区切りの一覧を使用します。たとえば、以下は、ユーザー 2 つと、他の構成のホスト 2 つを設定します。

```
# ipa netgroup-add-member --users=jsmith,bjensen --groups=ITadmin --
hosts=host1.example.com,host2.example.com --hostgroups=EngDev --netgroups=nisgroup2
example-group
```

13.4. 自動マウントマップの NIS クライアントへの公開

システムで NIS サービスが有効になっていると、IdM サーバーは、NIS ドメインを IdM ドメイン名に設定し、NIS ドメインの passwd、group および netgroup マップとして、IdM ユーザー、グループ、netgroup を追加できます。

自動マウントマップがすでに定義されている場合は、これらのマップを Identity Management の NIS 設定に手動で追加して、NIS クライアントに公開する必要があります。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で管理されます。これはコンテナーエントリーで、NIS サーバーによって使用される各 NIS ドメインとマップは、そのコンテナーの下にある子エントリーとして設定されます。NIS ドメインエントリーには、NIS ドメイン名、NIS マップ名、NIS マップのコンテンツとして使用するディレクトリーエントリーの検索方法、および NIS マップのキーおよび値として使用する属性が必要です。これら設定のほとんどは、各マップで同じものになります。

IdM サーバーは、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。

NIS のドメインおよびマップは、**ldapadd** などの LDAP ツールを使用してディレクトリーを直接編集して、追加します。たとえば、以下は、**default** という名前の場所に、**nisserver** という名前のサーバーに、**auto.example** という名前で指定した自動マウントマップを追加します。

```
[root@server ~]# ldapadd -h nisserver.example.com -x -D "cn=Directory Manager" -w secret

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
```

設定された全マップに対して、同様の追加操作を実行する必要があります。

13.5. NIS から IDM への移行

NIS から Identity Management への直接移行パスはありません。NIS から IdM への移行は手動のプロセスで、IdM への netgroup エントリーの設定、既存データの NIS からのエクスポート、そのデータの IdM へのインポートの 3 つの手順で主に構成されます。IdM 環境の設定方法やデータのエクスポート方法には、複数のオプションがあり、最適なオプションは、データの種類と、利用する全ネットワーク環境によって異なります。

13.5.1. IdM での netgroup エントリーの準備

最初のステップでは、NIS が管理するアイデンティティーの種類を特定します。NIS サーバーは、ユーザーエントリーまたはホストエントリーのいずれかに使用されることが多く、両方に使用されることはなく、データの移行プロセスを簡素化できます。

ユーザーエントリーの場合

NIS の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** のような) クライアントには NIS netgroup が必要ですが、多くのクライアントは代わりに Unix グループを使用できます。netgroup が必要ない場合は、IdM で対応するユーザーアカウントを作成し、netgroup を完全に削除するだけです。それ以外の場合は、IdM にユーザーエントリーを作成し、IdM 管理の netgroup を作成し、そのユーザーをメンバーとして追加します。この操作は、「Netgroups の作成」に説明があります。

ホストエントリーの場合

ホストグループが IdM に作成されるたびに、一致するシャドウの NIS グループが自動的に作成されます。これらの netgroup は、**ipa-host-net-manage** コマンドを使用して管理できます。

直接変換の場合

IdM に、すべての NIS ユーザーおよびホストに、完全に一致するエントリーがある状態で、正確な変換が必要になる場合があります。この場合には、元の NIS 名を使用して各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。または、ユーザーとホストを IdM グループまたは他の netgroup に追加し、それらのグループを netgroup に追加します。

13.5.2. Identity Management での NIS リスナーの有効化

IdM Directory Server は、制限ありの NIS サーバーとして機能します。**slapi-nis** プラグインは、NIS リスナーを設定し、着信 NIS 要求を受信して Directory Server 内の NIS マップを管理します。Identity Management は、以下の 3 つの NIS マップを使用します。

- passwd
- group
- netgroup

IdM を中間 NIS サーバーとして使用すると、NIS のクライアントとデータの移行時に、NIS の要求を妥当に処理できるようになります。

slapi-nis プラグインはデフォルトでは無効になっています。Identity Management の NIS を有効にするには、以下を実行します。

1. IdM 管理ユーザーとして新しい Kerberos 認証情報を取得します。

```
[root@ipaserver ~]# kinit admin
```

2. 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
```

3. DNS サービスおよび Directory Server サービスを再起動します。

```
[root@server ~]# service rpcbind restart
[root@server ~]# service dirsrv restart
```

13.5.3. 既存 NIS データのインポートおよびエクスポート

NIS には、ユーザー、グループ、DNS およびホスト、netgroups、および自動マウントマップの情報を追加できます。これらのエントリータイプはどれでも IdM サーバーに移行できます。

移行には、**ypcat** を使用してデータをエクスポートし、その出力をループして、対応の **ipa *-add** コマンドで IdM エントリーを作成します。これは手動で行うことができますが、スクリプト化するのが最も簡単です。これらの例では、シェルスクリプトを使用します。

13.5.3.1. ユーザーエントリーのインポート

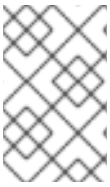
/etc/passwd ファイルには、すべての NIS ユーザー情報が含まれます。これらのエントリーは、NIS エントリーをミラーリングする UID、GID、gecos、shell、ホームディレクトリー、名前属性で IdM ユーザーアカウントを作成するのに使用できます。

たとえば、以下は **nis-user.sh** の例です。

```
#!/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
```

これは、特定の NIS ドメインに対して実行できます。

```
[root@nis-server ~]# kinit admin
[root@nis-server ~]# ./nis-user.sh nisdomain nis-master.example.com
```



注記

このスクリプトでは、ユーザーパスワードは移行されません。代わりに、一時パスワードが作成され、ユーザーの次回ログイン時に変更するようにプロンプトが表示されます。

13.5.3.2. グループエントリーのインポート

/etc/group ファイルには、全 NIS グループ情報が含まれます。このエントリーは、NIS エントリーをミラーリングする GID、gecos、shell、home ディレクトリー、および name 属性を使用して、IdM ユーザーグループアカウントを作成できます。

たとえば、以下は **nis-group.sh** の例です。

```
#!/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
```

これは、特定の NIS ドメインに対して実行できます。

```
[root@nis-server ~]# kinit admin
[root@nis-server ~]# ./nis-group.sh nisdomain nis-master.example.com
```

13.5.3.3. ホストエントリーのインポート

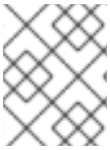
/etc/hosts ファイルには、すべての NIS ホスト情報が含まれます。このエントリーを使用して、NIS エントリーをミラーリングする IdM ホストアカウントを作成できます。

たとえば、以下は **nis-hosts.sh** の例です。

```
#!/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
```

これは、特定の NIS ドメインに対して実行できます。

```
[root@nis-server ~]# kinit admin
[root@nis-server ~]# ./nis-hosts.sh nisdomain nis-master.example.com
```



注記

このスクリプトの例では、エイリアスの使用など、特別なホストシナリオには対応していません。

13.5.3.4. Netgroup エントリーのインポート

/etc/netgroup ファイルには、全 NIS netgroup 情報が含まれます。このエントリーを使用して、NIS エントリーをミラーリングする IdM netgroup アカウントを作成できます。

たとえば、以下は **nis-netgroup.sh** の例です。

```
#!/bin/sh
# 1 is the nis domain, 2 is the nis master server
ypcat -k -d $1 -h $2 netgroup > /dev/shm/nis-map.netgroup 2>&1

IFS=$'\n'
for line in $(cat /dev/shm/nis-map.netgroup); do
        IFS=' '
        netgroupname=$(echo $line|awk '{print $1}')
        triples=$(echo $line|sed "s/^$netgroupname //")
        echo "ipa netgroup-add $netgroupname --desc=NIS_NG_$netgroupname"
        if [ $(echo $line|grep "(,"|wc -l) -gt 0 ]; then
                echo "ipa netgroup-mod $netgroupname --hostcat=all"
        fi
        if [ $(echo $line|grep ",,"|wc -l) -gt 0 ]; then
                echo "ipa netgroup-mod $netgroupname --usercat=all"
        fi

        for triple in $triples; do
                triple=$(echo $triple|sed -e 's/-//g' -e 's/(//' -e 's/)//')
                if [ $(echo $triple|grep ",.*,"|wc -l) -gt 0 ]; then
                        hostname=$(echo $triple|cut -f1 -d,)
                        username=$(echo $triple|cut -f2 -d,)
                        domain=$(echo $triple|cut -f3 -d,)
                        hosts=""; users=""; doms="";
                        [ -n "$hostname" ] && hosts="--hosts=$hostname"
                        [ -n "$username" ] && users="--users=$username"
                        [ -n "$domain"   ] && doms="--nisdomain=$domain"
                        echo "ipa netgroup-add-member $hosts $users $doms"
                else
                        netgroup=$triple
                        echo "ipa netgroup-add $netgroup --desc=NIS_NG_$netgroup"
                fi
        done
done
```

「NIS および Identity Management の概要」 で簡単に説明したように、NIS エントリーはトリプルと呼ばれる 3 つの値セットに存在します。トリプルは **host,user,domain** ですが、すべてのコンポーネントが必要な訳ではありません。通常、トリプルで、ホストとドメイン、またはユーザーとドメインのみを定義します。このスクリプトの記述の仕方から、**ipa netgroup-add-member** コマンドは常に、netgroup にホスト、ユーザー、ドメインのトリプルを追加します。

```
if [ $(echo $triple|grep ",.*,"|wc -l) -gt 0 ]; then
        hostname=$(echo $triple|cut -f1 -d,)
```

```
        username=$(echo $triple|cut -f2 -d,)
        domain=$(echo $triple|cut -f3 -d,)
        hosts=""; users=""; doms="";
        [ -n "$hostname" ] && hosts="--hosts=$hostname"
        [ -n "$username" ] && users="--users=$username"
        [ -n "$domain"   ] && doms="--nisdomain=$domain"
        echo "ipa netgroup-add-member $hosts $users $doms"
```

要素が抜けている箇所は、空白として追加されるので、トリプルは正しく移行されます。たとえば、**server,,domain** のトリプルの場合は、メンバー追加コマンドのオプションは、**--hosts=server --users="" --nisdomain=domain** です。

これは、NIS ドメインと NIS サーバーを指定して、特定の NIS ドメインに対して実行できます。

```
[root@nis-server ~]# kinit admin
[root@nis-server ~]# ./nis-hosts.sh nisdomain nis-master.example.com
```

13.5.3.5. Automount マップのインポート

自動マウントマップは実際には、場所 (親エントリー) と関連のキーおよびマップを定義する入れ子および相互関連のエントリーになります。

NIS および IdM エントリーのデータは同じですが、データの定義方法は異なります。NIS 情報は、エクスポートして、自動マウントの場所と関連マップの LDAP エントリー構築に使用します。次に、マップ用に設定されたすべてのキーのエントリーを作成します。

このスクリプトは、他の NIS 移行スクリプトの例とは異なり、移行する NIS ドメインおよびサーバー以外に自動マウントの場所、マップ名もオプションとして指定できます。

```
#!/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=nisdomain.example.com+nis-map=$4,cn=NIS Server,cn=plugins,cn=config
objectClass: extensibleObject
nis-domain: $3
nis-map: $4
nis-base: automountmapname=$4,cn=nis,cn=automount,$basedn
nis-filter: (objectclass=*)
nis-key-format: %{automountKey}
nis-value-format: %{automountInformation}
EOF
ldapadd -x -h $3 -D "cn=directory manager" -w secret -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
```

これは、特定の NIS ドメインに対して実行できます。

```
[root@nis-server ~]# kinit admin
[root@nis-server ~]# ./nis-hosts.sh location nisdomain nis-master.example.com map
```

13.5.4. IdM に NIS ユーザー認証の弱度のパスワード暗号化を設定する手順

NIS サーバーは、CRYPT パスワードハッシュを処理できます。既存の NIS サーバーを IdM (およびその基盤となる LDAP データベース) に移行した後も、NIS 対応の CRYPT パスワードを保持する必要がある場合があります。ただし、デフォルトでは LDAP サーバーは CRYPT ハッシュを使用しません。salted SHA (SSHA) または SSHA-256 を使用します。389 Directory Server パスワードのハッシュを変更しない場合には、NIS ユーザーは IdM ドメインに対して認証できず、パスワードが原因で **kinit** に失敗します。

基礎となる 389 Directory Server がパスワードハッシュとして CRYPT を使用するように設定するには、**ldapmodify** を使用して ***passwordStorageScheme*** 属性を変更します。

```
[root@server ~]# ldapmodify -D "cn=directory server" -w secret -p 389 -h ipaserver.example.com

dn: cn=config
changetype: modify
replace: passwordStorageScheme
passwordStorageScheme: crypt
```

注記

パスワードストレージスキームを変更すると、このスキームは新しいパスワードにのみ適用されます。遡って、既存のパスワードに使用する暗号化メソッドは変更されません。

パスワードハッシュに弱度の暗号化が必要な場合には、ユーザーのパスワードに弱度のパスワードハッシュを使用できるように、早い段階で設定を変更することを推奨します。

第14章 アイデンティティー: フォレスト間の信頼との統合 (テクノロジープレビュー)

Kerberos は *信頼* という概念を実装しています。信頼では、Kerberos レルムからのプリンシパルが別の Kerberos レルムのサービスにチケットを要求できます。プリンシパルはこのチケットを使って、別のレルムに属するマシン上のリソースに対して認証を行うことができます。

Kerberos には、*レルム間の信頼* と呼ばれる、2 つのレルム間の関係を作成する機能があります。この信頼の一部となっているレルムは、共有のチケットとキーのペアを使用します。1 つのレルムのメンバーが両方のレルムのメンバーとして認識されるようなります。

Active Directory と Identity Management の両方が、Kerberos、LDAP、DNS、証明書サービスなどのさまざまなコアサービスを管理します。このため、Kerberos レルム間の信頼を確立するだけでは、レルムのユーザーが別のレルムにあるリソースにアクセスするには不十分になります。別の通信レベルでのサポートも必要になってきます。このような目的を実現するため、IdM を使用すると、IdM ドメインと AD ドメインの間で *フォレスト間の信頼* を設定できます。フォレスト間の信頼は、2 つのフォレスト Root ドメイン間で確立された信頼のことで、異なるフォレストからのユーザーとサービスが通信できるようにします。

注記

複数の AD ドメインは、1 つの *Active Directory フォレスト* にまとめることができます。このフォレストの root ドメインは、フォレスト内で作成される最初のドメインになります。IdM ドメインは既存の AD フォレストに含めることができないので、常に別個のフォレストとみなされます。

Red Hat Enterprise Linux 6 のフォレスト間の信頼 (テクノロジープレビュー機能)

Red Hat Enterprise Linux 6 では、フォレスト間の信頼機能はテクノロジープレビュー機能として提供されます。Red Hat は、Red Hat Enterprise Linux 6 IdM クライアントを Red Hat Enterprise Linux 7 IdM サーバーに接続してフォレスト間の信頼機能を確保することを推奨します。信頼は、Red Hat Enterprise Linux 7 を実行するサーバーで完全にサポートされています。Red Hat Enterprise Linux 6 クライアントを Red Hat Enterprise Linux 7 サーバーに接続してフォレスト間の信頼を確立する設定も、完全にサポートされています。このような設定では、クライアント側に最新の Red Hat Enterprise Linux 6 を、サーバー側に最新の Red Hat Enterprise Linux 7 を使用することを推奨します。

Red Hat では、Red Hat Enterprise Linux 6 でのフォレスト間の信頼機能を、テクノロジープレビュー機能からサポート対象機能にアップグレードしていません。特定の AD デプロイメントで、Red Hat Enterprise Linux 6 のフォレスト間の信頼機能が動作しない場合には、最新の Red Hat Enterprise Linux 7 IdM バージョンを使用し、特定の設定で Red Hat Enterprise Linux 7 をアップグレードする必要があるかどうかを確認してください。

Red Hat Enterprise Linux 7 のフォレスト間の信頼の詳細にわたる説明が含まれるドキュメントについては、『Red Hat Enterprise Linux 7 『Windows 統合ガイド』』を参照してください。

Red Hat Enterprise Linux 6 のフォレスト間の信頼機能の概要

Red Hat Enterprise Linux 6 のフォレスト間の信頼機能には、以下のような機能が含まれます。

- 1 つの AD フォレストへの信頼を確立する。
- 信頼された AD フォレストの rootドメインからユーザーの IdM リソースにアクセスできる。

Red Hat Enterprise Linux 6 のフォレスト間の信頼機能は、以下の機能がありません。

- ログインシェルまたはホームディレクトリーなど AD ユーザーのデフォルトの属性を一元的に上書きする。これには、Red Hat Enterprise Linux 7 で IdM を使用して ID ビューをデプロイします。

- レガシークライアントの互換性ツリーを使用して、AD ユーザーおよびグループを公開する。レガシークライアントが AD ユーザーおよびグループにアクセスできるようにするには、Red Hat Enterprise Linux 7 で IdM を使用します。

信頼と同期
信頼と同期は、IdM ドメインと AD ドメイン統合するための、根本的に異なるアプローチです。どちらのアプローチも、AD ドメインのユーザーが透過的に Linux システムおよびサービスにアクセスできるようにすると共に、このようなシステムに関連する Linux システムとポリシーを一元管理できる利点があります。

信頼ベースおよび同期ベースのソリューションの比較については、『Red Hat Enterprise Linux 7 『Windows 統合ガイド』』を参照してください。Red Hat Enterprise Linux 6 での同期を使用した AD との統合に関する情報は、「15章 *アイデンティティー: 同期による Microsoft Active Directory との統合*」を参照してください。

第15章 アイデンティティー: 同期による MICROSOFT ACTIVE DIRECTORY との統合

Identity Management は有効な **同期機能** を使用して、Active Directory ドメインと、IdM ドメインに格納されているユーザーデータを統合します。パスワードなどの重要なユーザー属性はサービス間で同期されます。

Active Directory と IdM ドメインの同期機能は、IdM サーバーが初回インストール時に継承されます。同期プロセスは、IdM サーバーと Active Directory ドメインコントローラーとの間で **合意** を作成して設定します。

本章では、同期の設定方法、Active Directory と IdM を統合する設定方法、および Active Directory ドメイン内にある Windows システムで IdM ドメインを認識させる設定方法について説明します。

15.1. サポート対象の WINDOWS プラットフォーム

エントリーの同期は、Windows サーバーに接続してそこからディレクトリーデータを取得するのにフックを使用するレプリケーションと同様のプロセスで実行されます。

パスワードの同期は、Windows サーバーにインストールされ、Identity Management サーバーと通信する Windows サービスで実行されます。

次の Windows サーバーでは、エントリーとパスワードの両方の同期がサポートされています。

- Windows Server 2008 R2
- Windows Server 2012 R2

パスワード同期サービスで Windows と連携するバージョンは 1.1.5 です。これは、Red Hat Network の Red Hat Directory Server のダウンロードの部分で利用できます。

15.2. ACTIVE DIRECTORY および IDENTITY MANAGEMENT の概要

IdM ドメイン内では、情報はデータマスター (サーバーとレプリカ) 間で信頼性と予測性のある方法でコピーされ、複数のサーバーとレプリカ間で共有されます。このプロセスを **レプリケーション** といいます。

同様のプロセスは、IdM ドメインと Microsoft Active Directory ドメイン間でデータを共有するために使用できます。これが **同期** です。

同期は、Active Directory と Identity Management の間で、ユーザーデータをコピーするプロセスのことです。

同期は、IdM サーバーと Active Directory ドメインコントローラー間の **合意** で定義されます。同期合意は、同期可能なユーザーエントリー (同期するサブツリーや、ユーザーエントリーで必須のオブジェクトクラスなど) を特定するのに必要な全情報、アカウント属性の処理方法を定義します。同期合意は、デフォルト値で作成されますが、特定ドメインのニーズに合わせて調整が可能です。2 つのサーバーで同期が行われる場合に、この 2 つのサーバーは **ピア** と呼ばれます。

同期は通常、**双方向** で行われます。情報は、IdM ドメインと Windows ドメイン間で送受信され、このプロセスは IdM サーバーとレプリカが情報を共有する方法によく似ています。同期は、1 方向のみで行われるように設定することもできます。これは **一方向** の同期と呼ばれます。

データの競合が発生しないように、1 つの Identity Management サーバーと 1 つの Active Directory ドメインコントローラーの間で同期が設定されます。Identity Management サーバーは変更を IdM ドメインに伝播し直し、ドメインコントローラーは変更を Windows ドメインに伝播し直します。

IdM 同期には、以下のような主要な機能があります。

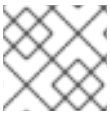
- 同期操作は 5 分ごとに実行されます。
- 同期が設定できるのは、Active Directory ドメイン 1 つのみとなっています。複数のドメインには対応していません。
- 同期が設定できるのは、Active Directory ドメイン **1 つ** のみとなっています。
- ユーザー情報のみが同期されます。
- ユーザー属性とパスワードの両方を同期することができます。
- 変更は双方向ですが (Active Directory から IdM、IdM から Active Directory の両方)、アカウントの作成または追加は、Active Directory から Identity Management への一方向のみになります。新しいアカウントが Active Directory に作成されると、自動的に IdM に対して同期されます。ただし、ユーザーアカウントを IdM で作成した場合には、同期の前に Active Directory にも作成する必要があります。
- アカウントロック情報はデフォルトで同期され、1 つのドメインで無効にされているユーザーアカウントは他方のドメインでも無効にされます。
- パスワードの変更は即時に有効になります。

Active Directory ユーザーが IdM に同期される場合に、特定の属性 (Kerberos および POSIX 属性を含む) では IPA 属性がユーザーエントリーに自動的に追加されます。この属性は、ドメイン内で IdM が使用します。対応する Active Directory ユーザーエントリーには、同期されません。

同期プロセスの一環で、同期データの一部が変更される可能性があります。たとえば、IdM ドメインに同期する場合に、特定の属性を自動的に Active Directory ユーザーアカウントに追加できます。このような属性の変更は、同期合意の一部として定義します。これについては、「「ユーザーアカウント属性の同期動作の変更」」で説明されています。

15.3. 同期された属性の概要

Identity Management は、IdM と Active Directory ユーザーエントリーの間で、ユーザー属性のサブセットを同期します。エントリーに含まれる他の属性は、Identity Management または Active Directory のどちらにある場合でも、同期時に無視されます。



注記

ほとんどの POSIX 属性は同期されません。

Active Directory の LDAP スキーマと、Identity Management で使用される 389 Directory Server の LDAP スキーマ間には、スキーマは大きな異なりますが、属性は同じものが多数あります。このようなの属性は、Active Directory と IdM ユーザーエントリー間で同期されるだけで、属性名や値の形式には変更が加えられません。

Identity Management および Windows サーバーで同一のユーザースキーマ

- cn[5]

- physicalDeliveryOfficeName
- description
- postOfficeBox
- destinationIndicator
- postalAddress
- facsimileTelephoneNumber
- postalCode
- givenname
- registeredAddress
- homePhone
- sn
- homePostalAddress
- st
- initials
- street
- l
- telephoneNumber
- mail
- teletexTerminalIdentifier
- mobile
- telexNumber
- o
- title
- ou
- usercertificate
- pager
- x121Address

一部の属性には異なる名前が使用されていますが、IdM (389 Directory Server を使用) と Active Directory の間には直接的な対応関係があります。このような属性は、同期プロセスで **マッピングされます** 。

表15.1 Identity Management と Active Directory との間でマッピングされるユーザースキーマ

ID 管理	Active Directory
cn[a]	name
nsAccountLock	userAccountControl
ntUserDomainId	sAMAccountName
ntUserHomeDir	homeDirectory
ntUserScriptPath	scriptPath
ntUserLastLogon	lastLogon
ntUserLastLogoff	lastLogoff
ntUserAcctExpires	accountExpires
ntUserCodePage	codePage
ntUserLogonHours	logonHours
ntUserMaxStorage	maxStorage
ntUserProfile	profilePath
ntUserParms	userParameters
ntUserWorkstations	userWorkstations

[a] Identity Management から Active Directory に同期時には、***cn*** は直接 (***cn*** から ***cn*** へ) マッピングされます。Active Directory から同期すると、***cn*** は、Active Directory の ***name*** 属性から Identity Management の ***cn*** 属性にマッピングされます。

15.3.1. Identity Management と Active Directory との間のユーザースキーマの相違点

属性が Active Directory と IdM の間で正常に同期される場合でも、Active Directory および Identity Management が基となる X.500 オブジェクトクラスを定義する方法には依然として違いがあり場合があります。この定義方法の相違点により、LDAP サービスが違うと、データの処理方法が異なる可能性があります。

このセクションでは、Active Directory および Identity Management のドメイン間で同期可能な属性を処理する方法に、Active Directory と Identity Management ではどのような違いがあるのかを説明します。

15.3.1.1. cn 属性の値

389 Directory Server では、***cn*** 属性に複数の値を設定できますが、Active Directory ではこの属性には単一の値しか設定できません。Identity Management の ***cn*** 属性が同期されると、単一の値のみが Active Directory ピアに送信されます。

これを同期との関連で見ると、***cn*** 値が Active Directory エントリーに追加され、その値が Identity Management の ***cn*** の値のいずれでもない場合には、Identity Management の ***cn*** 値はすべて単一の Active Directory 値で上書きされます。

もう 1 つの重要な相違点として、Active Directory では ***cn*** 属性をその命名属性として使用するのに対し、Identity Management は ***uid*** を使用する点があります。つまり、***cn*** 属性が Identity Management で編集する可能性がある場合には、エントリーの名前が完全に (および間違って) 変更されてしまう可能性があります。この ***cn*** の変更が Active Directory エントリーに書き込まれると、エントリーの名前が変更され、新しい名前のエントリーで Identity Management に書き込まれます。

15.3.1.2. street および streetAddress の値

Active Directory はユーザーの住所に ***streetAddress*** 属性を使用します。これは、389 Directory Server が ***street*** 属性を使用する方法に相当します。Active Directory および Identity Management が ***streetAddress*** および ***street*** 属性を使用する方法には 2 つの重要な相違点があります。

- 389 Directory Server では、***streetAddress*** は、 ***street*** のエイリアスです。Active Directory にも ***street*** 属性がありますが、***streetAddress*** のエイリアスではなく、独立した値を保持することができる個別の属性です。

- Active Directory は ***streetAddress*** と ***street*** を単一値の属性として定義しますが、389 Directory Server は RFC 4519 に指定されているように ***street*** を複数値の属性として定義します。

389 Directory Server および Active Directory が ***streetAddress*** および ***street*** 属性を処理する方法が異なるため、Active Directory と Identity Management で address 属性を設定する場合には以下の 2 つのルールに従う必要があります。

- 同期プロセスは、Active Directory エントリーの ***streetAddress*** から Identity Management の ***street*** にマッピングされます。競合を回避するために、***street*** 属性は Active Directory では使用しないようにしてください。

- Identity Management の ***street*** 属性値は 1 つだけ、Active Directory に同期されます。***streetAddress*** 属性が Active Directory で変更され、新しい値が Identity Management に存在しない場合には、Identity Management のすべての***street*** 属性値が新しい Active Directory の値に置き換えられます。

15.3.1.3. initials 属性の制約

initials 属性の場合には、Active Directory は最大長 6 文字の制限を課しますが、389 Directory Serve には長さ制限がありません。Identity Management に 7 文字以上の ***initials*** 属性が 追加されると、この値は Active Directory エントリーとの同期時にカットされます。

15.3.1.4. surname (sn) 属性の要求

Active Directory では、surname 属性なしで **person** エントリーを作成できます。ただし、RFC 4519 では、**person** オブジェクトクラスには surname 属性が必要と定義されていますが、これは、Directory Server で使用される定義です。

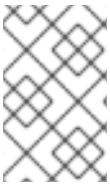
Active Directory の **person** エントリーが surname 属性なしで作成される場合には、このエントリーは、オブジェクトクラス違反で失敗するため、IdM には同期されません。

15.3.2. Active Directory エントリーおよび RFC 2307 属性

Windows は、無作為に選択された一意の **セキュリティー ID (SID)** を使用してユーザーを特定します。これらの SID はブロックまたは範囲で割り当てられ、Windows ドメインでさまざまなシステムユーザータイプを特定します。Identity Management と Active Directory 間でユーザーの同期を行うと、ユーザーの Windows SID は、Identity Management エントリーで使用される Unix UID にマッピングされます。言い換えると、Windows SID は Windows エントリーで唯一の ID で、対応の UNIX エントリーでは ID としてマッピングに使用されます。

Active Directory ドメインが Unix 形式のアプリケーションまたはドメインと対話すると、Active Directory ドメインは Unix または Unix のサービスを使用して Unix 形式の ***uidNumber*** および ***gidNumber*** 属性を有効化できます。これにより、Windows ユーザーエントリーは RFC 2307 のこれらの属性の仕様に準拠できます。

ただし、***uidNumber*** および ***gidNumber*** 属性は、Identity Management エントリーの ***uidNumber*** および ***gidNumber*** 属性として実際には使用されません。Identity Management の ***uidNumber*** と ***gidNumber*** 属性は、Windows ユーザーの同期時に生成されます。



注記

Identity Management で定義され、使用される ***uidNumber*** および ***gidNumber*** 属性は、Active Directory エントリーで定義され、使用される ***uidNumber*** と ***gidNumber*** 属性とは異なり、数字にも関連性はありません。

15.4. 同期用の ACTIVE DIRECTORY の設定

ユーザーアカウントのみの同期が IdM で有効になっているので、必要な操作は同期合意 (「同期合意の作成」) の設定だけです。ただし、Active Directory は、Identity Management サーバーが接続できるように設定する必要があります。

15.4.1. 同期用の Active Directory ユーザーの作成

Windows サーバーでは、IdM サーバーが Active Directory ドメインに接続するために使用するユーザーを作成する必要があります。

Active Directory でのユーザー作成プロセスは、Windows サーバーの文書 (http://technet.microsoft.com/en-us/library/cc732336.aspx) で説明されています。新規のユーザーアカウントには適切な権限を設定する必要があります。

- 同期用のユーザーアカウントには、同期先の Active Directory サブツリーに対して **ディレクトリーに加えられた変更を複製する** 権限を付与します。同期用のユーザーが同期操作を行うには、レプリケーターの権限が必要です。

  レプリケーターの権限は、http://support.microsoft.com/kb/303972 で説明されています。

- 同期ユーザーを **Account Operator** および **Enterprise Read-only Domain Controller** グループのメンバーとして追加します。このユーザーは、全 **Domain Admin** グループに所属する必要はありません。

15.4.2. Active Directory 認証局の設定

Identity Management サーバーは、セキュアな接続を使用して Active Directory サーバーに接続します。この接続には、Active Directory サーバーで利用可能な CA 証明書または CA 証明書チェーンがあることが条件となります。これらの証明書を Identity Management セキュリティーデータベースにインポートして、Windows サーバーを、信頼されるピアとなるように設定できます。

これは技術的には (Active Directory に対して) 外部の CA で実行できますが、大半のデプロイでは Active Directory で利用可能な証明書サービスを使用する必要があります。

Active Directory での証明書サービスの設定、構成手順は、Microsoft のドキュメント (http://technet.microsoft.com/en-us/library/cc772393(v=WS.10).aspx) に記載されています。

15.5. 同期合意の管理

15.5.1. Active Directory および IdM CA 証明書の信頼

Active Directory と Identity Management の両方で、サーバー認証に証明書が使用されます。Active Directory と IdM SSL サーバーの証明書を相互に信頼させるには、これらの証明書の発行元である CA の CA 証明書を、両サーバーで信頼する必要があります。つまり、Active Directory CA 証明書を IdM データベースにインポートし、IdM CA 証明書を Active Directory データベースにインポートする必要があります。

1. Active Directory サーバーで、**http://ipa.example.com/ipa/config/ca.crt** から IdM サーバーの CA 証明書をダウンロードします。

2. Active Directory 証明書データベースに IdM CA 証明書をインストールします。これは、Microsoft 管理コンソールまたは certutil ユーティリティー を使用して実行できます。以下に例を示します。

   ```
   certutil -installcert -v -config "ipaserver.example.com\Example Domain CA" c:\path\to\ca.crt
   ```

   詳細は、Active Directory のドキュメントを参照してください。

3. Active Directory CA 証明書をエクスポートします。

   a. **My Network Places** で CA の配信ポイントを開きます。

   b. セキュリティー証明書ファイル (**.crt** ファイル) をダブルクリックして、**証明書** ダイアログボックスを表示します。

   c. **詳細** タブで、 **ファイルにコピー** をクリックして **証明書のエクスポートウィザード** を起動します。

   d. **次へ** をクリックしてから、**Base-64 encoded X.509 (.CER)** を選択します。

e. エクスポートされたファイルに適切なディレクトリーおよびファイル名を指定します。**Next** をクリックして証明書をエクスポートし、**Finish** をクリックします。

4. Active Directory 証明書を IdM サーバーマシンにコピーします。

5. IdM サーバーの CA 証明書を **http://ipa.example.com/ipa/config/ca.crt** からダウンロードします。

6. Active Directory CA 証明書と IdM CA 証明書の両方を **/etc/openldap/cacerts/** ディレクトリーにコピーします。

7. 証明書のハッシュシンボリックリンクを更新します。

```
cacertdir_rehash /etc/openldap/cacerts/
```

8. **/etc/openldap/ldap.conf** ファイルを編集して、**/etc/openldap/cacerts/** ディレクトリーの証明書を参照および使用するための情報を追加します。

```
TLS_CACERTDIR /etc/openldap/cacerts/
TLS_REQCERT allow
```

15.5.2. 同期合意の作成

同期合意は、Active Directoryドメインへの **接続** を作成するため、IdM サーバー上では **ipa-replica-manage connect** コマンドを使用して作成します。同期合意の作成オプションは、表15.2「同期合意のオプション」に記載されています。

1. Active Directory サーバーと IdM サーバーが、「Active Directory および IdM CA 証明書の信頼」にあるように相互の CA 証明書を信頼していることを確認してください。

2. IdM サーバー上の既存の Kerberos 資格情報を削除します。

```
$ kdestroy
```

3. **ipa-replica-manage** コマンドを使用して Windows 同期合意を作成します。これには、**--winsync** オプションが必要です。パスワードとユーザーアカウントを同期する場合は、**--passsync** オプションも使用して、パスワード同期に使用するパスワードを設定します。

   --binddn および **--bindpwd** オプションを指定すると、IdM が Active Directory サーバーへの接続に使用する Active Directory サーバー上のシステムアカウントにユーザー名およびパスワードを設定します。

```
$ ipa-replica-manage connect --winsync
 --binddn cn=administrator,cn=users,dc=example,dc=com
 --bindpw Windows-secret
 --passsync secretpwd
 --cacert /etc/openldap/cacerts/windows.cer
 adserver.example.com -v
```

4. プロンプトが出されたら、Directory Manager のパスワードを入力します。

5. **任意。**「パスワード同期のセットアップ」に説明されているようにパスワードの同期を設定します。

表15.2 同期合意のオプション

オプション	Description
--winsync	同期合意として指定します。
--binddn	同期 ID の完全なユーザーの DN を指定します。これは、IdM LDAP サーバーが Active Directory にバインドするために使用するユーザーの DN です。このユーザーは Active Directory ドメインに存在しており、Active Directory サブツリーに replicator、read、search、write のパーミッションが必要です。
--bindpw	同期ユーザーのパスワードを指定します。
--passsync	同期を行う Windows ユーザーアカウントのパスワードを指定します。
--cacert	Active Directory CA 証明書の完全パスおよびファイル名を指定します。この証明書は、「Active Directory および IdM CA 証明書の信頼」にエクスポートされます。
--win-subtree	同期するユーザーが含まれる Windows ディレクトリーサブツリーの DN を指定します。デフォルト値は **cn=Users,$SUFFIX** です。

オプション	Description
AD_server_name	Active Directory ドメインコントローラーのホスト名を指定します。

15.5.3. ユーザーアカウント属性の同期動作の変更

同期合意が作成されると、同期プロセスでのユーザーアカウント属性の処理方法に関して特定のデフォルト動作が定義されます。動作のタイプには、ロックアウト属性の処理方法や異なる DN 形式の処理方法などが含まれます。この動作は、同期合意を編集することで変更できます。属性関連のパラメーターの一覧は、表15.3「同期属性の設定」にあります。

同期合意は LDAP サーバーの特殊なプラグインエントリーとして存在し、それぞれの属性動作は LDAP 属性から設定されます。同期の動作を変更するには、**ldapmodify** コマンドを使用して LDAP サーバーのエントリーを直接変更します。

たとえば、デフォルトで IdM と Active Directory との間のアカウントロックアウト属性が同期されますが、***ipaWinSyncAcctDisable*** 属性を編集すると無効にできます。(この属性を変更すると、Active Directory でアカウントが無効な場合でも、IdM で引き続き有効な状態となり、その逆も同様になります)。

```
[jsmith@ipaserver ~]$ ldapmodify -x -D "cn=directory manager" -w password

dn: cn=ipa-winsync,cn=plugins,cn=config
changetype: modify
replace: ipaWinSyncAcctDisable
ipaWinSyncAcctDisable: none

modifying entry "cn=ipa-winsync,cn=plugins,cn=config"
```

表15.3 同期属性の設定

パラメーター	説明	設定可能な値
一般ユーザーアカウントのパラメーター		
ipaWinSyncNewEntryFilter	新規ユーザーエントリーに追加するオブジェクトクラスの一覧を含むエントリーの検索に使用する検索フィルターを設定します。	デフォルトでは **(cn=ipaConfig)** です。
ipaWinSyncNewUserOCAttr	新規ユーザーエントリーに追加するオブジェクトクラスの一覧が実際に含まれる設定エントリーの属性を設定します。	デフォルトは **ipauserobjectclasses** です。
ipaWinSyncHomeDirAttr	POSIX ホームディレクトリーのデフォルトの場所を含むエントリー内の属性を識別します。	デフォルトは **ipaHomesRootDir** です。

パラメーター	説明	設定可能な値
ipaWinSyncUserAttr	Active Directory ユーザーを Active Directory ドメインから同期する時に、特定の値で別の属性を設定してAD ユーザーに追加します。複数値の属性の場合は、属性を複数回設定でき、同期プロセスで、値のすべてがエントリーに追加されます。 **注記** エントリーに属性が存在しない場合に属性値のみが設定されます。属性が存在する場合は、Active Directory エントリーの同期時にエントリーの値が使用されます。	ipaWinSyncUserAttr: **attributeName attributeValue**
ipaWinSyncForceSync	同期できるように、既存の Active Directory ユーザーに一致する既存 IdM ユーザーを自動編集する必要があるかどうかを設定します。IdM ユーザーアカウントに既存の Active Directory の ***samAccountName*** と同じ ***uid*** パラメーターがある場合に、アカウントはデフォルトでは同期 **されません**。この属性は、同期サービスに対して、***ntUser*** および ***ntUserDomainId*** を IdM ユーザーエントリーに自動的に追加し、同期されるように指示します。	true | false
ユーザーアカウントのロックパラメーター		

パラメーター	説明	設定可能な値
ipaWinSyncAcctDisable	アカウントロックアウト属性を同期する方法を設定します。有効にするアカウントロックアウト設定を制御できます。たとえば、**to_ad** は、アカウントロックアウト属性が IdM に設定される場合に、その値が Active Directory に対して同期され、ローカルの Active Directory 値を上書きすることを意味します。デフォルトでは、アカウントロックアウト属性は両ドメインから同期されます。	• both (デフォルト) • to_ad • to_ds • none
ipaWinSyncInactivatedFilter	非アクティブ化された (無効にされた) ユーザーを保持するために使用されるグループの DN 検索用のフィルターを設定します。これは、ほとんどの実装では変更する必要はありません。	デフォルトは **(&(cn=inactivated)(objectclass=groupOfNames))** です。
ipaWinSyncActivatedFilter	アクティブなユーザーを保持するために使用されるグループの DN 検索用のフィルターを設定します。これは、ほとんどの実装では変更する必要はありません。	デフォルトは **(&(cn=activated)(objectclass=groupOfNames))** です。
グループのパラメーター		
ipaWinSyncDefaultGroupAttr	ユーザーのデフォルトグループを確認するために参照する新規ユーザーアカウントの属性を設定します。その後、エントリーのグループ名がユーザーアカウントの ***gidNumber*** の検索に使用されます。	デフォルトは **ipaDefaultPrimaryGroup** です。
ipaWinSyncDefaultGroupFilter	検索フィルターを設定して、グループ名を POSIX の ***gidNumber*** にマッピングします。	デフォルトは **(&(gidNumber=*)(objectclass=posixGroup)(cn=**groupAttr_value**))** です。
レルムのパラメーター		
ipaWinSyncRealmAttr	レルムエントリーにレルム名を含む属性を設定します。	デフォルトは ***cn*** です。
ipaWinSyncRealmFilter	IdM レルム名を含むエントリーの検索に使用する検索フィルターを設定します。	デフォルトは **(objectclass=krbRealmContainer)** です。

15.5.4. 同期された Windows サブツリーの変更

同期合意を作成すると、同期されたユーザーデータベースとして使用する 2 つのサブツリーが自動設定されます。IdM の場合、デフォルトは **cn=users,cn=accounts,$SUFFIX** となり、Active Directory の場合、デフォルトは **CN=Users,$SUFFIX** となります。

--win-subtree オプションを使用して同期合意が作成されると、Active Directory サブツリーの値はデフォルト以外の値に設定できます。この合意の作成後に、**ldapmodify** コマンドを使用し、同期合意エントリー内の ***nsds7WindowsReplicaSubtree*** 値を編集して Active Directory サブツリーを変更できます。

1. **ldapsearch** を使用して、同期合意の名前を取得します。この検索は、エントリー全体ではなく、***dn*** および ***nsds7WindowsReplicaSubtree*** 属性の値のみを返します。

```
[jsmith@ipaserver ~]$ ldapsearch -xLLL -D "cn=directory manager" -w password -p 389 -h
ipaserver.example.com -b cn=config objectclass=nsdswindowsreplicationagreement dn
nsds7WindowsReplicaSubtree

dn:
cn=meToWindowsBox.example.com,cn=replica,cn=dc\3Dexample\2Cdc\3Dcom,cn=mapping
tree,cn=config
nsds7WindowsReplicaSubtree: cn=users,dc=example,dc=com

... 8< ...
```

2. 同期合意の変更

```
[jsmith@ipaserver ~]$ ldapmodify -x -D "cn=directory manager" -W -p 389 -h
ipaserver.example.com <<EOF
 dn:
cn=meToWindowsBox.example.com,cn=replica,cn=dc\3Dexample\2Cdc\3Dcom,cn=mapping
tree,cn=config
 changetype: modify
 replace: nsds7WindowsReplicaSubtree
 nsds7WindowsReplicaSubtree: cn=alternateusers,dc=example,dc=com
 EOF

 modifying entry
"cn=meToWindowsBox.example.com,cn=replica,cn=dc\3Dexample\2Cdc\3Dcom,cn=mapping
tree,cn=config"
```

新規のサブツリー設定は即時に有効になります。同期操作が実行中の場合は、現在の操作が完了するとすぐに有効になります。

15.5.5. 一方向同期の設定

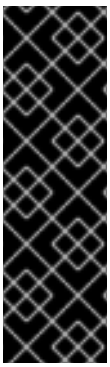
デフォルトでは、すべての変更および削除は双方向で行われます。Active Directory の変更が Identity Management に同期され、Identity Management のエントリーへの変更が Active Directory に同期されます。基本的にこれは、同等のマルチマスターの関係で、Active Directory と Identity Management はどちらも同期時は同等のピアであり、データマスターでもあります。

ただし一部のデータ構造または IT デザインでは、一方のドメインのみをデータマスターとし、他方のドメインでは更新を受け入れられるようにする必要があります。この場合には、マルチマスターの関係 (ピアサーバーが同等) からマスター対コンシュマーの関係に同期関係が変更されます。

これには、同期合意に ***oneWaySync*** パラメーターを設定します。使用可能な値は、 ***fromWindows*** (Active Directory から Identity Management への同期) と ***toWindows*** (Identity Management から Active Directory の同期) です。

たとえば、Active Directory から Identity Management への変更を同期するには、次のコマンドを実行します。

```
[jsmith@ipaserver ~]$ ldapmodify -x -D "cn=directory manager" -w password -p 389 -h
ipaserver.example.com

dn: cn=windows.example.com,cn=replica,cn=dc\3Dexample\2Cdc\3Dcom,cn=mapping tree,cn=config
changetype: modify
add: oneWaySync
oneWaySync: fromWindows
```



重要

一方向同期を有効にすると、同期されていないサーバーで自動的に変更ができなくなる**わけではないため**、同期更新間の同期ピア間で不整合が生じる可能性があります。たとえば、一方向同期は Active Directory から Identity Management に送信されるように設定されるので、(基本的には) Active Directory がデータマスターになります。Identity Management でエントリーを変更または削除すると、Identity Management の情報が異なるため、その変更は Active Directory に引き継がれなくなります。次の同期更新時に、編集内容は Directory Server で上書きされ、エントリーを削除していても再び追加されます。

15.5.6. 同期合意の削除

同期を停止するには、同期合意を削除し、IdM と Active Directory サーバーの接続を **切断** します。同期合意の作成とは逆で、同期合意の削除には、**ipa-replica-manage disconnect** コマンドと Active Directory サーバーのホスト名を使用します。

1. 同期合意を削除します。

```
# ipa-replica-manage disconnect adserver.example.com
```

2. IdM サーバーのデータベースから Active Directory CA 証明書を削除します。

```
# certutil -D -d /etc/dirsrv/slapd-EXAMPLE.COM/ -n "Imported CA"
```

15.5.7. Winsync 合意のエラー

Active Directory サーバーに接続できないため、同期合意の作成に失敗します。

同期合意での最も一般的なエラーの 1 つとして、IdM サーバーが Active Directory サーバーに接続できない点が挙げられます。

```
"Update failed! Status: [81  - LDAP error: Can't contact LDAP server]
```

これは、合意の作成時に正しくない Active Directory CA 証明書が指定される場合に生じる可能性があります。これにより、IdM LDAP データベース (**/etc/dirsrv/slapd-DOMAIN/** ディレクトリー内) に **Imported CA** という名前で重複した証明書が作成されます。これは、**certutil** を使用して確認できます。

```
$ certutil -L -d /etc/dirsrv/slapd-DOMAIN/

Certificate Nickname                                         Trust Attributes
SSL,S/MIME,JAR/XPI

CA certificate                                               CTu,u,Cu
Imported CA                                                   CT,,C
Server-Cert                                                  u,u,u
Imported CA                                                   CT,,C
```

この問題を解決するには、証明書データベースを削除します。

```
# certutil -d /etc/dirsrv/slapd-DOMAIN-NAME -D -n "Imported CA"
```

これにより、LDAP データベースから CA 証明書が削除されます。

エントリーが存在することを示すため、パスワードが同期されていないことを示すエラーがあります。

ユーザーデータベースの一部のエントリーについて、エントリーがすでに存在するためにパスワードはリセットされないという情報のエラーメッセージが表示される可能性があります。

```
"Windows PassSync entry exists, not resetting password"
```

これはエラーではありません。このメッセージは、適用除外ユーザー、パスワード同期ユーザーが変更されていない場合に生じます。パスワード同期ユーザーは、IdM でパスワードを変更するためにサービスで使用される操作上のユーザーです。

15.6. パスワード同期の管理

ユーザーエントリーの同期が同期合意と設定されている。ただし、Active Directory と Identity Management の両方のパスワードは保存時にハッシュ化され、ユーザー同期プロセスの一部として復号化できません。ユーザーアカウントの作成またはパスワードの変更時にパスワードを取り込み、同期更新でそのパスワード情報を転送できるようにするには、別のクライアントが Active Directory サーバー上にインストールされる必要があります。

重要

IdM は現在、ユーザーアカウントの初期パスワード同期に対応していないことに注意してください。IdM にパスワードを同期するには、最初にパスワードを手動で変更する必要があります。

15.6.1. パスワード同期のための Windows Server のセットアップ

パスワードの同期には、以下の 2 つの項目が必要です。

- Active Directory が SSL で実行されている必要があります。
- パスワード同期サービスは、**各** Active Directory ドメインコントローラーにインストールする必要があります。

パスワード同期サービスは、パスワードの変更を記録し、セキュアな接続で IdM エントリーに同期します。

ヒント

エンタープライズルートモードで Microsoft Certificate System をインストールします。Active Directory は自動的に登録され、SSL サーバー証明書を取得します。

1. Active Directory パスワードの複雑さのポリシーが有効になっていることを確認し、パスワード同期サービスを実行します。

   a. コマンドライン **secpol.msc** から実行します。

   b. **セキュリティー設定** を選択します。

   c. **アカウントポリシー** を開き、**パスワードポリシー**を開きます。

   d. **Password must meet complexity requirements** オプションを有効にし、保存します。

2. SSL がまだ有効になっていない場合は、Active Directory サーバーに SSL を設定します。LDAPS の設定に関する詳細は、http://support.microsoft.com/kb/321051 の Microsoft ナレッジベースを参照してください。

   a. プログラムの **追加/削除** の **Windows コンポーネント** セクションに認証局をインストールします。

   b. **Enterprise Root CA** オプションを選択します。

   c. Active Directory サーバーを再起動します。IIS Web サービスを実行している場合は、**http://servername/certsrv** を開いて CA 証明書にアクセスできます。

   d. SSL サーバー証明書を使用するように Active Directory サーバーを設定します。

      i. Active Directory の完全修飾ドメイン名を証明書サブジェクトとして使用し 、証明書要求 **.inf** を作成します。たとえば、以下のようになります。

```
;----------------- request.inf -----------------

[Version]

Signature="$Windows NT$

[NewRequest]

Subject = "CN=ad.server.example.com, O=Engineering, L=Raleigh, S=North
Carolina, C=US"
KeySpec = 1
KeyLength = 2048
Exportable = TRUE
MachineKeySet = TRUE
SMIME = False
PrivateKeyArchive = FALSE
UserProtected = FALSE
UseExistingKeySet = FALSE
ProviderName = "Microsoft RSA SChannel Cryptographic Provider"
ProviderType = 12
RequestType = PKCS10
KeyUsage = 0xa0
```

```
[EnhancedKeyUsageExtension]

OID=1.3.6.1.5.5.7.3.1

;----------------------------------------------
```

.inf リクエストファイルの詳細は、http://technet.microsoft.com/en-us/library/cc783835.aspx などの Microsoft ドキュメントを参照してください。

ii. 証明書要求を生成します。

```
certreq -new request.inf request.req
```

iii. Active Directory CA にリクエストを送信します。以下に例を示します。

```
certreq -submit request.req certnew.cer
```

注記

コマンドラインツールがエラーメッセージを返す場合は、Web ブラウザーを使用して CA にアクセスし、証明書要求を送信します。IIS が実行されている場合、CA URL は **http://servername/certsrv** になります。

iv. 証明書要求を受け入れます。以下に例を示します。

```
certreq -accept certnew.cer
```

v. サーバー証明書が Active Directory サーバーに存在していることを確認します。

File メニューで **Add/Remove** をクリックし、**Certificates** と **Personal>Certificates** をクリックします。

vi. Directory Server から Active Directory に CA 証明書をインポートします。**Trusted Root CA** をクリックし、続いて **Import** をクリックして Directory Server CA 証明書を参照します。

e. ドメインコントローラーを再起動します。

15.6.2. パスワード同期のセットアップ

Windows パスワードを同期するために、Active Directory ドメインのすべてのドメインコントローラーにパスワード同期サービスをインストールします。

1. Active Directory マシンに **PassSync.msi** ファイルをダウンロードします。

   a. カスタマーポータルにログインします。

   b. **Downloads** タブをクリックします。

   c. ページの中心の **Red Hat Enterprise Linux** ダウンロードボタンをクリックします。

   d. **Directory Server** などの検索用語を使用してダウンロードをフィルタリングし、Red Hat Enterprise Linux バージョンの 1 つを展開します。

e. Directory Server のリンクをクリックします。

f. Directory Server ページで、WinSync Installer の適切なバージョンをダウンロードします。これは、パスワード同期 MSI ファイル (**RedHat-PassSync-1.1.5-*arch*.msi**) です。

注記

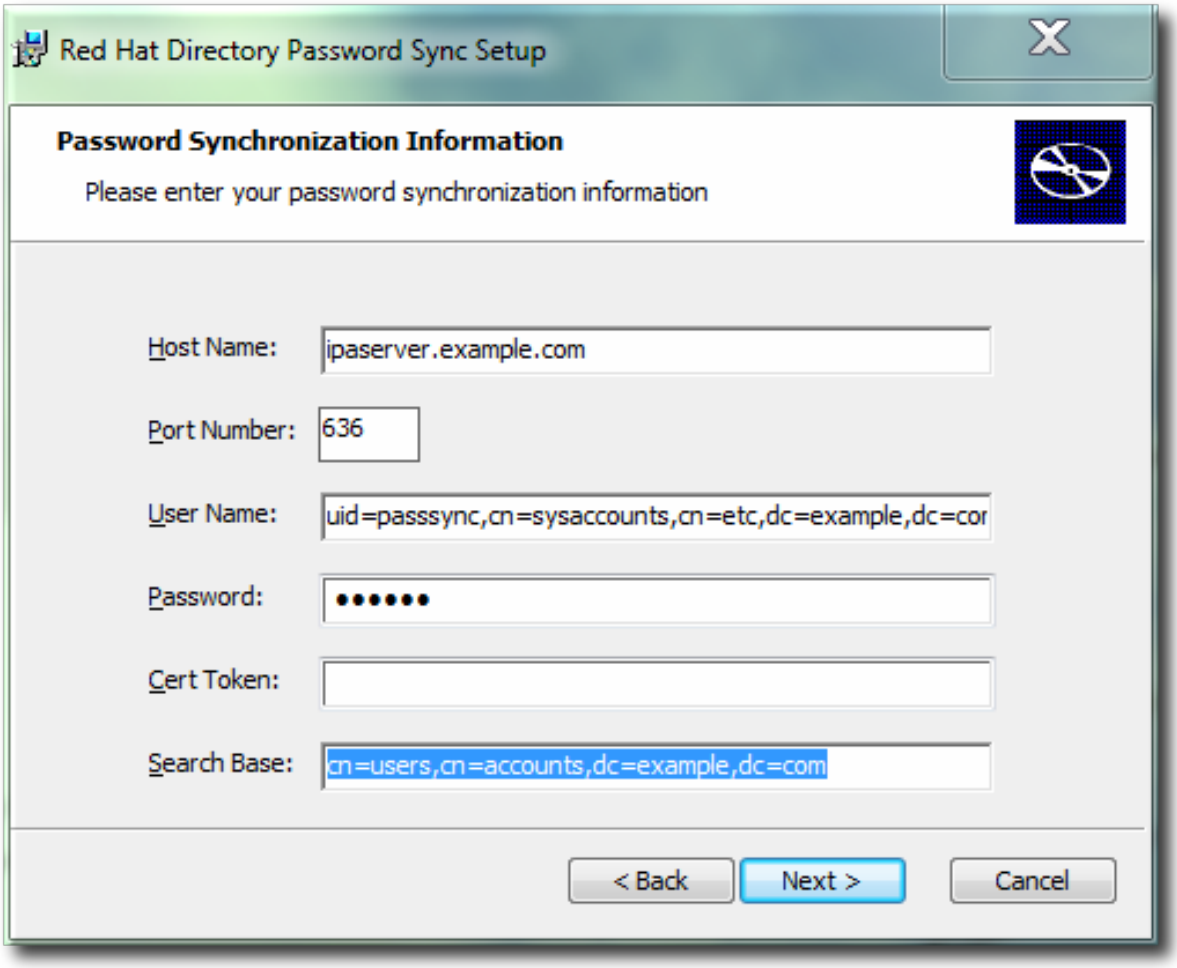
Red Hat Enterprise Linux アーキテクチャーに関係なく、32 ビットの Windows サーバー用と 64 ビット用にそれぞれ PassSync パッケージが 2 つあります。Windows プラットフォームに適したパッケージを選択してください。

2. Password Sync MSI ファイルをダブルクリックしてインストールします。

3. **パスワード同期セットアップ画面** が表示されます。**Next** を押して、インストールを開始します。

4. IdM サーバーへの接続を確立するための情報を入力します。

   - ホスト名およびセキュアなポート番号を含む IdM サーバー接続情報。
   - Active Directory が IdM マシンへの接続に使用するシステムユーザーのユーザー名。このアカウントは、同期が IdM サーバーに設定されている場合に自動的に設定されます。デフォルトのアカウントは **uid=passsync,cn=sysaccounts,cn=etc,dc=example,dc=com** です。
   - 同期合意の作成時に **--passsync** オプションに設定されたパスワード。
   - IdM サーバーの people サブツリーの検索ベース。Active Directory サーバーは、**ldapsearch** またはレプリケーション操作と似た IdM サーバーに接続するため、IdM サブツリーでユーザーアカウントを検索する場所を知っている必要があります。ユーザーサブツリーは **cn=users,cn=accounts,dc=example,dc=com** です。
   - 証明書トークンはこの時点では使用されないため、このフィールドは空白にする必要があります。



Next に進み、**完了** してパスワード同期をインストールします。

5. IdM サーバーの CA 証明書を Active Directory 証明書ストアにインポートします。

   a. IdM サーバーの CA 証明書を **http://ipa.example.com/ipa/config/ca.crt** からダウンロードします。

   b. IdM CA 証明書を Active Directory サーバーにコピーします。

   c. **Run as Administrator** を使用してコマンドプロンプトを開きます。

   d. パスワード同期データベースに IdM CA 証明書をインストールします。以下に例を示します。

   ```
   cd "C:\Program Files\Red Hat Directory Password Synchronization"

   certutil.exe -d . -A -n "IPASERVER.EXAMPLE.COM IPA CA" -t CT,, -a -i ipaca.crt
   ```

   ```
   cd "C:\Program Files\389 Directory Password Synchronization"

   certutil.exe -d . -A -n "IPASERVER.EXAMPLE.COM IPA CA" -t CT,, -a -i ipaca.crt
   ```

6. Windows マシンを再起動して、パスワード同期を開始します。

注記

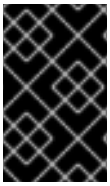
Windows マシンは再起動されている必要があります。再起動しないと **PasswordHook.dll** は有効にされず、パスワードの同期は機能しません。

パスワード同期アプリケーションのインストール時におけるパスワード同期の初回の試行は、Directory Server と Active Directory 同期ピア間の SSL 接続により常に失敗します。証明書およびキーデータベースを作成するためのツールは **.msi** でインストールされます。

15.6.3. 他のユーザーのパスワードのクリーンな変更を許可する

デフォルトでは、管理者がユーザーパスワードを変更するたびに、そのユーザー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。ただし、この動作を変更して、管理者が即時にパスワードをリセット**せず**にパスワードをリセットできます。

passSyncManagersDNs 属性は、パスワード変更操作を実行できる管理者アカウントの一覧を表示します。**これは**パスワードのリセットを必要としません。



重要

これは、パスワードの同期に必要になります。これは、パスワードが同期されるたびに、IdM サーバーがパスワード変更操作として解釈し、次回のログイン時にパスワードの変更を求めるためです。

パスワードの同期エントリー **cn=ipa_pwd_extop,cn=plugins,cn=config** を編集して、ユーザーの名前で ***passSyncManagersDNs*** 属性を追加します。この属性は多値です。以下に例を示します。

```
$ ldapmodify -x -D "cn=Directory Manager" -w secret -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
```

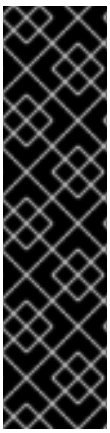
警告

ユーザーパスワードを設定する必要がある管理者アカウントにのみ、リストされている DN を制限してください。ここでリストしたユーザーは、すべてのユーザーパスワードへのアクセスが許可されます。これは非常に強力なユーザーパスワードです。

[5] ***cn*** は、他の同期属性とは異なる処理がされます。Identity Management から Active Directory に同期時には、直接 (***cn*** から ***cn*** へ) マッピングされます。ただし、Active Directory から Identity Management に同期する場合には、***cn*** は、Windows の ***name*** 属性から Identity Management の ***cn*** 属性にマッピングされます。

第16章 ID: ID ビューおよび既存の環境から信頼への移行

Red Hat Identity Management に含まれる *ID ビュー* メカニズムにより、管理者はユーザーまたはグループの POSIX 属性を指定できます。新しい ID ビューが作成されると、管理者は上書きするユーザーまたはグループ属性を定義できます。これらの新たに定義された属性はユーザーまたはグループに適用されます。これにより、ID ビューは、他の ID 管理およびシステム統合ソリューションからの移行中に既存の環境を維持するソリューションを提供します。



重要

この機能を活用するには、Red Hat Enterprise Linux 7.1 移行をベースにした IdM にクライアントが登録されている必要があります。

ID ビューは、Red Hat Enterprise Linux 6.7 以降を実行している Red Hat Enterprise Linux 6 クライアントでのみ使用できます。

管理者はサーバー側の ID ビューのみを管理できます。Red Hat Enterprise Linux 6 クライアントでは設定できません。本章では、クライアント側の ID ビューについて説明します。サーバー側の機能を含む ID ビューの詳細は、Windows Integration Guide for Red Hat Enterprise Linux 7 を参照してください。

IdM サーバーで **ipa-adtrust-install** コマンドを実行すると、デフォルトの *Default Trust View* が作成されます。Default Trust View は常に Active Directory ユーザーおよびグループに適用されます。AD 自体がどのように定義されているかに関わらず、管理者は AD ユーザーおよびグループの POSIX 属性を定義できます。AD ユーザーまたはグループを上書きするホスト固有の ID ビューを追加する場合、ホスト固有の ID ビューの属性が Default Trust View の上部に適用されます。新しい ID ビューは Default Trust View を上書きしますが、デフォルトのビュー自体は削除できません。特定の ID ビューがクライアントに適用されていない場合は、Default Trust View は常に適用されます。

注記

ipa-adtrust-install が実行されていない場合は、純粋な IdM 環境で ID ビュー機能を使用して IdM ユーザーの ID ビューおよびオーバーライドを管理できます。

同期ベースの AD 統合を使用したセットアップでは、ログイン名、UID、GID、またはシェルなどの生成された POSIX 属性を使用して、すべてのユーザーが IdM サーバーにコピーされます。管理者は、AD ユーザー用に AD が以前に生成した POSIX 属性を変更できるため、ID ビュー機能は、既存の環境を信頼ベースの AD 統合に移行するソリューションを提供します。

注記

同期ベースと信頼ベースのアプローチの比較は、Red Hat Enterprise Linux 7 『Windows Integration Guide』 を参照してください。

ID ビューのユースケースには以下が含まれます。

AD ユーザーの POSIX 属性と SSH 鍵を保存する

AD ユーザーの POSIX 属性、SSH キー、および SSH ログイン情報を定義します。そして、ID ビューサポートのある SSSD を実行しているクライアントに対して AD ユーザーが認証する場合や、AD ユーザーが *LDAP 互換ツリー* を使用して認証する際に適用されるようにします。これにより、レガシークライアントのユーザーおよびグループデータでシンプルな LDAP ツリーが提供されます。

この機能は、同期ベースのソリューションからの移行や、Linux 管理者が AD ユーザーの POSIX 属性を手動で定義したい場合に便利ですが、AD ポリシーはこれを許可しません。

同期ベースから信頼ベースの統合への移行

以前に使用した UID や他のツールを指定する ID ビューオーバーライドを作成して、同期ベースの環境にあるユーザーの POSIX 属性を設定します。次に、ユーザーを AD に戻します。

IdM ユーザー POSIX 属性のホストごとのグループのオーバーライドを実行する

AD との IdM 統合に移行している NIS ベースのインフラストラクチャーでは、一部の NIS ドメインで元の POSIX データが変更されていない状態である必要があります。そうでなければ、企業のポリシーにより、AD に元の POSIX データを直接設定できない場合があります。このような状況では、ID ビューを使用して、Identity Management サーバーで直接 POSIX データを設定できます。

環境ごとに異なる POSIX 属性または SSH キーを設定する

対応するホストグループに応じて、さまざまな POSIX 属性またはさまざまなユーザー SSH 公開鍵を各種プロダクション環境 (開発、テスト、またはプロダクション) に対して設定します。

16.1. ユーザーオーバーライドおよびグループのオーバーライド

各 *ID view* は、指定したホストに適用される*user overrides* と *group overrides* のオーバーライドのコレクションです。上書きにより、以前の属性を上書きする新しいユーザーまたはグループ属性が提供されます。これにより、以前に生成された属性を新しい属性に置き換えることができます。すべてのオーバーライドは、AD または IdM のユーザーまたはグループに関連付けられます。

注記

IdM 以外の統合システムは、IdM で使用されるアルゴリズムとは異なるアルゴリズムを使用して、UID および GID 属性を生成できます。以前生成された属性を上書きして、IdM システムに従うようにすることで、別の統合システムのメンバーとなるように使用されたクライアントを IdM と完全に統合できます。

以下のユーザー属性は ID ビューで上書きできます。

- **uid**: ユーザーログイン名
- **uidNumber**: ユーザー UID 番号
- **gidNumber**: ユーザーの GID 番号
- **loginShell**: ユーザーログインシェル
- **GECOS**: ユーザー GECOS エントリー
- **homeDirectory**: ユーザーのホームディレクトリー
- **ipaSshPubkey**: ユーザー SSH 公開鍵またはキー

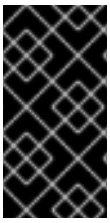
以下のグループ属性は ID ビューで上書きできます。

- **cn**: グループ名
- **gidNumber**: グループの GID 番号

注記

IdM は *ID の範囲* を使用して、異なるドメインからの POSIX ID の競合を回避します。IdM は他の種類の ID 範囲との重複を許可する必要があるので、ID ビューの POSIX ID は特別な範囲タイプを使用しません。例えば、同期で作成された AD ユーザーは、IdM ユーザーと同じ ID 範囲からの POSIX ID を持つことになります。競合が発生した場合は、POSIX ID は IdM の ID ビューで手動で管理されるため、競合する ID を変更することで簡単に修正できます。

16.2. サーバー側での ID ビューの管理

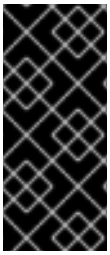


重要

管理者はサーバー側の ID ビューのみを管理できます。Red Hat Enterprise Linux 6 クライアントでは設定できません。本章では、クライアント側の ID ビューについて説明します。サーバー側の機能を含む ID ビューの詳細は、Windows Integration Guide for Red Hat Enterprise Linux 7 を参照してください。

サーバーから ID ビューを追加、変更、または削除できます。管理者は、ID ビューが上書きする ID 属性や、適用する必要のあるクライアントホストを定義できます。

16.3. クライアント側の ID ビュー



重要

ID ビューは、Red Hat Enterprise Linux 6.7 以降を実行している Red Hat Enterprise Linux 6 クライアントでのみ使用できます。

この機能を活用するには、Red Hat Enterprise Linux 7.1 移行をベースにした IdM にクライアントが登録されている必要があります。

クライアント側では、クライアント自体が、クライアントシステムが起動または再起動した後に所属する ID ビューを決定します。その後、クライアントは適用された ID ビューによって定義されたデータの使用を開始します。ID ビューはクライアント側に適用されるため、Red Hat Enterprise Linux 7.0 以前のバージョンを実行するクライアントは Default Trust View のみを表示します。クライアントに別の ID ビューが必要な場合は、クライアントの SSSD を ID View サポートのあるバージョンに更新するか、クライアントが compat LDAP ツリーを使用している。

管理者は、クライアントで別の ID ビューを適用するたびに、クライアントと、この ID ビューを適用する他のすべてのクライアントが SSSD サービスを再起動する必要があります。

注記

ID ビューを適用すると、特定の最適化と ID ビューが同時に実行できなくなるので、SSSD パフォーマンスにマイナス影響が出る可能性があります。

たとえばID ビューは、SSSD によるサーバー上でグループルックアップのプロセス最適化を妨げます。ID ビューを使用すると、グループ名が上書きされた場合、SSSD は返されたグループメンバー名リストの各メンバーをチェックする必要があります。ID ビューを使用しないと、SSSD はグループオブジェクトのメンバー属性からユーザー名を収集するだけで済みます。この負の効果は、SSSD キャッシュが空であるか、または、すべてのエントリーが無効である場合に、ャッシュをクリアした後に多くの場合で影響します。

16.4. SYNCHRONIZATION-BASED からトラストベースのソリューションへの移行

ID ビューを使用して、同期ベースのインテグレーションから信頼ベースのインテグレーションに移行できます。移行は、IdM サーバーで実行できます。これは、Red Hat Enterprise Linux 7 のWindows 統合ガイドを参照してください。

第17章 アイデンティティー: DNS の管理

DNS が設定された状態で IdM サーバーがインストールされている場合は、IdM ツールを使用して、ドメインの DNS エントリー (ホストエントリー、場所、レコード) をすべて管理できます。

17.1. IDM の DNS について

DNS は、IdM ドメインで設定および維持できるサービスの 1 つです。DNS は、IdM ドメインのパフォーマンスに不可欠です。DNS は、すべてのサーバーおよびクライアントの Kerberos サービスおよび SSL 接続に使用され、LDAP などのドメインサービスへの接続に使用されます。

IdM は外部 DNS サービスを使用できますが、ドメイン内で DNS サービスを設定する際に、IdM に対する柔軟性と制御が非常に高くなります。たとえば、DNS レコードとゾーンは、IdM ツールを使用してドメイン内で管理できます。また、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。ホストが IdM に追加されると、そのホストマシンの IdM の DNS サービスに DNS レコードが自動的に作成されます。

IdM は、すべての DNS 情報を LDAP エントリーとして保存します。各マシンのリソースレコードはすべてドメインに保存されます。たとえば、client1 リソースには、3 つの IPv4(A) レコードと 1 つの IPv6(AAAA) レコードがあります。

```
dn: idnsname=client1,idnsname=example.com,cn=dns,dc=example,dc=com
idnsname: client1
arecord: 10.0.0.1
arecord: 10.0.0.2
arecord: 10.0.0.3
aaaarecord: fc00::1
objectclass: top
objectclass: idnsrecord
```

DNS エントリーを定義するために使用されるスキーマは **/usr/share/ipa/60basev2.ldif** スキーマファイルにあります。[6].

BIND サービスは、システム **bind-dyndb-ldap** プラグインを使用して Directory Server と通信します。DNS を管理するために Identity Management を設定すると、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 (**named**) サービスの **bind-dyndb-ldap** プラグインが設定されます。

このプラグインが適切に設定されている場合は、Directory Server から **named** サービスに DNS レコードを提供します。設定を変更して、プラグインの動作に合わせて LDAP-BIND の対話を行います。

17.2. 既存の DNS 設定での IDM および DNS サービス検出の使用

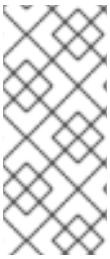
適切な DNS 設定を作成および設定するために、IdM インストールスクリプトにより、サンプルゾーンファイルが作成されます。インストール時に、IdM は以下のようなメッセージが表示されます。

```
Sample zone file for bind has been created in /tmp/sample.zone.F_uMf4.db
```

DNS サーバーがネットワークに設定されている場合は、IdM が生成したファイルの設定を、既存の DNS ゾーンファイルに追加できます。これにより、IdM クライアントが検索できるようになります。たとえば、この DNS ゾーン設定は、KDC および DNS サーバーがすべて EXAMPLE.COM レルムにある同じマシンに作成されます。

例17.1 デフォルトの IdM DNS ファイル

```
; ldap servers
_ldap._tcp              IN SRV 0 100 389        ipaserver.example.com.

;kerberos realm
_kerberos               IN TXT EXAMPLE.COM

; kerberos servers
_kerberos._tcp          IN SRV 0 100 88         ipaserver.example.com.
_kerberos._udp          IN SRV 0 100 88         ipaserver.example.com.
_kerberos-master._tcp   IN SRV 0 100 88         ipaserver.example.com.
_kerberos-master._udp   IN SRV 0 100 88         ipaserver.example.com.
_kpasswd._tcp           IN SRV 0 100 464        ipaserver.example.com.
_kpasswd._udp           IN SRV 0 100 464        ipaserver.example.com.
```



ヒント

DNS サービスが IdM ドメイン外のサーバーでホストされる場合、管理者は 例17.1「デフォルトの IdM DNS ファイル」の行を既存の DNS ゾーンファイルに追加できます。これにより、IdM クライアントおよびサーバーは DNS サービス検出を引き続き使用して、IdM ドメインに参加するために必要な LDAP および Kerberos サーバー (IdM サーバー) を検索できます。

17.3. DNS に関する注意事項

- DNS 名の設定時にはワイルドカードを使用できません。明示的な DNS ドメイン名のみがサポートされます。
- **--setup-dns** オプションを指定しても、**rndc** サービスは設定されません。このサービスは、IdM サーバーの設定後に手動で設定する必要があります。

17.4. インストール後の DNS サービスの追加または更新

DNS は、単に **--setup-dns** オプションを使用して、IdM サーバーのインストールの一部として設定できます。DNS が設定されていない場合は、**ipa-dns-install** コマンドを使用して後で設定できます。

この **ipa-dns-install** コマンドは、IdM サーバーの DNS サービスも更新します。

```
[root@server ~]# ipa-dns-install -p secret --ip-address=1.2.34.56 --no-forwarders
```

- **-p** では、389 Directory Server の Directory Manager ユーザーのパスワードを指定します。すべての DNS エントリーは LDAP ディレクトリーに格納されるため、このディレクトリーにアクセスして DNS 設定を追加する必要があります。
- **--ip-address** では、マスター DNS サーバーの IP アドレスを取得します。
- **--no-forwarders** は、DNS サービスと使用されるフォワーダーがルートサーバーのみであることを意味します。また、**--forwarder** オプションを指定して、使用するフォワード定義します。複数のフォワーダーを指定するには、**--forwarder** オプションを複数回指定します。
- 逆引き DNS は自動的に設定されます。**--no-reverse** オプションを使用して逆引き DNS を無効にすることができます。

既存の逆引き DNS ゾーンが設定されている場合は、この **--no-reverse** オプションを指定して、新しい逆引きゾーンを作成するのではなく、既存の逆引きゾーンを使用します。

- IdM サーバーが明示的に無効になっている場合を除き、Directory Server で永続的な検索を開き、新しいゾーンの変更を即座にキャプチャーします。

17.5. RNDC サービスの設定

この **ipa-dns-install** コマンドは、システムの **rndc** サービスを自動的に設定しません。このサービスは、DNS を IdM に設定した後に手動で設定する必要があります。

1. **rndc** 設定ファイルとキーを作成します。

```
[root@server ~]# /usr/sbin/rndc-confgen -a
[root@server ~]# /sbin/restorecon /etc/rndc.conf
```

これには、キーの作成時にユーザーが入力してエントロピーを作成する必要がある場合があります。

2. **rndc** キーファイルの所有者と権限を変更します。

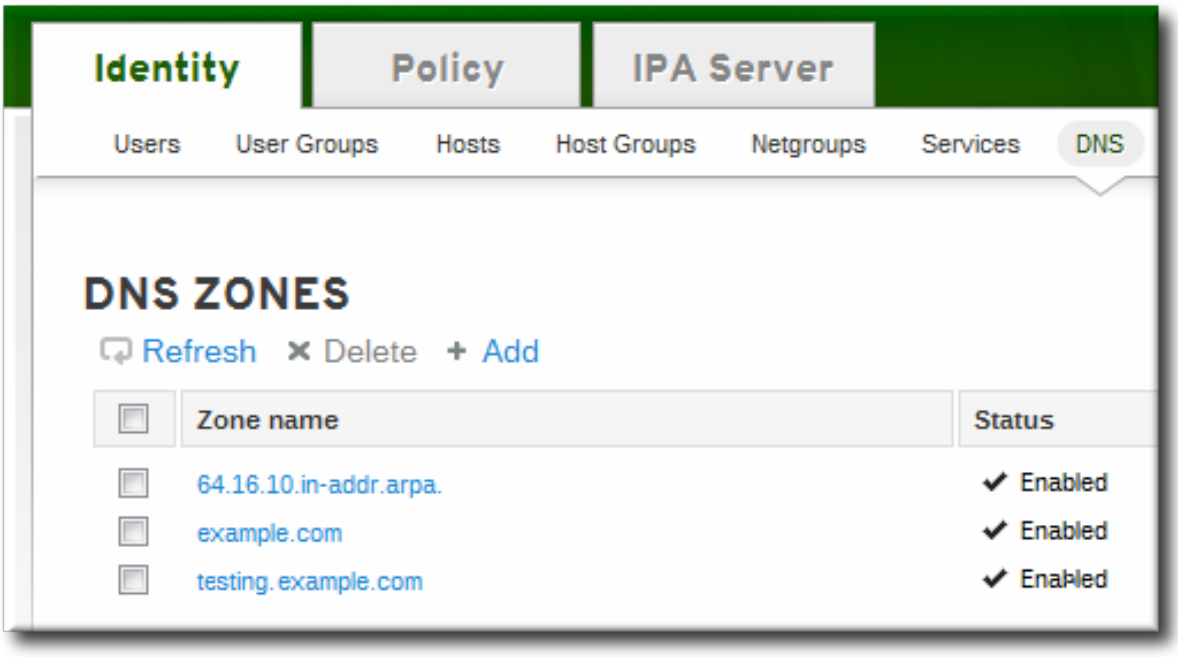
```
[root@server ~]# chown root:named /etc/rndc.key
[root@server ~]# chmod 0640 /etc/rndc.key
```

17.6. DNS ゾーンエントリーの管理

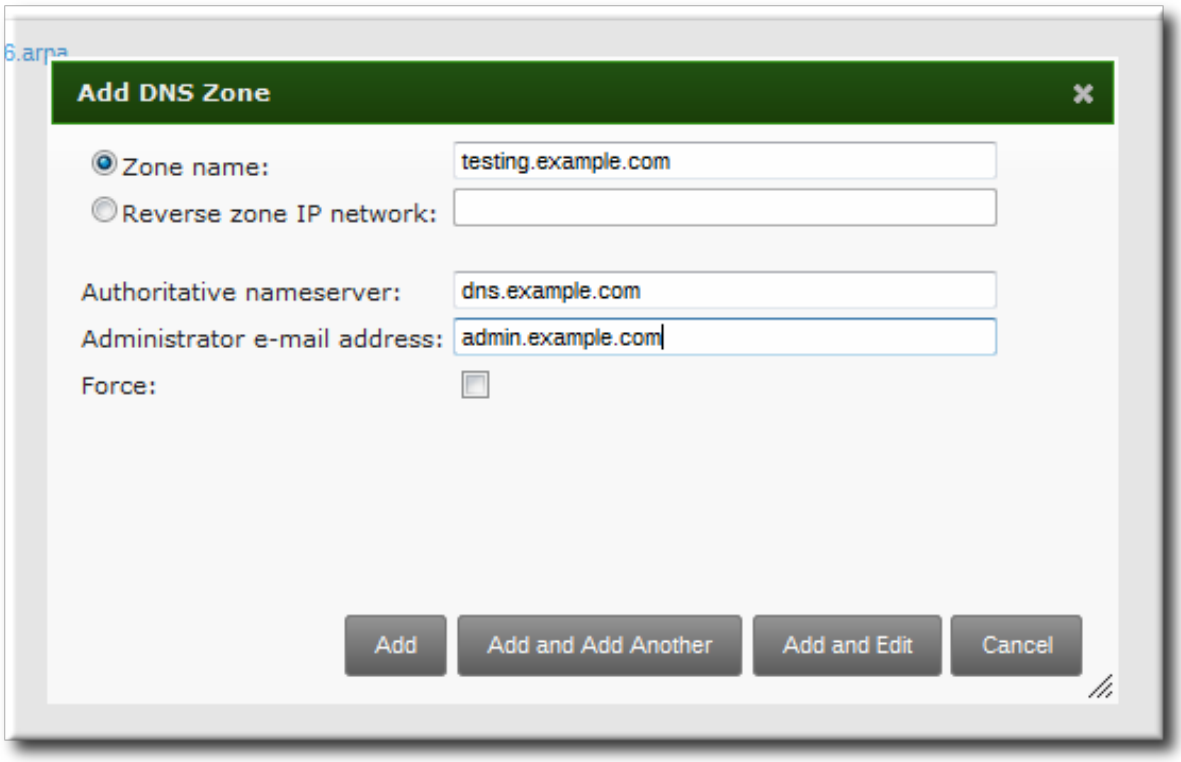
17.6.1. 正引き DNS ゾーンの追加

17.6.1.1. Web UI での操作

1. **Identity** タブを開き、**DNS** サブタブを選択します。
2. DNS ゾーンの一覧の上部にある **Add** リンクをクリックします。



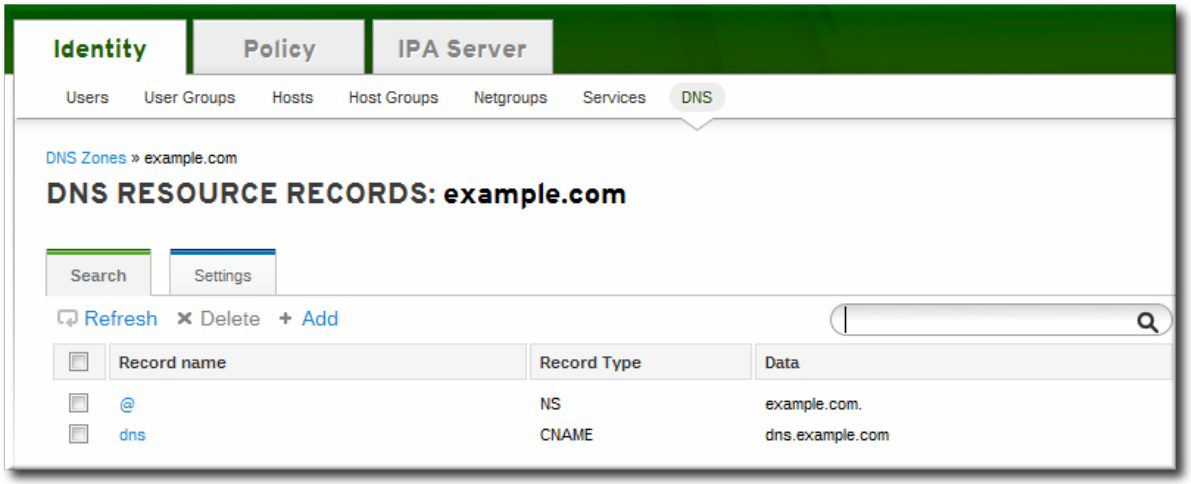
3. 新しい DNS ゾーンに関する情報を入力します。**ゾーン名**が必要です。これは実際のドメイン名です。管理者メールおよび権威ネームサーバーに関するその他の情報は任意です。



注記

管理者にメールがある場合は、at 記号 (@) をピリオド (.) に置き換え、ゾーンファイルとの互換性を維持します。

4. **追加および編集** ボタンをクリックして、DNS ゾーンページに直接移動します。**Settings** タブで、デフォルトのゾーン設定をリセットして、動的バインド (「Web UI での動的 DNS 更新の有効化」) を有効にするか、他のデフォルトレコード情報 (「Web UI でのゾーン設定編集」) を変更できます。**DNS Resource Records** タブで新しい DNS リソースレコード (「Web UI での DNS リソースレコードの追加」) の追加を開始することもできます。



17.6.1.2. コマンドラインでの操作

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。少なくとも、新しいサブドメインの名前が必要です。

```
$ ipa dnszone-add domainName
```

名前が指定されていない場合、スクリプトはその名前の入力を要求します。**ipa dnszone-add** コマンドでは、他のコマンドラインオプションもコマンドで渡すこともできます。

ゾーンエントリーを追加するには、次のコマンドを実行します。

1. 新しいゾーンを追加します。以下に例を示します。

```
[root@server ~]# ipa dnszone-add newserver.example.com --admin-email=admin@example.com --minimum=3000 --dynamic-update
```

2. ネームサービスを再読み込みします。

```
[root@server ~]# rndc reload
```

ヒント

name サービスを再起動せずに新しいリソースレコードを即座に解決できるようにするには、**named** サービスを使用した永続的な検索を有効にするか、BIND サービスを設定してゾーンの変更を自動的にポーリングします。「永続検索の無効化」を参照してください。

17.6.2. DNS ゾーンの追加設定の追加

ゾーンが、更新期間、転送設定、キャッシュ設定などの設定量が一定で設定されて作成され、デフォルト値に設定されます。

例17.2 DNS ゾーンのデフォルトのエントリー設定

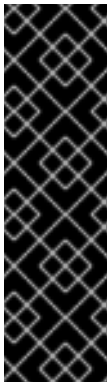
```
[root@server ~]# ipa dnszone-show server.example.com
Zone name: server.example.com
Authoritative nameserver: dns.example.com
Administrator e-mail address: admin.example.com.
SOA serial: 1377691702
SOA refresh: 3600
SOA retry: 900
SOA expire: 1209600
SOA minimum: 3000
Active zone: TRUE
Allow query: any;
Allow transfer: none;
```

17.6.2.1. DNS ゾーン設定の属性

可能なゾーン設定はすべて 表17.1「ゾーン属性」に一覧表示されます。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17.1 ゾーン属性

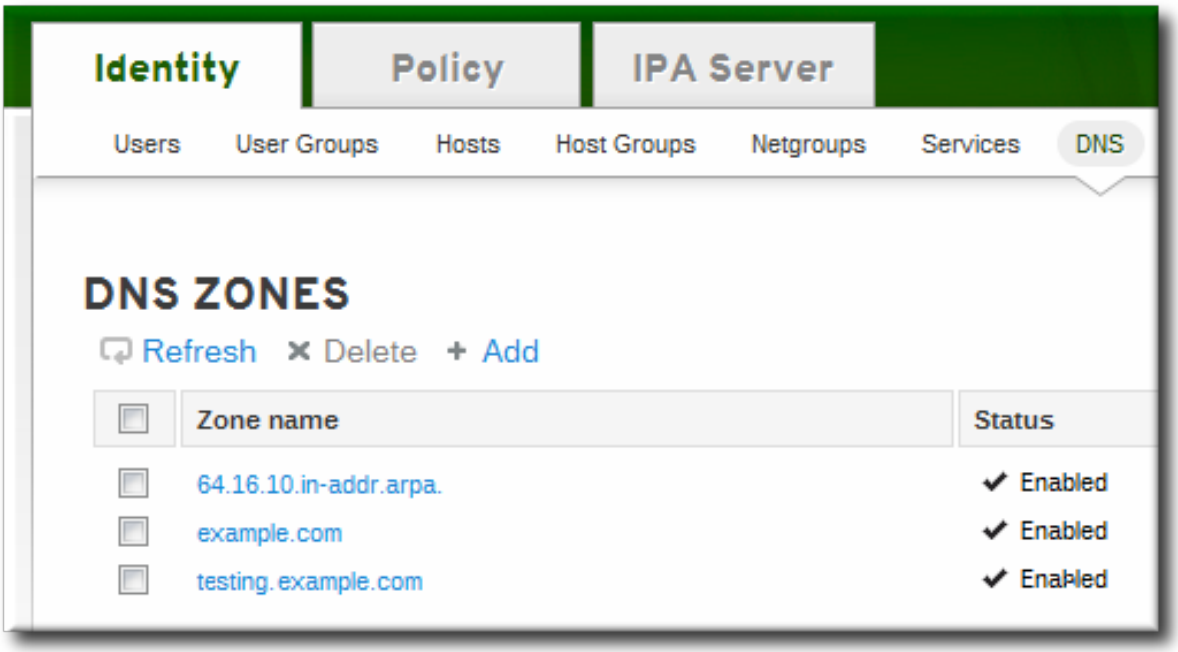
属性	コマンドラインオプション	説明
ゾーン名	--name	ゾーンの名前を設定します。
権威ネームサーバー	--name-server	DNS ネームサーバーの完全修飾ドメイン名を設定します。
管理者の電子メールアドレス	--admin-email	ゾーン管理者が使用する電子メールアドレスを設定します。デフォルトでは、ホストの root アカウントになります。
SOA serial	--serial	SOA レコードファイルのバージョン番号を設定します。
SOA refresh	--refresh	セカンダリー DNS サーバーがプライマリ DNS サーバーから更新を要求するまでの待機時間を秒単位で設定します。
SOA retry	--retry	失敗したリフレッシュ動作を再試行するまでの待機時間を秒単位で設定します。
SOA expire	--expire	セカンダリー DNS サーバーがリフレッシュ更新を試行して、その動作を停止するまでの時間を秒単位で設定します。
SOA minimum	--minimum	データがキャッシュに保持される最小時間 (秒単位) を設定します。
SOA time to live	--ttl	情報がデータキャッシュに保持される最大時間 (秒単位) を設定します。
SOA クラス	--class	レコードのタイプを設定します。これはほとんどの場合で、インターネットを表します。
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。

属性	コマンドラインオプション	説明
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。  **重要** 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ります。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。
ネームサーバー	--ip-address	IP アドレスで DNS ネームサーバーを追加します。
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一覧を指定します。
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一覧を指定します。
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。
Zone forwarders	--forwarder=*string*	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に固有の場合は、オプションを複数回使用します。

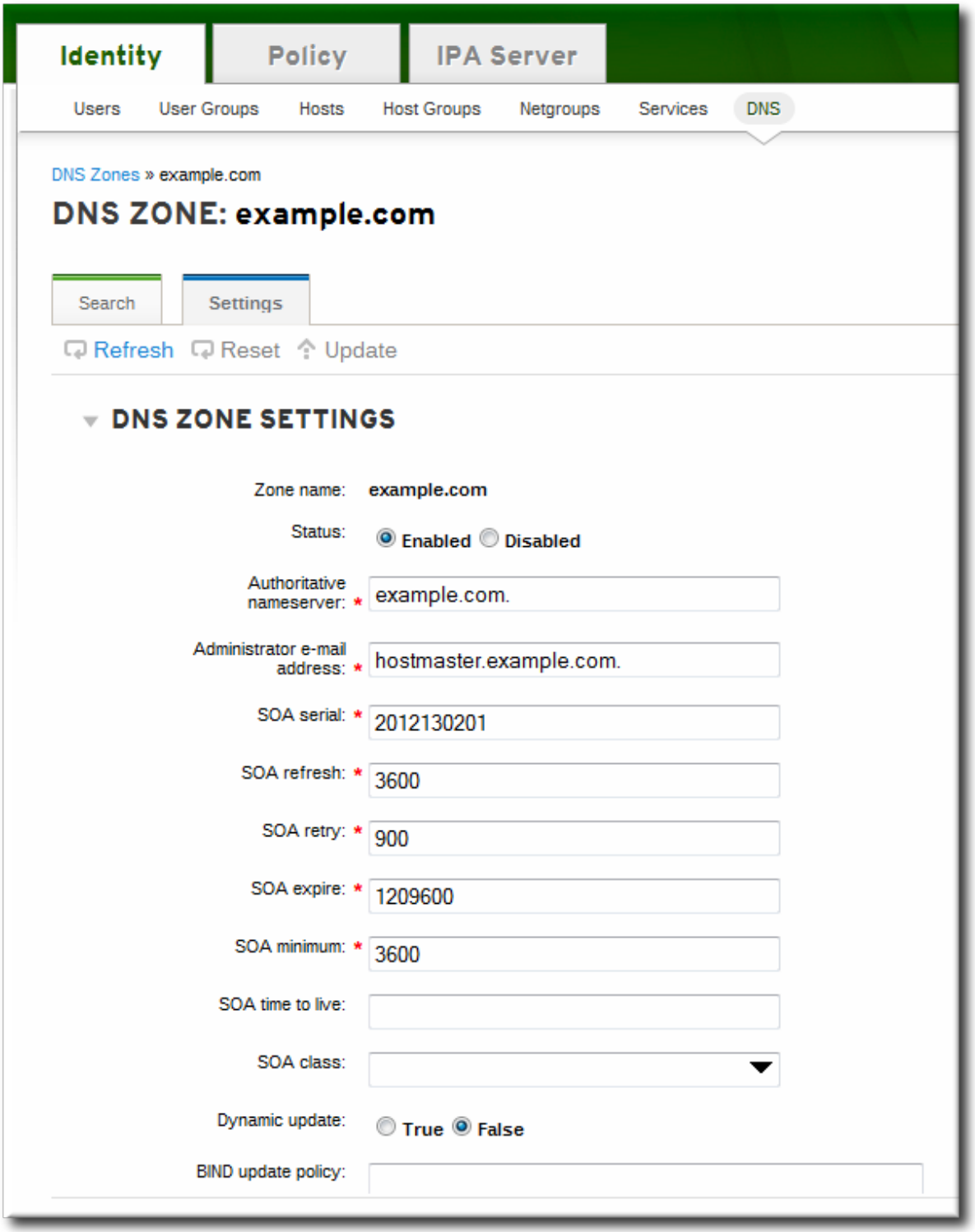
属性	コマンドラインオプション	説明
Forward policy	--forward-policy=only|first	ゾーンが DNS ネームサーバー (**正引きのみ**のゾーン) へのリクエストのみを転送するかどうかを設定します。または、最初に DNS レコードをチェックしてから、独自のローカルレコードを確認するかどうかを設定します。

17.6.2.2. Web UI でのゾーン設定編集

1. **Identity** タブを開き、**DNS** サブタブを選択します。
2. 編集する DNS ゾーンの名前をクリックします。



3. **Settings** タブを開きます。
4. DNS ゾーン設定を変更します。属性の完全なリストは、表17.1「ゾーン属性」に記載されています。変更すべき一般的な属性はいくつかあります。
   - **権威ネームサーバー**。DNS ネームサーバーの完全修飾ドメイン名です。
   - クライアントの DNS レコードへの **動的更新** を有効にするための動的更新。
   - **SOA 更新**。セカンダリー DNS サーバーがプライマリ DNS サーバーから更新を要求するまでの待機時間を秒単位で設定します。



5. 設定ページの上部にある **Update** リンクをクリックします。

17.6.2.3. コマンドラインでのゾーン設定の編集

ゾーンは、**dnszone-add** コマンドで追加オプションを渡すことで、デフォルトとは異なる値で作成できます。同様に、**dnszone-mod** コマンドで同じ属性オプションを渡すと、ゾーンエントリーで属性を追加または変更できます。これらについては、表17.1「ゾーン属性」に一覧表示されます。

DNS ゾーンエントリーに属性が存在しない場合は、**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、現在の値を指定された値で上書きします。

たとえば、SOA レコードにタイムアウトを設定し、DNS ゾーンエントリーに新しい属性を追加します (以前のデフォルト値がないため)。

```
[jsmith@server ~]$ kinit admin
[jsmith@server ~]$ ipa dnszone-mod server.example.com --ttl=1800

Zone name: server.example.com
Authoritative nameserver: dns.example.com
Administrator e-mail address: admin.example.com.
SOA serial: 1377691702
SOA refresh: 3600
SOA retry: 900
SOA expire: 1209600
SOA minimum: 3000
SOA time to live: 1800
Active zone: TRUE
Allow query: any;
Allow transfer: none;
```

17.6.3. 逆引き DNS ゾーンの追加

逆引きゾーンを追加するプロセスは、「正引き DNS ゾーンの追加」で説明するように正引きゾーンと同じです。ただし、必要な情報は異なります。

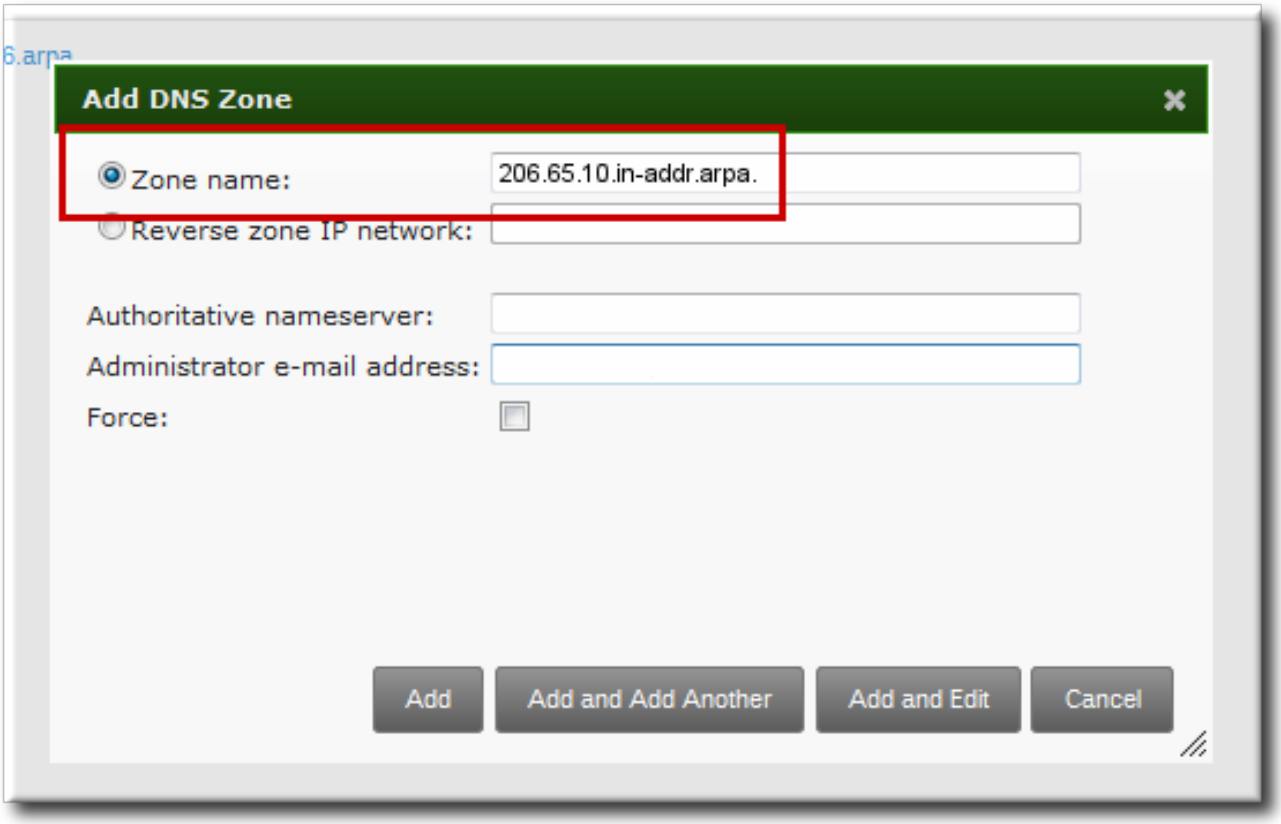
逆引き DNS ゾーンを識別する方法は 2 つあります。

- ゾーン名で、**reverse_ip_address.in-addr.arpa.** 形式で指定します。
- **network_ip_address/subnet_mask_bit_count** の形式でのネットワークアドレス。

ゾーン名で逆引きゾーンを作成する場合は、正引きゾーンの作成と全く同じ設定を行い、IP アドレスのコンポーネントの順番のみを逆にします。たとえば、IP アドレスが **1.2.3.4** の場合、逆引きゾーン名は **3.2.1.in-addr.arpa.** になります (末尾はピリオド)。

Web UI では、これは **Zone name** フィールドに設定されます。

図17.1 名前での逆引きゾーンの作成



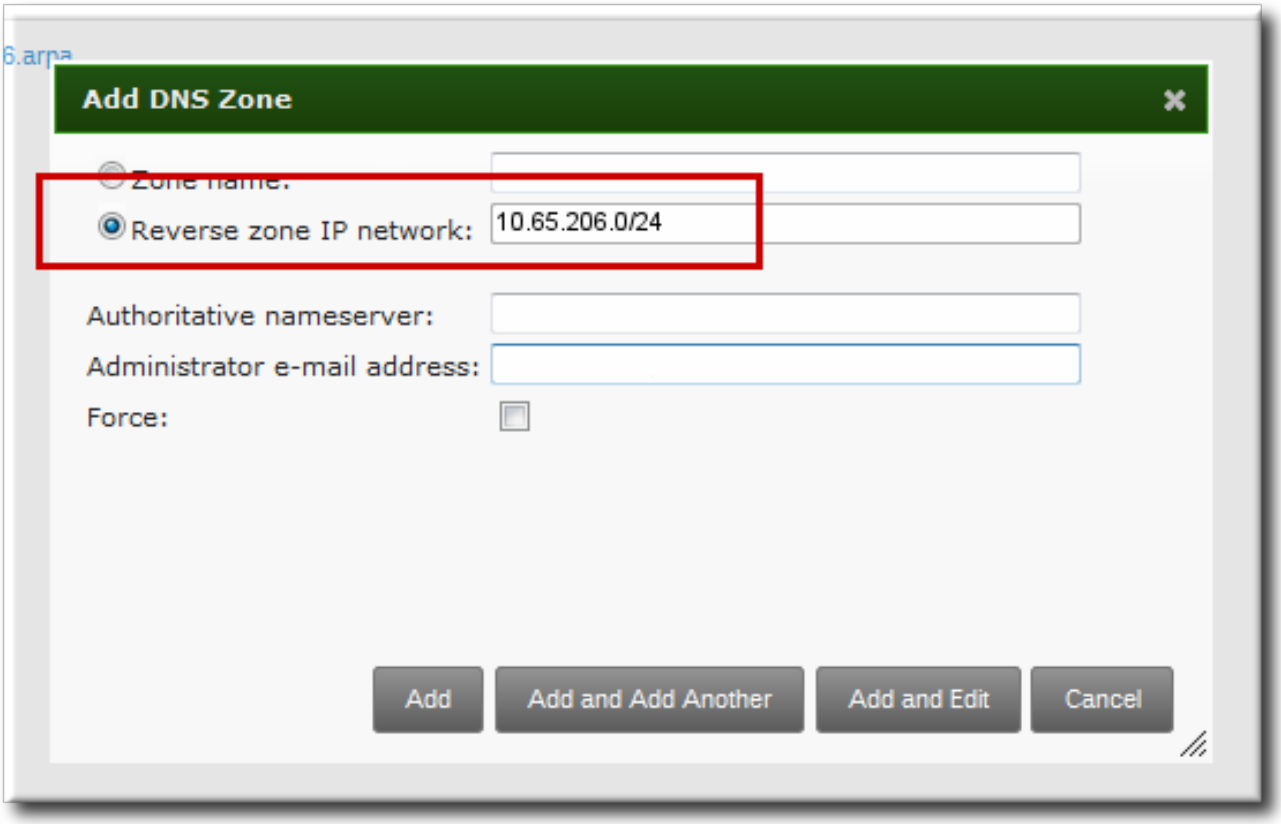
コマンドラインツールでは、ゾーンは以下のような名前で作成されます。

```
[bjensen@server ~]$ kinit
[bjensen@server]$ ipa dnszone-add 206.65.10.in-addr.arpa.
```

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 では、これは **Reverse zone IP network** フィールドで設定されます。

図17.2 IP ネットワークでの逆引きゾーンの作成



コマンドラインツールを使用すると、ゾーンは次のような IP ネットワークで作成されます。

```
[bjensen@server ~]$ kinit
[bjensen@server]$ ipa dnszone-add 10.65.206.0/24
```

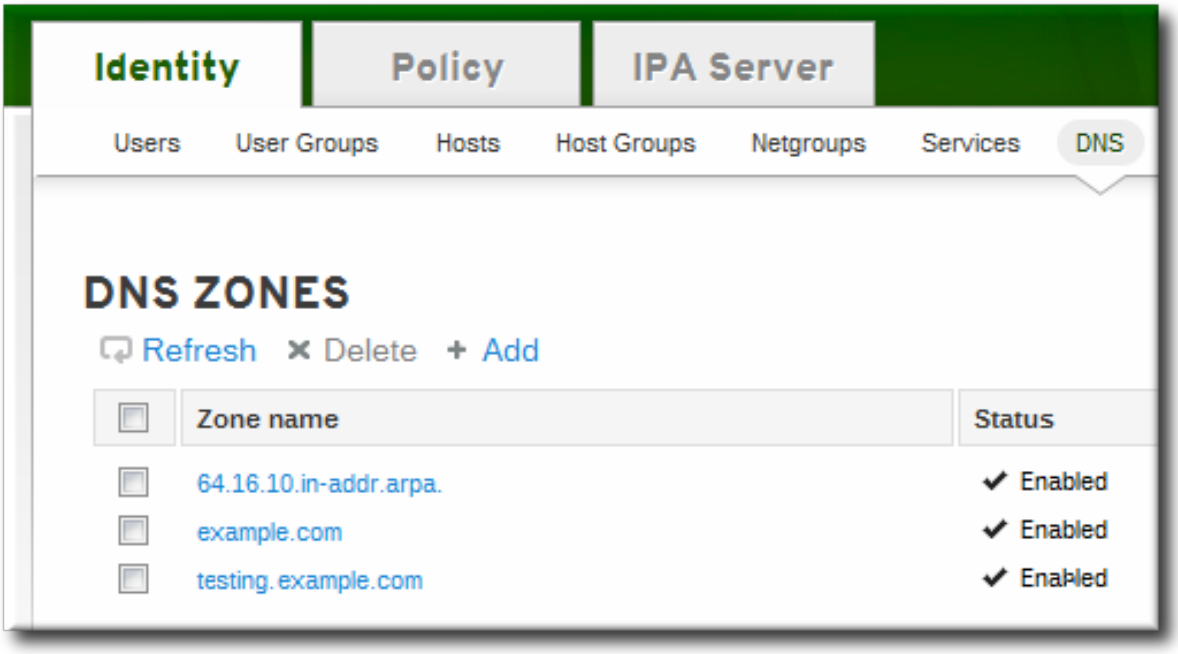
17.6.4. ゾーンの有効化と無効化

アクティブゾーンはクライアントを追加でき、検索に利用できるほか、Kerberos などの IdM サービスで使用できます。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削除されます。

ゾーンを永続的に削除せずに、ゾーンをアクティビティーから削除する必要がある場合も考えられます。これは、ゾーンを **無効** にすることで行います。

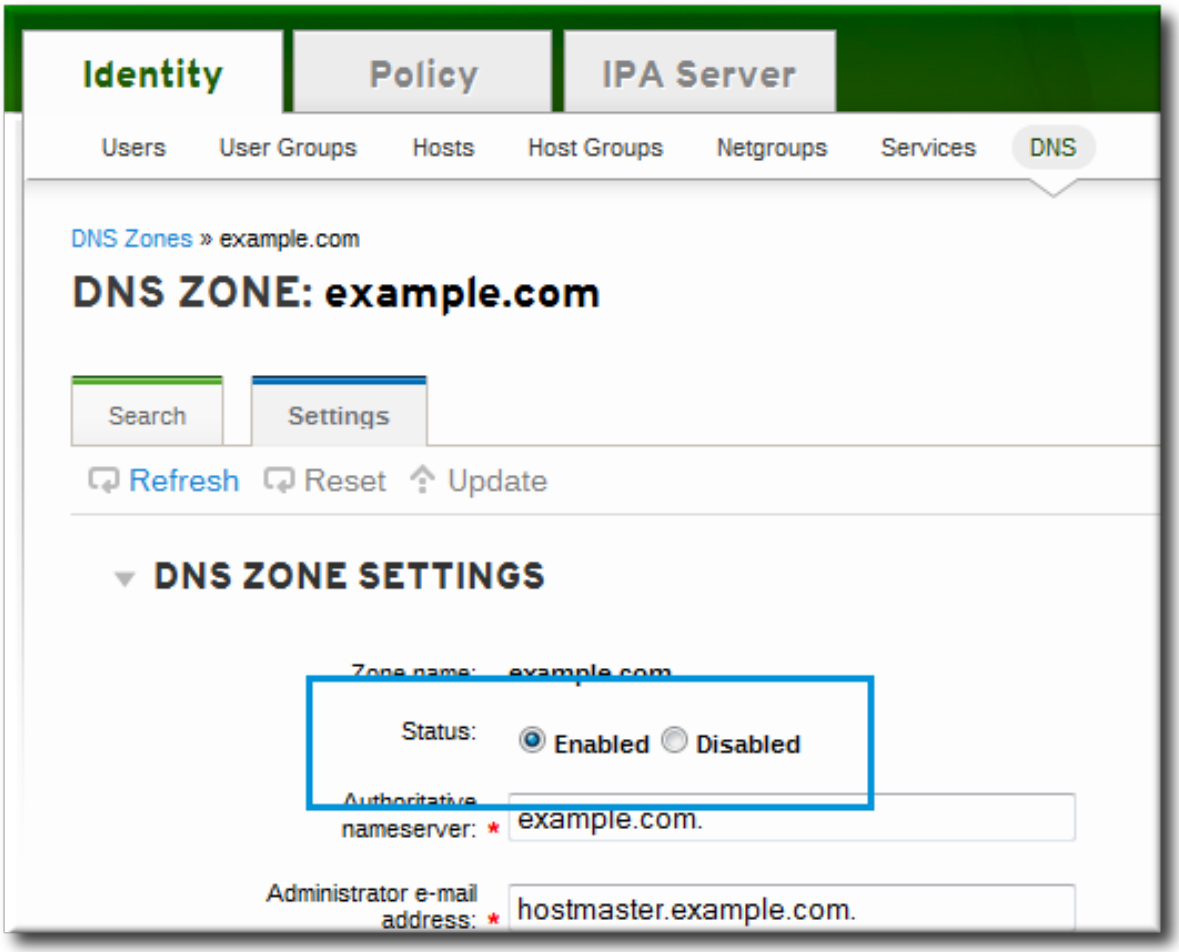
17.6.4.1. Web UI でのゾーンの無効化

1. **Identity** タブを開き、**DNS** サブタブを選択します。
2. 編集する DNS ゾーンの名前をクリックします。



3. **Settings** タブを開きます。

4. **Active zone** フィールドまでスクロールします。ゾーンを無効にするには、値を **Disabled** に設定します。



5. 設定ページの上部にある **Update** リンクをクリックします。

17.6.4.2. コマンドラインでのゾーンの無効化

この **dnszone-disable** コマンドを使用して、ゾーンの無効化を行います。

たとえば、以下のようになります。

```
[jsmith@server ~]$ kinit admin
[jsmith@server ~]$ ipa dnszone-disable server.example.com
----------------------------------------
Disabled DNS zone "server.example.com"
----------------------------------------
```

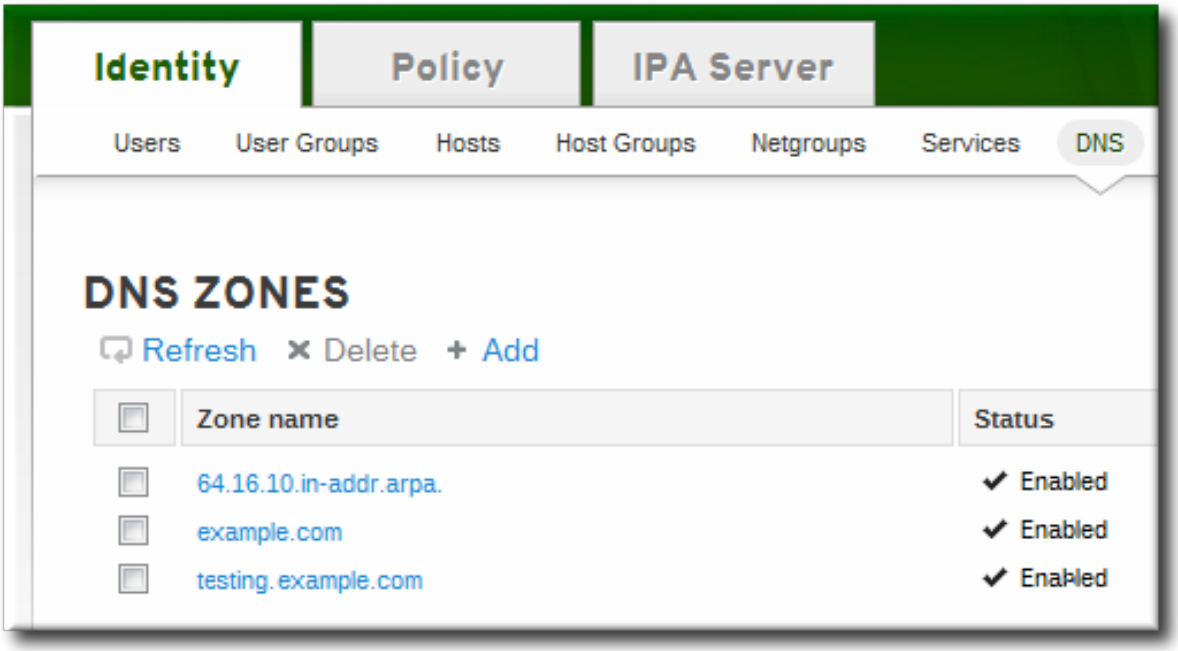
ゾーンをオンラインに戻す必要がある場合は、**dnszone-enable** コマンドを使用してゾーンを再度有効にできます。

17.6.5. ダイナミック DNS 更新の有効化

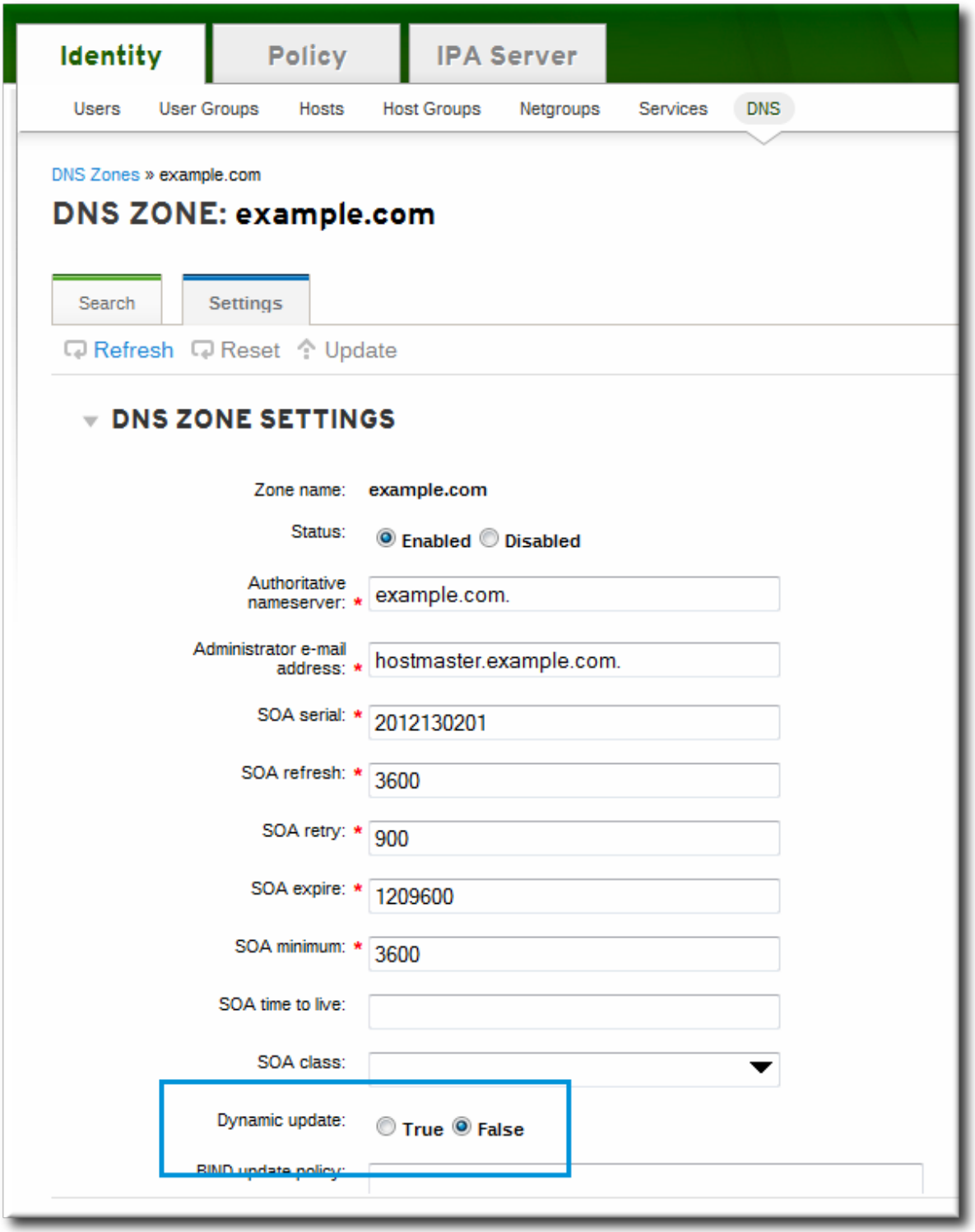
動的 DNS 更新は、IdM の新規 DNS ゾーンに対してデフォルトで有効になっていません。動的更新が許可されていない場合は、新しいクライアントを参照する DNS レコードを追加できないため、**ipa-client-install** スクリプトがクライアントをドメインに参加できない可能性があります。

17.6.5.1. Web UI での動的 DNS 更新の有効化

1. **Identity** タブを開き、**DNS** サブタブを選択します。
2. 編集する DNS ゾーンの名前をクリックします。



3. **Settings** タブを開きます。
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。



5. 設定ページの上部にある **Update** リンクをクリックします。

17.6.5.2. コマンドラインでの動的 DNS 更新の有効化

DNS ゾーンへの動的更新を許可するには、**--dynamic-update** オプションを設定します。

```
$ ipa dnszone-mod server.example.com --dynamic-update=TRUE
```

17.6.6. フォワーダーおよび Forward ポリシーの設定

DNS フォワーダー は、解決のために別の外部 DNS ネームサーバーに DNS クエリーを渡すサーバーです。IdM DNS ドメインには、フォワーダーの使用方法を定義する 3 つの設定プロパティーがあります。

- IdM のすべてのゾーンが使用するグローバルフォワーダーの一覧
- ゾーン設定の一部として、1 つの特定ゾーンによって使用されるフォワーダーの一覧
- ゾーンがフォワーダーに要求を送信する方法を定義するポリシー

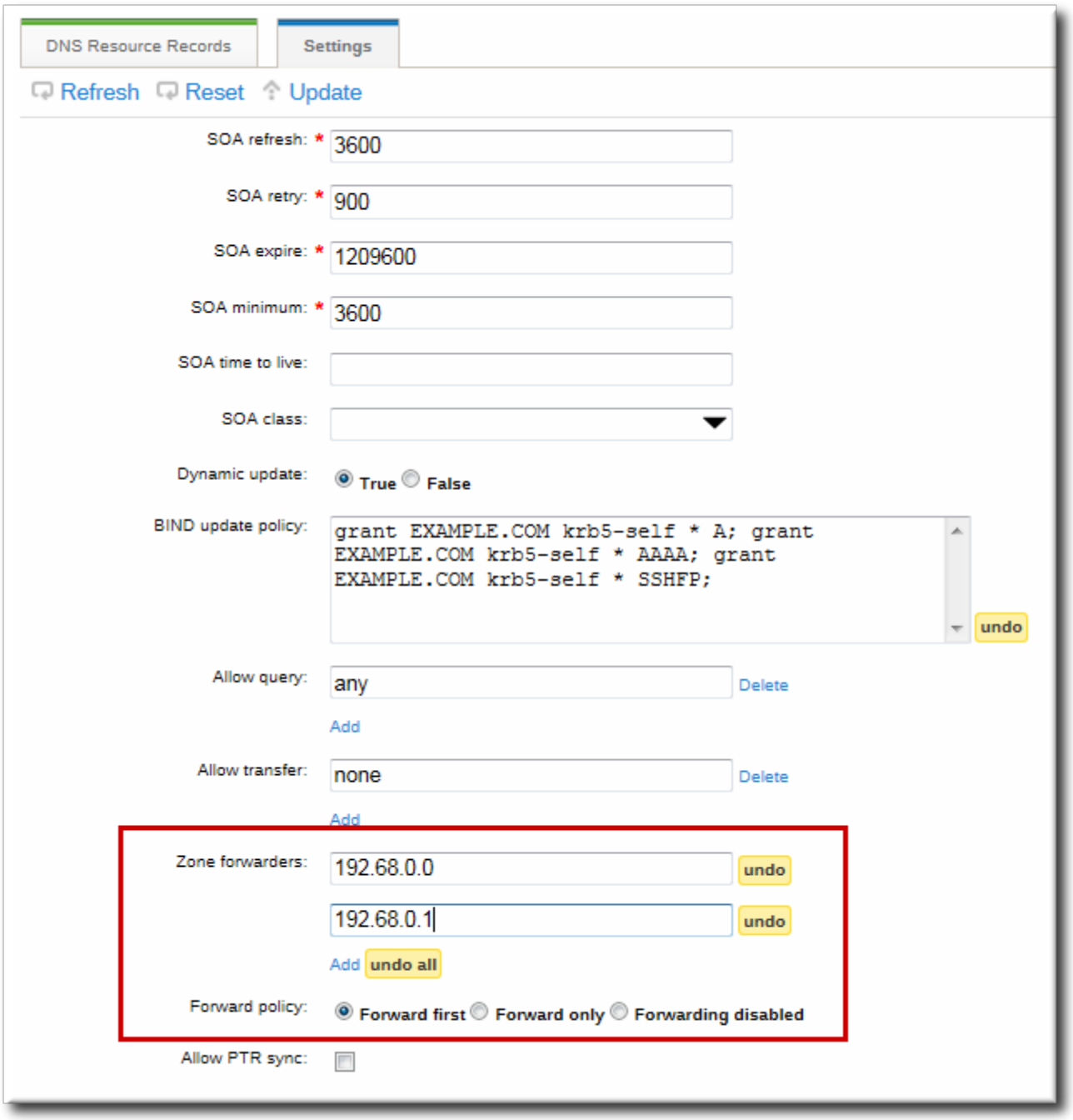
17.6.6.1. UI でのフォワーダーの設定

他の DNS ゾーン設定 (「Web UI でのゾーン設定編集」) と同様に、フォワーダー設定は指定の DNS ゾーンの **Settings** タブにあります。

編集するエリアは 2 つあります。

- フォワーダーを追加するには、フィールドを入力するか、または **Add** をクリックして、新しい IP アドレスをフォワーダー一覧に追加します。
- デフォルトでは、ゾーンは、名前解決要求のサービスにのみフォワーダーを使用します。これは **正引きのみのゾーン** と呼ばれます。前方のみのゾーンは、独自の名前レコードを確認しません。フォワーダーサーバーレコードのみがチェックされます。設定されたフォワーダーにレコードが存在しない場合は、ゾーンはクライアントに異常な状態を返します。また、ゾーンは最初にフォワーダーレコードを確認し、次に独自のリソースレコードにフォールバックできます。これには、**最初** のポリシーがあります。

図17.3 DNS ゾーン設定のフォワーダー



17.6.6.2. コマンドラインでのフォワーダーの設定

フォワーダー設定は、**dnszone-mod** コマンドを使用してゾーン設定を更新すると編集できます。これは、UI のように DNS フォワーダーおよび転送ポリシーの一覧を設定するために使用できます。

また、この **dnsconfig** コマンドを使用して、DNS 設定ファイルを編集して、全ゾーンのフォワーダーのグローバルリストを設定できます。

> **例17.3 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、IdM サーバー設定の一部として設定されます。フォワーダーは、**setup-dns** オプションを指定してサーバーをインストールする場合や **ipa-dns-install** スクリプトが使用される場合に設定されます (任意)。

サーバーの設定後、**dnsconfig-mod** コマンドを使用してグローバルフォワーダーの一覧を編集できます。たとえば、以下のようになります。

```
[jsmith@server ~]$ ipa dnsconfig-mod --forwarder=0.9.8.7
  Global forwarders: 0.9.8.7
```

例17.4 ゾーンフォワーダーの設定

フォワーダーは、ゾーン設定の一部として、特定の DNS ゾーンで使用するように設定できます。この **--forwarder** オプションは、ゾーンで使用するフォワーダーの一覧を作成するために複数回使用できます。

たとえば、以下のようになります。

```
[jsmith@server ~]$ ipa dnszone-mod --forwarder=192.0.2.0 --forwarder=198.51.100.0
example.com

  Zone name: example.com
...
  Zone forwarders: 192.0.2.0, 198.51.100.0
```

注記

DNS フォワーダーは、ホスト名としてではなく、IP アドレスとして指定する必要があります。

例17.5 ゾーンのフォワーダーポリシーの設定

フォワーダーが設定されると、ゾーンを使用してリクエストを処理する方法が異なります。

ゾーンは、名前解決要求のサービスにのみフォワーダーを使用できます。これは **正引きのみのゾーン** と呼ばれます。前方のみのゾーンは、独自の名前レコードを確認しません。フォワーダーサーバーレコードのみがチェックされます。設定されたフォワーダーにレコードが存在しない場合は、ゾーンはクライアントに異常な状態を返します。

また、ゾーンは最初にフォワーダーレコードを確認し、次に独自のリソースレコードにフォールバックできます。これには、**最初** のポリシーがあります。

この設定は、**only** または **first** のいずれかのポリシーを使用して **--forward-policy** オプションで設定されます。たとえば、以下のようになります。

```
[jsmith@server ~]$ ipa dnszone-mod --forward-policy=only example.com

  Zone name: example.com
...
  Zone forwarders: 1.2.3.4;5.6.7.8
  Forward policy: only
```

17.6.7. 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、これらの変更は 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** は、リソースレコードを 1 つのネームサーバーから別のネームサーバーに移動します。**権威転送** (AXFR)は、ゾーンの権限のあるデータを含むゾーン転送です (増分転送とは逆で、即時ゾーンの変更のみを提供します)。

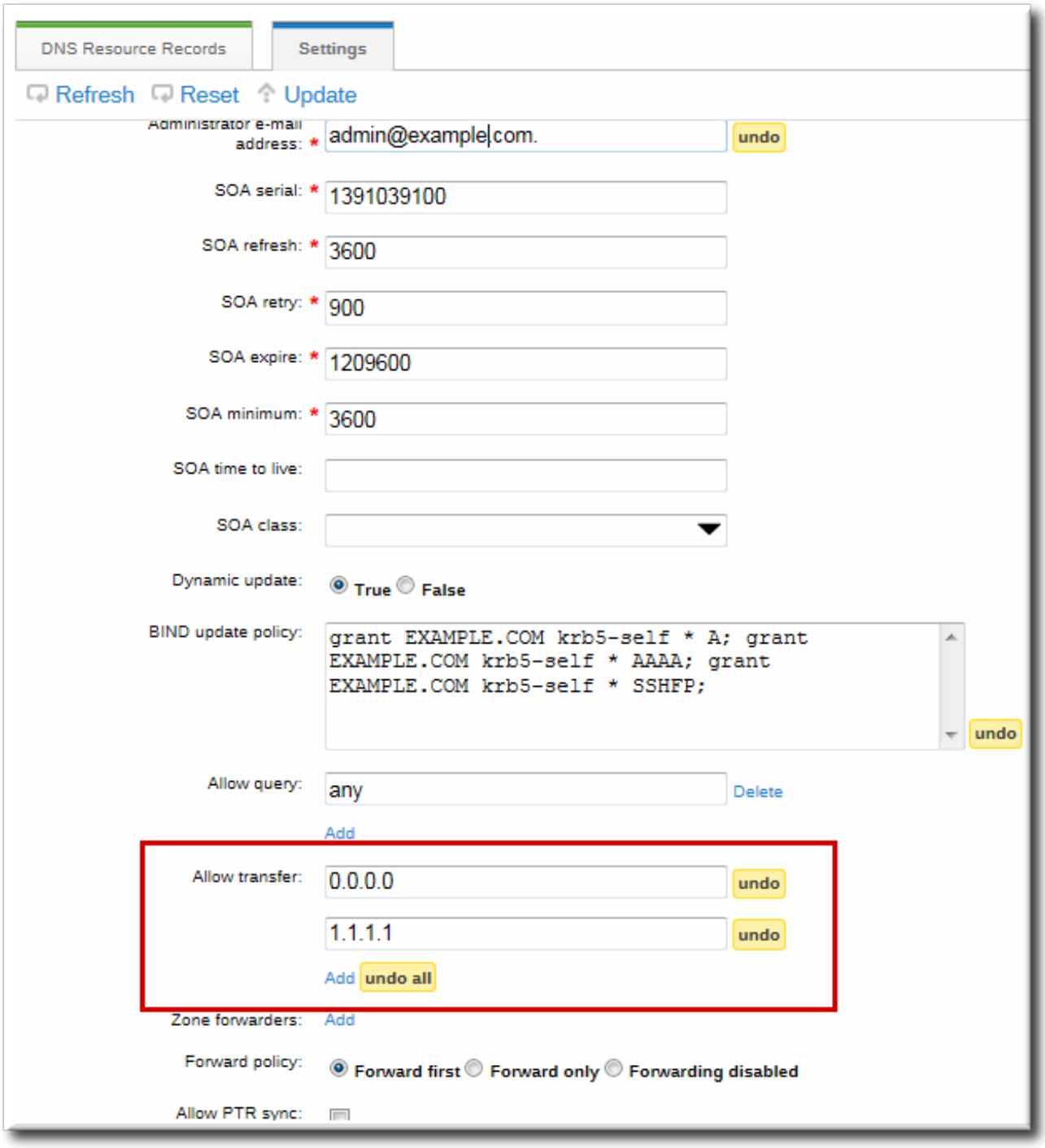
ゾーン転送は RFC 1034 および RFC 5936 で定義されます。

17.6.7.1. UI でのゾーン転送の有効化

他の DNS ゾーン設定 (「Web UI でのゾーン設定編集」) と同様に、ゾーン転送設定は指定の DNS ゾーンの **Settings** タブにあります。

ゾーンレコードを転送できるネームサーバーの一覧を設定します。フィールドに入力するか、**Add** をクリックして新規 IP アドレスをネームサーバー一覧に追加します。

図17.4 DNS ゾーンの転送設定



17.6.7.2. コマンドラインでゾーンの転送の有効化

ゾーン転送は、ゾーンが作成されたときや、**--allow-transfer** オプションを使用してゾーンレコードを転送できるネームサーバーの一覧を設定するときに有効化できます。

たとえば、以下のようになります。

```
[jsmith@server ~]$ ipa dnszone-mod --allow-transfer="0.0.0.0;1.2.3.4;5.6.7.8" example-zone
```

デフォルトは **any** で、DNS ドメインのどこにでも転送されるゾーンです。

bind サービスで有効にすると、**dig** のようなクライアントにより、名前で IdM DNS ゾーンを転送できます。

```
[root@server ~]# dig @ipa-server zone_name AXFR
```

17.6.8. DNS クエリーの定義

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** オプションを使用してクエリーを発行できるクライアントの一覧を設定するときに設定できます。

たとえば、以下のようになります。

```
[jsmith@server ~]$ ipa dnszone-mod --allow-query=0.0.0.0;1.2.3.4;5.6.7.8 example-zone
```

デフォルは **any** です。これは、ゾーンをどのクライアントでもクエリーできます。

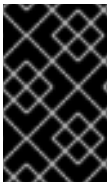
17.6.9. 前方および逆引きゾーンエントリーの同期

前方エントリー (A および AAAA) は、リバースエントリー (PTR) とは別に設定されます。これらのエントリーは独立して設定されるため、フォワードエントリーは、対応するリバースエントリーなしで存在できます。

PTR 同期が機能するには、以下の DNS 設定が必要になります。

- 正引きおよび逆引きゾーンの両方が IdM サーバーで管理されていること。
- 両方のゾーンで動的更新が有効になっていること。

  動的更新の有効化については、「ダイナミック DNS 更新の有効化」で説明されています。
- 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って IP アドレスを変更すると、PTR レコードの上書きが可能になります。

17.6.9.1. UI でのゾーンエントリー同期の設定

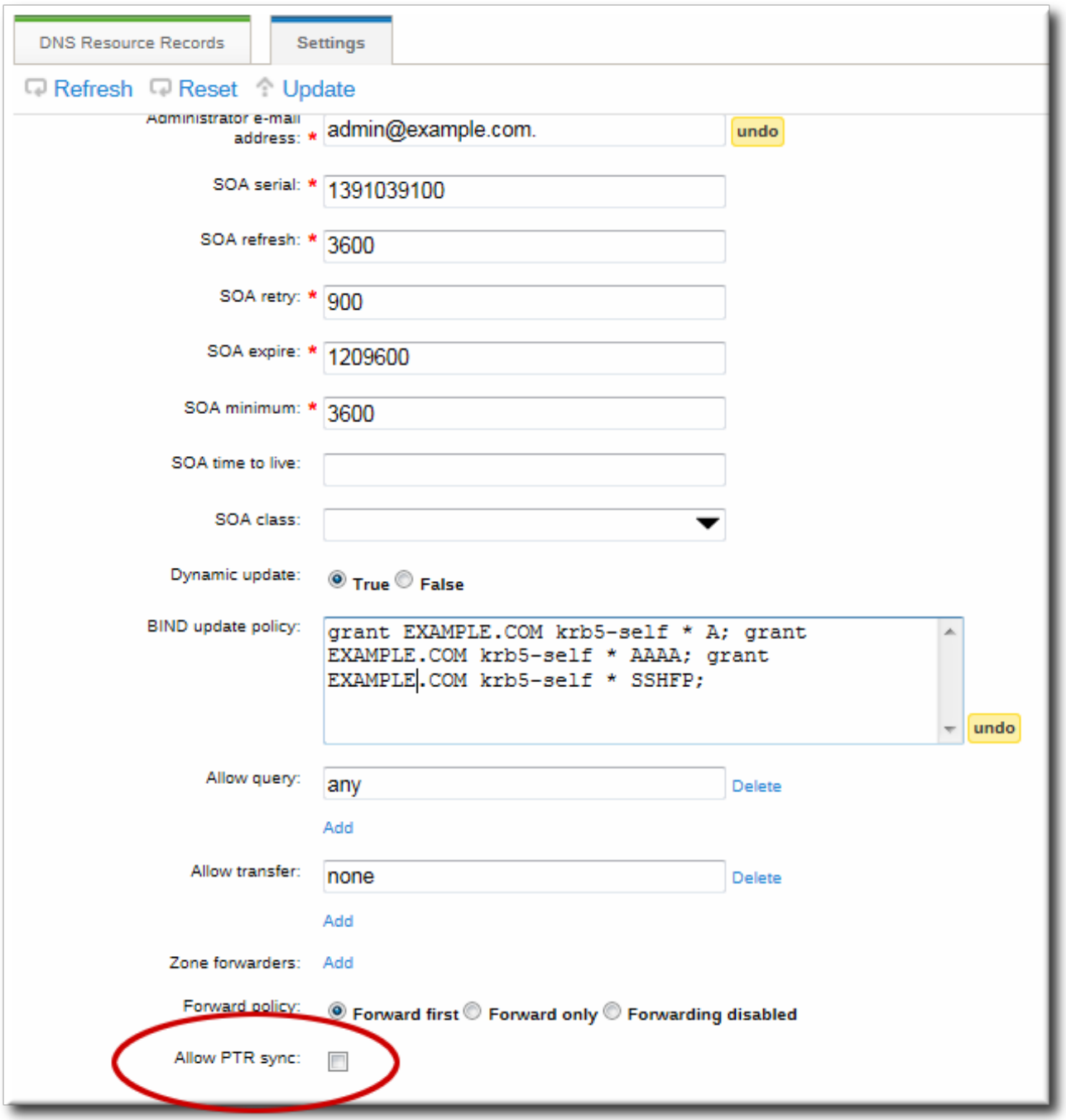
注記

これは、逆引き DNS サーバーではなく、**正引き**ゾーンサーバーに設定されます。

他の DNS ゾーン設定 (「Web UI でのゾーン設定編集」) と同様に、ゾーン転送設定は指定の DNS ゾーンの **Settings** タブにあります。

PTR 同期を有効にするには、**Allow PTR Sync** チェックボックスを選択します。

図17.5 DNS ゾーンの同期設定



17.6.9.2. コマンドラインでゾーンエントリー同期の設定

DNS ゾーンは、**--allow-sync-ptr** オプションを **1** に設定して、正引きおよび逆のエントリーを自動的に同期するように設定できます。これは、ゾーンの作成時または編集時に実行できます。

注記

これは、逆引き DNS サーバーではなく、**正引き**ゾーンサーバーに設定されます。

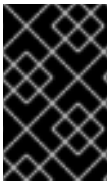
たとえば、既存のエントリーを編集するには、以下のコマンドを実行します。

```
[jsmith@server ~]$ ipa dnszone-mod --allow-sync-ptr=1 example-zone
```

デフォルトは **0** で、同期を無効にし、サーバーのパフォーマンスが向上します。

17.6.10. DNS アクセスポリシーの設定

IdM DNS ドメインは、ゾーンの付与/拒否ルールに基づいてアクセス制御を定義できます。これにより、**update-policy** ステートメントが **/etc/named.conf** ファイルに作成されます。これは DNS アクセスルールを定義します。



重要

更新ポリシーを 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ります。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。

17.6.10.1. UI での DNS アクセスポリシーの設定

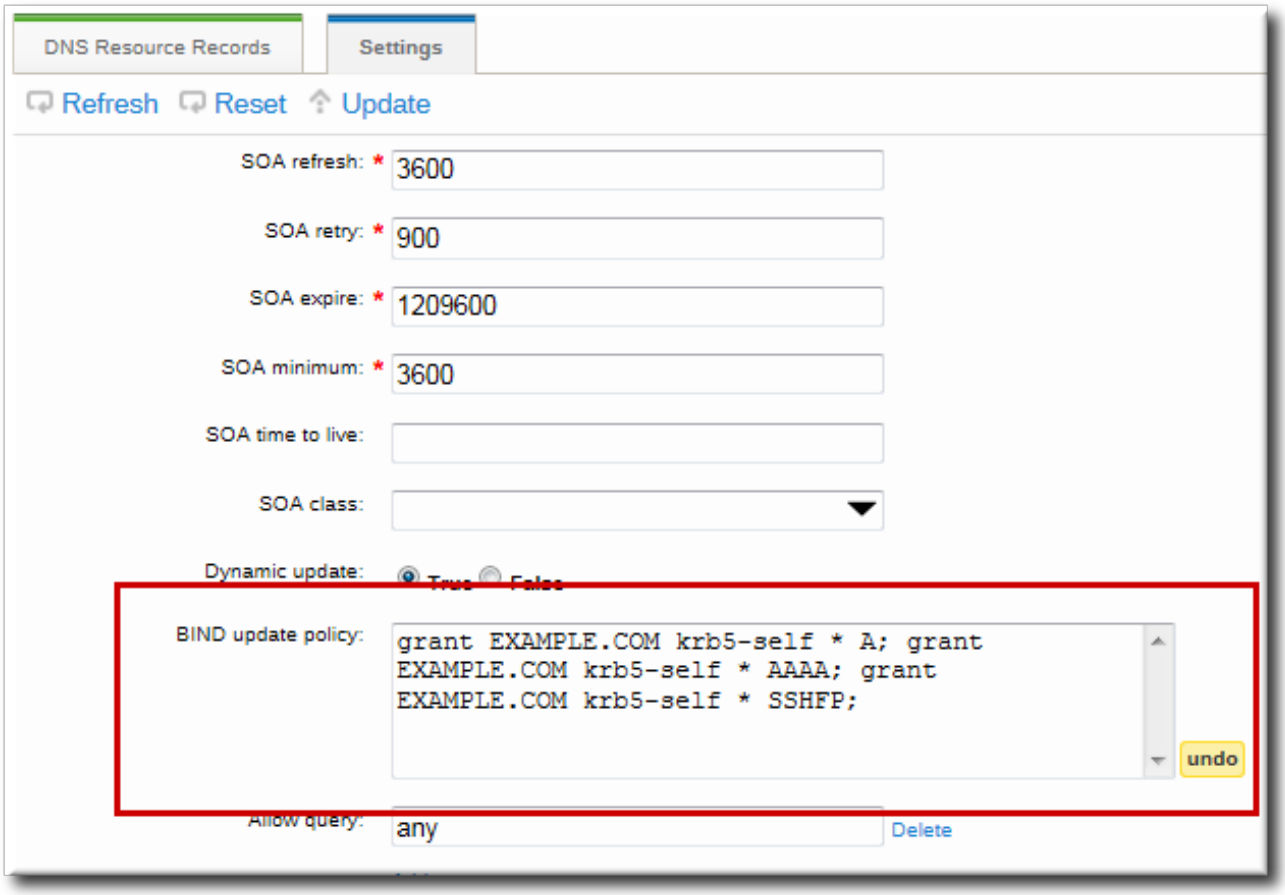
ゾーンアクセスポリシーは、DNS ゾーンの特定の部分に対する一般的な付与または拒否ルールを設定する **アクセス制御命令** です。フルステートメントは、ゾーン名と、クライアントがゾーン内の特定のレコードおよびレコードタイプを編集できるようにする方法を示します。

grant|deny *zoneName policyName recordName recordType*

他の DNS ゾーン設定 (「Web UI でのゾーン設定編集」) と同様に、ゾーン転送設定は指定の DNS ゾーンの **Settings** タブにあります。

アクセスポリシーは、**BIND update policy** のテキストボックスのセミコロン区切りリストで設定されます。

図17.6 DNS 更新ポリシーの設定



サポートされるレコードタイプの完全リストは、表17.2「DNS レコードタイプ」にあります。

17.6.10.2. コマンドラインで DNS アクセスポリシーの設定

コマンドラインツールを使用する場合、ポリシーは **--update-policy** オプションを指定して設定され、その後のステートメントにアクセス制御ルールを使用します。

```
--update-policy "grant|deny zoneName policyName recordName recordType"
```

- **ZoneName** は、ルールを適用する IdM DNS ゾーンです。
- **PolicyName** は、BIND ルールに使用する名前です。
- **recordName** は、ルールを適用するリソースレコードを設定します。アスタリスク (*) は自己ルールに使用されます。
- **RecordType** は、ルールが適用されるレコードタイプです。更新アクセスルールは、同じ DNS ゾーンエントリー内であっても、レコードタイプごとに個別に適用されます。

  サポートされるレコードタイプの完全リストは、表17.2「DNS レコードタイプ」にあります。

たとえば、**EXAMPLE.COM** ゾーンに独自の A および AAAA リソースレコードエントリーを編集する機能を許可するには、次のコマンドを実行します。

```
$ ipa dnszone-mod example.com --update-policy="grant EXAMPLE.COM krb5-self * A; grant EXAMPLE.COM krb5-self * AAAA;"
```

17.7. DNS レコードエントリーの管理

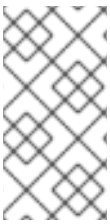
17.7.1. DNS ゾーンへのレコードの追加

IdM は、表17.2「DNS レコードタイプ」に記載されているさまざまなタイプの DNS レコードに対応します。

表17.2 DNS レコードタイプ

A	CERT	KX	NS	SIG
AAAA	CNAME	LOC	NSEC	SRV
A6	DNAME	MX	PTR	SSHFP
AFSDB	DS	PNATR	RRSIG	TXT

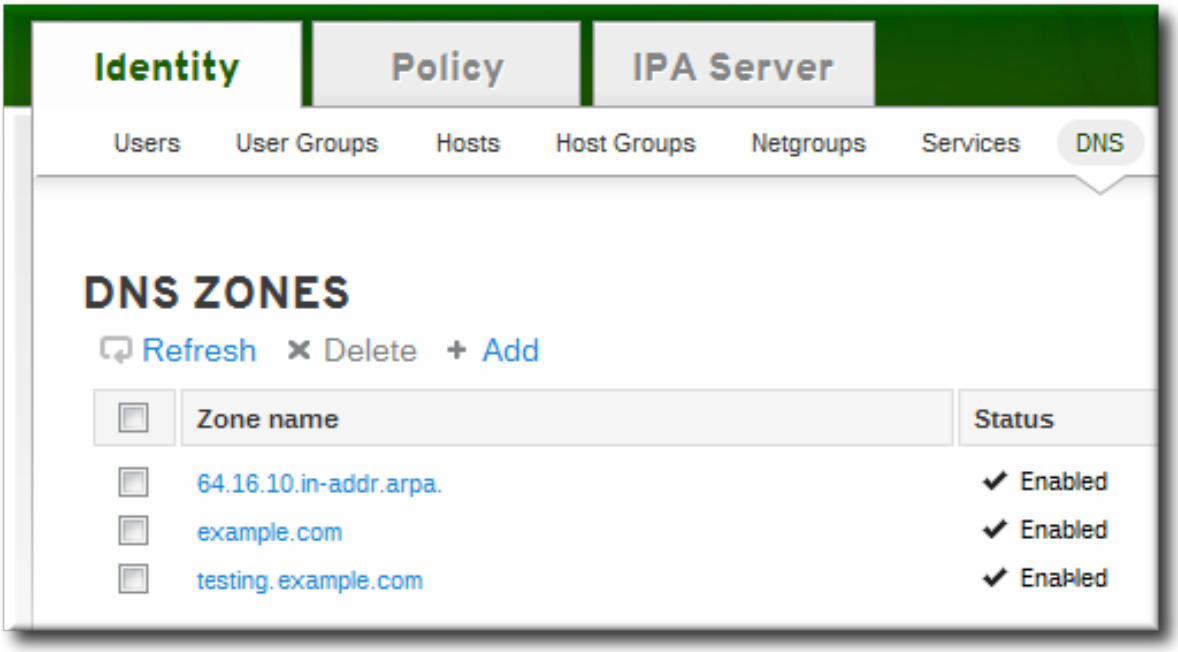
17.7.1.1. Web UI での DNS リソースレコードの追加



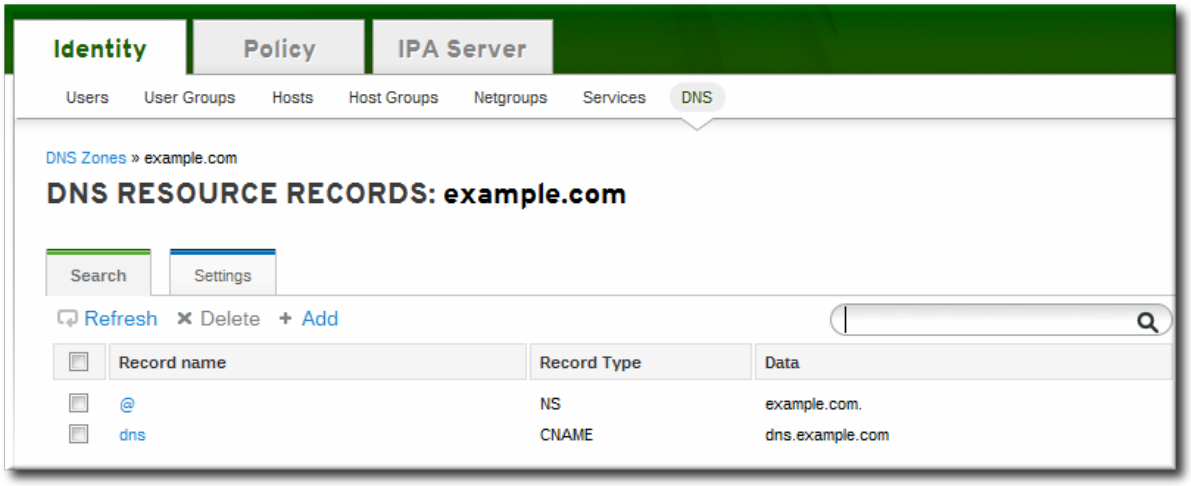
ヒント

name サービスを再起動せずに新しいリソースレコードを即座に解決できるようにするには、**named** サービスを使用した永続的な検索を有効にするか、BIND サービスを設定してゾーンの変更を自動的にポーリングします。「永続検索の無効化」を参照してください。

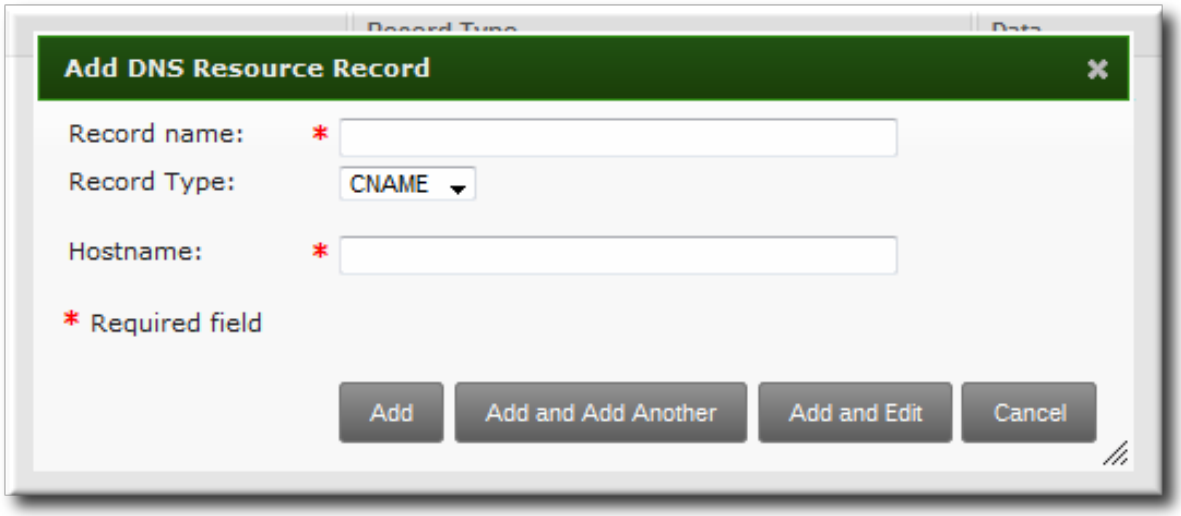
1. **Identity** タブを開き、**DNS** サブタブを選択します。
2. レコードを追加する DNS ゾーンの名前をクリックします。



3. **DNS Resource Record** タブで、**Add** リンクをクリックします。



4. **Record Type** ドロップダウンメニューで作成するレコードのタイプを選択します。レコードタイプに応じて、必要なデータは異なります。たとえば、CNAME レコードにはホスト名が必要です。データフィールド名は、どのような情報を提供すべきかを示すために自動的に更新されます。



IdM は多くの異なるレコードタイプをサポートしますが、使用されている 4 つの頻繁にレコードタイプが使用されます。

- **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。**レコード名** は **www** などのホスト名です。**IP アドレス** の値は、**192.168.1.2** などの標準の 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。**レコード名** は **www** などのホスト名です。**IP アドレス** の値は、**fe80::20c:29ff:fe02:a1b3** などの標準の 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。**レコード名** のフォーマットは **_service._protocol** です (例: **_ldap._tcp**)。ターゲットサービスの優先順位、重み、ポート番号、およびホスト名を設定する個々のフィールドがあります。

  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。この場合、**Record Name** はリソースの DNS エントリーのレコード ID 番号で、**Hostname** の値は、**server.example.com.** などの端末期間が含まれるホスト名になります。

  PTR レコードの詳細は RFC 1035 を参照してください。

5. **Add** ボタンをクリックして、新しいリソースレコードを保存します。

17.7.1.2. コマンドラインでの DNS リソースレコードの追加

同じスクリプト **ipa dnsrecord-add** は、すべてのタイプのリソースレコードを追加するために使用されますが、スクリプトと必要なデータのオプションは、リソースレコードタイプによって異なります。

17.7.1.2.1. DNS レコードを追加するコマンドについて

この **ipa dnsrecord-add** コマンドは、種類に基づいてレコードを DNS ゾーンに追加します。レコードの追加は、基本的なコマンド形式と同じです。

```
$ ipa dnsrecord-add zoneName recordName --recordType-option=data
```

zoneName は、レコードを追加する DNS ゾーンの名前です。 **recordName** は、新しい DNS リソースレコードの識別子です。

表17.3「一般的な dnsrecord-add オプション」 では、A (IPv4)、AAAA (IPv6)、SRV、および PTR という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。対応しているその他のレコードタイプのオプションは、**ipa dnsrecord-add** help および man ページに記載されています。

注記

この **ipa dnsrecord-add** コマンドは、リバースエントリーではなく、フォワードエントリーのみを作成します。

表17.3 一般的な 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。
--class=IN | CS | CH | HS	レコードのクラスを設定します。これは通常 IN です (インターネットプロトコルの場合)。
--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 レコードのコンマ区切りリストを渡します。
--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) レコードのコンマ区切りリストを渡します。
--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 レコードのコンマ区切りリストを渡します。
--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 レコードのコンマ区切りリストを渡します。
--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-target=**string**	ターゲットホストのドメイン名を提供します。該当サービスがドメイン内で利用可能でない場合は、単一のピリオド (.) にすることもできます。

17.7.1.2.2. DNS リソースレコードの追加例

ヒント

name サービスを再起動せずに新しいリソースレコードを即座に解決できるようにするには、**named** サービスを使用した永続的な検索を有効にするか、BIND サービスを設定してゾーンの変更を自動的にポーリングします。「永続検索の無効化」を参照してください。

> **例17.6 IPv4 レコード**
>
> タイプ A リソースレコードはホスト名を IPv4 アドレスにマップします。これらのコマンドの **レコード** 値は、標準の IPv4 アドレスになります。URL ラベルは通常 www です。

```
$ ipa dnsrecord-add example.com www --a-rec 10.64.14.165
```

これにより、IP アドレスが 10.64.14.165 のレコード www.example.com が作成されます。

A レコードの詳細は RFC 1035 を参照してください。

例17.7 IPv4 レコードの変更

A レコードの値を指定するオプションは 2 つあります。レコードを作成する場合、オプションは **--a-record** になります。ただし、A レコードを変更すると、**--a-record** オプションは A レコードの **古い** 値を表示します。新しい値は、**--ip-address** オプションで設定します。

```
$ ipa dnsrecord-mod example.com www --a-rec 10.1.1.1 --ip-address 10.1.1.2
```

例17.8 IPv6 レコード

AAAA リソースレコード (*quad-A レコード*) はホスト名を IPv6 アドレスにマップします。これらのコマンドの **record** の値は、IPv6 アドレスです。タイプ A レコードと同様に、URL ラベルは通常 www です。

```
$ ipa dnsrecord-add example.com www --aaaa-rec fe80::20c:29ff:fe02:a1b3
```

これにより、IP アドレス fe80::20c:29ff:fe02:a1b3 のレコード www.example.com が作成されます。AAAA レコードの詳細は RFC 3596 を参照してください。

例17.9 SRV レコード

*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。

タイプ A および Type AAAA レコードと同様に、SRV レコードはサービスへの接続および特定方法を指定しますが、レコード形式は異なります。

recordName は、**_service._protocol** 形式で、サービスタイプと接続プロトコルを識別します。

レコード情報には、**「優先順位の重みポートのターゲット」**の形式があります。

```
[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."
```

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 レコードの詳細は、RFC 2782 を参照してください。

例17.10 PTR レコード

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。

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。

```
$ ipa dnsrecord-add reverseZone recordName --ptr-rec FQDN
```

recordName および **reverseZone** は、次の方法で連結したときに有効な逆名を作成する必要があります (**recordName.reverseZone**)。

たとえば、これにより、IP アドレス **192.0.1.2** のホスト **server2.example.com** の **1.0.192.in-addr.arpa.** 逆引きゾーンに逆引き DNS エントリーが追加されます。

```
$ ipa dnsrecord-add 1.0.192.in-addr.arpa. 2 --ptr-rec server2.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.
```

注記

PTR レコードの詳細は、以下のリソースを参照してください。

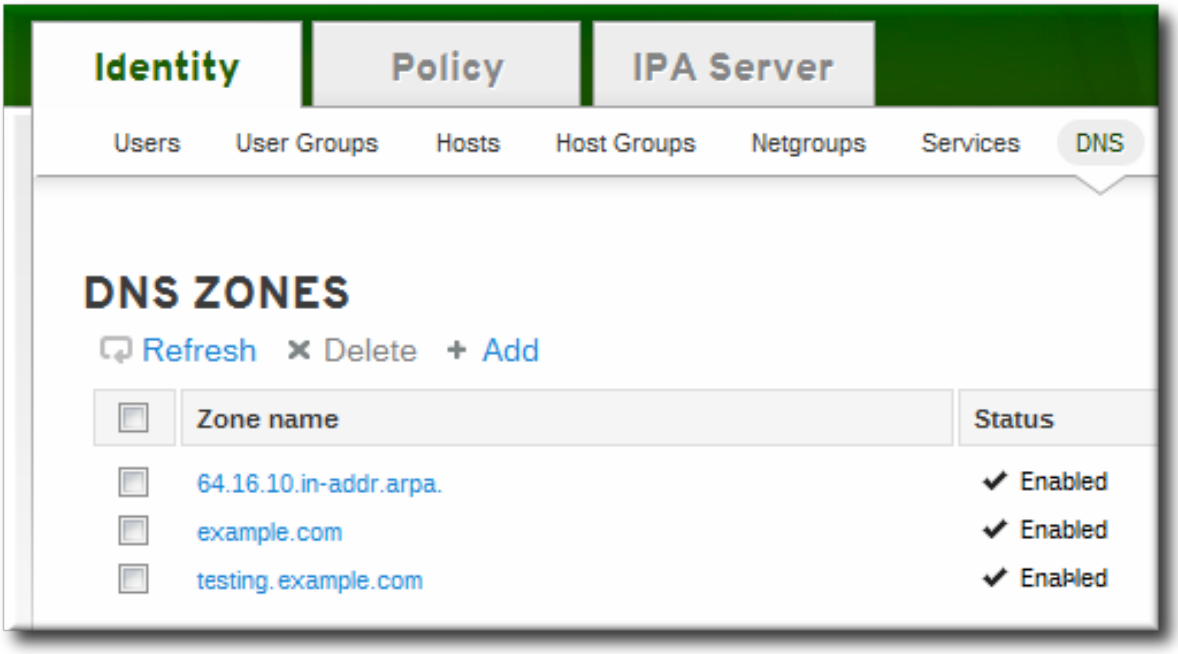
- RFC 1035 は、IPv4 in-addr.arpa ドメインの仕様を記述します。
- RFC 2317 では、addr.arpa 委譲の IPv4 クラスレスが説明されています。
- RFC 3596 では、IPv6 をサポートする拡張が説明されています。

17.7.2. DNS ゾーンからレコードを削除する

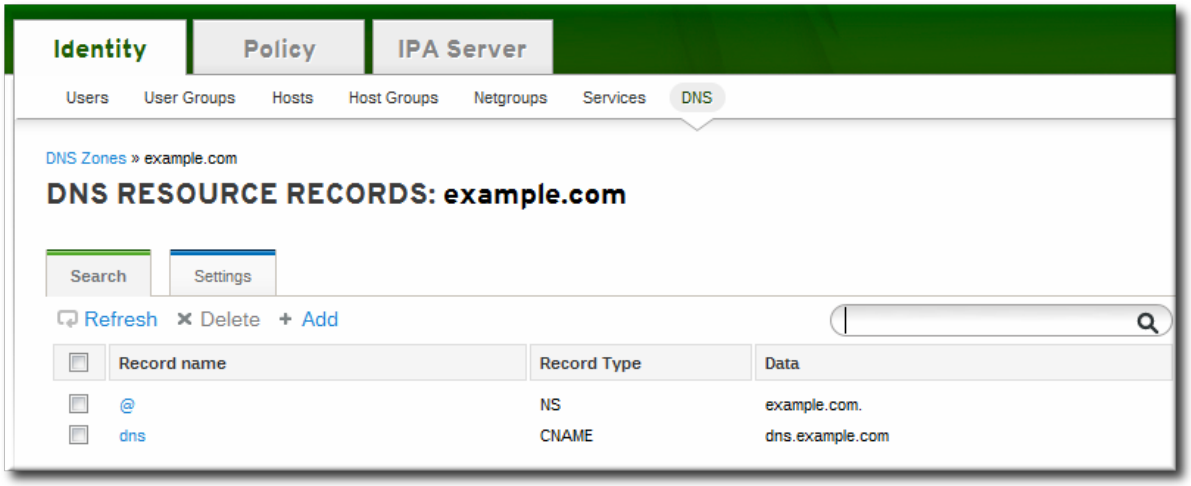
17.7.2.1. Web UI でレコードの削除

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。

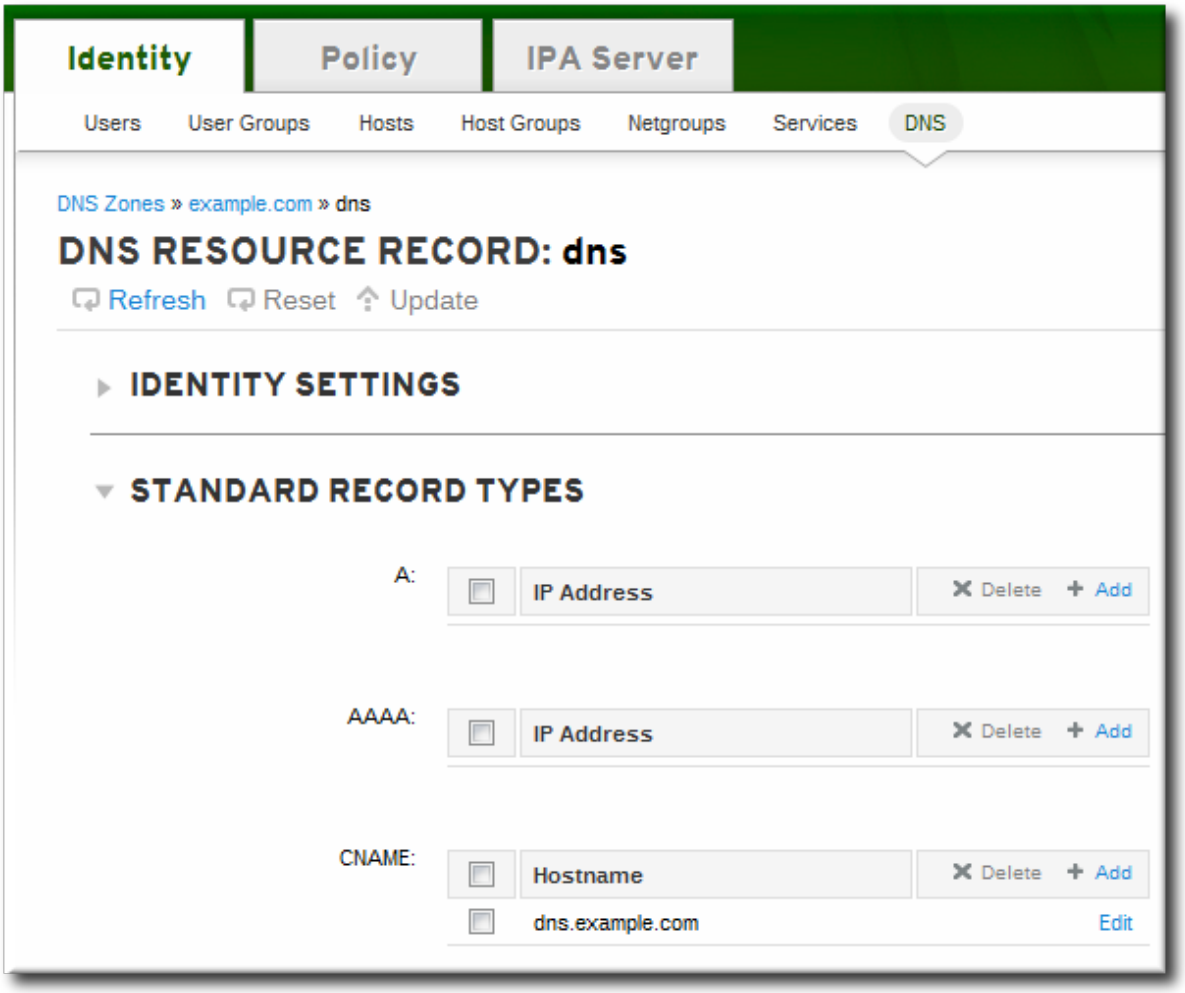
1. **Identity** タブを開き、**DNS** サブタブを選択します。
2. DNS ゾーンの名前をクリックします。



3. **DNS Resource Record** タブで、リソースレコードの名前をクリックします。



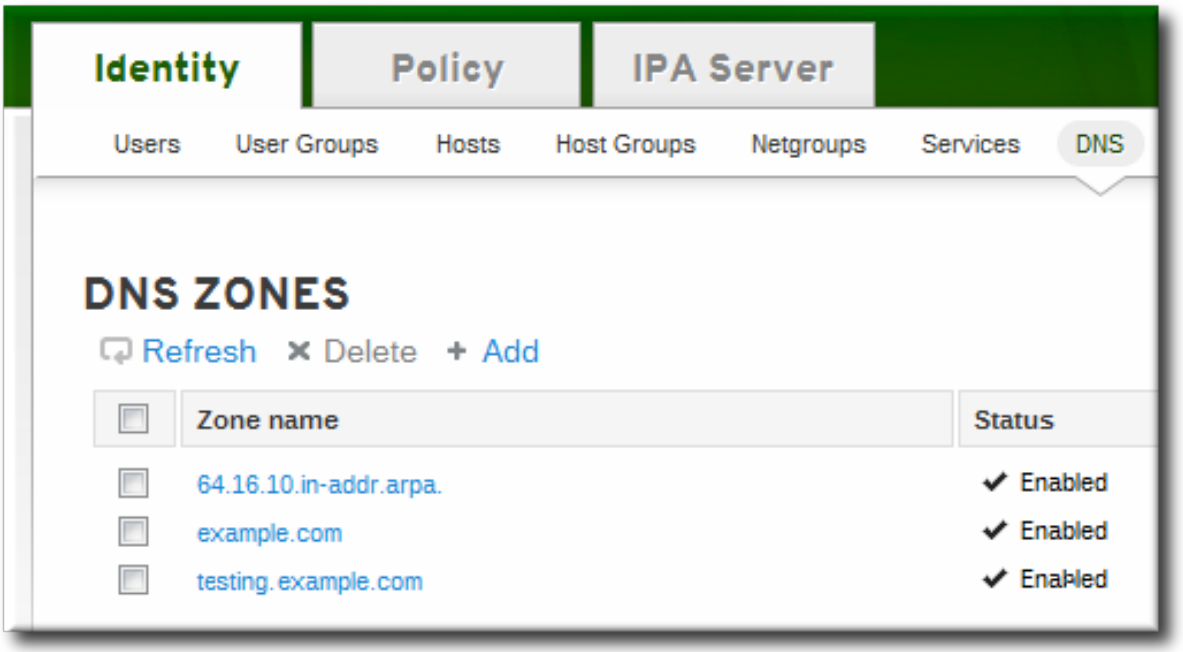
4. 削除するレコードタイプ名のチェックボックスをクリックし、一覧の上部にあるアクティブな **Delete** リンクをクリックします。



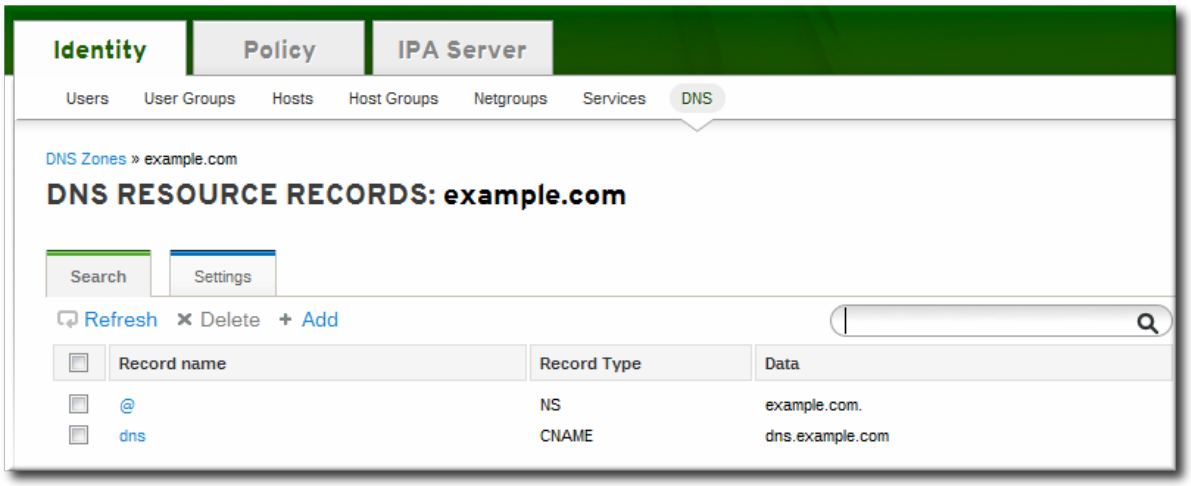
これにより、他の設定をそのまま残すと、そのレコードタイプのみが削除されます。

ゾーン内のリソースのすべてのレコードを削除します。

1. **Identity** タブを開き、**DNS** サブタブを選択します。
2. DNS ゾーンの名前をクリックします。



3. **DNS Resource Record** タブで、削除するリソースレコードの名前でチェックボックスを選択します。これにより、レコード全体が削除されます。



4. ゾーンレコードページの上部にある **Delete** リンクをクリックします。

17.7.2.2. コマンドラインでレコードの削除

ipa dnsrecord-del コマンドを使用すると、レコードがゾーンから削除されます。レコードの追加と同様に、レコードの種類 (**--recordType-rec**) とレコード値を指定するオプションを使用してレコードが削除されます。

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 10.64.14.213
```

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が要求されます。

あるいは、**--del-all** オプションを使用して、ゾーンに関連付けられたすべてのレコードを削除します。

17.8. BIND-DYNDB-LDAP プラグインの設定

bind-dyndb-ldap システムプラグインには、ゾーン用の DNS レコードキャッシュと、成功した DNS 解決の履歴が含まれます。新しい DNS リクエストが存在するたびにディレクトリーサービスのクエリーを実行する必要がなくなるため、キャッシュを維持すると Directory Server でルックアップパフォーマンスが向上します。

このプラグインがインストールされ、IdM が DNS を管理するように設定すると、プラグイン設定に新しい設定セクションが追加されます。

例17.11 デフォルトの dynamic-db 設定

```
dynamic-db "ipa" {
        library "ldap.so";
        arg "uri ldapi://%2fvar%2frun%2fslapd-EXAMPLE.socket";
        arg "base cn=dns,dc=example,dc=com";
        arg "fake_mname server.example.com.";
        arg "auth_method sasl";
        arg "sasl_mech GSSAPI";
        arg "sasl_user DNS/server.example.com";
```

```
        arg "zone_refresh 0";
        arg "psearch yes";
        arg "serial_autoincrement 1";
};
```

この設定は、キャッシュが維持される期間など、**他** のプラグインの動作に暗黙的なデフォルト値を使用します。仮定すると、**dynamic-db "ipa"** エントリーに引数を追加することにより、デフォルト設定を変更できます。

```
arg "argument value";
```

追加パラメーターは、表17.4「追加の bind-dyndb-ldap 設定パラメーター」 に一覧表示されています。

注記

ネームサーバーをリロードすることで、キャッシュの更新と新しいゾーンの検出の両方を強制できます。

```
# rndc reload
```

表17.4 追加の bind-dyndb-ldap 設定パラメーター

パラメーター	説明	デフォルト値
cache_ttl	Directory Server の DNS 設定に新しいゾーンがあるかどうかを確認します。	120 (秒)。これは bind-dyndb-ldap プラグインで定義されます。
zone_refresh	サーバーが新しいゾーンについて Directory Server の DNS 設定を確認する頻度 (秒単位) をチェックします。	0 (無効)
psearch	Directory Server の永続的な検索を有効にし、BIND サービスが新しい DNS ゾーンが追加されると直ちに更新通知を受け取ります。	yes

17.8.1. DNS キャッシュ設定の変更

DNS のパフォーマンスを向上させるためには、キャッシュ設定を変更する必要がある場合があります。デフォルトでは、DNS レコードはキャッシュに保持され、120 秒間有効とみなされます。つまり、DNS レコードが変わると、 最大 120 秒間にわたりネームサーバーに伝播されない (される必要がない) ことを意味します。Directory Server にトラフィックボリュームが大きい場合や、レコードが頻繁に変更されない場合は、***cache_ttl*** パラメーターを追加してパフォーマンスを向上させるためにキャッシュ時間を増やすことができます。

```
dynamic-db "ipa" {
...
    arg "cache_ttl 1800";
```

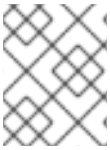
```
};
```

17.8.2. 永続検索の無効化

DNS サービスは、**bind-dyndb-ldap** プラグインを介してその情報を受信します。プラグインは、ネームサーバーの起動時に Directory Server で設定および有効化されたゾーンのみを解決します。ネームサービスが再起動すると、プラグインはその設定を再読み込みし、新しいゾーンまたは新しいリソースレコードを特定します。

ただし、**bind-dyndb-ldap** プラグインは、IdM LDAP ディレクトリーからゾーンおよびリソースレコード情報をプルします。また、プラグインを再起動するだけで、そのディレクトリーから情報をプルできます。**bind-dyndb-ldap** プラグインは、Directory Server への永続的な接続を開き、変更を即座にキャッチすることで、ゾーンの変更をアクティブに検索します。

永続的な検索により、変更を即時に通知し、設定データのローカルキャッシュを維持します。



注記

永続的な検索では、ゾーンとゾーンリソースレコードの両方に更新がキャッチされます。

永続的な検索では、Directory Server との接続が継続されるため、パフォーマンスの問題が発生する可能性があります。パフォーマンスへの影響は、Red Hat Directory Server Administrator's Guide に記載されています。

永続的な検索はデフォルトで有効になっていますが、***psearch*** 引数で無効にできます。

```
dynamic-db "ipa" {
...
        arg "psearch no";
 };
```

17.9. 再帰クエリーの変更

ipa-client-install スクリプトは、IdM DNS ドメイン外にあるホストに対して名前解決を可能にする設定ステートメントを **/etc/named.conf** ファイルに設定します。(これには、DNS が設定され、フォワーダーが設定されている状態で IdM サーバーを設定する必要があります。) つまり、すべてのホストが、設定されたフォワーダーに対して再帰クエリーを発行できることを意味します。

デフォルトでは、設定されたフォワーダーに対して再帰クエリーを実行することが許可されます。IdM インストールスクリプトは、**/etc/named.conf** ファイルに自動的に行を追加して、再帰クエリーを許可します。

```
        forward first;
        forwarders { 10.16.36.29; };
        allow-recursion { any; };
```

この動作は **allow-recursion** ステートメントで変更できます。

1. **/etc/named.conf** ファイルを開きます。
2. **allow-recursion** ステートメントをリセットします。これは、デフォルトで **any** に設定され、すべてのホストがすべてのフォワーダーに対して名前を解決できるようにします。

```
forward first;
forwarders { 10.16.36.29; };
allow-recursion { any; };
```

3. **named** サービスを再起動します。

```
service named restart
```

ネームサーバーのドキュメントでは、設定ステートメントの編集に関する詳細が記載されています。

17.10. IDM ドメインのホスト名の解決

この **dns-resolve** コマンドを使用して、IdM ドメインメンバーの DNS エントリーを確認することができます。レコードが存在し、DNS 設定で適切にフォーマットされると、コマンドは DNS レコードを返します。そうでない場合は、ホスト名が DNS サービス内で認識されないというエラーが返されます。

```
$ipa dns-resolve server1.example.com
```

これは、サーバー、クライアント、およびサービス間の接続問題のトラブルシューティングに役立ちます。

---

[6] 更新された DNS スキーマ要素を含む更新済みスキーマファイルは、すべて **/usr/share/ipa/updates** ディレクトリーに置かれます。

第18章 ポリシー: 自動マウントの使用

自動マウントは、ユーザーによる要求時に、異なるサーバーでディレクトリーを自動的に利用できるようにする方法です。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」)。

IdM では、自動マウントは内部 LDAP ディレクトリーで動作します。また、設定されている場合は DNS サービスと動作します。

18.1. 自動マウントと IDM

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。自動マウントは、リソースが要求されるたびにディレクトリーを自動的にマウントします。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または **マウントポイント** は **鍵** 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 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 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。

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**、コマンドの **ipa automountlocation*** 使用
- **ipa automountmap*** コマンドを使用したダイレクトマップと間接・直接 **マップ** の両方
- **キー**、コマンドの **ipa automountkey*** 使用

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。

18.2. 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。

ヒント

自動マウント設定を変更する前に、コマンドラインから **/home** ディレクトリーを正常にマウントできることをテストします。NFS が既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解決が容易になります。

18.2.1. 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** コマンドは 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@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=raleigh
```

この **ipa-client-automount** コマンドは、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
```

必要な NFS 設定ファイルすべて - ファイルの一覧は、SSSD を使用しないと若干異なります。

- 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。

18.2.2. 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一覧に追加します。

```
[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 ~]# service sssd restart
```

6. SSSD が自動マウント設定のソースとして一覧表示されるように、**/etc/nsswitch.conf** ファイルを確認します。

```
automount: sss files
```

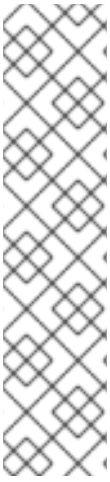
7. Restart autofs:

```
[root@server ~]# service autofs restart
```

8. ユーザーの **/home** ディレクトリーを一覧表示して、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。



ヒント

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。

18.2.3. 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 構成に使用します。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** ディレクトリーを一覧表示します。

```
# ls /home/userName
```

18.3. KERBERIZED NFS サーバーの設定

Identity Management を使用して Kerberized NFS サーバーを設定できますが、Red Hat Enterprise Linux で実行する必要はありません。

18.3.1. Kerberized NFS サーバーの設定

1. IdM ユーティリティを実行する前に、Kerberos チケットを取得します。

```
[user@server ~]$ kinit admin
```

2. NFS ホストマシンが IdM ドメインにクライアントとして追加されていない場合は、「ホストエントリーを追加する他の例」の説明に従って GUI でホストエントリーを作成するか、以下のようなコマンドを実行します。

```
[user@server ~]$ ipa host-add --ip-address 192.0.2.10 nfs-server.example.org
```

3. IdM ドメインに NFS サービスエントリーを作成します。以下に例を示します。

```
[user@server ~]$ ipa service-add nfs/nfs-server.example.com
```

詳細は「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。

4. **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生成します。

NFS サーバーは、IdM ドメインの Red Hat Enterprise Linux マシンまたは別の Unix マシン上にある場合があります。Red Hat Enterprise Linux マシンでは、NFS サーバーマシンで **ipa-getkeytab** コマンドを実行できます。それ以外の場合は、IdM ドメインの Red Hat Enterprise Linux マシンで **ipa-getkeytab** コマンドを実行してから、NFS サーバーにコピーする必要があります。

ipa-getkeytab コマンドが NFS サーバーで実行されている場合は、キーをホストキータブに直接保存します。たとえば、以下のようになります。

```
[user@server ~]$ ipa-getkeytab -s server.example.com -p nfs/nfs-server.example.com -k
/etc/krb5.keytab
```

Red Hat Enterprise Linux マシンでは、必要なのはそれだけです。

別のシステムにコピーするキーを生成する場合は、鍵を生成しますが、その鍵はホストのキータブに保存されません。キーは、NFS サーバーにコピーした後にキーをキータブに個別に追加する必要があります。

a. キータブを一時ファイルに保存します。以下に例を示します。

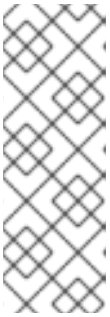
```
[user@server ~]$ ipa-getkeytab -s server.example.com -p nfs/nfs-server.example.com -k
/root/nfs-server.keytab
```

b. キータブを NFS サーバーにコピーします。

c. ファイルのパーミッションを **0700** に設定します。

d. サービスキーをキータブファイルに追加します。

```
[root@nfs-server ~]# ( echo rkt /root/nfs-server.keytab; echo wkt /etc/krb5.keytab ) |
ktutil
```



注記

NFS サービスがキータブで IdM で適切に設定されていることを確認するには、次のコマンドを実行してサービスエントリーを確認します。

```
[user@server ~]$ ipa service-show nfs/ipaclient2.example.com
Principal: NFS/ipaclient2.example.com@EXAMPLE.COM
Keytab: True
```

5. NFS パッケージをインストールします。以下に例を示します。

```
[root@nfs-server ~]# yum install nfs-utils
```

6. 弱い暗号化サポートを設定します。ドメインのクライアント (Red Hat Enterprise Linux 5 クライアントなど) が DES などの古い暗号化オプションを使用する場合は、NFS クライアントごとに必要です。

   a. 以下の行を追加して **krb5.conf** ファイルを編集して、弱い暗号化を有効にします。

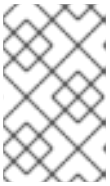
   ```
   allow_weak_crypto = true
   ```

   b. IdM サーバーの Kerberos 設定を更新して、DES 暗号化タイプに対応します。

   ```
   [user@ipaserver ~]$ ldapmodify -x -D "cn=directory manager" -w password -h
   ipaserver.example.com -p 389

   dn: cn=EXAMPLEREALM,cn=kerberos,dc=example,dc=com
   changetype: modify
   add: krbSupportedEncSaltTypes
   krbSupportedEncSaltTypes: des-cbc-crc:normal
   -
   add: krbSupportedEncSaltTypes
   krbSupportedEncSaltTypes: des-cbc-crc:special
   -
   add: krbDefaultEncSaltTypes
   krbDefaultEncSaltTypes: des-cbc-crc:special
   ```

7. **ipa-client-automount** コマンドを実行して、NFS 設定を構成します。

   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。



注記

サーバーが IdM ドメインのメンバーではない場合は (ipa-client パッケージがインストールされていない)、この手順を手動で行う必要があります。詳細は、ストレージ管理ガイドの NFS 設定セクションを参照してください。

8. **/etc/exports** ファイルを編集し、Kerberos 情報を追加します。

```
/export  *(rw,sec=krb5:krb5i:krb5p)
```

9. NFS サーバーおよび関連サービスを再起動します。

```
[root@nfs-server ~]# service nfs restart
[root@nfs-server ~]# service rpcsvcgssd restart
```

10. NFS サーバーを NFS クライアントとして設定する場合は、「Kerberized NFS クライアントの設定」を参照してください。

18.3.2. Kerberized NFS クライアントの設定

1. IdM ツールを実行する前に、Kerberos チケットを取得します。

```
[user@server ~]$ kinit admin
```

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、「ホストエントリーを追加する他の例」の説明に従って GUI で必要なホストエントリーを設定するか 、以下のようなコマンドを実行します。

```
[user@server ~]$ ipa host-add --ip-address 192.0.2.20 nfs-client.example.org
```

3. この **ipa-getkeytab** ユーティリティーを使用して、NFS クライアントの NFS サービスキータブを生成します。

   NFS クライアントは、IdM ドメインの Red Hat Enterprise Linux マシンまたは別の Unix マシン上にある場合があります。Red Hat Enterprise Linux マシンでは、NFS クライアントマシンで **ipa-getkeytab** コマンドを実行できます。それ以外の場合は、IdM ドメインの Red Hat Enterprise Linux マシンで **ipa-getkeytab** コマンドを実行してから、NFS サーバーにコピーする必要があります。

   ipa-getkeytab コマンドが NFS クライアントで実行している場合は、キーをホストキータブに直接保存します。たとえば、以下のようになります。

```
[user@server ~]$ ipa-getkeytab -k /etc/krb5.keytab -s ipa-server.example.org -p nfs/nfs-client-server.example.com@EXAMPLE.COM
```

   Red Hat Enterprise Linux マシンでは、必要なのはそれだけです。

   別のシステムにコピーするキーを生成する場合は、鍵を生成しますが、その鍵はホストのキータブに保存されません。キーは、NFS サーバーにコピーした後にキーをキータブに個別に追加する必要があります。

   a. キータブを一時ファイルに保存します。以下に例を示します。

```
[user@server ~]$ ipa-getkeytab -s ipa-server.example.org -p host/nfs-client-server.example.com@EXAMPLE.COM -k /root/nfs-client.keytab
```

   b. キータブを NFS クライアントにコピーします。

   c. ファイルのパーミッションを **0700** に設定します。

   d. サービスキーをキータブファイルに追加します。

```
[root@nfs-client-server ~]# ( echo rkt /root/nfs-client.keytab; echo wkt /etc/krb5.keytab ) |
ktutil
```

4. **ipa-client-automount** コマンドを実行して、NFS 設定を構成します。

   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。

注記

クライアントが IdM ドメインのメンバーではない場合は (ipa-client パッケージがインストールされていない)、この手順を手動で行う必要があります。詳細は、ストレージ管理ガイドの NFS 設定セクションを参照してください。

5. GSS デーモンを起動します。

```
[root@nfs-client-server ~]# service rpcgssd start
[root@nfs-client-server ~]# service rpcbind start
[root@nfs-client-server ~]# service rpcidmapd start
```

6. ディレクトリーをマウントします。

```
[root@nfs-client-server ~]# echo "$NFSSERVER:/this /mnt/this nfs4
sec=krb5i,rw,proto=tcp,port=2049"  >>/etc/fstab
[root@nfs-client-server ~]# mount -av
```

18.4. 場所の設定

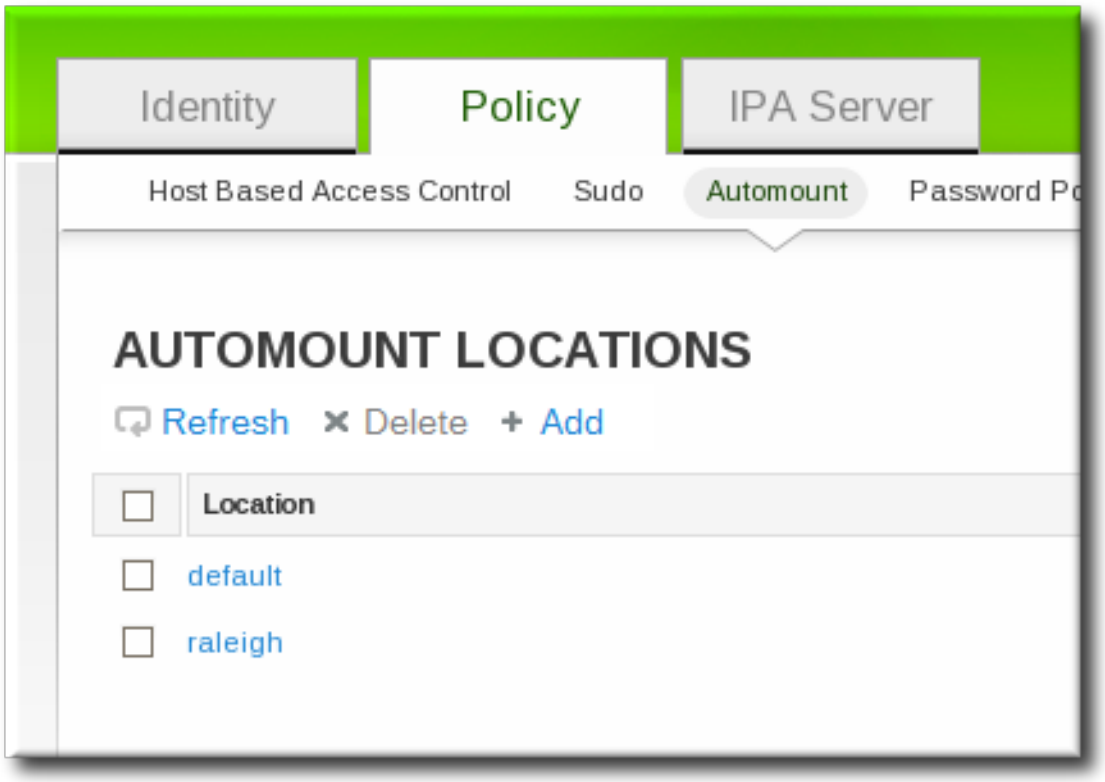
場所はマップのセットで、すべて **auto.master** に保存され、場所は複数のマップを保存できます。また、場所には複数のマップを保存できます。場所のエントリーは、マップエントリーのコンテナーとしてのみ機能します。それ自体は、自動マウント設定ではありません。

重要

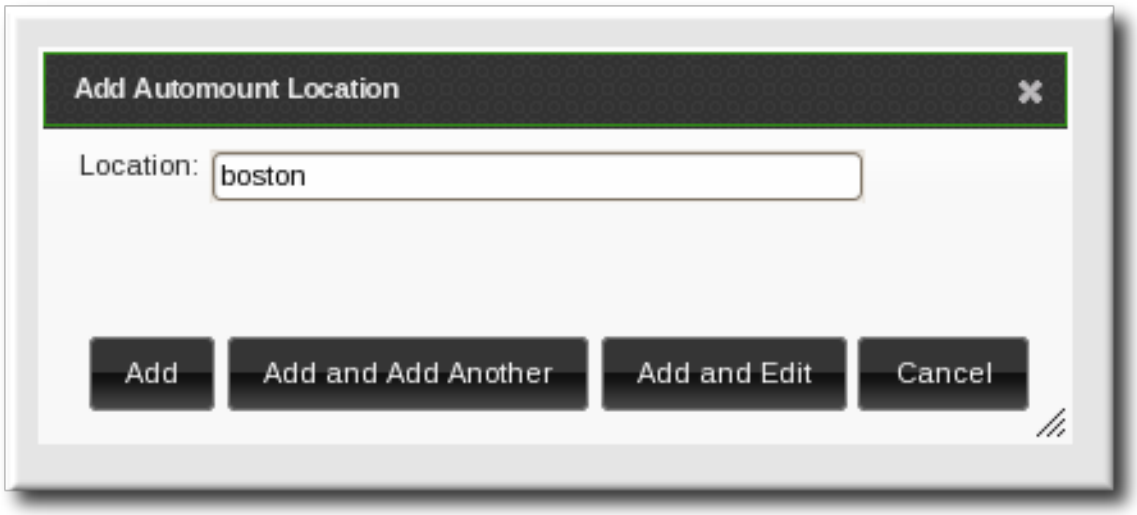
Identity Management は autofs を設定または設定しません。これは個別に行う必要があります。Identity Management は、既存の autofs デプロイメントと動作します。

18.4.1. Web UI での場所の設定

1. **Policy** タブをクリックします。
2. **Automount** サブタブをクリックします。
3. 自動マウントの場所一覧の上部にある **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。

18.4.2. コマンドラインでの場所の設定

マップを作成するには、**automountlocation-add** を使用して場所名を指定します。

```
$ ipa automountlocation-add location
```

たとえば、以下のようになります。

```
$ 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。

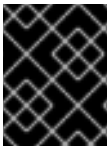
```
$ ipa automountlocation-tofiles raleigh
/etc/auto.master:
/-      /etc/auto.direct
--------------------------
/etc/auto.direct:
```

18.5. マップの設定

マップを設定するとマップが作成されるだけでなく、キーによってマウントポイントに関連付けられ、ディレクトリーにアクセスした際に使用するマウントポイントが割り当てられます。IdM は、ダイレクト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。

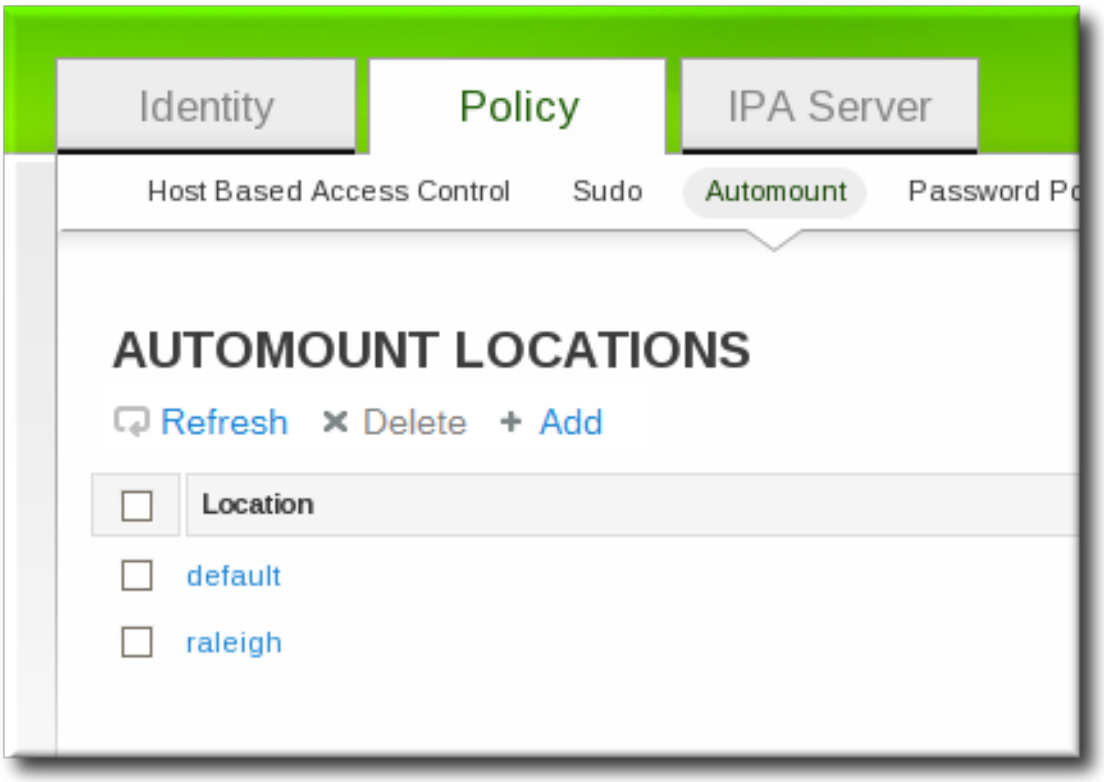
18.5.1. ダイレクトマップの設定

ダイレクトマップは、ファイルマウントへの正確な場所、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。

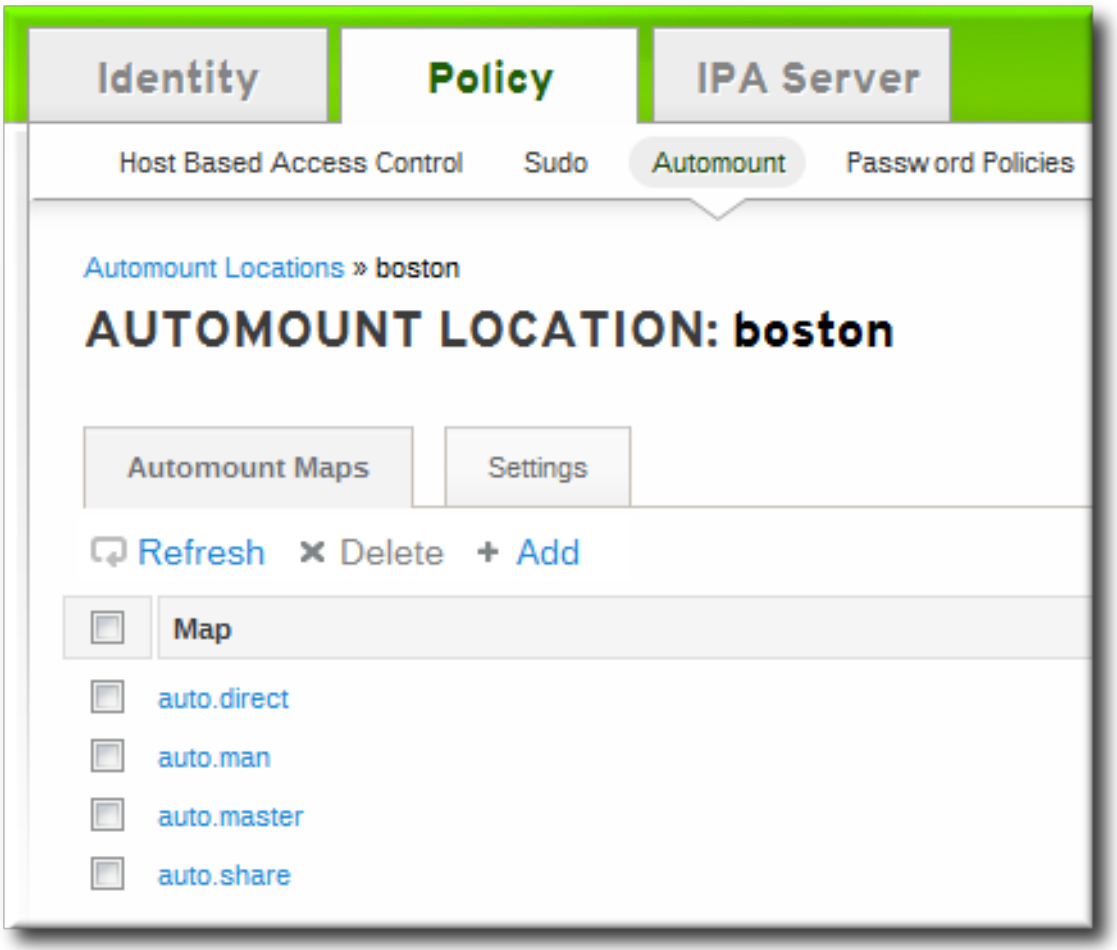
```
--------------------------
/etc/auto.direct:
/shared/man server.example.com:/shared/man
```

18.5.1.1. Web UI でのダイレクトマップの設定

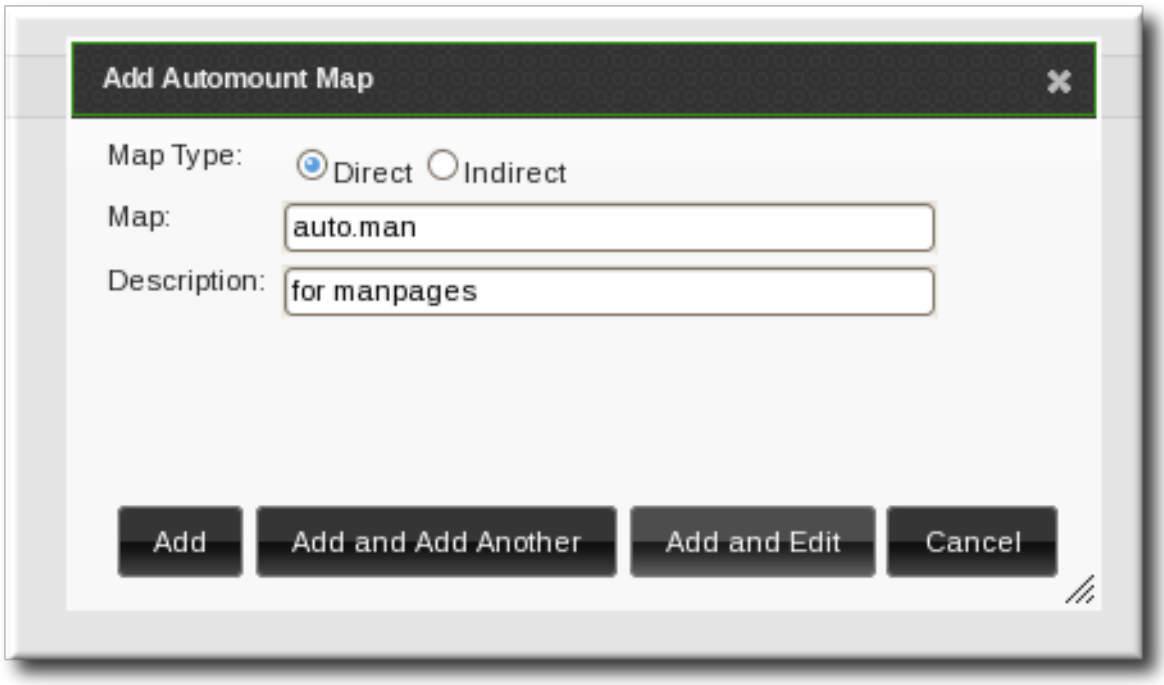
1. **Policy** タブをクリックします。
2. **Automount** サブタブをクリックします。
3. マップの追加先となる automount の場所の名前をクリックします。



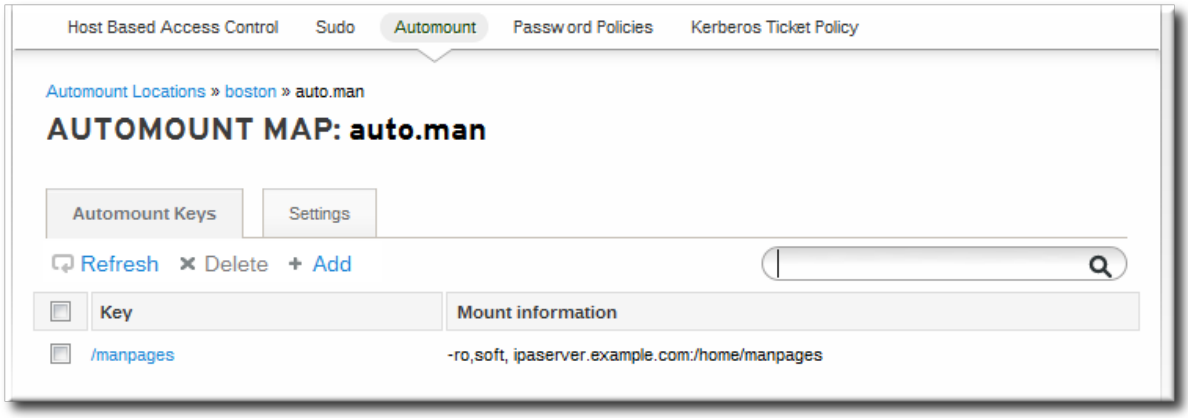
4. **Automount Maps** タブで **Add** をクリックして新規マップを作成します。



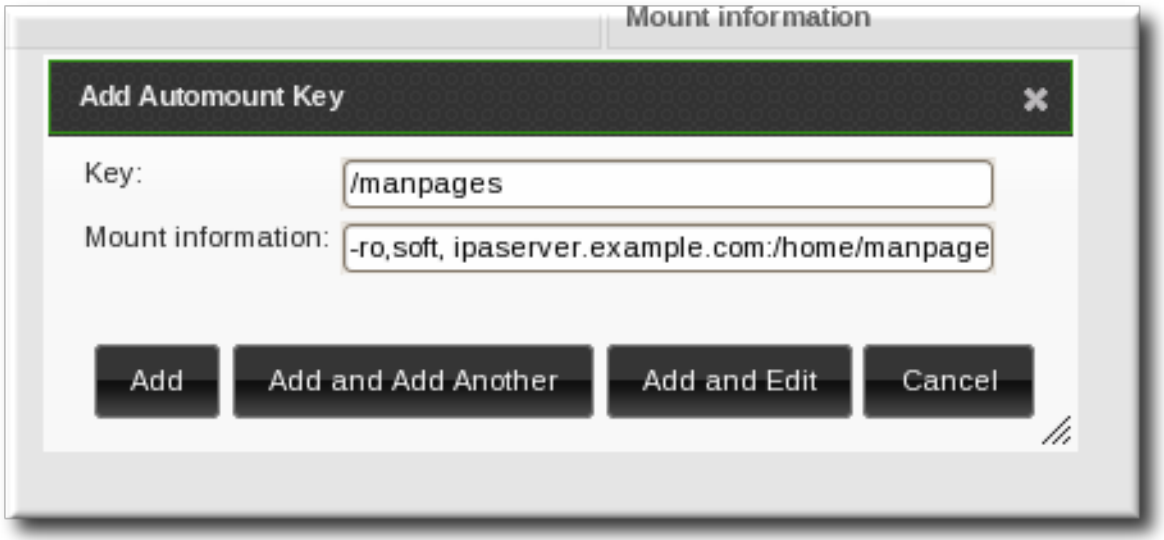
5. ポップアップウィンドウで **Direct** ラジオボタンを選択し、新規マップの名前を入力します。



6. **Automount Keys** タブで **+Add** をクリックしてマップの新規キーを作成します。



7. マウントポイントを入力します。key では、実際のマウントポイントを key の名前で定義します。**Info** フィールドは、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。

18.5.1.2. コマンドラインでのダイレクトマップの設定

key では、実際のマウントポイントとオプションを key の名前で定義します。キーの形式に基づいて、マップはダイレクトまたは間接マップになります。

各場所は **auto.direct** アイテムと共に作成されます。最もシンプルな設定では、automount キーを既存のダイレクトマップエントリーに追加することでダイレクトマップを定義します。異なるダイレクトマップエントリーを作成することも可能です。

ダイレクトマップのキーを場所の **auto.direct** ファイルに追加します。**--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。たとえば、以下のようになります。

```
$ 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。

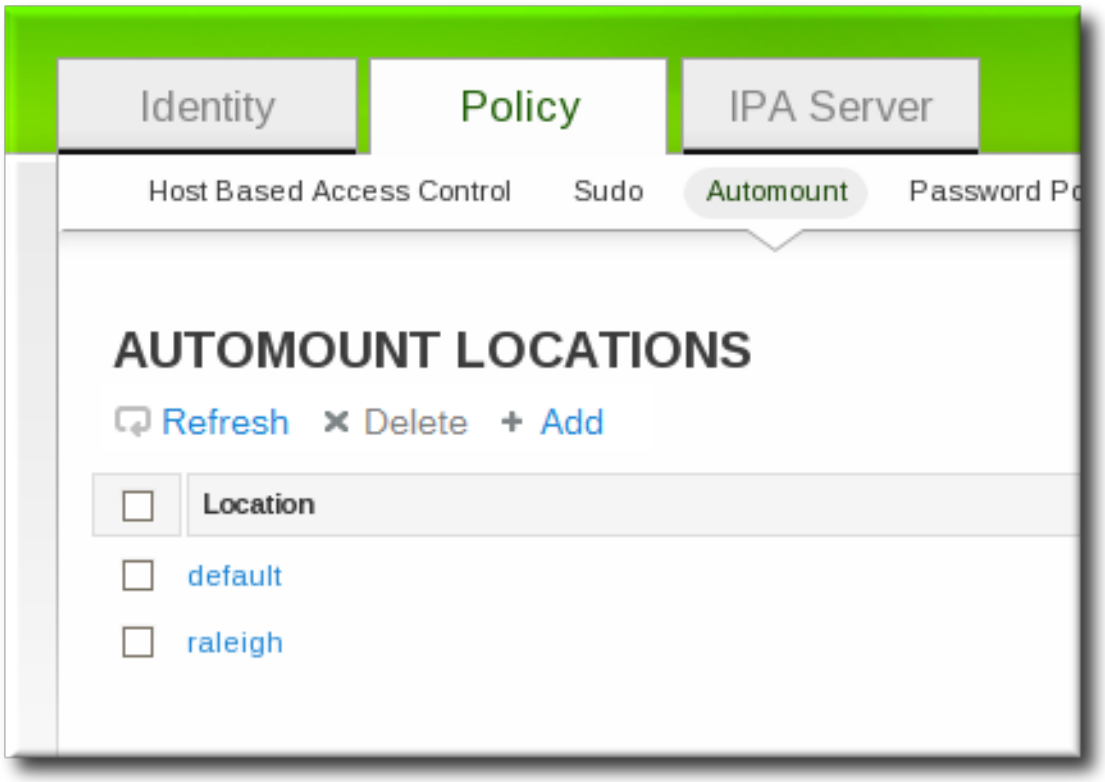
```
ldapclient -a
serviceSearchDescriptor=auto_direct:automountMapName=auto.direct,cn=location,cn=automount,dc
=example,dc=com?one
```

18.5.2. 間接マップの設定

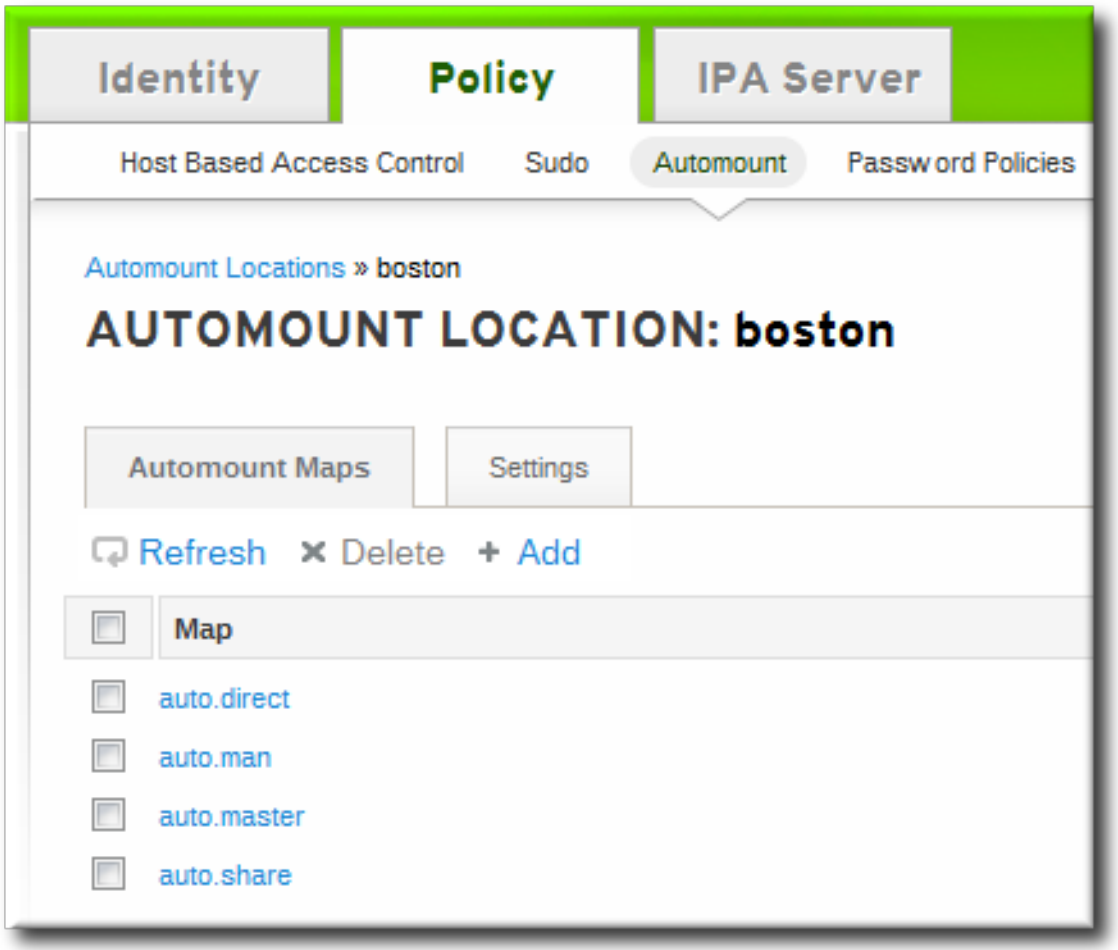
間接マップは、基本的にマップの相対パスを指定するものです。親エントリーがすべての間接マップのベースディレクトリーを設定します。間接マップキーはサブディレクトリーを設定します。間接マップの場所がロードされたときに常に、キーがベースディレクトリーに追加されます。たとえば、ベースディレクトリーが **/docs** で、キーが **man** の場合は、マップは **/docs/man** になります。

18.5.2.1. Web UI での間接マップの設定

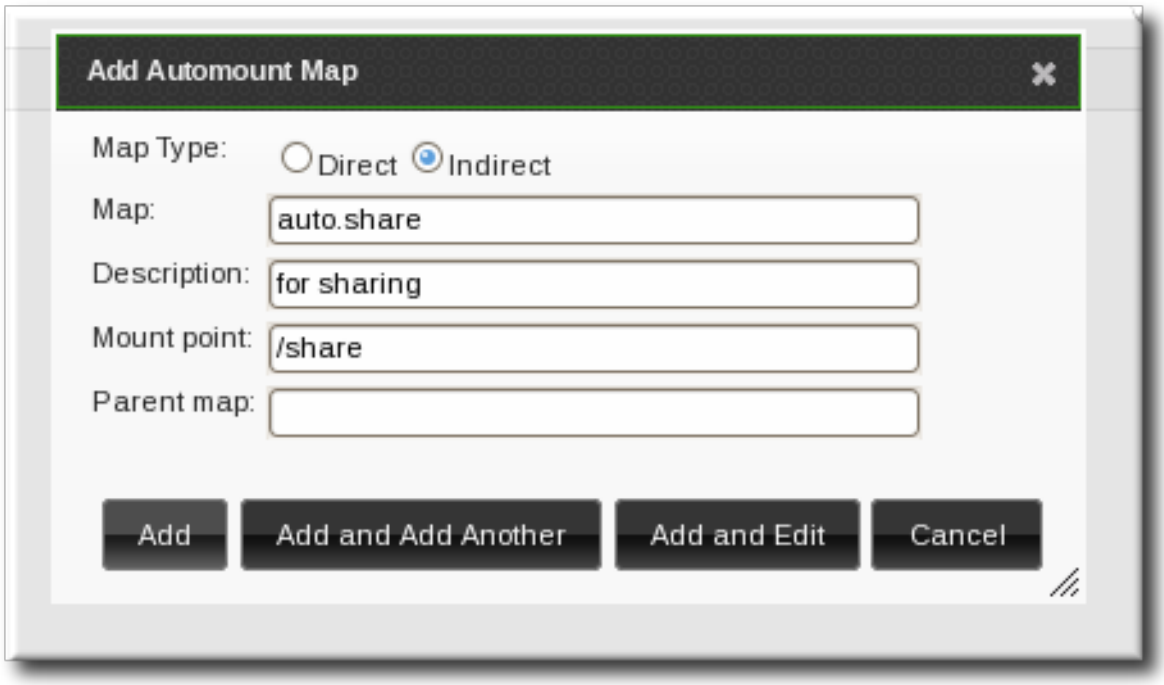
1. **Policy** タブをクリックします。
2. **Automount** サブタブをクリックします。
3. マップの追加先となる automount の場所の名前をクリックします。



4. **Automount Maps** タブで **Add** をクリックして新規マップを作成します。



5. ポップアップウィンドウで **Indirect** ラジオボタンを選択し、以下の必要な間接マップの情報を入力します。



- 新規マップの名前
- マウントポイント。**Mount** フィールドでは、すべての間接マップキーに使用するベースディレクトリーを設定します。
- オプションで親マップ。デフォルトの親は **auto.master** ですが、使用する別のマップがある場合は、**Parent Map** フィールドでそれを指定できます。

6. **Add** をクリックして新規キーを保存します。

18.5.2.2. コマンドラインでの間接マップの設定

ダイレクトマップと間接マップの主な違いは、間接キーの前にはフォワードスラッシュがないことです。

```
---------------------------
/etc/auto.share:
man     ipa.example.com:/docs/man
---------------------------
```

1. **automountmap-add-indirect** コマンドを使用して、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]
   ```

   たとえば、以下のようになります。

   ```
   $ 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、その場所ファイル一覧を確認します。

```
$ 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
```

18.5.3. 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]
```

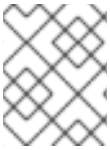
必要となる情報は、IdM 自動マウントの場所とマップファイルの完全パスおよびファイル名のみです。この **--continuous** オプションでは、**automountlocation-import** コマンドに対して、エラーが発生した場合でも、マップファイルを継続するように指示します。

たとえば、以下のようになります。

```
$ ipa automountlocation-import raleigh /etc/custom.map
```

第19章 ポリシー: パスワードポリシーの定義

すべてのユーザーには、Kerberos ドメインへの認証に使用するパスワードが必要です。Identity Management は、セキュリティーを維持するために、パスワードの複雑さ、パスワード履歴、およびアカウントのロックアウトに関するルールを定義し、実施します。



注記

デフォルトでは、IdM は、システムセキュリティーのためにハッシュ化されたパスワードであっても、クライアントにパスワードを公開しません。

19.1. パスワードポリシーとポリシー属性

パスワードポリシー は、パスワードの複雑性やパスワード変更のルールなど、パスワードの特定標準を設定します。パスワードポリシーは、ブルートフォース攻撃を阻止するための適切で複雑な標準を確実に満たし、パスワードを発見または検出するリスクを軽減するのに十分な頻繁でパスワードを変更することで、パスワードの使用に関するリスクを最小限に抑えます。

パスワードポリシーには、主に 3 つの設定エリアがあります。

- 強度または複雑さの要件
- 履歴
- アカウントのロックアウト

IdM パスワードポリシーは、KDC と LDAP サーバーが共同で実施します。パスワードポリシーは LDAP ディレクトリーで設定され、389 Directory Server のパスワードポリシー属性を基にしていますが、ポリシーは最終的に KDC パスワードポリシーフレームワークによって制限されます。KDC ポリシーは 389 Directory Server ポリシーフレームワークよりも柔軟性が低いため、IdM パスワードポリシーは KDC でサポートされるパスワードポリシー要素のみを実装することができます。389 Directory Server 内で行われたその他のポリシー設定は、Identitity Management で表示されたり、強制されたりしません。

パスワードポリシーは、個々のユーザーではなく、IdM のグローバルまたはグループに割り当てられます。パスワードポリシーには優先順位が割り当てられるため、ユーザーが異なるパスワードポリシーを持つ複数のグループに所属すると、優先度が高いポリシーが優先されます。

設定できるさまざまなポリシー属性は、表19.1「パスワードポリシー設定」に一覧表示されています。

表19.1 パスワードポリシー設定

設定プロパティー	コマンドラインオプション	説明
UI と CLI の両方のオプション		
パスワードの最小ライフタイム	--minlife	ユーザーがユーザーのパスワードを変更できる前に、ユーザーのパスワードが有効でなければならない最低限の時間を時で設定します。これにより、ユーザーがパスワードを変更できず、即座に元の値に変更される可能性があります。デフォルト値は 1 時間です。

設定プロパティー	コマンドラインオプション	説明
パスワードの最大有効期間	--maxlife	ユーザーのパスワードの変更前に有効になる最大期間を日数単位で設定します。デフォルト値は 90 日です。
文字クラスの最小数	--minclasses	有効とみなされる前にパスワードに存在する必要がある異なるクラス、タイプ、文字の最小値を設定します。たとえば、この値を 3 に設定すると、承認には、パスワードに最低 3 つのカテゴリーからの文字が必要となります。デフォルト値はゼロ (0) で、必要なクラスがないことを意味します。 6 つの文字クラスがあります。 ● 大文字 ● 小文字 ● 数字 ● 特殊文字 (例: 区切り) ● 8 ビット文字 (10 進数コードが 128 以下で始まる文字) ● 繰り返す文字数 この重みが反対の方向にあるため、繰り返した文字が多すぎると、krbPwdMinDiffChars で表現される「レベル」を満たすためにクォーラムに合致します。
パスワードの最小長	--minlength	パスワードの最小文字数を設定します。デフォルト値は 8 文字です。

設定プロパティー	コマンドラインオプション	説明
パスワード履歴	--history	保存する以前のパスワードの数と、ユーザーが使用できないパスワード数を設定します。たとえば、これが 10 に設定されている場合は、IdM により、ユーザーは以前の 10 つのパスワードを再利用できなくなります。デフォルト値はゼロ (0) で、パスワード履歴を無効にします。  **注記** パスワード履歴がゼロに設定された場合でも、ユーザーは **現在** のパスワードを再利用できます。
CLI のみのオプション		
優先度	--priority	有効なポリシーを決定する優先度を設定します。数値が小さいほど優先度が高くなります。 この優先順位は、UI でポリシーを最初に作成する際に必要ですが、UI でリセットすることはできません。これは CLI を使用してのみリセットできます。
連続不具合の最大数	--maxfail	ユーザーのアカウントがロックされる前に、正しいパスワードを入力する最大失敗数を指定します。
失敗間隔	--failinterval	障害数がリセットされる期間 (秒単位) を指定します。
ロックアウト時間	--lockouttime	ロックアウトの実施期間 (秒単位) を指定します。

19.2. パスワードポリシーの表示

IdM には、複数のパスワードポリシーを設定できます。サーバーの作成時に設定されるグローバルポリシーは常にあります。IdM のグループに追加ポリシーを作成できます。

UI は、**Password Policies** ページのすべてのグループパスワードポリシーとグローバルポリシーを一覧表示します。

CLI を使用して、グローバルおよびグループレベルのパスワードポリシーの両方を **pwpolicy-show** コマンドで表示できます。CLI は、ユーザーに有効なパスワードポリシーを表示することもできます。

19.2.1. グローバルパスワードポリシーの表示

グローバルパスワードポリシーは、初期の IdM サーバー設定の一部として作成されます。このポリシーは、グループレベルのパスワードポリシーが優先されるまですべてのユーザーに適用されます。

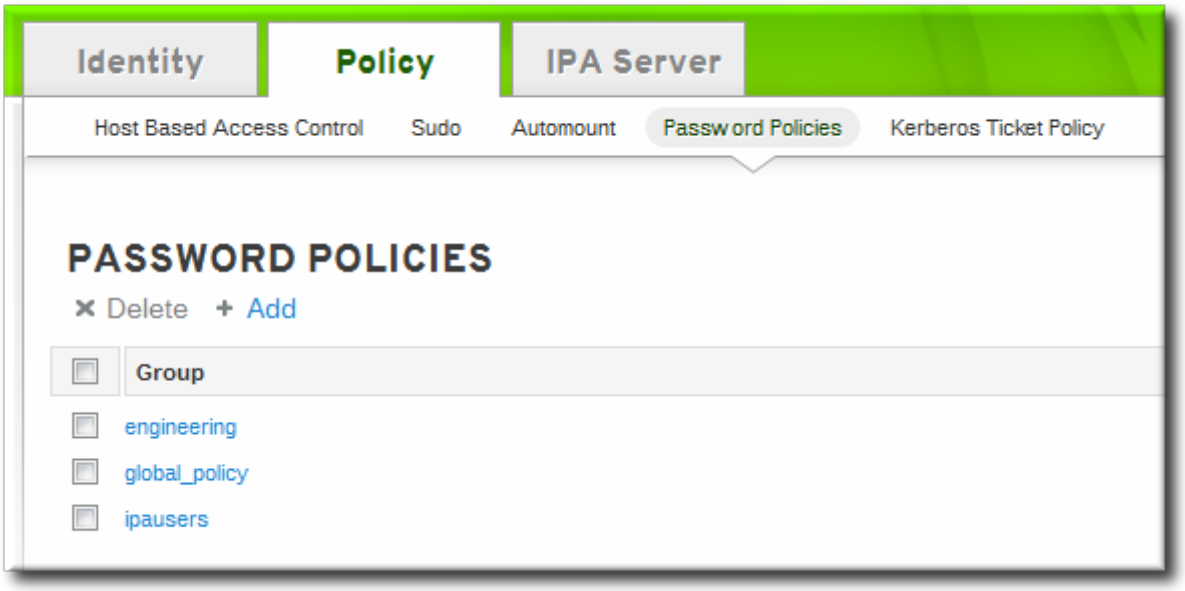
グローバルパスワードポリシーのデフォルト設定が 表19.2「デフォルトのグローバルパスワードポリシー」に一覧表示されています。

表19.2 デフォルトのグローバルパスワードポリシー

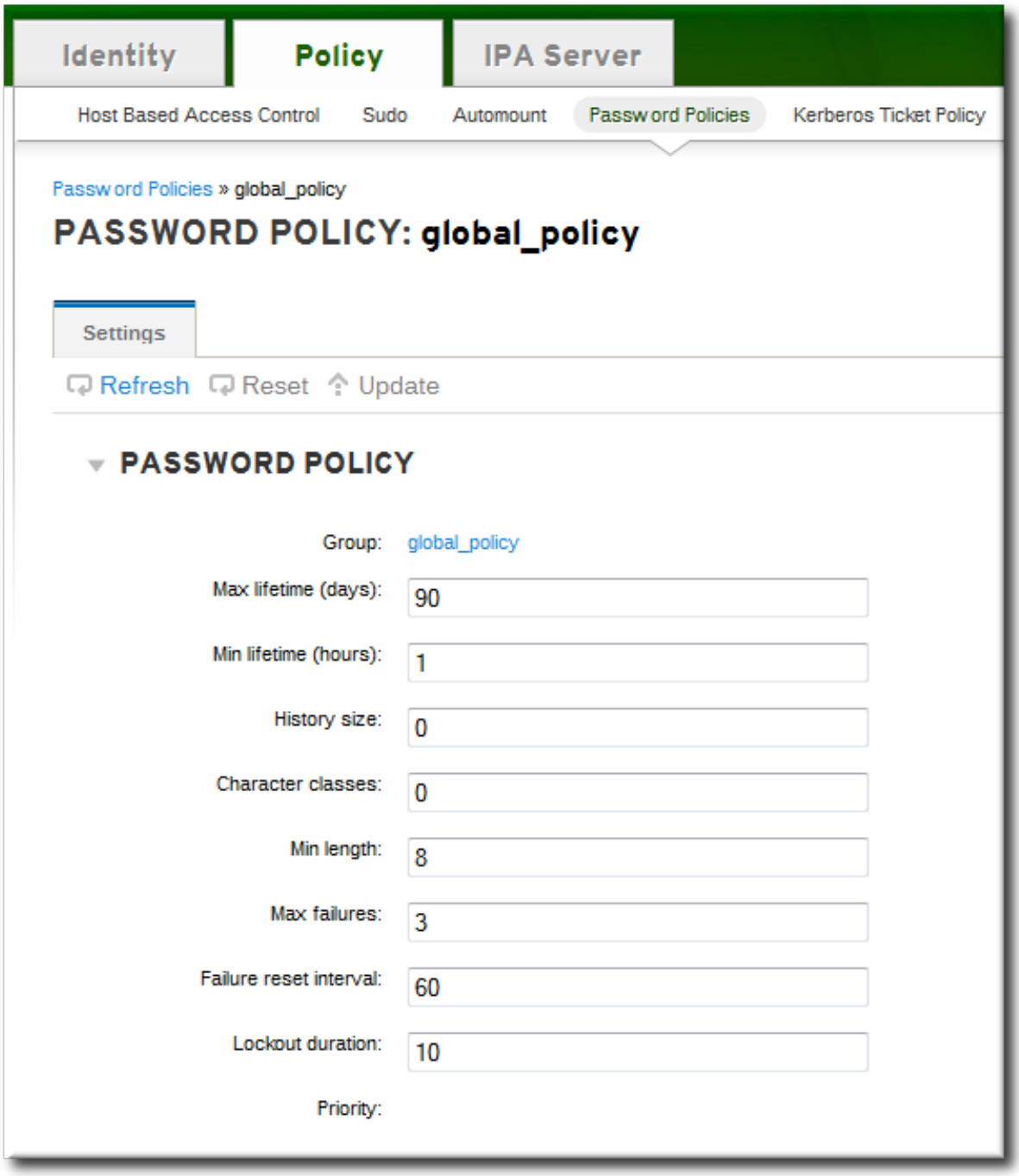
属性	値
Max lifetime	90 (日)
Min lifetime	1 (時間)
History size	0 (未設定)
Character クラス	0 (未設定)
Min length	8
Max failures	6
Failure reset interval	60
Lockout duration	600

19.2.1.1. Web UI の使用

1. **Policy** タブをクリックし、**Password Policies** サブタブをクリックします。
2. UI のすべてのポリシーがグループごとに一覧表示されます。グローバルパスワードポリシーは、**global_policy** グループによって定義されます。グループリンクをクリックします。



3. グローバルポリシーが表示されます。



19.2.1.2. コマンドラインの使用

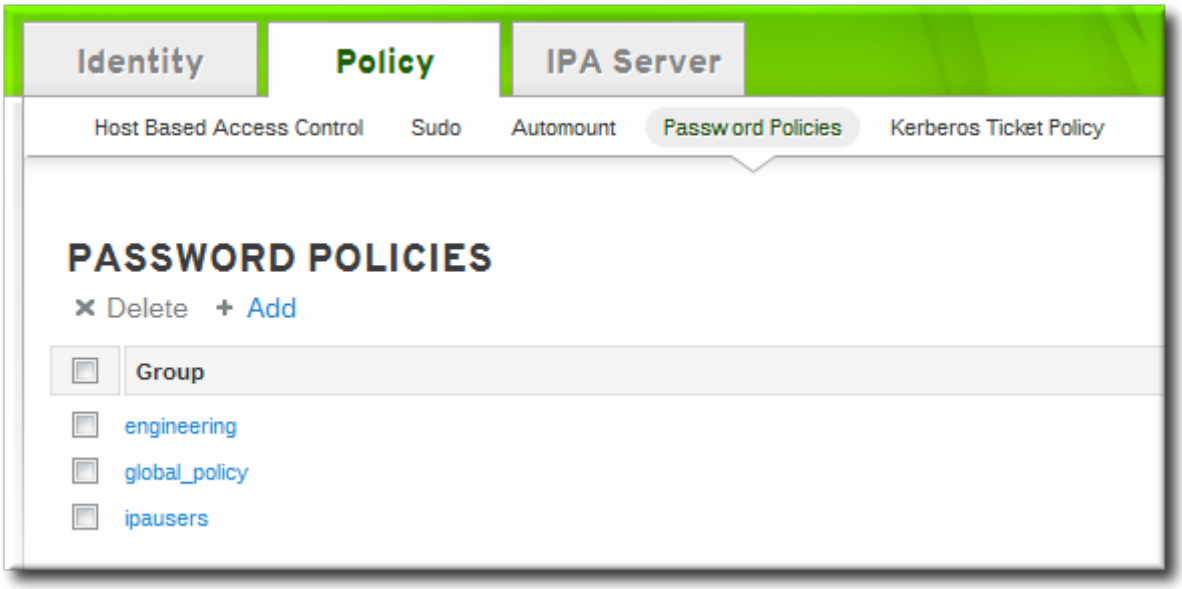
グローバルポリシーを表示するには、引数なしで **pwpolicy-show** コマンドを実行します。

```
[root@server ~]# kinit admin
[root@server ~]# ipa pwpolicy-show

  Group: global_policy
  Max lifetime (days): 90
  Min lifetime (hours): 1
  History size: 0
  Character classes: 0
  Min length: 8
  Max failures: 6
  Failure reset interval: 60
  Lockout duration: 600
```

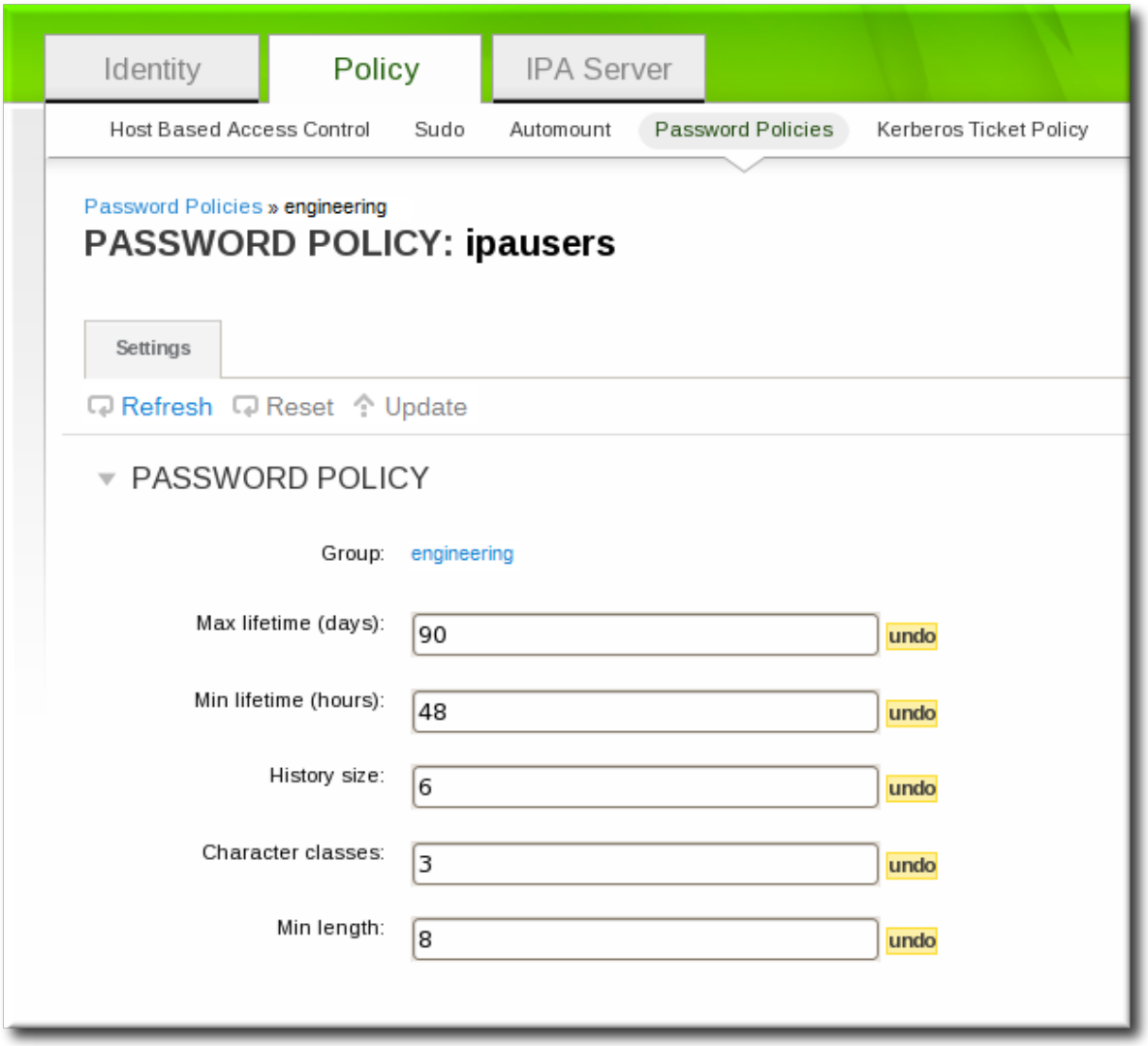
19.2.2. グループレベルのパスワードポリシーの表示

19.2.2.1. Web UI の使用

1. **Policy** タブをクリックし、**Password Policies** サブタブをクリックします。
2. UI のすべてのポリシーがグループごとに一覧表示されます。ポリシーを割り当てられたグループの名前をクリックします。



3. グループポリシーが表示されます。



19.2.2.2. コマンドラインの使用

グループレベルのパスワードポリシーの場合は、コマンドでグループ名を指定します。

```
[root@server ~]# kinit admin
[root@server ~]# ipa pwpolicy-show ipausers
Group: ipausers
Max lifetime (days): 120
Min lifetime (hours): 10
Min length: 10
Priority: 50
```

19.2.3. ユーザーの有効なパスワードポリシーの表示

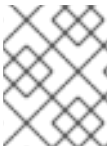
ユーザーは、それぞれが個別のパスワードポリシーを持つ複数のグループに所属することができます。これらのポリシーは追加されません。一度に有効になっているポリシーは 1 つだけで、すべてのパスワードポリシー属性に適用されます。特定のユーザーに有効になっているポリシーを確認するには、特定のユーザーに対して **pwpolicy-show** コマンドを実行できます。結果には、そのユーザーに**有効なグループポリシー**も表示されます。

```
[root@server ~]# kinit admin
[root@server ~]# ipa pwpolicy-show --user=jsmith
  Group: global_policy
```

```
Max lifetime (days): 90
Min lifetime (hours): 1
History size: 0
Character classes: 0
Min length: 8
Max failures: 6
Failure reset interval: 60
Lockout duration: 600
```

19.3. パスワードポリシーの作成および編集

パスワードポリシーは選択可能で、特定の要素のみを定義できます。**グローバル**パスワードポリシーは、グループポリシーが優先されない限り、すべてのユーザーエントリーに使用されるデフォルトを設定します。



注記

グローバルポリシーは常に存在するため、グローバルパスワードポリシーを追加する必要はありません。

グループのポリシーはグローバルポリシーを上書きし、グループメンバーにのみ適用される特定のポリシーを提供します。パスワードポリシーは累積されません。グループポリシーまたはグローバルポリシーは、ユーザーまたはグループに対して有効になりますが、両方を同時に行うことはできません。

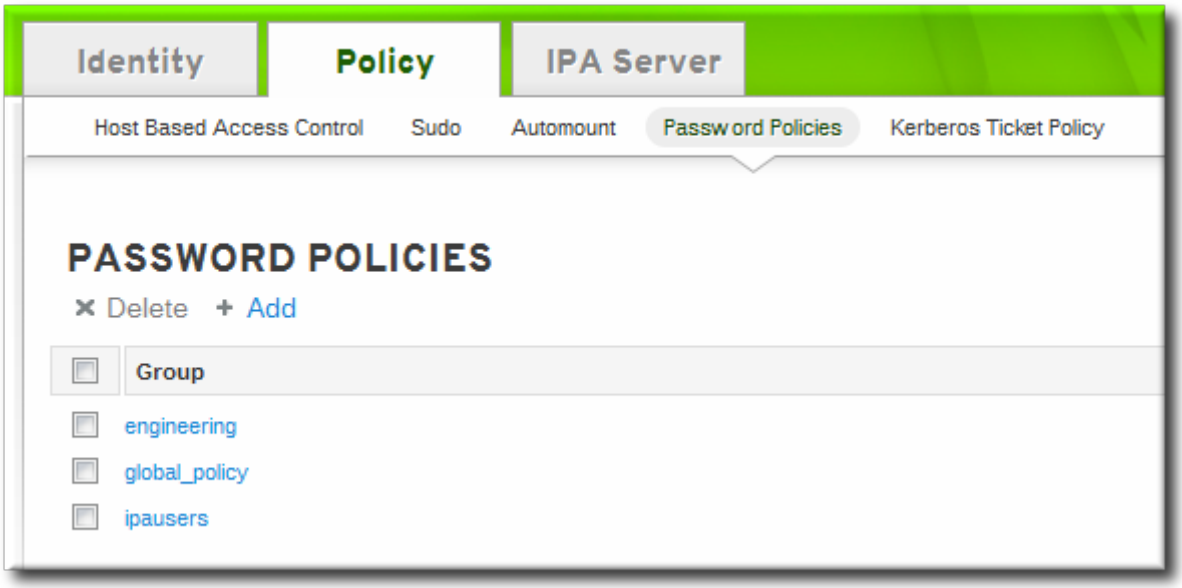
グループレベルのポリシーはデフォルトでは存在しないため、手動で作成する必要があります。

注記

存在しないグループにパスワードポリシーを設定することはできません。

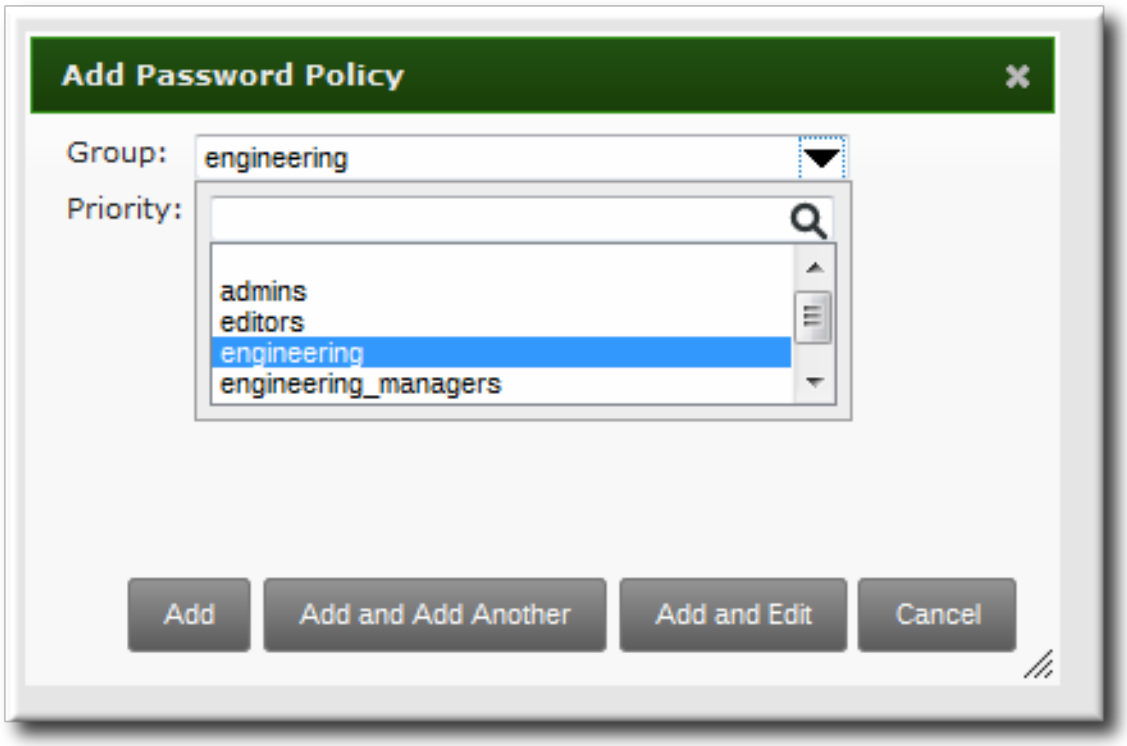
19.3.1. Web UI でのパスワードポリシーの作成

1. **Policy** タブをクリックし、**Password Policies** サブタブをクリックします。
2. UI のすべてのポリシーがグループごとに一覧表示されます。グローバルパスワードポリシーは、**global_policy** グループによって定義されます。グループリンクをクリックします。



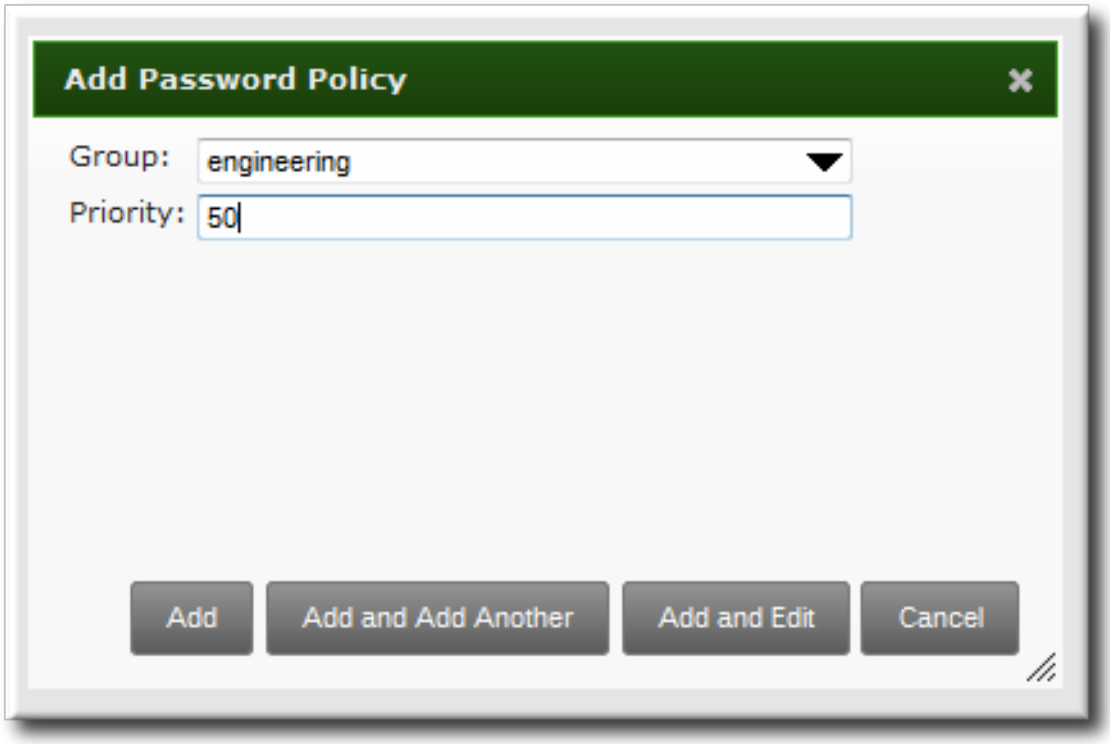
3. 上部の **Add** リンクをクリックします。

4. ポップアップボックスで、パスワードポリシーを作成するグループを選択します。



5. ポリシーの優先度を設定します。数値が大きいほど優先度が低くなります。最も優先度ポリシーの番号は最も低くなります。

   ユーザーに対して有効になっているパスワードポリシーは 1 つだけで、最も優先度が高いポリシーです。



注記

ポリシーの作成後に優先順位は UI で変更できません。

6. **Add and Edit** ボタンをクリックして、ポリシーフォームをすぐに開くようにします。

7. ポリシーフィールドを設定します。フィールドを空白のままにすると、属性がパスワードポリシー設定に追加されないことを意味します。

   - **Max lifetime** は、パスワードのリセットが必要になるまでの、パスワードの最長有効期間を日数単位で設定します。
   - **最小ライフタイム** は、パスワードの変更が許可される前に、パスワードが有効である必要のある最小時間 (時間単位) を設定します。これにより、ユーザーがパスワードをすぐに古いパスワードに戻さなくしたり、パスワード履歴をサイクリングしないようにします。
   - **履歴サイズ** は、保存する以前のパスワードの数を設定します。ユーザーは、パスワード履歴にあるパスワードを再使用することはできません。
   - **文字クラス** は、パスワードで使用する必要があるさまざまなカテゴリーの文字 **数** を設定します。これは、使用する必要があるクラスを設定しません。パスワードで使用する必要がある異なる (指定されていない) クラスの数を設定します。たとえば、文字クラスには数字、特殊文字、大文字を使用できます。カテゴリーの完全なリストは 表19.1「パスワードポリシー設定」にあります。これは、複雑な要件の設定の一部です。
   - **最小長** は、パスワードで必要な文字数を設定します。これは、複雑な要件の設定の一部です。

19.3.2. コマンドラインでのパスワードポリシーの作成

pwpolicy-add コマンドを使用すると、パスワードポリシーが追加されます。

```
[root@server ~]# kinit admin
[root@server ~]# ipa pwpolicy-add groupName --attribute-value
```

たとえば、以下のようになります。

```
[root@server ~]# kinit admin
[root@server ~]# ipa pwpolicy-add exampleGroup --minlife=7 --maxlife=49 --history= --priority=1
Group: exampleGroup
Max lifetime (days): 49
Min lifetime (hours): 7
Priority: 1
```

ヒント

属性を空の値に設定すると、その属性がパスワードポリシーから効果的に削除されます。

19.3.3. コマンドラインでパスワードポリシーの編集

多くの IdM エントリーと同様に、パスワードポリシーは ***-mod** コマンド、**pwpolicy-mod**、そしてポリシー名を使用して編集されます。ただし、パスワードポリシーの編集には 1 つの違いがあります。常に存在するグローバルポリシーがあります。グループレベルのパスワードポリシーの編集は、グローバル

パスワードポリシーの編集と若干異なります。

グループレベルのパスワードポリシーの編集は、***-mod** コマンドの標準的な構文に従います。これは、**pwpolicy-mod** コマンド、ポリシーエントリーの名前、および変更する属性を使用します。たとえば、以下のようになります。

```
[jsmith@ipaserver ~]$ ipa pwpolicy-mod exampleGroup --lockouttime=300 --history=5 --minlength=8
```

グローバルパスワードポリシーを編集するには、**パスワードポリシー名** を指定せずに、変更する属性とともに **pwpolicy-mod** コマンドを使用します。たとえば、以下のようになります。

```
[jsmith@ipaserver ~]$ ipa pwpolicy-mod --lockouttime=300 --history=5 --minlength=8
```

19.4. パスワード有効期限の制限の管理

パスワードポリシーは**パスワードが変更されたとき**に適用されます。したがって、パスワードを設定すると、その時点で有効なパスワードポリシーに準拠します。パスワードポリシーが後で変更されると、その変更はパスワードに適用されず、パスワードに遡って適用されません。

パスワードの有効期限の設定は、グループパスワードポリシーの一部として設定されます。パスワードポリシーの作成および編集 (ポリシーの expiration 属性を含む) は、「パスワードポリシーの作成および編集」で説明されます。

パスワードの有効期限では、関連する属性が 2 つあります。

- パスワードポリシー (**--maxlife**) で指定される最大有効期間設定
- 指定のユーザーのパスワードが期限切れになる実際の日付 (***krbPasswordExpiration***)

パスワードポリシーでパスワードの有効期限を変更しても、ユーザーパスワードが変更されるまで、ユーザーの有効期限は影響を受けません。パスワードの有効期限をすぐに変更する必要がある場合は、ユーザーエントリーを編集して変更できます。

有効期限を強制的に変更するには、ユーザーの ***krbPasswordExpiration*** 属性値をリセットします。**これは、ldapmodify を使用してのみ実行できます。**単一ユーザーの場合、以下のようにします。

```
[bjensen@ipaserver ~]$ ldapmodify -D "cn=Directory Manager" -w secret -h ipaserver.example.com -
p 389 -vv

dn: uid=jsmith,cn=users,cn=accounts,dc=example,dc=com
changetype: modify
replace: krbpasswordexpiration
krbpasswordexpiration: 20140202203734Z
-
```

-f オプションで LDIF ファイルを **ldamodify** コマンドで参照して、複数のエントリーを同時に編集できます。

ヒント

管理者がパスワードをリセットすると、以前のパスワードは期限切れになり、ユーザーはパスワードを強制的に更新します。ユーザーがパスワードを更新すると、新しい有効期限を含む新しいパスワードポリシーが自動的に使用されます。

19.5. グループパスワードポリシーの優先順位の変更

ユーザーは、それぞれ異なるパスワードポリシーを持つ複数のグループに所属することができます。ユーザーに対して有効なポリシーは 1 つしかないため、ポリシーに優先順位を割り当てる方法が必要です。これは **優先順位** で行われます。

最も優先度はゼロ (0) です。数値が小さいほど優先度が高くなります。

これは、パスワードポリシーの作成時に最初に設定されます。この **--priority** オプションをリセットすると、ポリシーの作成後に変更できます。

```
[root@server ~]# kinit admin
[root@server ~]# ipa pwpolicy-mod examplegroup --priority=10
```

ユーザーが複数のグループに属する場合、最優先度の低い**数値** を持つグループパスワードポリシーが最も優先されます。

19.6. アカウントロックアウトポリシーの設定

ブルートフォース攻撃は、悪意のあるユーザーが複数のログイン試行でサーバーにアクセスすることでパスワードの推測を試みる際に発生します。アカウントのロックアウトポリシーにより、一定数ログインが失敗すると、アカウントがシステムにログインできなくなり、これによってブルートフォース攻撃を防ぎます。これは、正しいパスワードを入力してもログインできなくなります。

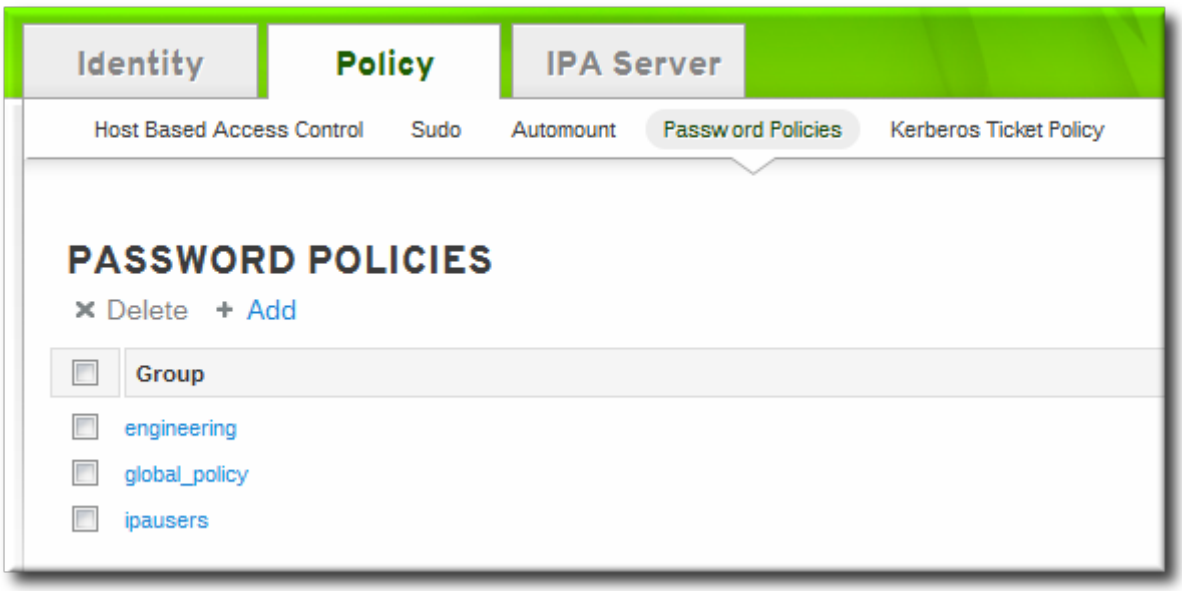
注記

ユーザーアカウントは、**ipa user-unlock** を使用して管理者が手動でロックを解除できます。「ログイン失敗後のユーザーアカウントのロック解除」を参照してください。

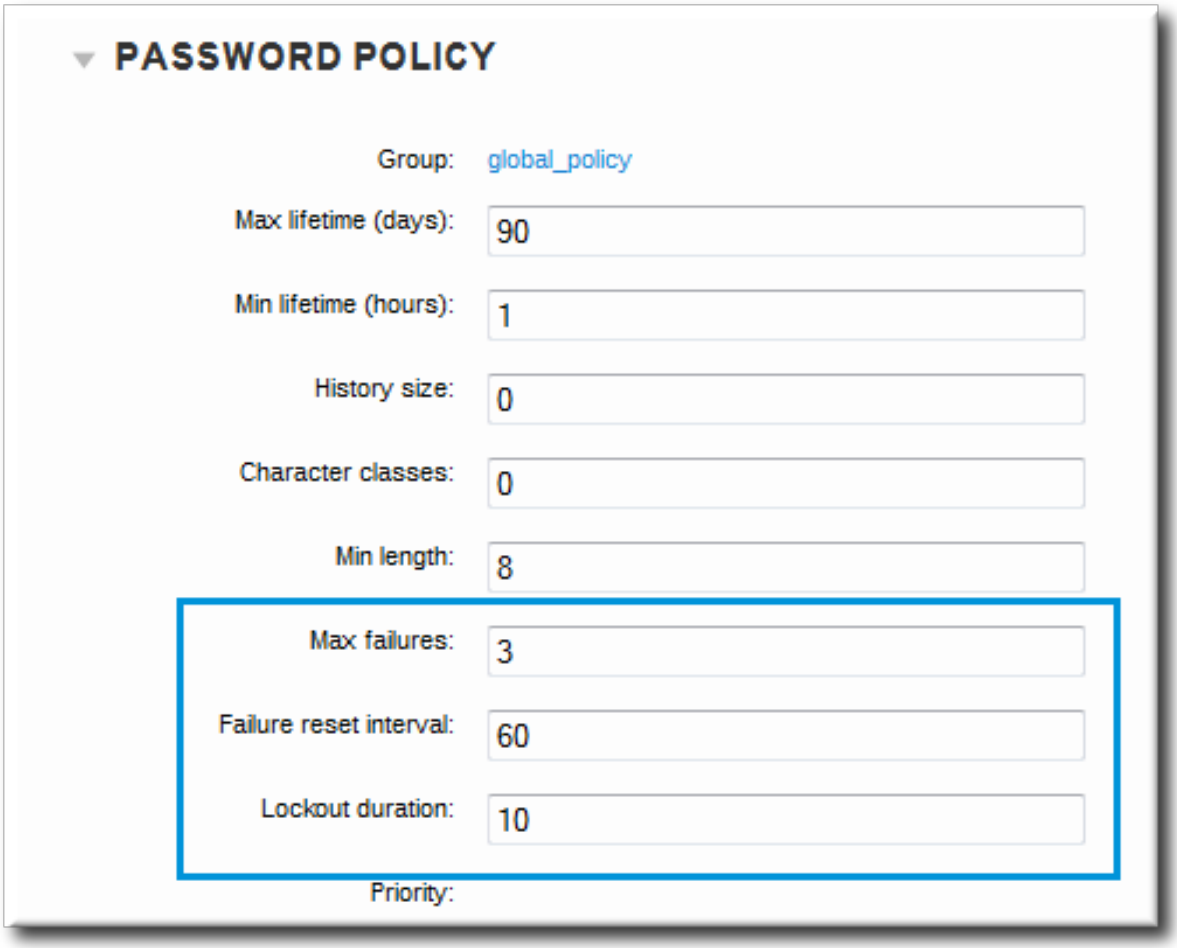
19.6.1. UI で

これらの属性は、グループレベルのパスワードポリシーが作成された場合、またはグローバルパスワードポリシーを含むパスワードポリシーが編集されると、パスワードポリシーフォームで利用できます。

1. **Policy** タブをクリックし、**Password Policies** サブタブをクリックします。
2. 編集するポリシーの名前をクリックします。



3. アカウントロックアウト属性の値を設定します。



アカウントロックアウトポリシーには、以下の 3 つの部分があります。

- アカウントがロックされるまでの失敗したログイン試行回数 (**最大失敗**)。
- カウンターをリセットする前にログインに失敗した時間 (**Failure reset interval**)。ミスが生じるため、失敗した試行の数は永久に保持されず、一定時間が経過すると、警告に経過します。これは、特定の時間が経過したときに自然に起こります。これは秒単位です。
- 最大失敗回数 (**Lockout duration**) に達した後にアカウントがロックされる時間。これは秒単位です。

19.6.2. コマンドラインでの設定

アカウントロックアウトポリシーには、以下の 3 つの部分があります。

- アカウントがロックされるまでの失敗したログイン試行回数 (**--maxfail**)。
- 最大失敗数に達した後にアカウントがロックされる時間 (**--lockouttime**)。これは秒単位です。
- カウンターをリセットするまでのログイン試行に失敗する時間 (**--failinterval**)。ミスが生じるため、失敗した試行の数は永久に保持されず、一定時間が経過すると、警告に経過します。これは、特定の時間が経過したときに自然に起こります。これは秒単位です。

これらのアカウントのロックアウト属性はすべて、パスワードポリシーが後に使用して追加されるか、**pwpolicy-add** で作成されるか、**pwpolicy-mod** を使用して後で追加されるときに設定できます。たとえば、以下のようになります。

```
[jsmith@ipaserver ~]$ kinit admin
[jsmith@ipaserver ~]$ ipa pwpolicy-mod examplegroup --maxfail=4 --lockouttime=600 --
failinterval=30
```

19.7. パスワード変更ダイアログの有効化

Identity Management にユーザーが存在する場合はありますが、有効な Kerberos チケットがない場合もあります。つまり、IdM ドメインに対して認証できません。これは、新規ユーザーまたはドメインパスワードの有効期限が切れたユーザーの場合に可能です。Web UI でパスワード認証を有効にするのと同様に、クライアントへのパスワードベースの認証を有効にすることもできます。これにより、パスワード変更ダイアログボックスが表示され、期限切れのパスワードをリセットできるようになります。

パスワード変更ダイアログは、OpenSSH の **challenge-response** 認証を使用して有効にします。

challenge-response ダイアログは任意です。多くの環境では、必要な PAM モジュールを呼び出すことで、SSSD が期限切れのパスワードを処理できるため、必須ではありません。ただし、OpenSSH で challenge-response オプションを使用すると、直接 PAM でパスワードの変更を行い、完全な PAM 対話に対応することができます。

これはデフォルトでは有効になっていませんが、OpenSSH 設定を編集して有効にできます。

1. **/etc/ssh/sshd_config** ファイルを開きます。
2. ***ChallengeResponseAuthentication*** を **yes** に設定します。

第20章 ポリシー: KERBEROS ドメインの管理

Kerberos 認証は、IdM ドメイン内の認証の中核です。IdM サーバーは実際には、内部で Kerberos サーバーを実行します。この Kerberos サーバーは、チケットおよびキータブを管理するカスタムポリシー用に設定できます。

Kerberos の概念に関する詳細情報は、http://web.mit.edu/kerberos/www/ を参照してください。

重要

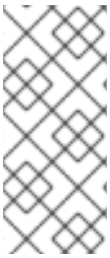
Identity Management には、Kerberos ポリシーの管理に使用する独自のコマンドラインツールがあります。IdM Kerberos 設定の管理には、**kadmin** または **kadmin.local** は **使わないで** ください。

20.1. KERBEROS について

Kerberos は、サービスとユーザーの間で認証層を提供します。Kerberos は認証を 1 つの場所に集中化します。ユーザーが Kerberos サーバーに対して認証を行い、そのユーザーがネットワーク上のリソースにアクセスしようとすると、そのリソースは保存されたユーザー認証情報の **キー配布センター** (KDC) を確認できます。これにより、ユーザーは認証情報を個別に指定しなくても、複数のリソースにアクセスできます。

相互を認識しているユーザーとサービス、組み合わせたすべての KDC および Kerberos サーバーが **レルム** を構成します。レルム内の各ユーザー、マシン、およびサービスは、**プリンシパル** と呼ばれる一意の名前で識別されます。ユーザーまたはサービスはプリンシパルと検証認証情報 (通常はパスワード) を使用して KDC に対する認証を行います。KDC と共有される認証情報は**キー**であり、**キーテーブル**または**キータブ**と呼ばれるファイルに保存されます。

KDC がユーザーのアイデンティティーを検証すると、**チケット** が発行されます。チケットは、レルムのサービスおよびマシンへの長期パスです。KDC は、TGT (**Ticket-granting Ticket**)と呼ばれる特殊な種類のチケットを発行します。ユーザーが Kerberos レルム内のリソースにアクセスしようとすると、リソースはチケットに対して要求を送信します。TGT は、リソースがユーザーの認証およびアクセス許可すのためのリソース固有のチケット発行に使用されます。



注記

IdM クライアントを最初に設定すると、ホストプリンシパルがセットアップスクリプトによって自動的に取得され、**/etc/krb5.keytab** ファイルに保存されます。このホストプリンシパルはホストレコードに格納されるため、ローカルサービスコマンドをこのプリンシパルと使用できません。これにより、IdM レルムで機能するクライアントが準備されます。

20.1.1. プリンシパル名

プリンシパルはユーザーやサービスだけでなく、そのエンティティーが属するレルムも特定します。プリンシパル名は、識別子とレルムの 2 つからなります。

```
identifier@REALM
```

ユーザーの場合、**識別子** は Kerberos ユーザー名のみになります。サービスの場合、**識別子** はサービス名と、それが実行するマシンのホスト名の組み合わせです。

```
service/FQDN@REALM
```

サービス名は、host、ldap、http、dns など、サービスタイプに固有の大文字と小文字を区別する文字列です。すべてのサービスに明らかなプリンシパル識別子があるわけではありません。たとえば、**sshd** デーモンはホストサービスプリンシパルを使用します。

ホストプリンシパルは通常、**/etc/krb5.keytab** に保存されます。

Kerberos がチケットを要求する際は常に、ドメイン名のエイリアス (DNS CNAME レコード) を対応する DNS アドレス (A または AAAA レコード) に解決します。アドレスレコードからのホスト名は、サービスまたはホストプリンシパルが作成される際に使用されます。

以下に例を示します。

```
www.example.com  CNAME  web-01.example.com
web-01.example.com  A  192.0.2.145
```

サービスは、ホストの CNAME エイリアスを使ってホストに接続を試みます。

```
$ ssh www.example.com
```

Kerberos サーバーは解決されたホスト名 **web-01.example.com@EXAMPLE.COM** のチケットを要求するため、ホストプリンシパルは **host/web-01.example.com@EXAMPLE.COM** である必要があります。

20.1.2. キータブの保護について

キータブファイルを保護するには、パーミッションと所有権をリセットして、ファイルへのアクセスをキータブ所有者のみに制限します。たとえば、Apache キータブ (**/etc/httpd/conf/ipa.keytab**) の所有者を **apache** に設定し、モードを **0600** に設定します。

20.2. KERBEROS チケットポリシーの設定

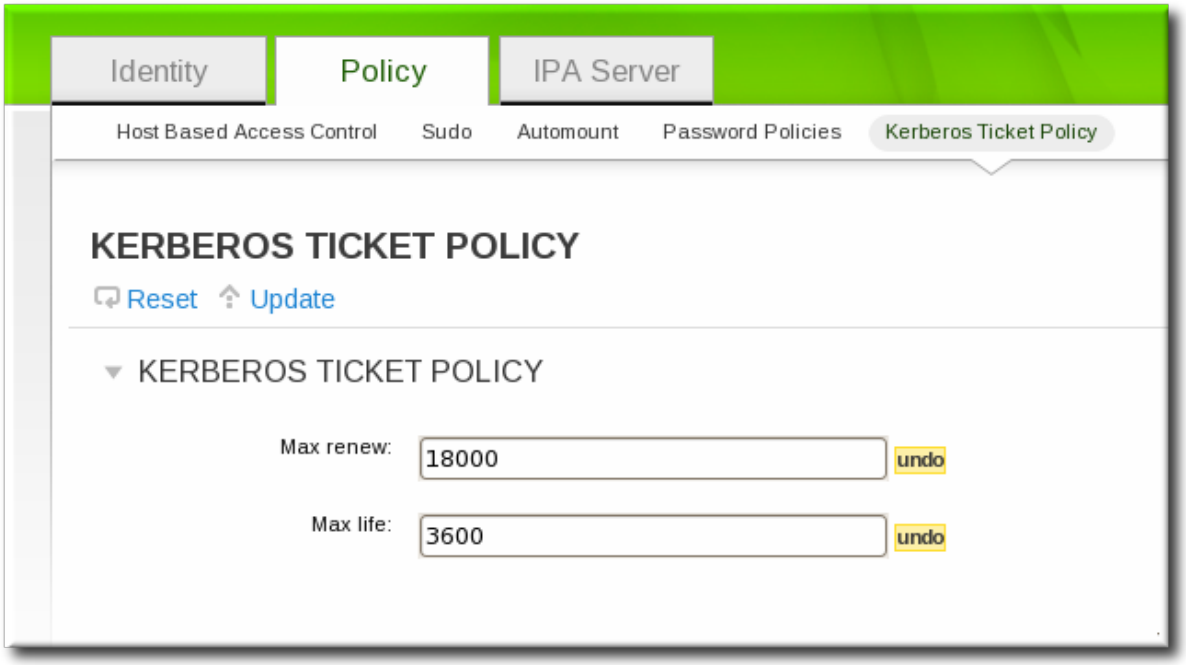
Kerberos **チケットポリシー** は、チケットの最大有効期間や更新期間 (チケットを更新することのできる期間) など、Kerberos レルム内のチケット管理に基本的な制限を設定します。

Kerberos チケットポリシーはグローバルに設定され、レルム内で発行されたすべてのチケットに適用されます。IdM には、グローバルポリシーを上書きするユーザーレベルのチケットポリシーを設定する機能もあります。これは、たとえば、管理者の有効期限を設定したり、社員の有効期限を短くするために使用できます。

20.2.1. グローバルチケットポリシーの設定

20.2.1.1. Web UI での操作

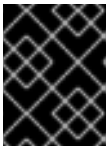
1. **Policy** タブをクリックし、**Kerberos Ticket Policy** サブタブをクリックします。
2. チケットライフタイムポリシーを変更します。



- **最大更新** は、チケットの期限が切れた後に更新できる期間を設定します。
- **Maximum life** は、Kerberos チケットのアクティブな期間 (ライフサイクル) を設定します。

3. ポリシーページの上部にある **Update** リンクをクリックします。
4. KDC を再起動します。

```
# service krb5kdc restart
```



重要

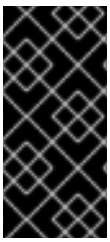
グローバル Kerberos チケットポリシーを変更するには、KDC を再起動して変更を反映する必要があります。

20.2.1.2. コマンドラインでの操作

この **ipa krbtpolicy-mod** コマンドはポリシーを変更します。一方、**ipa krbtpolicy-reset** コマンドでは、ポリシーがデフォルト値にリセットされます。

たとえば、以下のようになります。

```
# ipa krbtpolicy-mod --maxlife=3600 --maxrenew=18000
  Max life: 3600
  Max renew: 18000
```



重要

グローバル Kerberos チケットポリシーを変更するには、KDC を再起動して変更を反映する必要があります。KDC を再起動します。

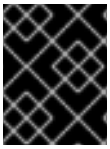
```
# service krb5kdc restart
```

20.2.2. ユーザーレベルのチケットポリシーの設定

ユーザーレベルの Kerberos チケットポリシーは、グローバルポリシーと同じコマンドを使用して設定されますが、ユーザーはコマンドで指定されます。

以下に例を示します。

```
# ipa krbtpolicy-mod jsmith --maxlife=3600
  Max life: 3600
```



重要

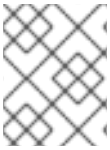
ユーザーレベルのポリシーは、KDC サービスを再起動しなくても、次に要求されたチケット (実行中など **kinit**) で即座に適用されます。

20.3. KERBEROS チケットの更新

Kerberos キーはパスワードに類似しています。パスワードポリシーと同様に、Kerberos チケットはセキュリティーポリシーの対象となります。このポリシーでは、指定した間隔後に手動で更新する必要があります。

キーのバージョンは、**キーバージョン番号** (KVNO)に表示されます。更新 (ローテーションとも呼ばれる) が、キータブエントリーの KVNO をインクリメントします。キーを更新すると、新しいエントリーが KVNO のキータブに追加されます。元の鍵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。

IdM レルムの各キータブには、IdM LDAP サーバーにエントリーがあり、これには最後の変更時間が含まれます。更新が必要なプリンシパルは、**ipa-getkeytab** コマンドを使用して再生成できます。



注記

この **ipa-getkeytab** コマンドは、ファイルにすでに存在する場合に古いキータブを削除しません。

1. 必須の日付の前に発行されたすべてのキータブを見つけます。たとえば、2010 年 1 月 1 日から、2010 年 12 月 31 日の午後 11:59 の間に作成されたプリンシパルを探します。

```
# ldapsearch -x -b "cn=computers,cn=accounts,dc=example,dc=com" "(&
(krblastpwdchange>=20100101000000)(krblastpwdchange<=20101231235959))" dn
krbprincipalname

# ldapsearch -x -b "cn=services,cn=accounts,dc=example,dc=com" "(&
(krblastpwdchange>=20100101000000)(krblastpwdchange<=20101231235959))" dn
krbprincipalname
```

   - ホスト (マシン) 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。
   - 最終変更日 (***krblastpwdchange***) で絞り込みます。

- ***dn krbprincipalname*** 属性を指定して、検索結果の情報をエントリー名とプリンシパルにのみ制限します。

日付は、YYYYMMDD 形式で表現され、時刻は HHMMSS 形式 (GMT) で表されます。

2. **ipa-getkeytab** コマンドを使用して、プリンシパルの新しいキータブを取得します。これには、サービスまたはホストの元のキータブの場所 (**-k**)、プリンシパル (**-p**)、および IdM サーバーのホスト名 (**-s**) が必要です。

   たとえば、これにより、以下のように **/etc/krb5.keytab** のデフォルトのキータブでホストプリンシパルが更新されます。

   ```
   # ipa-getkeytab -p host/client.example.com@EXAMPLE.COM -s ipa.example.com -k
   /etc/krb5.keytab
   ```

   これにより、Apache サービスのキータブが、**/etc/httpd/conf/ipa.keytab** のデフォルトの場所にあるキータブで更新されます。

   ```
   # ipa-getkeytab -p HTTP/client.example.com@EXAMPLE.COM -s ipa.example.com -k
   /etc/httpd/conf/ipa.keytab
   ```

3. すべてのサービスに使用するキータブ **ipa-getkeytab** を再生成します。

この **klist** コマンドは、更新されたキータブの新しいキーバージョン番号を表示します。元のキータブはデータベースに存在し、以前の KVNO で一覧表示されます。

```
# klist -kt /etc/krb5.keytab
Keytab: WRFILE:/etc/krb5.keytab
KVNO Timestamp         Principal
---- ----------------- --------------------------------------------------------
   1 06/09/10 11:23:01 host/client.example.com@EXAMPLE.COM(aes256-cts-hmac-sha1-96)
   2 06/09/11 05:58:47 host/client.example.com@EXAMPLE.COM(aes256-cts-hmac-sha1-96)
   1 03/09/11 13:57:16 krbtgt/EXAMPLE.COM@EXAMPLE.COM(aes256-cts-hmac-sha1-96)
   1 03/09/11 13:57:16 HTTP/ipa.example.com@EXAMPLE.COM(aes256-cts-hmac-sha1-96)
   1 03/09/11 13:57:16 ldap/ipa.example.com@EXAMPLE.COM(aes256-cts-hmac-sha1-96)
```

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。

重要

NFSv4 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。

20.4. KERBEROS パスワードのキャッシュ

マシンは常に IdM ドメインと同じネットワークに存在するとは限りません。たとえば、マシンが IdM ドメインにアクセスする前に VPN にログインしなければならない場合があります。ユーザーがオフライン時にシステムにログインし、後で IdM サービスへの接続を試みると、そのユーザーの IdM Kerberos チケットがないため、ブロックされます。IdM は、SSSD を使用して SSSD キャッシュに Kerberos パスワードを保存することで、この制限を回避します。

これは、**ipa-client-install** スクリプトによりデフォルトで設定されます。設定パラメーターがファイルに追加されます。この **/etc/sssd/sssd.conf** ファイルは、IdM ドメインの Kerberos パスワードを保存するように SSSD に指示します。

```
[domain/example.com]
cache_credentials = True
ipa_domain = example.com
id_provider = ipa
auth_provider = ipa
access_provider = ipa
chpass_provider = ipa
ipa_server = _srv_, server.example.com
krb5_store_password_if_offline = true
```

このデフォルトの動作は、**--no-krb5-offline-passwords** オプションを使用してクライアントのインストール時に無効にできます。

この動作は、**/etc/sssd/sssd.conf** ファイルを編集し、**krb5_store_password_if_offline** 行を削除したり、その値を false に変更したりして無効にすることもできます。

```
[domain/example.com]
...
krb5_store_password_if_offline = false
```

Kerberos 認証の SSSD 設定オプションは Red Hat Enterprise Linux デプロイメントガイドの SSSD の設定セクションを参照してください。

20.5. キータブの削除

Kerberos チケットを更新すると、新しいキーがキータブに追加されますが、キータブはクリアされません。ホストが登録解除されて IdM ドメインに再追加されている場合、または Kerberos 接続エラーがある場合は、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る場合があります。

これは、**ipa-rmkeytab** コマンドを使用して行います。ホストのすべてのプリンシパルを削除するには、**-r** オプションでレルムを指定します。

```
# ipa-rmkeytab -r EXAMPLE.COM -k /etc/krb5.keytab
```

特定のサービスのキータブを削除するには、**-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 -p ldap/client.example.com -k /etc/krb5.keytab
```

第21章 ポリシー: 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。**sudo** ポリシーは、ドメインユーザーおよびドメインホストに適用されます。

21.1. SUDO および IPA について

この **sudo** ユーティリティーを使用すると、システム管理者は特定のユーザーに権限を委譲して、特定のコマンドを root または別の指定されたユーザーとして実行できます。このユーティリティーは、コマンドとその引数の監査証跡を提供するため、アクセスを追跡できます。

21.1.1. Identity Management の全般的な sudo 設定

この **sudo** ユーティリティーは、ローカル設定ファイルを **/etc/sudoers** 使用します。このファイルは、コマンドや **sudo** アクセスのあるユーザーを定義します。このファイルはマシン間で共有できますが、マシン間で **sudo** 設定ファイルを分散するネイティブの方法はありません。

Identity Management は一元管理された LDAP データベースを使用して **sudo** 設定が含まれるため、すべてのドメインホストでグローバルに利用できるようになります。また、Identity Management には **sudo** エントリー用の特別な LDAP スキーマがあり、より柔軟でシンプルな設定が可能です。このスキーマは、2 つの主要な機能を追加します。

- Identity Management スキーマは、**sudo** netgroups に加え、ホストグループに対応します。一方、**sudo** は netgroups のみに対応します。

  Identity Management は、すべてのホストグループに対応するシャドウ netgroup も作成します。これにより、IdM 管理者はホストグループを参照する **sudo** ルールを作成できますが、ローカルの **sudo** コマンドは対応する netgroup を使用します。

- Identity Management では、**sudo コマンドグループ** の概念が導入されました。グループには複数のコマンドが含まれ、コマンドグループは **sudo** 設定で参照できます。

sudo はホストグループおよび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 Identity Management は IdM **sudo** 設定を **sudo** 設定に変換します。

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、Identity Management は、LDAP/**sudo** 設定ファイル **/etc/sudo-ldap.conf** で設定できるデフォルトの **sudo** ユーザー **uid=sudo,cn=sysaccounts,cn=etc,$SUFFIX** を定義します。

Identity Management **sudo** と Identity Management の両方で、 **sudo** 設定の一部としてユーザーグループがサポートされます。ユーザーグループは Unix または非 POSIX グループのいずれかになります。POSIX グループ以外を作成すると、グループのユーザーはグループから POSIX 以外の権限を継承するため、アクセスの問題が発生する可能性があります。Unix グループと非 POSIX グループを選択すると、管理者はグループのフォーマットで選択でき、継承されたパーミッションまたは GID 情報の問題を回避することができます。

21.1.2. sudo および Netgroups

前述の「Identity Management の全般的な **sudo** 設定」ように、Identity Management の sudo エントリーに使用される LDAP スキーマは、netgroups のほかにもホストグループをサポートしています。実際に、Identity Management は、見えるホストグループとシャドウ netgroup の 2 つのグループを作成します。**sudo** 自体は、グループフォーマットの NIS 形式のネットグループのみに対応します。

1 つの重要な点として、**sudo** が NIS netgroups を使用している場合でも、 **NIS サーバーまたは NIS クラ**

イアントを設定する必要はありません。グループが **sudo** 用に作成されると、NIS オブジェクトが Directory Server インスタンスに作成され、その情報は NSS_LDAP または SSSD によって取得されます。クライアント (この場合 **sudo**) は、Identity Management の Directory Server によって提供される情報から必要な NIS 情報を抽出します。[7]

簡単にして、**sudo** 設定には NIS 形式の netgroups が必要です。NIS は必要ありません。

ただし、IdM **sudo** がホストグループと連携するには、**nisdomainname** コマンドを使用して、**sudo** ルールで使用する NIS ドメイン名を設定します。**nisdomainname** の使用方法やその他の設定機能の設定については、「IdM **sudo** ポリシーを使用するようにホストを設定」を参照してください。

21.1.3. サポートされる sudo クライアント

IdM クライアントシステムとして対応しているシステムは、IdM で **sudo** クライアントとして設定できます。

21.2. SUDO コマンドおよびコマンドグループの設定

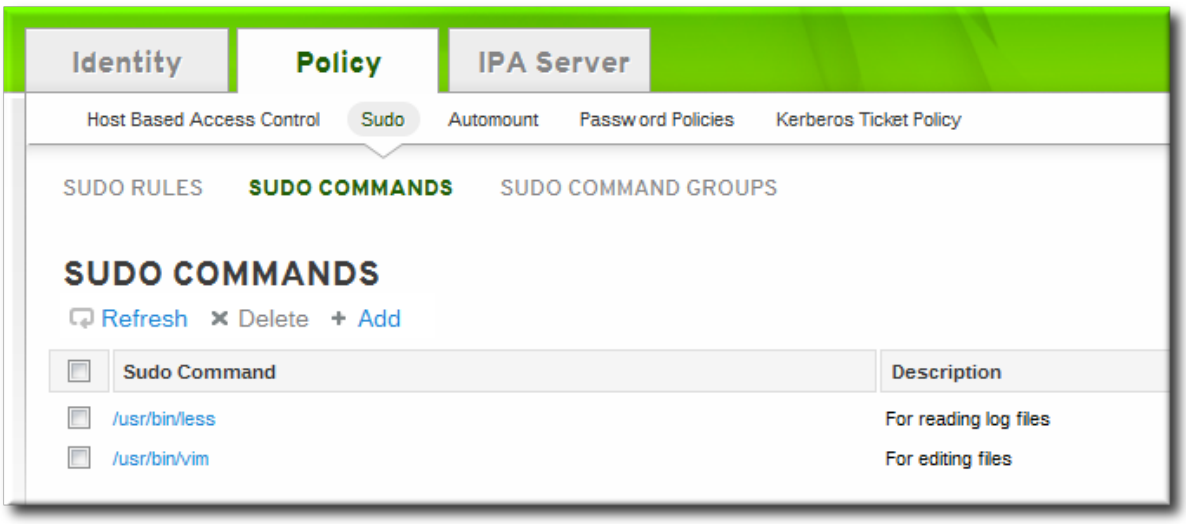
通常の **sudo** 設定と同様に、**sudo** アクセスで管理されるコマンドは、設定に一覧表示する必要があります。Identity Management は、**sudo コマンドグループ** にさらなる制御手段を追加します。これにより、コマンドのグループ を定義し、その後に 1 つのコマンドとして **sudo** 設定に適用することができます。

コマンドまたはコマンドグループを追加すると、IdM が **sudo** ルールで定義できるようになります。ただし、コマンドを追加するだけでは、**sudo** ルールに自動的に追加されません。

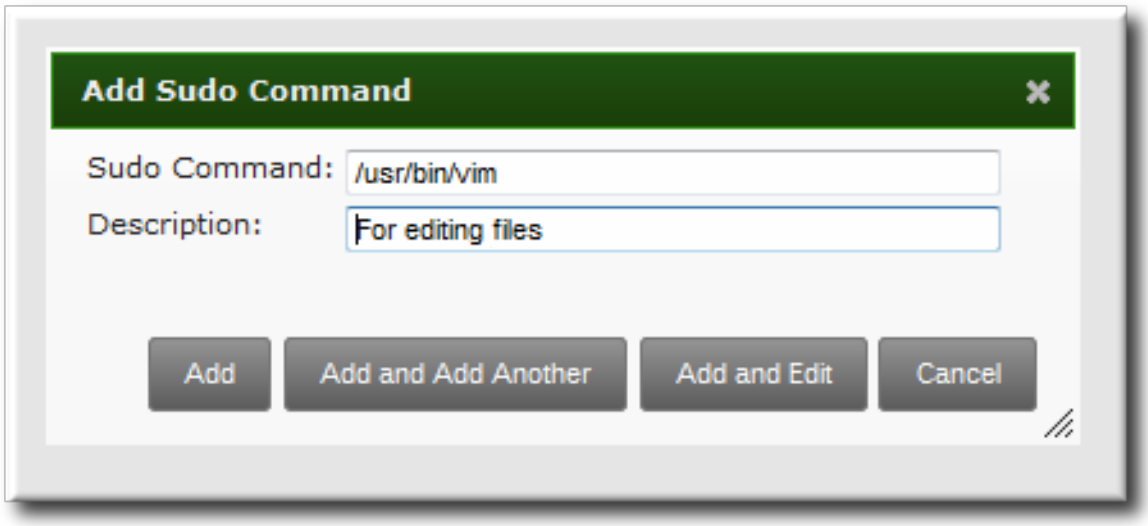
21.2.1. sudo コマンドの追加

21.2.1.1. Web UI を使用した sudo コマンドの追加

1. **Policy** タブをクリックします。
2. **Sudo** サブタブをクリックし、**Sudo Commands** のリンクを選択します。
3. コマンドリストの上部にある **Add** リンクをクリックします。



4. コマンドの完全なシステムパスと名前を入力し、必要に応じて説明を入力します。



5. **Add and Edit** ボタンをクリックして、コマンドの設定ページに即座に移動します。
6. **Sudo Command Groups** タブで、**Add** ボタンをクリックして sudo コマンドをコマンドグループに追加します。
7. 参加するコマンドの groups のチェックボックスをクリックし、右向き矢印ボタン **>>** をクリックしてグループを選択ボックスに移動します。
8. **追加** ボタンをクリックします。

21.2.1.2. コマンドラインでの sudo コマンドの追加

1 つのコマンドを追加するには、**sudocmd-add** コマンドを使用します。これには、コマンド実行ファイルへの完全、ローカルパス、およびコマンドの説明が必要です。

```
$ ipa sudocmd-add --desc "description" /local/path/to/command
```

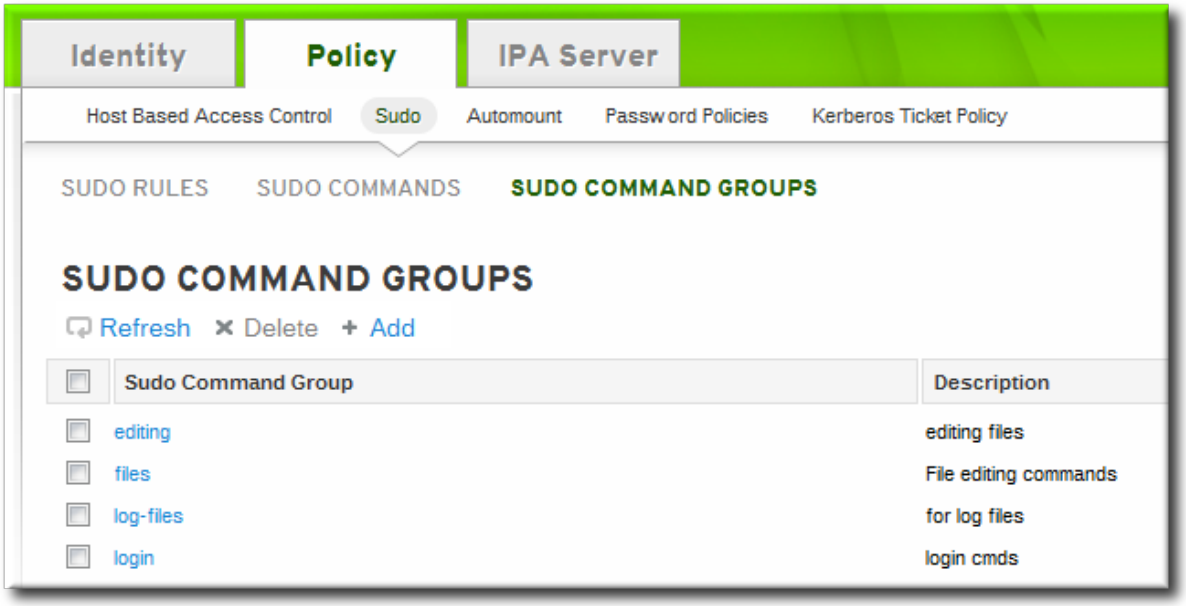
以下に例を示します。

```
$ ipa sudocmd-add --desc 'For reading log files' '/usr/bin/less'
----------------------------------
Added sudo command "/usr/bin/less"
----------------------------------
  sudo Command: /usr/bin/less
  Description: For reading log files
```

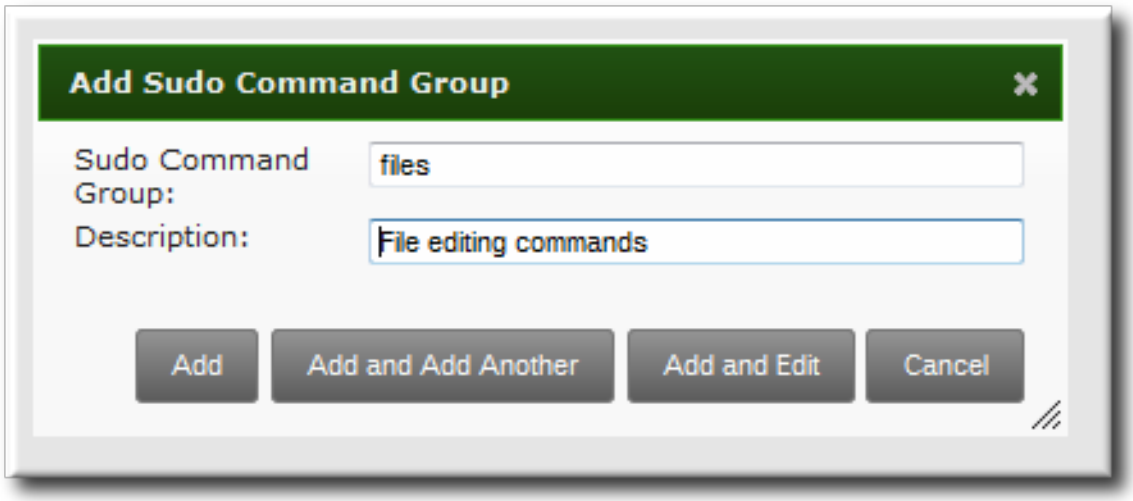
21.2.2. sudo コマンドグループの追加

21.2.2.1. Web UI を使用した sudo コマンドグループの追加

1. **Policy** タブをクリックします。
2. **Sudo** サブタブをクリックし、**Sudo Command Groups** のリンクを選択します。
3. コマンドグループ一覧の上部にある **Add** リンクをクリックします。

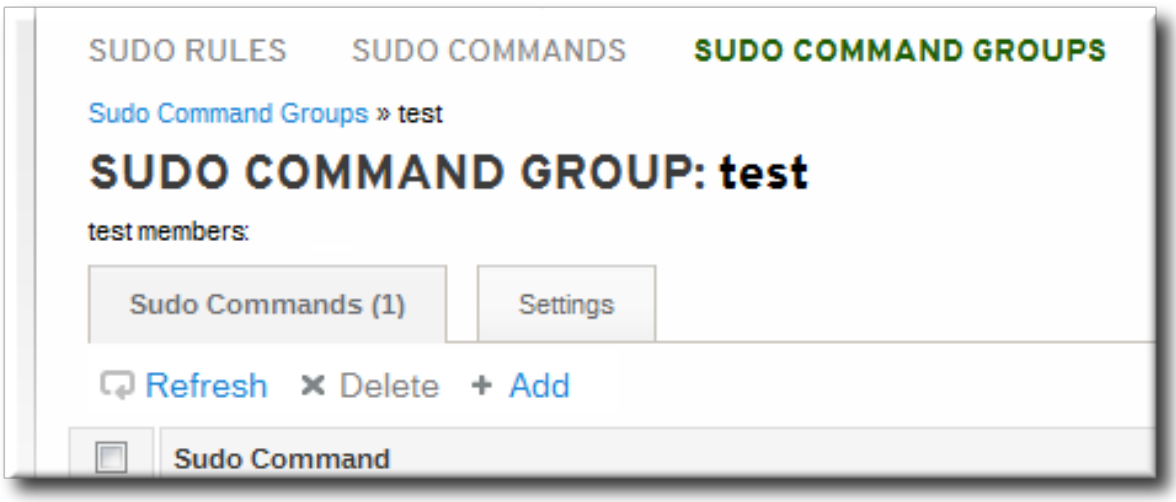


4. 新しいコマンドグループの名前と説明を入力します。

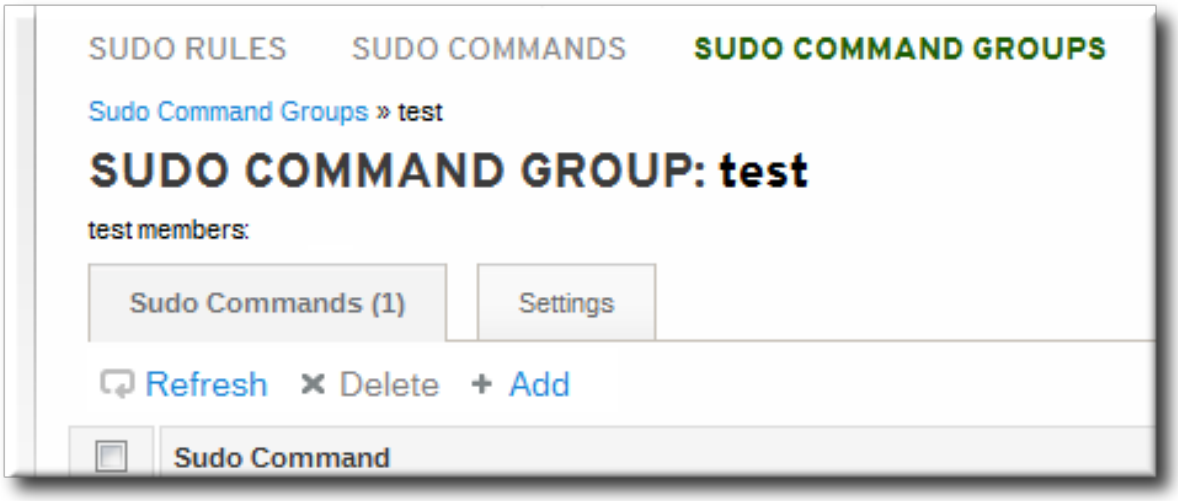


5. **Add and Edit** ボタンをクリックして、グループの設定ページに即座に移動します。

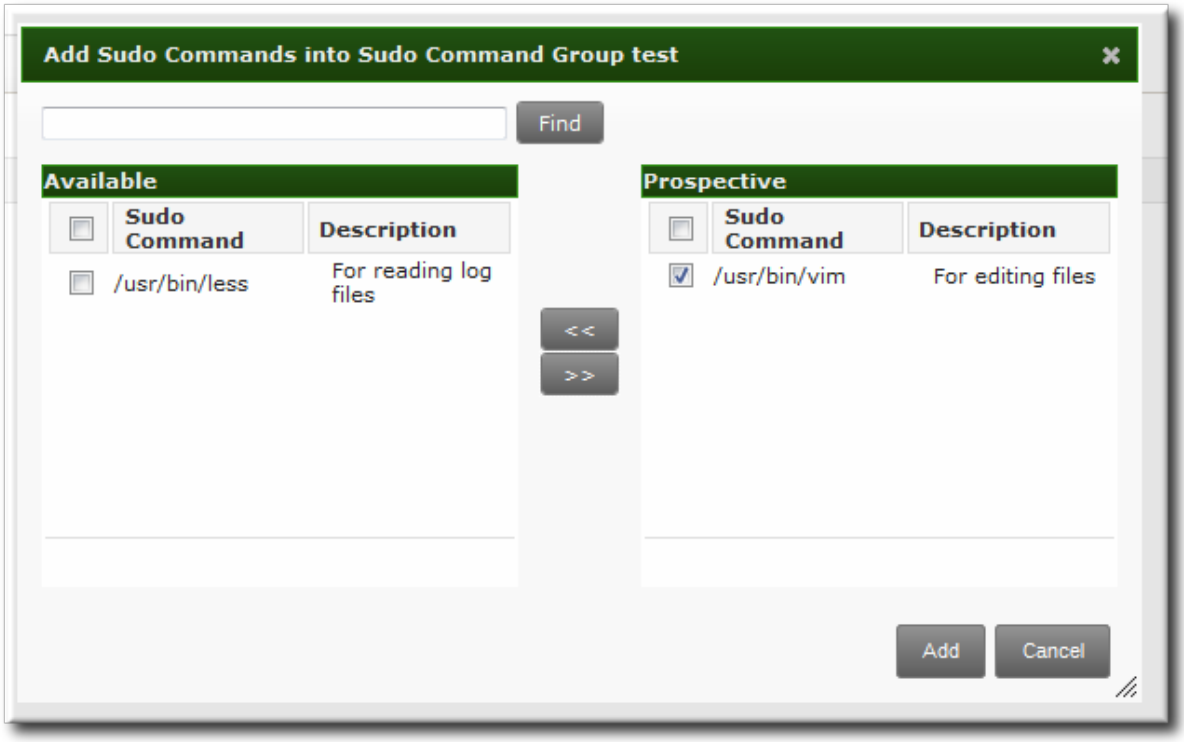
6. **Sudo Commands** タブで **Add** ボタンをクリックして、グループに sudo コマンドを追加します。



7. **Sudo Commands** タブで **Add** ボタンをクリックして、グループに sudo コマンドを追加します。



8. 追加するコマンドの名前の横にあるチェックボックスをクリックし、右向きの矢印 **>>** をクリックして選択ボックスに移動します。



9. **追加** ボタンをクリックします。

21.2.2.2. コマンドラインで sudo コマンドグループの追加

コマンドグループを作成するには、グループ用と、コマンド自体用のエントリーを 2 つ作成する必要があります。

1. **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。

```
$ ipa sudocmdgroup-add --desc 'File editing commands' files
----------------------------------
Added sudo command group "files"
```

```
-----------------------------------
  sudo Command Group: files
  Description: File editing commands
```

2. **sudocmd-add** コマンドを使用して、コマンドエントリーを作成します。

```
$ ipa sudocmd-add --desc 'For editing files' '/usr/bin/vim'
-----------------------------------
Added sudo command "/usr/bin/vim"
-----------------------------------
  sudo Command: /usr/bin/vim
  Description: For editing files
```

3. **sudocmdgroup-add-member** コマンドを使用して、完全なディレクトリーの場所を名前として使用し、コマンドをコマンドグループに追加します。

```
$ ipa sudocmdgroup-add-member --sudocmds '/usr/bin/vim' files
  sudo Command Group: files
  Description: File editing commands
  Member sudo commands: /usr/bin/vim
-------------------------
Number of members added 1
-------------------------
```

21.3. SUDO ルールの定義

sudo ルールは、アクセス制御ルールと似ています。アクセスが付与されたユーザー、ルールの範囲内にあるコマンド、ルールが適用されるターゲットホストを定義します。IdM では、**sudoers** オプションや run-as 設定など、ルールで追加情報を設定できます。ただし、基本要素は常にユーザー、対象 (サービス)、および場所 (ホスト) を定義します。

21.3.1. 外部ユーザーについて

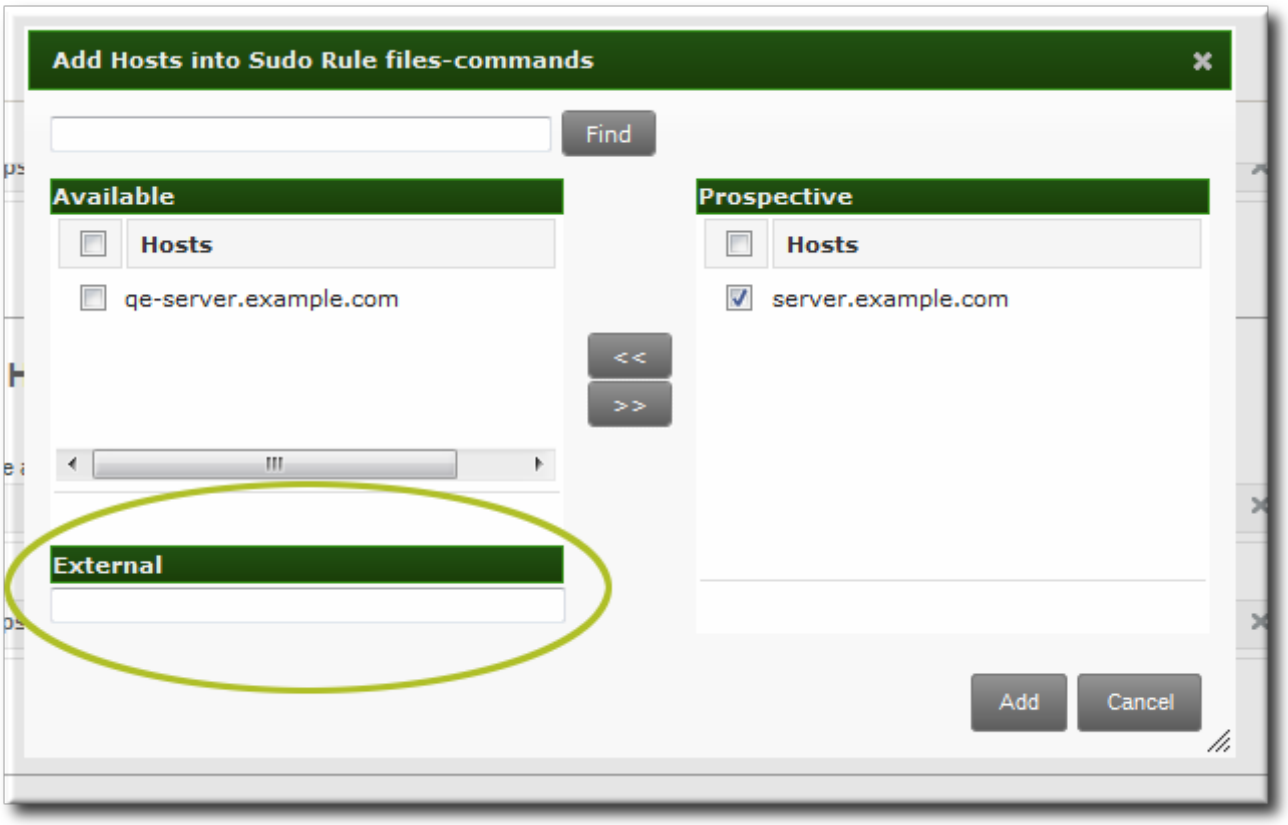
sudo ルールは、**誰** が **何を**、**どこで** で **誰として**行うことができるかという 4 つの要素を定義します。**who** が通常のユーザーであり、**as whom** は、ユーザーがタスクの実行に使用するシステムまたは他のユーザー ID です。これらのタスクは、ターゲットマシンで実行できる (実行しない) システムコマンドです。

これら 3 つの要素 who、who、where はアイデンティティーです。これらはユーザーです。多くの場合、環境内のシステムユーザーと、IdM ドメインに属するユーザーおよびホストとの間に重複があるため、これらのアイデンティティーは IdM ドメイン内のエンティティーになります。

ただし、**sudo** ポリシーが現実的にカバーできるすべてのアイデンティティーが必ずしも当てはまるわけではありません。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、root ユーザーは IdM のユーザーではありません。別の例では、ネットワーク上にあるものの IdM ドメインの一部ではない特定ホストへのアクセスを管理者はブロックする場合もあります。

Identity Management の **sudo** ルールは、**外部** ユーザーの概念 (つまり、保存され、IdM 設定外に存在するユーザー) をサポートします。

図21.1 外部エンティティー



sudo ルールを設定する際に、ユーザーおよび run-as 設定は **sudo** ルールに含めて評価できるように、外部アイデンティティーを参照することができます。

21.3.2. sudo オプションのフォーマットについて

sudo ルールは、サポートされる **sudoers** オプションを使用するよう設定できます。オプションの完全なリストは **sudoers** man ページにあります。

ただし、Identity Management の **sudo** ルール設定では、**/etc/sudoers** ファイルの設定と同じ形式は**使用できません**。特に、Identity Management では、UI または CLI で設定されるかどうかに関係なく、オプションパラメーターには空白文字は使用できません。

たとえば、**/etc/sudoers** ファイルでは、以下のように空白文字が付いたコンマ区切りリストのオプションをリストできます。

```
mail_badpass, mail_no_host, mail_no_perms, syslog = local2
```

ただし、Identity Management では、同じ設定が別の引数として解釈されます。これには、等号記号 (=) にスペースがあるためです。代わりに、UI またはコマンドラインツールを使用して、各オプションを個別に追加する必要があります。

```
[jsmith@server ~]$ ipa sudorule-add-option readfiles
Sudo Option: mail_badpass
-----------------------------------------------------
Added option "mail_badpass" to Sudo rule "readfiles"
-----------------------------------------------------
[jsmith@server ~]$ ipa sudorule-add-option readfiles
Sudo Option: syslog=local2
-----------------------------------------------------
```

```
Added option "syslog=local2" to Sudo rule "readfiles"
----------------------------------------------------
...
```

同様に、**/etc/sudoers** ファイルで無視される改行は、Identity Management 設定では許可されません。

```
env_keep = "COLORS DISPLAY EDITOR HOSTNAME HISTSIZE INPUTRC
        KDEDIR LESSSECURE LS_COLORS MAIL PATH PS1 PS2
        QTDIR USERNAME LANG LC_ADDRESS LC_CTYPE LC_COLLATE
        LC_IDENTIFICATION LC_MEASUREMENT LC_MESSAGES
        LC_MONETARY LC_NAME LC_NUMERIC LC_PAPER LC_TELEPHONE
        LC_TIME LC_ALL LANGUAGE LINGUAS _XKB_CHARSET
    XAUTHORITY"
```

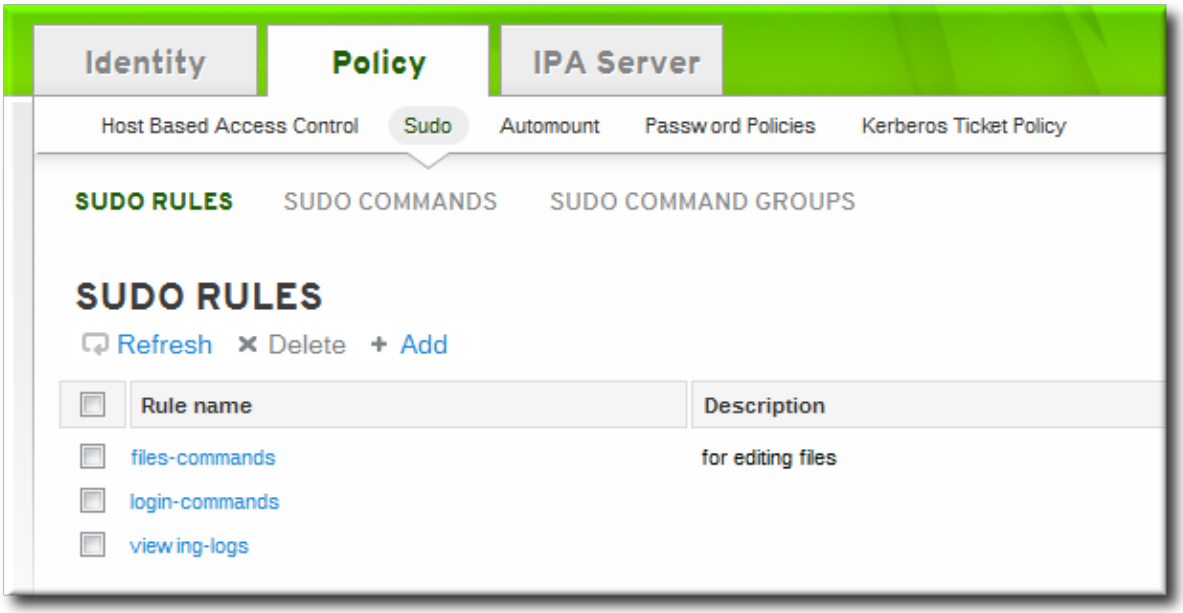
たとえば、IdM コマンドラインの同じコマンドには、1 行上の変数がすべて含まれ、等号記号の周りにはスペースがありません。

```
[jsmith@server ~]$ ipa sudorule-add-option readfiles
Sudo Option: env_keep="COLORS DISPLAY EDITOR HOSTNAME HISTSIZE INPUTRC KDEDIR
LESSSECURE LS_COLORS MAIL PATH PS1 PS2 ... XAUTHORITY"
```

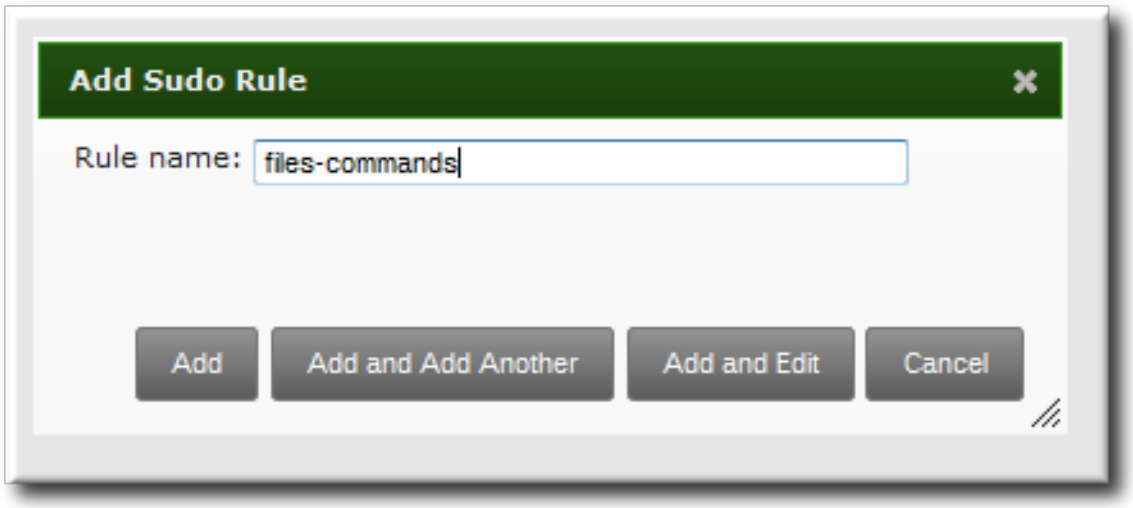
Identity Management で複数の **sudoers** オプションを指定するには、各オプションを 1 行すべてではなく、個別のオプション設定として設定します。

21.3.3. Web UI での sudo ルールの定義

1. **Policy** タブをクリックします。
2. **Sudo** サブタブをクリックし、**Sudo Rules** のリンクをクリックします。
3. sudo ルールの一覧の上部にある **Add** リンクをクリックします。



4. ルールの名前を入力します。



5. **Add and Edit** ボタンをクリックして、すぐにルールの設定を設定します。

   ルールには設定エリアが多数あります。最も基本的な要素は、**Who**、**Access This Host**、**Run Commands** で設定されています。もう 1 つ はオプションで、ルールを改良するために使用されます。

6. **任意。Options** エリアで、**sudoers** オプションを追加します。オプションの完全なリストは **sudoers** man ページにあります。

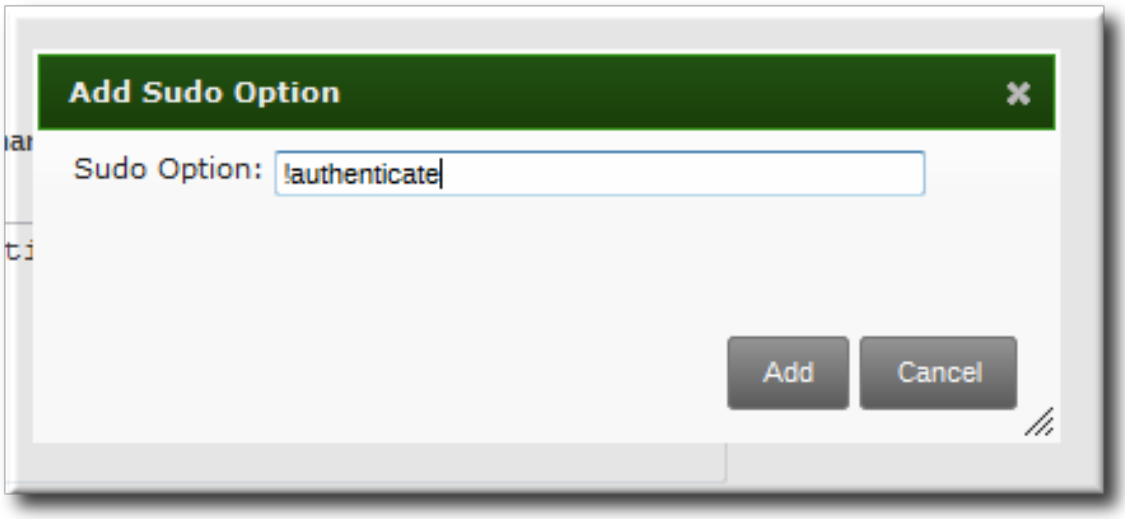
   注記

   「sudo オプションのフォーマットについて」で説明されているように、値に空白があるオプションを使用しないでください。1 行にオプションの一覧を追加するのではなく、目的のオプションごとに 1 つのオプション設定を追加します。

   a. オプション一覧の右側にある **+ Add** リンクをクリックします。

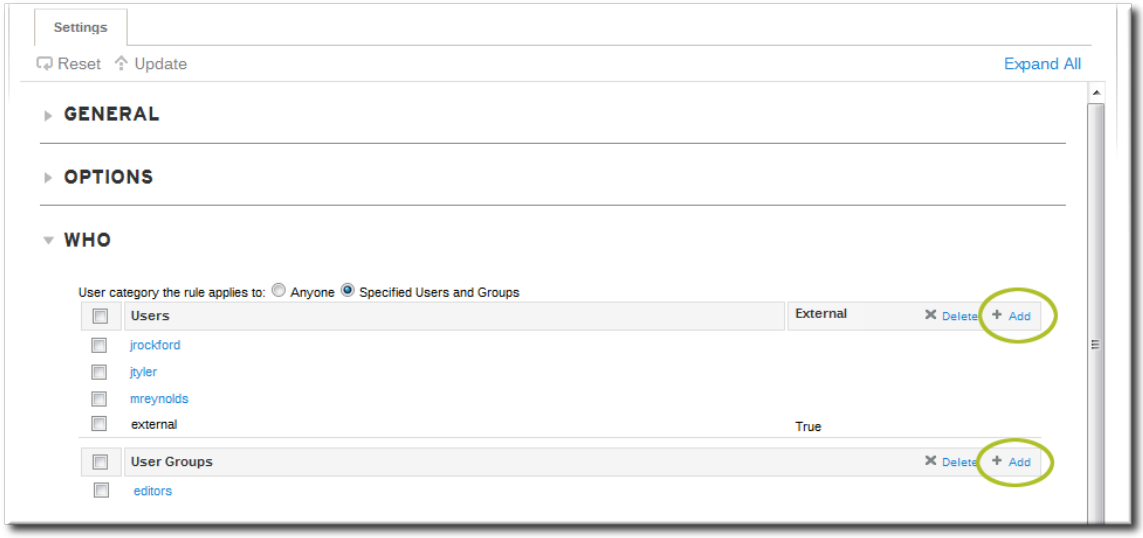
   b. **sudoers** オプションを入力します。



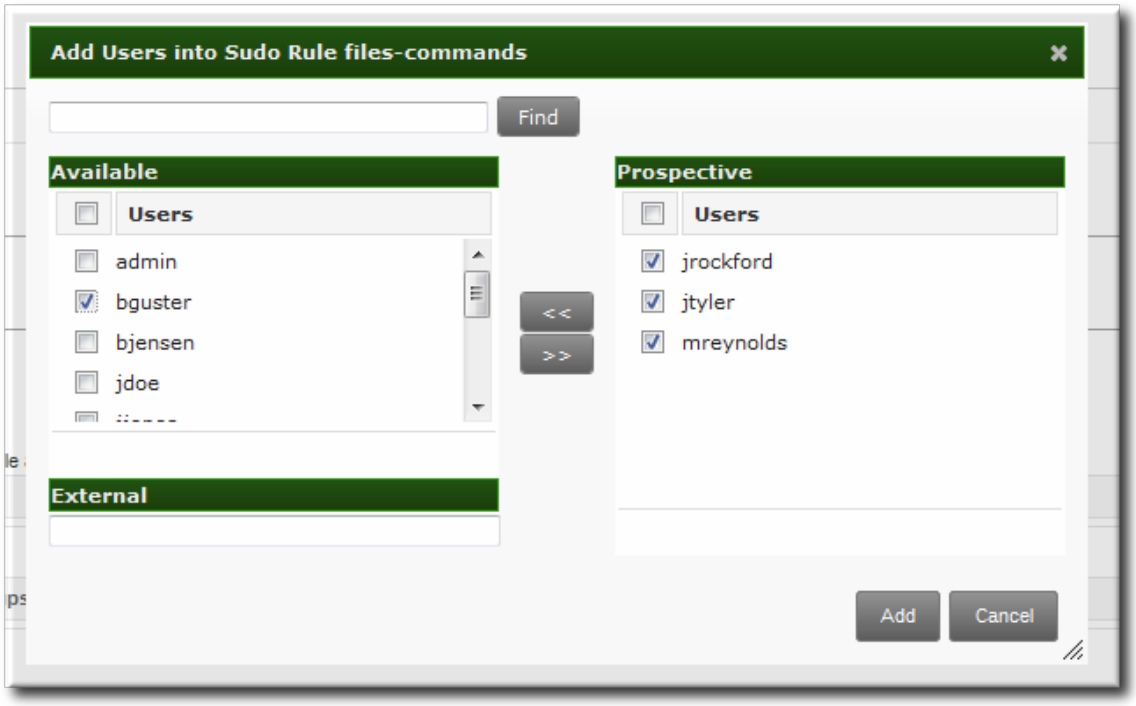
c. **追加** をクリックします。

7. **Who** エリアで、sudo ルールが適用されるユーザーまたはユーザーグループを選択します。

a. ユーザー一覧の右側にある **+ Add** リンクをクリックします。



b. ルールに追加するユーザーのチェックボックスをクリックし、右向きの矢印 (**>>**) をクリックして選択ボックスにユーザーを移動します。

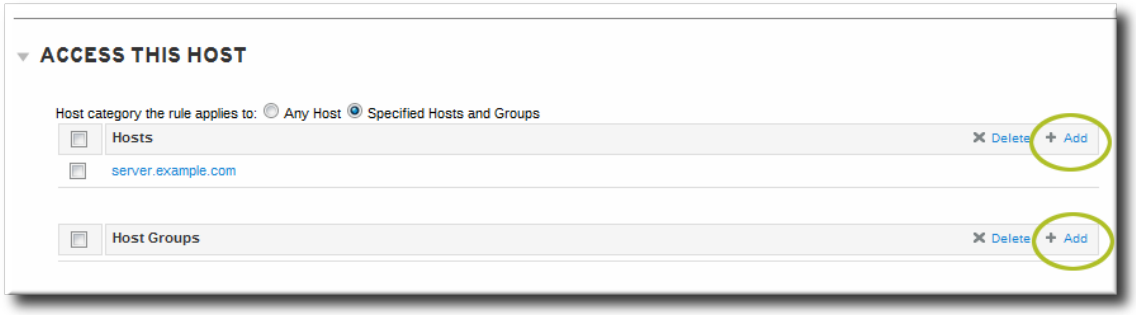


c. **追加** をクリックします。

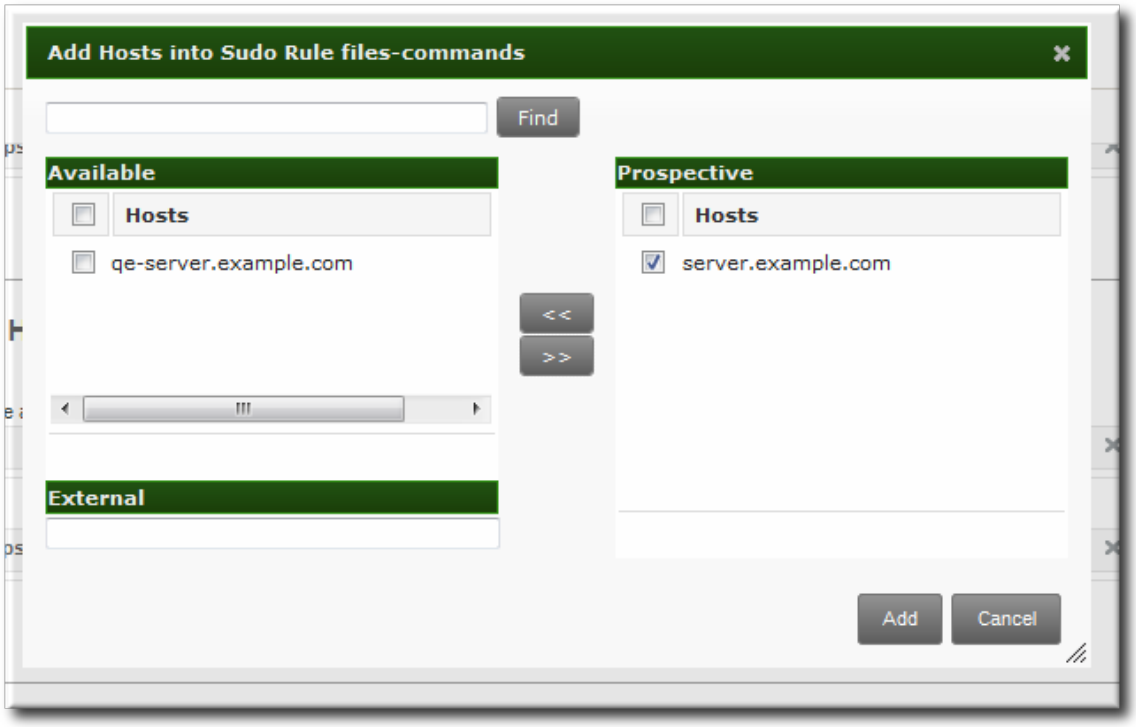
IdM ユーザーと外部システムユーザー (「外部ユーザーについて」) の両方を設定できます。

8. **Access This Host** エリアで、sudo ルールが有効なホストを選択します。

a. ホスト一覧の右側にある **+ Add** リンクをクリックします。



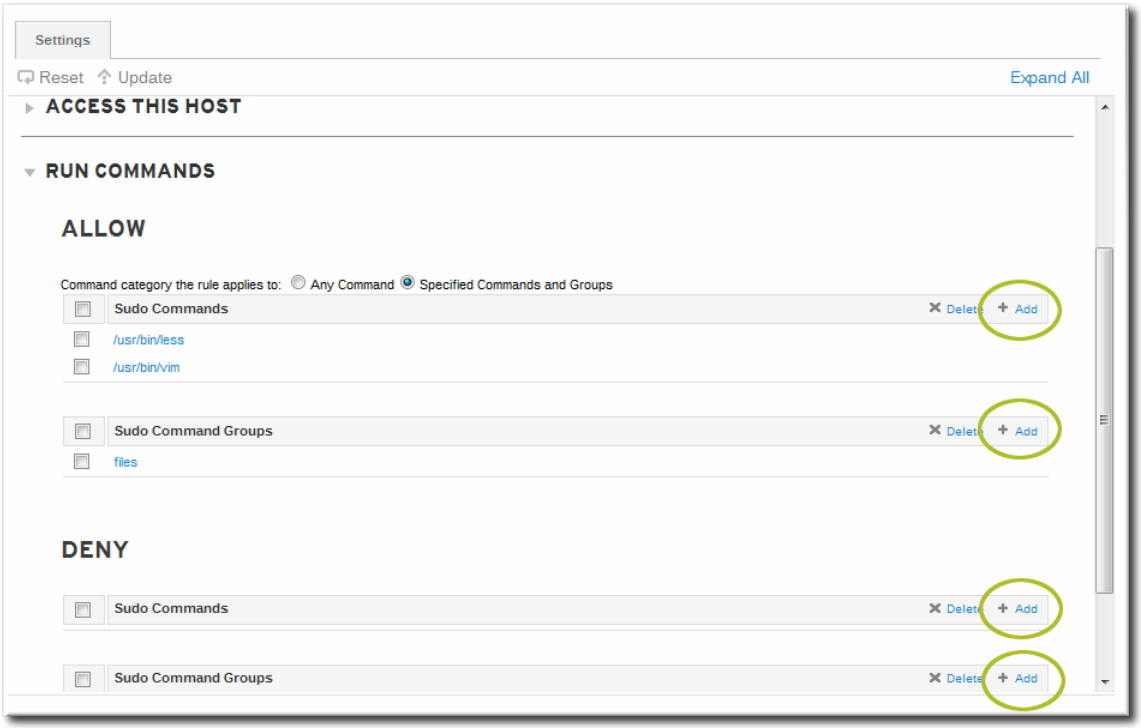
b. ルールとともに追加するホストのチェックボックスをクリックし、右矢印ボタン (**>>**) をクリックして、ホストを選択項目のボックスに移動します。



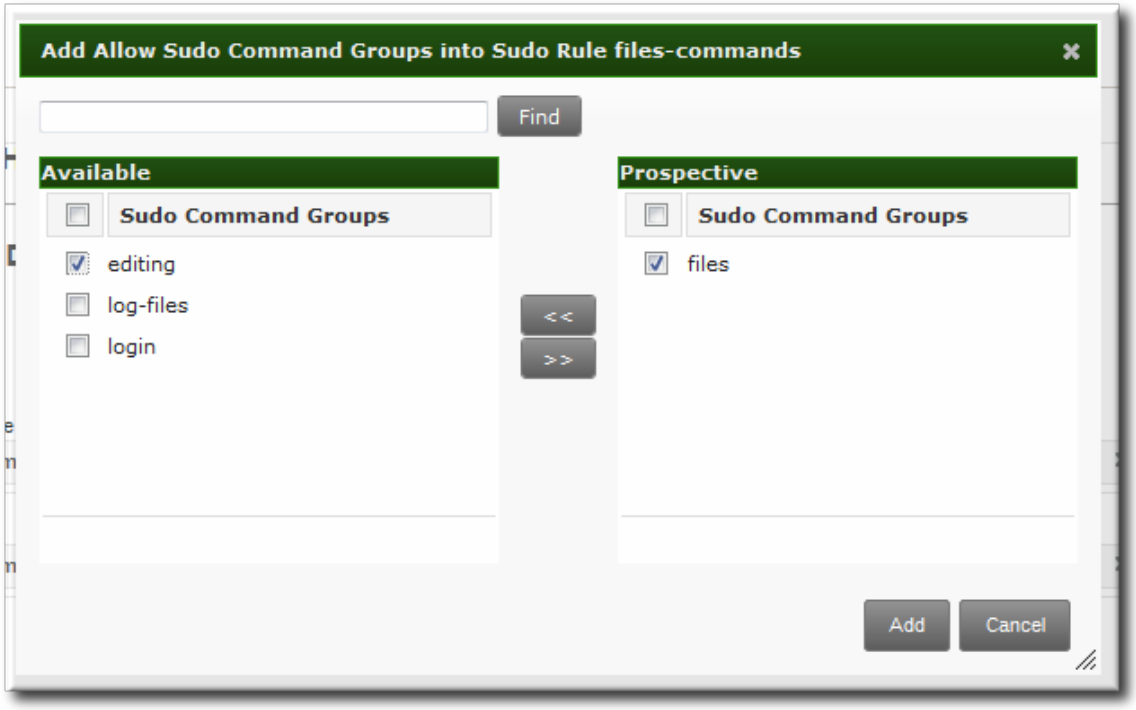
c. **追加** をクリックします。

9. **Run Commands** エリアで、sudo ルールに含まれるコマンドを選択します。**sudo** ルールは、アクセスを許可またはコマンドへのアクセスを拒否します。また、あるコマンドへのアクセスを許可し、別のコマンドへのアクセスを拒否することができます。

a. **Allow/Deny** エリアで、コマンド一覧の右側にある **+ Add** リンクをクリックします。



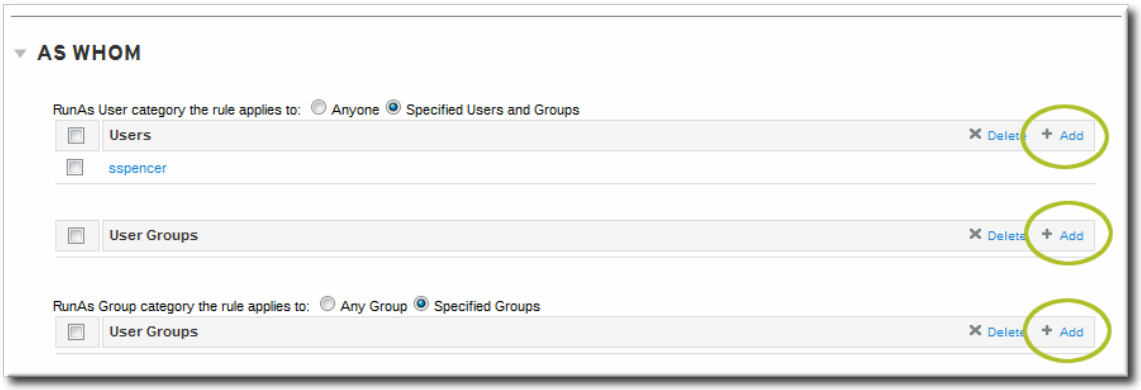
b. ルールとともに追加する、複数のコマンドまたは単一のコマンドまたはグループのチェックボックスをクリックし、右矢印ボタン (**>>**) をクリックして、コマンドを選択項目のボックスに移動します。



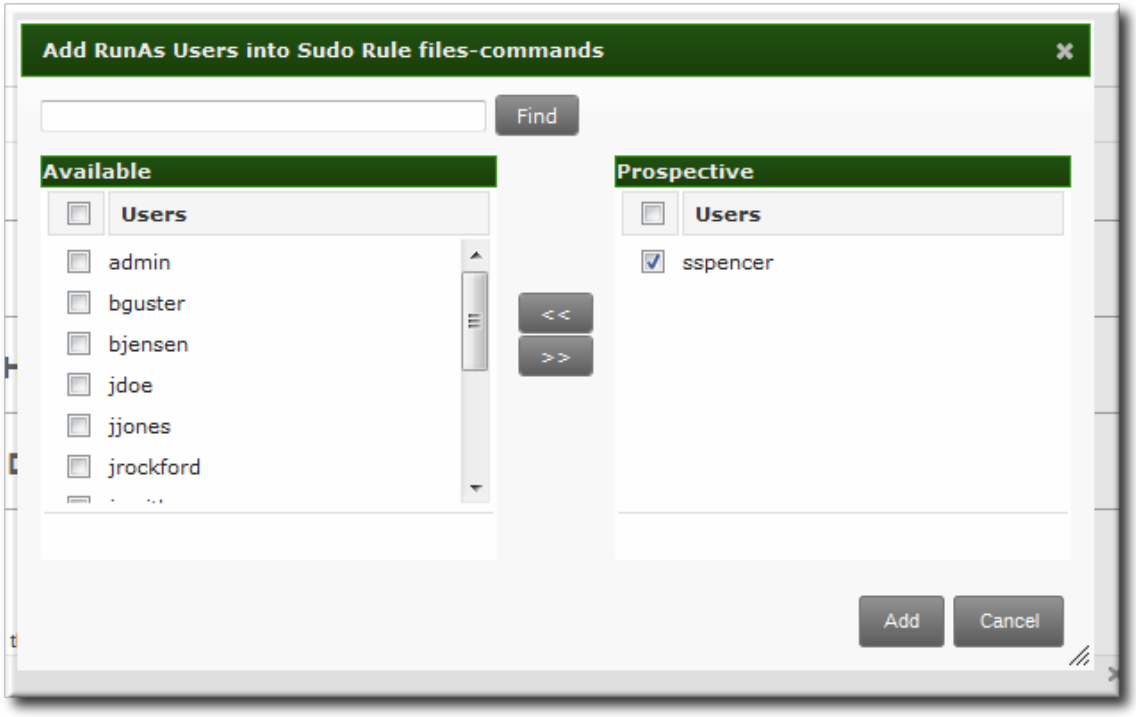
c. **追加** をクリックします。

10. **任意。** sudo ルールを設定して、指定したコマンドを特定の root 以外のユーザーとして実行できます。

a. **As Whom** エリアで、ユーザー一覧の右側にある **+ Add** リンクをクリックします。



b. ユーザーのチェックボックスをクリックしてコマンドを実行し、右向きの矢印 (**>>**) をクリックして選択ボックスにユーザーを移動します。



c. **追加** をクリックします。

21.3.4. コマンドラインでの sudo ルールの定義

各要素は、異なるコマンド (表21.1「sudo コマンド」 に一覧あり) を使用して、ルールコマンドに追加されます。

sudo ルールコマンドの基本的な概要は、以下のとおりです。

```
$ ipa sudorule-add* options ruleName
```

例21.1 基本的な sudo ルールの作成

最も基本的なケースでは、**sudo** 設定が、1 つのホストの 1 つのコマンドに対して、1 つのユーザーに適切な権限を付与します。

最初のステップでは、最初のルールエントリーを追加します。

```
[jsmith@server ~]$ kinit admin
[jsmith@server ~]$ ipa sudorule-add files-commands
---------------------------------
Added sudo rule "files-commands"
---------------------------------
  Rule name: files-commands
  Enabled: TRUE
```

次に、アクセス権限を**付与**するコマンドを追加します。これは、**--sudocmds** の単一のコマンド、または **--sudocmdgroups** を使用したコマンドグループで行うことができます。

```
[jsmith@server ~]$ ipa sudorule-add-allow-command --sudocmds "/usr/bin/vim" files-commands
  Rule name: files-commands
  Enabled: TRUE
  sudo Commands: /usr/bin/vim
```

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

ルールにホストまたはホストグループを追加します。

```
[jsmith@server ~]$ ipa sudorule-add-host --host server.example.com files-commands
  Rule name: files-commands
  Enabled: TRUE
  Hosts: server.example.com
  sudo Commands: /usr/bin/vim
-------------------------
Number of members added 1
-------------------------
```

最後に、ユーザーまたはグループをルールに追加します。これは、ルールで定義されたとおりに **sudo** を使用可能なユーザーです。「run-as」ユーザーが指定されていない場合、このユーザーは root として **sudo** コマンドを実行します。

```
[jsmith@server ~]$ ipa sudorule-add-user --user jsmith files-commands
  Rule name: files-commands
  Enabled: TRUE
  Users: jsmith
  Hosts: server.example.com
  sudo Commands: /usr/bin/vim"
-------------------------
Number of members added 1
-------------------------
```

例21.2 コマンドの許可と拒否

sudo ルールは、コマンドへのアクセスを許可したり、アクセスを拒否することができます。たとえば、このルールではファイルへの読み取りアクセスが許可されますが、編集はできません。

```
[jsmith@server ~]$ kinit admin
[jsmith@server ~]$ ipa sudorule-add-allow-command --sudocmds "/usr/bin/less" readfiles
[jsmith@server ~]$ ipa sudorule-add-allow-command --sudocmds "/usr/bin/tail" readfiles
[jsmith@server ~]$ ipa sudorule-add-deny-command --sudocmds "/usr/bin/vim" readfiles
```

例21.3 sudoers オプションの使用

この **sudoers** ファイルには、**sudo** ユーザーの動作を制御するようにセットできる見込みフラグが多くあります。これは、ユーザーが **sudo** に対する認証を行うためにパスワードを必須としたり (あるいは必須としない)、**sudoers** ファイルで完全に資格のあるドメイン名を使用するような動作です。オプションの完全なリストは **sudoers** man ページにあります。

この **sudorule-add-option** コマンドを使用して、IdM **sudo** ルールにオプションのいずれかを設定できます。コマンドを実行すると、オプションを追加するプロンプトが表示されます。

```
[jsmith@server ~]$ kinit admin
[jsmith@server ~]$ ipa sudorule-add-option readfiles
```

```
Sudo Option: !authenticate
-------------------------------------------------------
Added option "!authenticate" to Sudo rule "readfiles"
-------------------------------------------------------
```

注記

「sudo オプションのフォーマットについて」で説明されているように、値に空白があるオプションを使用しないでください。1行にオプションの一覧を追加するのではなく、目的のオプションごとに1つのオプション設定を追加します。

例21.4 他のユーザーとしての実行

sudo ルールには、root 以外のユーザーまたはグループを指定してコマンドを実行することもできます。最初のルールには、**--sudorule-add-runasuser** または **--sudorule-add-runasgroup** コマンドを使用してそれぞれユーザーまたはグループが指定されています。

```
$ ipa sudorule-add-runasuser --users=jsmith readfiles
$ ipa sudorule-add-runasgroup --groups=ITadmins readfiles
```

ルールの作成時に、**sudorule-add-runasuser** または **sudorule-add-runasgroup** コマンドは **特定**のユーザーまたはグループのみを設定できます。ただし、ルールを編集する場合は、**--runasusercat** または **--runasgroupcat** を使用して、すべてのユーザーまたはすべてのグループとして **sudo** を実行できます。たとえば、以下のようになります。

```
$ ipa sudorule-mod --runasgroupcat=all ruleName
```

注記

--sudorule-add-runasuser および **--sudorule-add-runasgroup** コマンドは、特定のユーザー名またはグループ名のみに対応しており、**all** オプションには対応していません。すべてのユーザーまたはすべてのグループの指定は、**sudorule-mod** コマンドでオプションとともにのみ使用できます。

例21.5 外部ユーザーの参照

sudo ルールの「who」は IdM ユーザーですが、論理的で有用なルールが多数あります。この指定の1つがシステムユーザーです。同様に、ルールは、IdM クライアントではないネットワーク上のホストマシンへのアクセスを許可または拒否する必要がある場合があります。

このような場合には、この **sudo** ポリシーは、**外部**ユーザーを参照できます。これは、IdM (「外部ユーザーについて」) 外で作成・保存されているアイデンティティーです。

外部アイデンティティーを **sudo** ルールに追加するオプションは次のとおりです。

- --externaluser
- --runasexternaluser

以下に例を示します。

```
$ ipa sudorule-add-user --externaluser=ITAdmin readfiles
$ ipa sudorule-add-runasuser --runasexternaluser=root readfiles
```

表21.1 sudo コマンド

コマンド	説明
sudorule-add	sudo ルールエントリーを追加します。
sudorule-add-user	ユーザーまたはユーザーグループを sudo ルールに追加します。このユーザー (またはグループのすべてのメンバー) は、ルール内のコマンドのいずれかを sudo することができます。
sudorule-add-host	ルールのターゲットホストを追加します。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。
sudorule-add-runasgroup	sudo コマンドを実行するには、グループを設定します。これは特定のユーザーである必要があります。すべてのユーザーを指定するには、**sudo-rule** を使用してルールを変更します。
sudorule-add-runasuser	sudo コマンドを実行するには、ユーザーを設定します。これは特定のユーザーである必要があります。すべてのユーザーを指定するには、**sudo-rule** を使用してルールを変更します。
sudorule-add-allow-command	ルールのユーザーが実行に sudo パーミッションを持つコマンドを追加します。
sudorule-add-deny-command	ルールのユーザーが、実行する sudo パーミッションを **拒否**されたコマンドを追加します。
sudorule-add-option	sudo ルールに sudoers フラグを追加します。
sudorule-disable	sudo ルールエントリーを一時的に非アクティブにします。
sudorule-enable	以前に一時停止した sudo ルールをアクティベートします。
sudorule-del	sudo ルールを完全に削除します。

例21.6 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
```

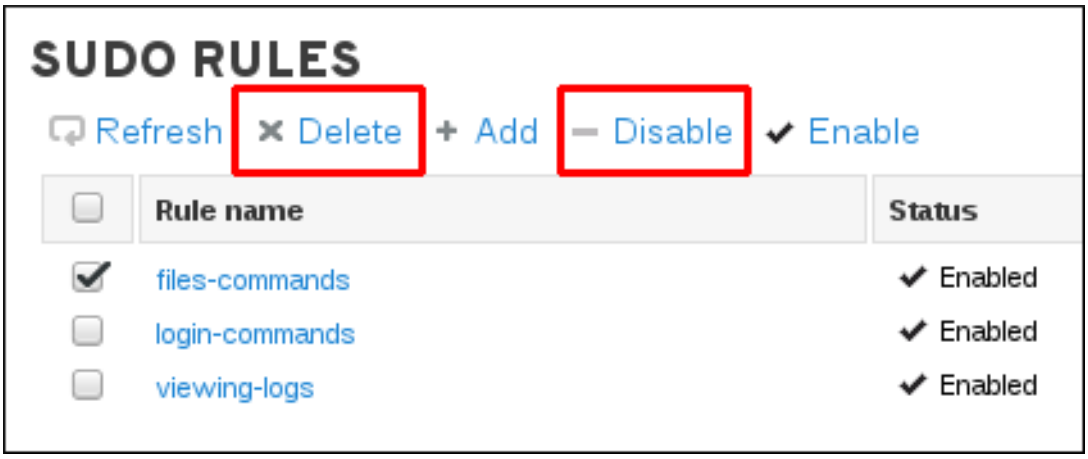
21.3.5. sudo ルールの一時停止および削除

定義した **sudo** ルールは、Web UI またはコマンドラインから一時的に非アクティブ化または完全に削除できます。中断されたルールは、サーバーを再起動しなくても **ou=sudoers** compat ツリーから削除されます。

Web UI からの sudo ルールの一時停止および削除

Web UI からルールを一時停止または完全に削除するには、**sudo** ルールの一覧の上部にある **Disable** または **Delete** ボタンを使用します。

図21.2 Web UI からのsudo ルールの一時停止または削除



コマンドラインからの sudo ルールの一時停止および削除
コマンドラインからルールを一時停止するには、以下のようなコマンドを実行します。

```
ipa sudorule-disable files-commands
```

コマンドラインからルールを完全に削除するには、以下のようなコマンドを実行します。

```
ipa sudorule-del files-commands
```

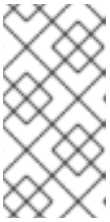
21.4. IDM SUDO ポリシーを使用するようにホストを設定

実際に **sudo** ポリシーを実装するのは、IdM でルールを作成するよりも複雑です。これらのルールはすべてのローカルマシンに適用する必要があります。つまり、IdM ドメインの各システムがポリシーに対して IdM を参照するように設定する必要があります。

SSSD または LDAP を使用して、ホストに **sudo** ポリシーを適用できます。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。

21.4.1. SSSD を使用した sudo ポリシーのホストへの適用

1. IdM でホストおよび **sudo** エントリーを設定します。
    1. 「sudo コマンドおよびコマンドグループの設定」の説明に従って、**sudo** コマンドおよびコマンドグループを設定します。
    2. 「sudo ルールの定義」の説明に従って、**sudo** ルールを設定します。
    3. **任意。**「ホストグループの管理」の説明に従って、ホストグループを設定します。
    4. **任意。**「ユーザーグループの作成」で説明されているようにユーザーグループを作成し、ユーザーを追加します。
2. **sudo** ルールに SSSD を使用するように、IdM ドメインのすべてのシステムを設定します。



注記

このステップは、Red Hat Enterprise Linux 6.5 以前に基づいたシステムでのみ実行してください。Red Hat Enterprise Linux 6.6 以降では、**ipa-client-install** ユーティリティーが SSSD を自動的に **sudo** のデータプロバイダーとして設定します。

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一覧に追加します。

```
[root@server ~]# 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。

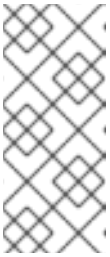
    1. **sudo** ルールで使用する NIS ドメイン名を設定します。

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

    2. 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。たとえば、以下のようになります。

    ```
    [root@server ~]# echo "NISDOMAIN=example.com.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。



注記

sudo は NIS 形式のネットグループを使用しますが、NIS サーバーをインストールする必要はありません。netgroups では、NIS ドメインを設定で命名する必要があるため、**sudo** では、netgroups に NIS ドメインという名前を付ける必要があります。ただし、NIS ドメインが存在する必要はありません。

3. 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** ファイルに記録されます。

21.4.2. 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 Red Hat Enterprise Linux 6.3 以前またはクライアントには、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。

1. IdM でホストおよび sudo エントリーを設定します。
    1. **任意。**「ホストグループの管理」の説明に従って、ホストグループを設定します。
    2. **任意。**「ユーザーグループの作成」で説明されているようにユーザーグループを作成し、ユーザーを追加します。
    3. 「sudo コマンドおよびコマンドグループの設定」の説明に従って、**sudo** コマンドおよびコマンドグループを設定します。
    4. 「sudo ルールの定義」の説明に従って、**sudo** ルールを設定します。
2. デフォルトの IdM **sudo** ユーザーのパスワードを設定して、バインド (認証) ユーザーを設定します。ユーザーがサーバーへの認証が可能でなければなりません。**sudo** ポリシーでは、匿名アクセスはサポートされません。

    LDAP ツールを使用して、**sudo ユーザーのパスワード uid=sudo,cn=sysaccounts,cn=etc,dc=example,dc=com** を設定します。以下に例を示します。

```
[jsmith@server ~]$ ldappasswd -Y GSSAPI -S -h ipaserver.example.com
uid=sudo,cn=sysaccounts,cn=etc,dc=example,dc=com
  New password:
  Re-enter new password:
  Enter LDAP Password:
```

3. sudo ルールに SSSD を使用するように、IdM ドメインのすべてのシステムを設定します。
    1. **sudo** を設定して、**sudoers** ファイルの LDAP を検索します。

```
vim /etc/nsswitch.conf

sudoers:  files ldap
```

    この **files** オプションをそのままにすると、**sudo** は LDAP ベースの IdM 設定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. **/etc/ldap.conf** ファイル内の **sudo** 操作のデバッグロギングを有効にします。このファイルが存在しない場合は作成できます。

```
vim /etc/ldap.conf

sudoers_debug: 1
```

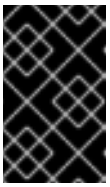
注記

sudoers_debug パラメーターを追加すると、トラブルシューティングに役立ちます。このパラメーターの有効な値は 0、1、および 2 です。**sudo** ドキュメント (http://www.gratisoft.us/sudo/readme_ldap.html) には、プロセスのデバッグに関する詳細情報が記載されています。

3. NSS/LDAP 設定ファイルを編集し、以下の **sudo** 関連の行を **/etc/sudo-ldap.conf** ファイルに追加します。

```
binddn uid=sudo,cn=sysaccounts,cn=etc,dc=example,dc=com
bindpw sudo_password

ssl start_tls
tls_cacertfile /etc/ipa/ca.crt
tls_checkpeer yes

bind_timelimit 5
timelimit 15

uri ldap://ipaserver.example.com ldap://backup.example.com:3890
sudoers_base ou=SUDOers,dc=example,dc=com
```

複数の LDAP サーバーをスペースで区切って設定できます。その他のオプション (SSL や非標準ポートなど) は LDAP URL と併用できます。**sudo** LDAP 設定は sudooers.ldap(8) man ページで説明されています。



重要

uri ディレクティブは、IP アドレスではなく、LDAP サーバーの完全修飾ドメイン名を提供する必要があります。それ以外の場合は、**sudo** は LDAP サーバーへの接続に失敗します。

4. **任意。** SSSD でのデバッグを有効にして、使用している LDAP 設定を表示します。

```
[root@server ~]# vim /etc/sssd/sssd.conf

[domain/LDAPDOMAIN]
debug_level = 6
....
```

SSSD が操作に使用する LDAP 検索ベースは、**sssd_***DOMAINNAME***.log** ファイルに記録されます。

5. **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。

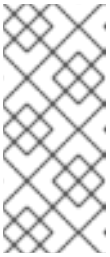
    1. **sudo** ルールで使用する NIS ドメイン名を設定します。

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

    2. 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。たとえば、以下のようになります。

    ```
    [root@server ~]# 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。



注記

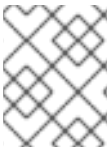
sudo は NIS 形式のネットグループを使用しますが、NIS サーバーをインストールする必要はありません。netgroups では、NIS ドメインを設定で命名する必要があるため、**sudo** では、netgroups に NIS ドメインという名前を付ける必要があります。ただし、NIS ドメインが存在する必要はありません。

[7] Identity Management Directory Server インスタンスは、RFC 2307 で定義されている NIS オブジェクトの標準の LDAP スキーマを使用します。

第22章 ポリシー: ホストベースのアクセス制御の設定

IdM は、IdM ドメイン内のマシンおよびそれらのマシンのサービスの両方へのアクセスを制御できます。ルールは、(システムやアプリケーション設定で定義される) アクセスのレベルではなく、ドメイン内の対象にアクセスできるユーザーを定義します。これらのアクセス制御ルールにより、他のすべてのユーザーとホストが暗黙的に拒否されたアクセスを許可します。

ルールが、ユーザーがアクセスできるドメイン内のホスト (**ターゲット**) を定義するため、**ホストベースのアクセス制御**と呼ばれます。このアクセスは、それらのホストのユーザーおよびサービスにさらに分類できます。



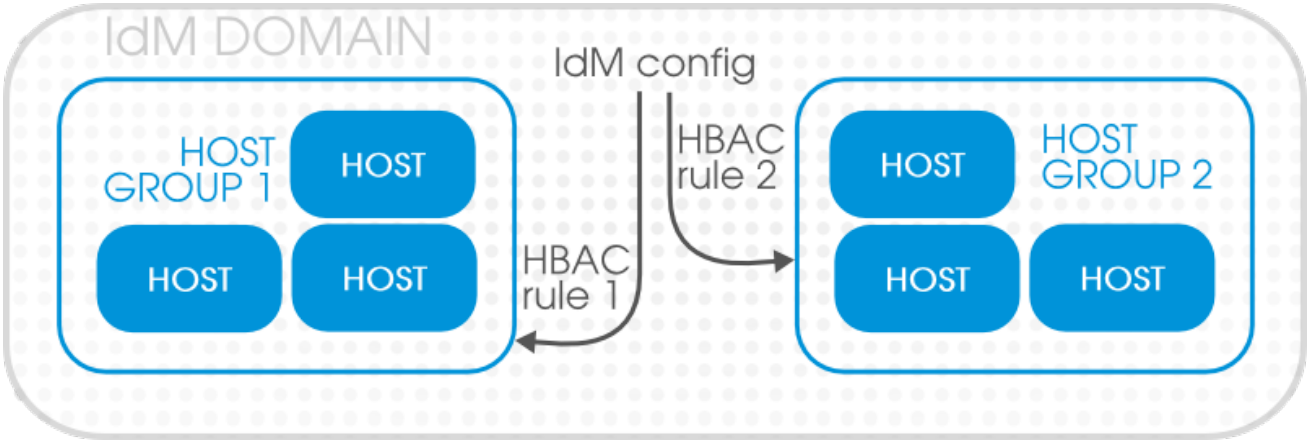
注記

ホストベースのアクセス制御を使用する場合は、SSSD を IdM クライアントマシンにインストールし、設定する必要があります。

22.1. ホストベースのアクセス制御

ホストベースのアクセス制御ルール (「*22章 ポリシー: ホストベースのアクセス制御の設定*」で説明) は、個別のホストに適用できます。ただし、ホストグループを使用すると、アクセス制御ルールを一度だけ定義してからグループ内のすべてのホストに一貫して適用するため、集中された (場合によっては簡素化された) アクセス制御管理が可能になります。

図22.1 ホストグループとホストベースのアクセス制御



注記

IdM ドメイン内のユーザーおよびホストへのアクセスは明示的に付与する必要がありますが、IdM サーバーは、ドメイン内のすべてのホストに対するアクセスを許可する **allow all** アクセス制御ルールでデフォルトで設定されます。

デフォルトの **allow all** ルールを使用せずに IdM サーバーを作成するには、**--no_hbac_allow** オプションを指定して **ipa-server-install** を実行します。

ルール は最初にアクセス可能なものを定義します。以下の 2 つのタイプのエンティティーがあります。

- IdM ドメイン内のホスト**または** ターゲットホスト。
- ターゲットホストの**サービス**複数のサービスを **サービスグループ** に統合できます。サービスグループは、アクセス制御ルール自体を編集しなくても変更できます。

ルールは、**アクセスできる** ユーザー (IdM ドメインユーザー) も設定します。

ヒント

アクセス制御ルールに個別に追加する代わりに、ユーザーおよびターゲットホストにカテゴリーを使用することができます。サポートされるカテゴリーは **all** のみです。

ホストベースのアクセス制御ルールのエンタイトルは、Kerberos プリンシパルエントリーユーザー、ホスト (マシン)、サービスに従います。ユーザーおよびターゲットホストは、ホストベースのアクセス制御ルールに直接追加できます。ただし、まずホストベースのアクセス制御設定にサービスを追加して、ルールを使用できるようにし、次にアクセス制御ルールに追加する必要があります。

22.2. サービスおよびサービスグループのホストベースのアクセス制御エントリーの作成

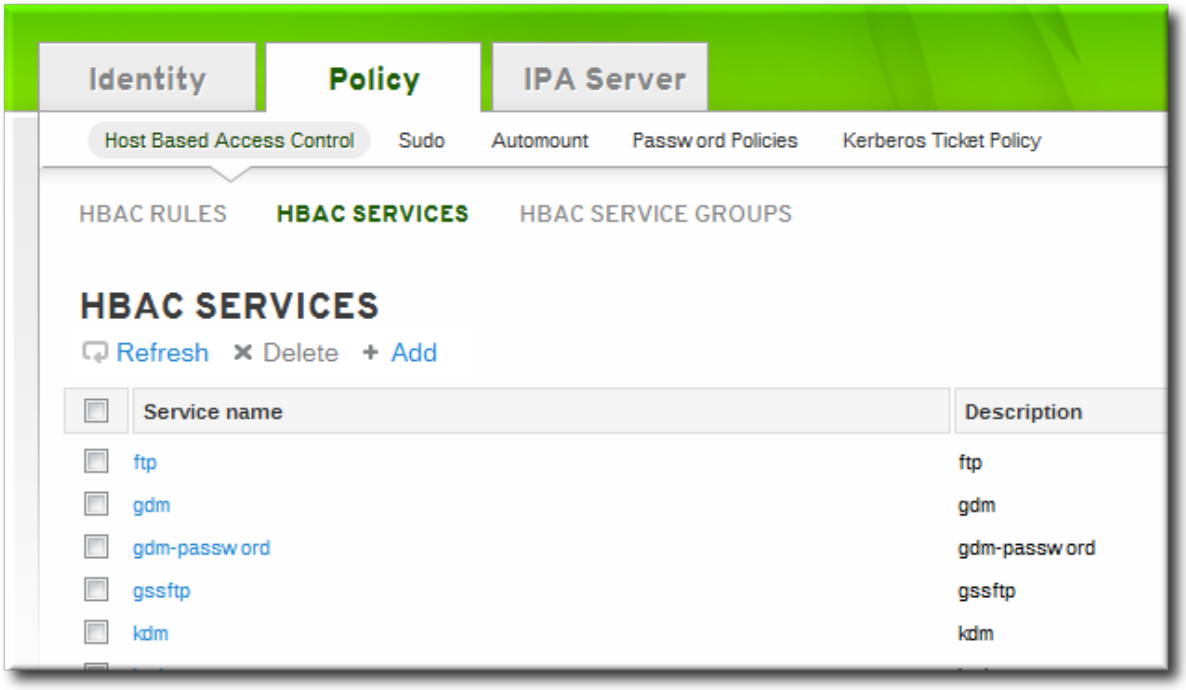
PAM サービスは、IdM のホストベースのアクセス制御 (HBAC) システムと識別できます。ホストベースのアクセス制御で使用されるサービスエントリーは、IdM ドメインへのサービスの追加とは異なります。サービスをドメインに追加すると、他のリソースが利用できる認識リソースになります。ホストベースのアクセス制御設定にドメインリソースを追加すると、管理者はドメインユーザーと、そのサービスにアクセスできるドメインクライアントの制御を適切に制御できます。

一部の共通サービスは HBAC サービスとして設定されているため、ホストベースのアクセス制御ルールで使用できます。さらにサービスを追加でき、管理を簡素化するためにサービスグループに追加することができます。

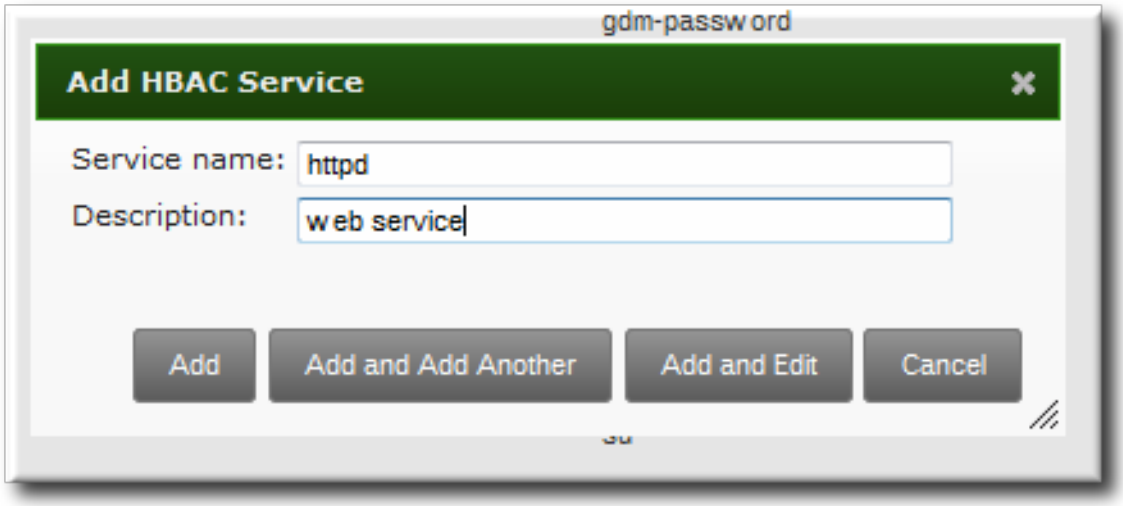
22.2.1. HBAC サービスの追加

22.2.1.1. Web UI での HBAC サービスの追加

1. **Policy** タブをクリックします。
2. **Host-Based Access Control** サブタブをクリックし、**HBAC Services** のリンクを選択します。
3. サービス一覧の上部にある **Add** リンクをクリックします。



4. サービス名と説明を入力します。



5. **Add** ボタンをクリックして新規サービスを保存します。

6. サービスグループが存在する場合は、「Web UI でのサービスグループの追加」の説明に従って、サービスを必要なグループに追加します。

22.2.1.2. コマンドラインでサービスの追加

このサービスは、**hbacsvc-add** コマンドを使用してアクセス制御システムに追加され、サービスを評価するのに PAM が使用する名前でサービスを指定します。

たとえば、これにより **tftp** サービスが追加されます。

```
# ipa hbacsvc-add --desc="TFTP service" tftp
----------------------------
Added HBAC service "tftp"
```

```
-------------------------
Service name: tftp
Description: TFTP service
```

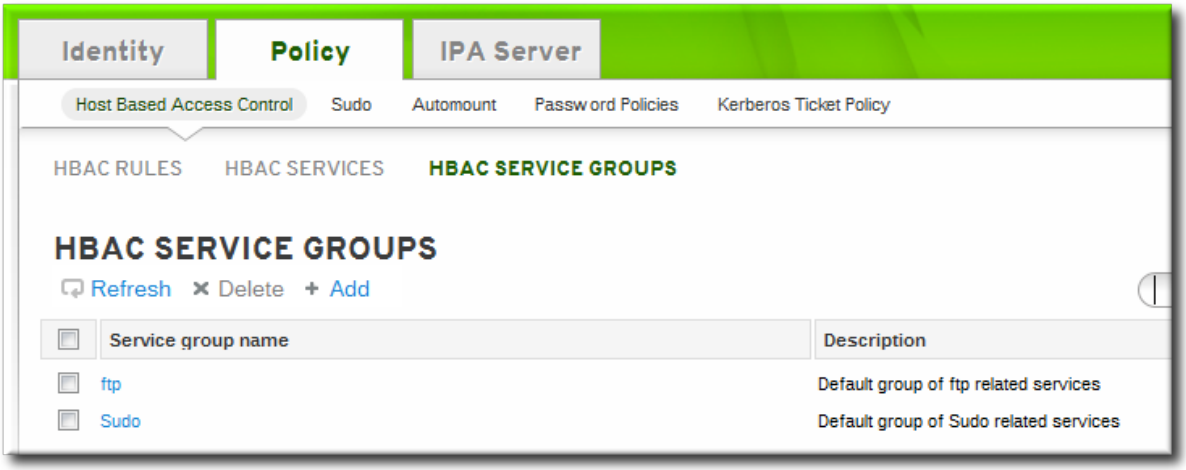
サービスグループが存在する場合は、「コマンドラインでサービスグループの追加」にあるように **hbacsvcgroup-add-member** コマンドを使用してサービスをグループに追加できます。

22.2.2. サービスグループの追加

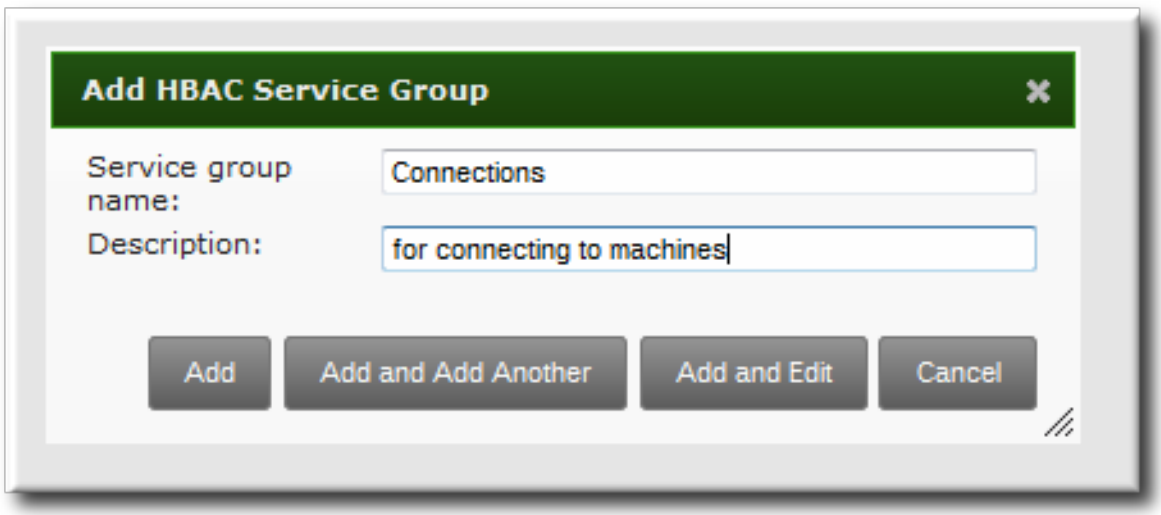
個々のサービスを追加したら、アクセス制御ルールに追加できます。ただし、サービスが多数ある場合は、サービスが変更される際に、アクセス制御ルールに頻繁に更新が必要になる場合があります。また、ID 管理により、サービスのグループをアクセス制御ルールに追加できます。これにより、サービスグループのメンバーはルール自体を編集しなくても変更できるため、アクセス制御の管理が非常に容易になります。

22.2.2.1. Web UI でのサービスグループの追加

1. **Policy** タブをクリックします。
2. **Host-Based Access Control** サブタブをクリックし、**HBAC Services** グループリンクを選択します。
3. サービスグループ一覧の上部にある **Add** リンクをクリックします。

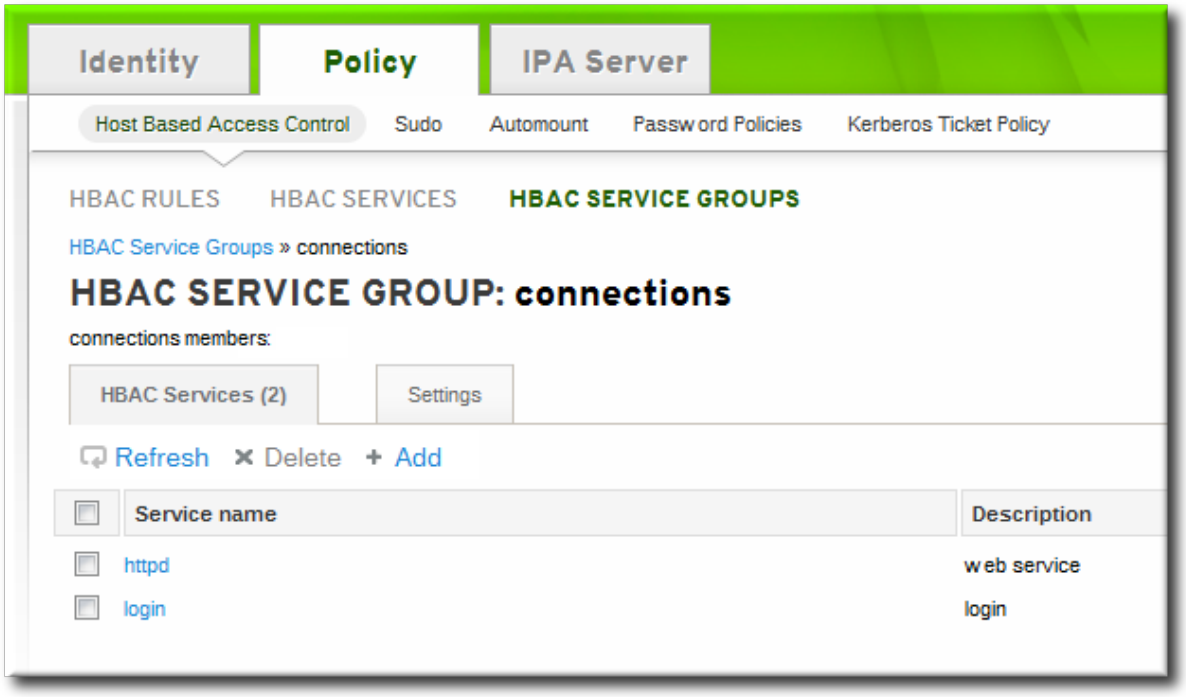


4. サービスグループ名と説明を入力します。

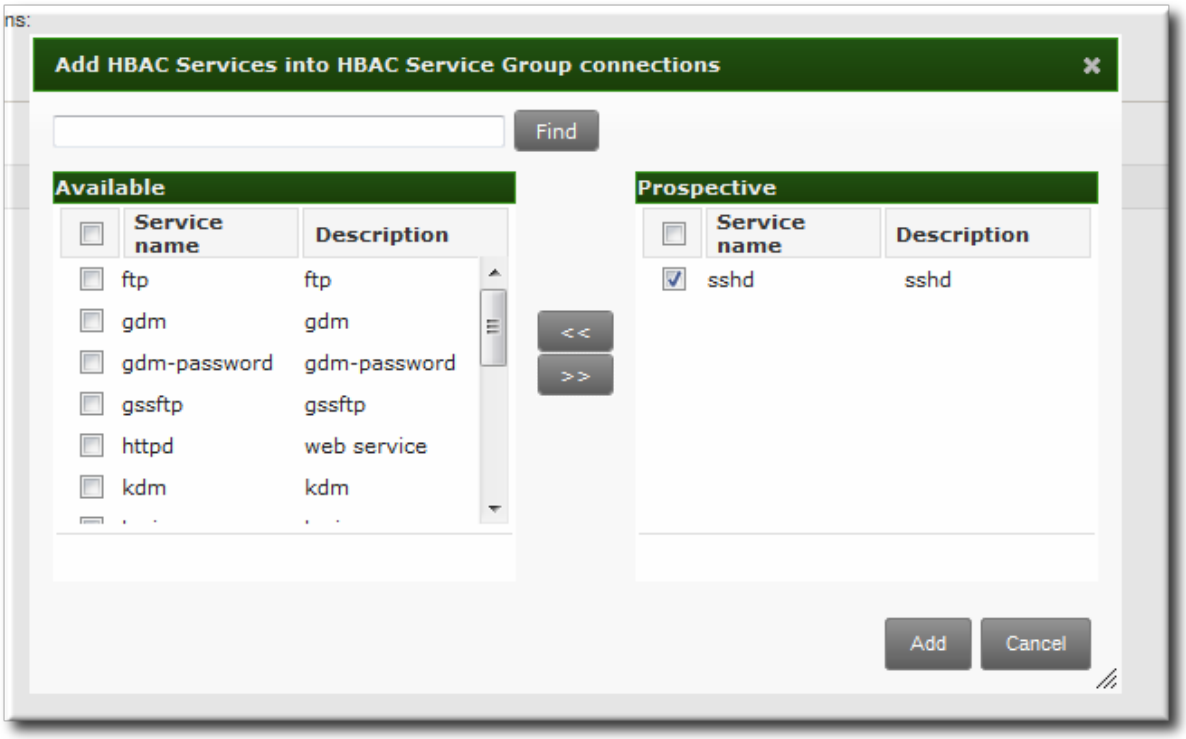


5. **Add and Edit** ボタンをクリックして、サービスグループの設定ページに即座に移動します。

6. **HBAC サービス** タブの上部にある **Add** リンクをクリックします。



7. 追加するサービスの名前の横にあるチェックボックスをクリックし、右向きの矢印 **>>** をクリックして選択ボックスに移動します。

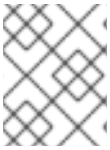


8. **追加** ボタンをクリックしてグループメンバーシップを保存します。

22.2.2.2. コマンドラインでサービスグループの追加

サービスグループエントリーを作成してからサービスを作成し、そのサービスをメンバーとしてサービスグループに追加します。以下に例を示します。

```
[jsmith@server ~]$ kinit admin
[jsmith@server ~]$ ipa hbacsvcgroup-add --desc="login services" login
--------------------------------
Added HBAC service group "login"
--------------------------------
  Service group name: login
  Description: login services

[jsmith@server ~]$ ipa hbacsvc-add --desc="SSHD service" sshd
------------------------
Added HBAC service "sshd"
------------------------
Service name: sshd
Description: SSHD service

[jsmith@server ~]$ ipa hbacsvcgroup-add-member --hbacsvcs=sshd login
  Service group name: login
  Description: login services
------------------------
Number of members added 1
------------------------
```



注記

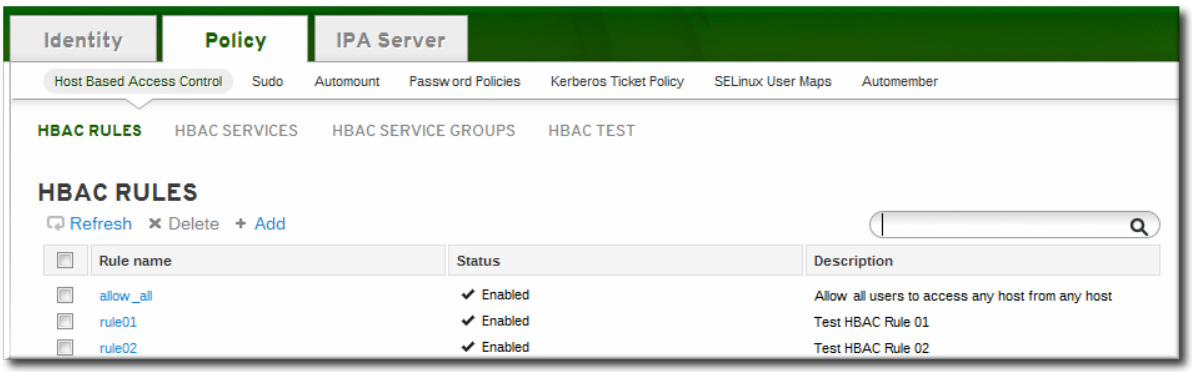
IdM は、2 つのデフォルトサービスグループを定義します。たとえば、sudo サービスの **SUDO** と、**FTP** アクセスを提供するサービスの FTP です。

22.3. ホストベースのアクセス制御ルールの定義

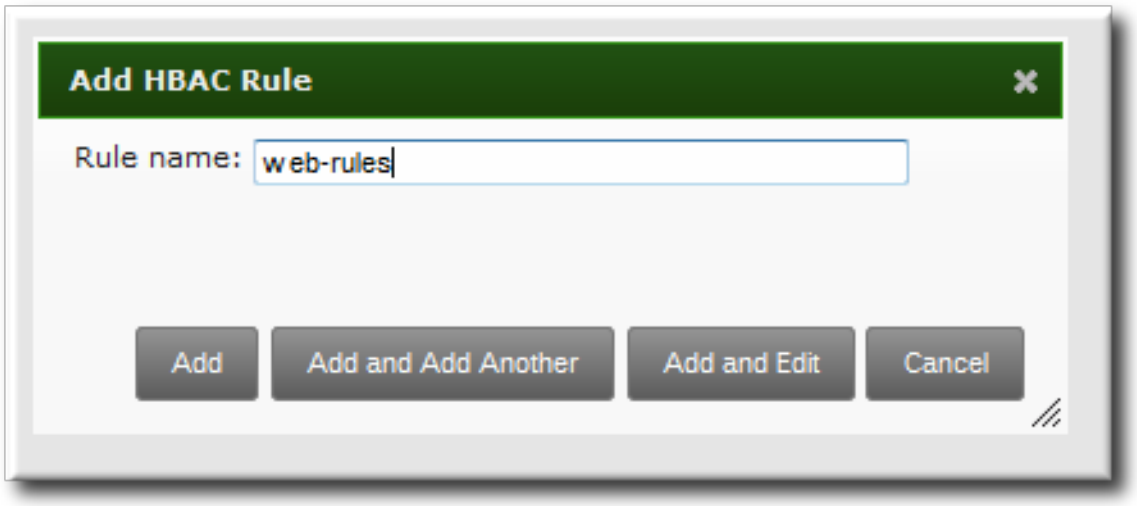
アクセス制御の概要は、**誰** が **何にアクセスできるか** を定義します。**who** は IdM ユーザー、およびホスト (ターゲットホスト)、サービスグループ、または 3 つの組み合わせである**もの**になります。

22.3.1. Web UI でのホストベースのアクセス制御ルールの設定

1. **Policy** タブをクリックします。
2. **Host-Based Access Control** サブタブをクリックし、**HBAC Rules** のリンクを選択します。
3. ホストベースのアクセス制御ルール一覧の上部にある **Add** リンクをクリックします。



4. ルールの名前を入力します。



5. **Add and Edit** ボタンをクリックして、すぐにルールの設定を設定します。

   ルールには設定エリアが多数あります。3 つの基本的な要素は、**誰**がルールを適用するか、どのホストがアクセスを許可するか (ターゲット)、そして任意で、どのサービスがアクセスできるかです。

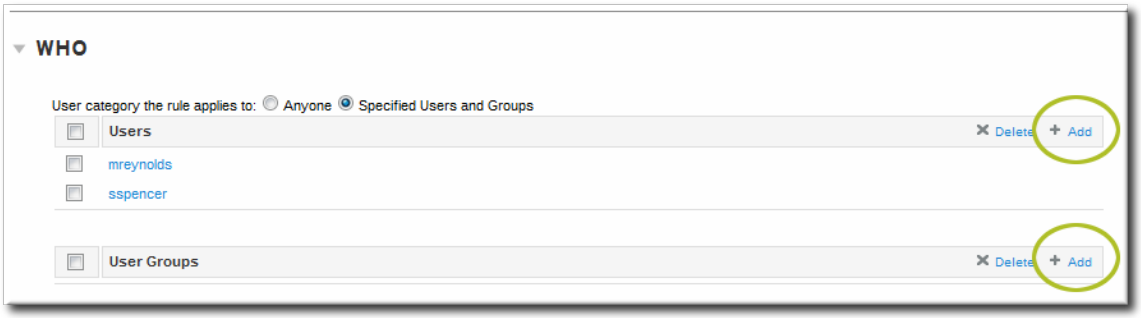
6. **Who** エリアで、アクセス制御ルールが適用されるユーザーまたはユーザーグループを選択します。

   すべての IdM ユーザーにルールを適用する場合は、**Anyone** ラジオボタンを選択します。

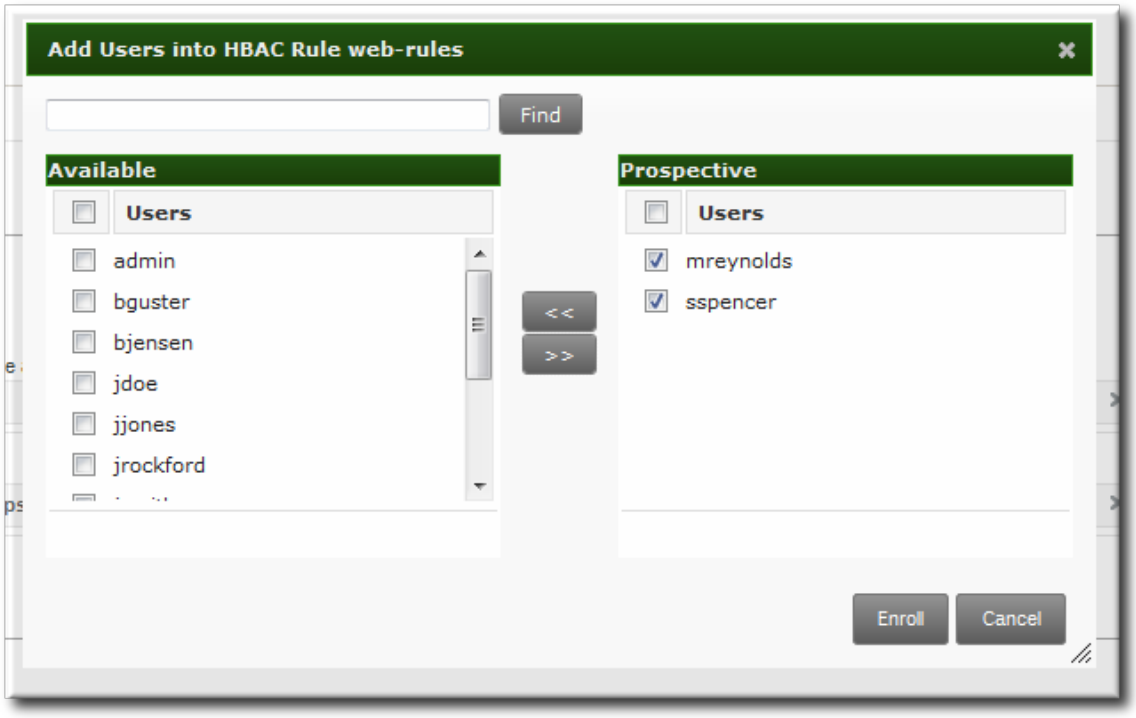
   ルールを特定のユーザーまたはユーザーグループのセットに適用するには、以下を実行します。

   a. **指定したユーザーとグループラジオボタンを** 選択します。

   b. ユーザー一覧の右側にある **+ Add** リンクをクリックします。



   c. ルールに追加するユーザーのチェックボックスをクリックし、右向きの矢印 (**>>**) をクリックして選択ボックスにユーザーを移動します。



d. **追加** をクリックします。

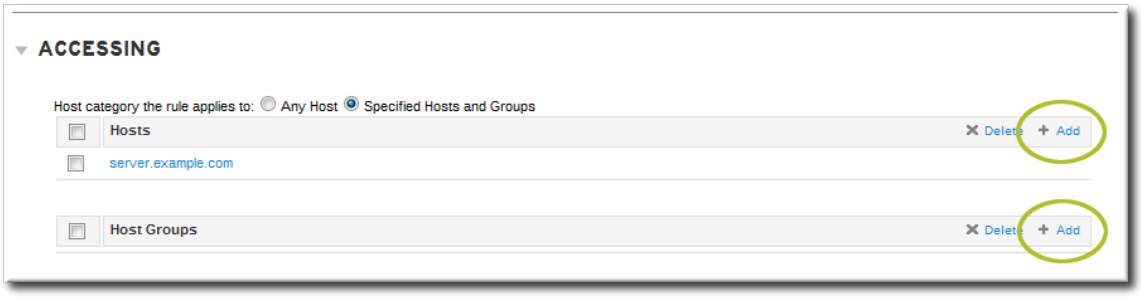
7. **Accessing** エリアで、このアクセス制御ルールでアクセスできるターゲットホストを選択します。

   すべての IdM ホストにルールを適用する場合は、**Any Host** ラジオボタンを選択します。

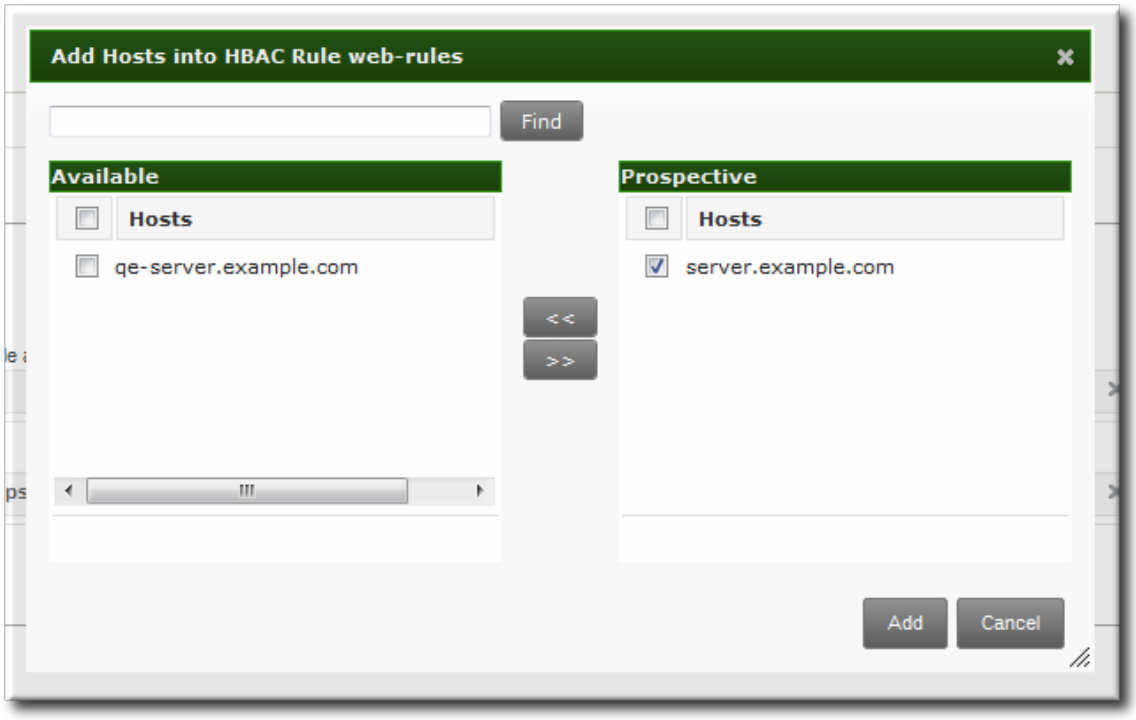
   特定のホストまたはホストのグループセットにルールを適用するには、次のコマンドを実行します。

   a. **指定したホストとグループラジオボタンを** 選択します。

   b. ホスト一覧の右側にある **+ Add** リンクをクリックします。



   c. ルールとともに追加するホストのチェックボックスをクリックし、右矢印ボタン (**>>**) をクリックして、ホストを選択項目のボックスに移動します。



d. **追加** をクリックします。

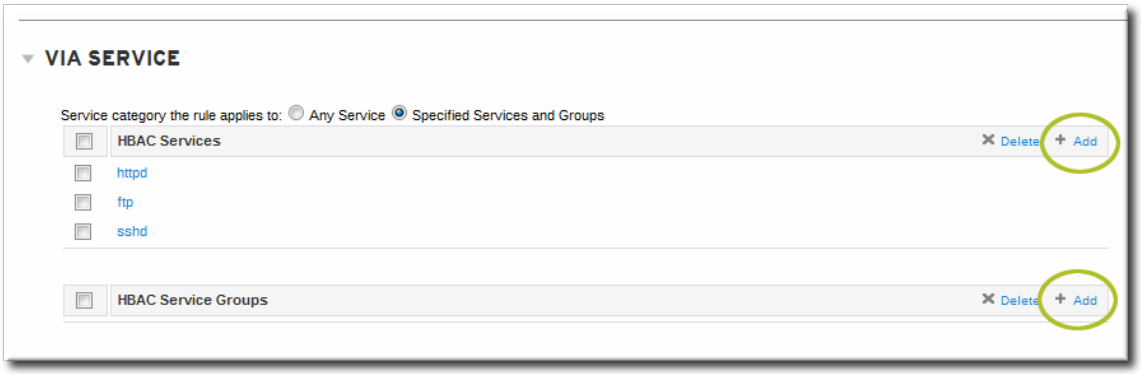
8. **Via Service** エリアで、ユーザーがターゲットマシンにアクセスできるターゲットホストで特定のサービスを選択します。

   すべての IdM ホストにルールを適用する場合は、**Any Service** ラジオボタンを選択します。

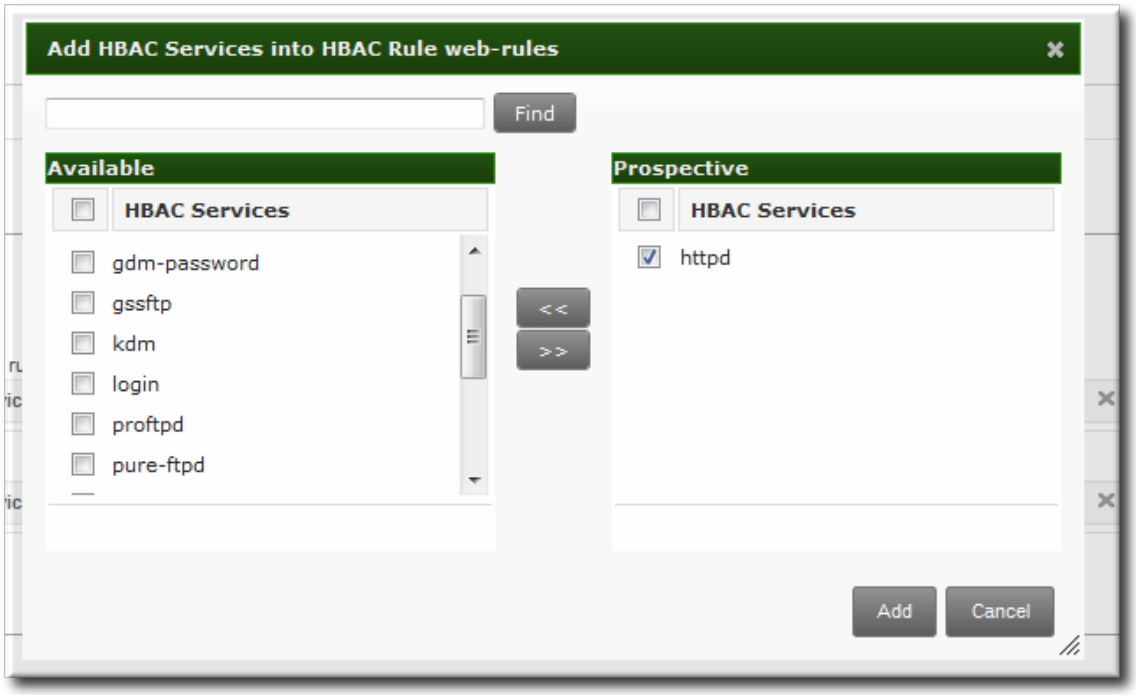
   特定のホストまたはホストのグループセットにルールを適用するには、次のコマンドを実行します。

   a. **指定したサービスとグループ** ラジオボタンを選択します。

   b. コマンド一覧の右側にある **+ Add** リンクをクリックします。



   c. ルールとともに追加するサービスまたはグループのチェックボックスをクリックし、右矢印ボタン (**>>**) をクリックして、サービスを選択項目のボックスに移動します。



d. **追加** をクリックします。

22.3.2. コマンドラインでのホストベースのアクセス制御ルールの設定

アクセス制御ルールは、**hbacrule-*** コマンド (表22.1「ホストベースのアクセス制御コマンドおよびオプション」) を使用して作成されます。最初のステップでは、コンテナーエントリーを作成します。そこから、ユーザー、ホスト、およびサービスをアクセス制御エントリーに追加できます。

すべてのアクセス制御コマンドの基本的な概要は以下のとおりです。

```
$ ipa hbacrule-add* options ruleName
```

ヒント

すべてのユーザーまたはすべてのホストをターゲットとして設定するには、**--usercat=all** などのカテゴリーオプションを使用します。

例22.1 つのホストへのすべてのアクセス権限の付与

1 つの単純なルールの 1 つは、すべてのユーザーが 1 台のサーバーへのアクセスを許可することです。最初のコマンドはエントリーを作成し、カテゴリーのオプションを使用して全ユーザーを適用します。

```
$ ipa hbacrule-add --usercat=all allGroup
---------------------------
Added HBAC rule "allGroup"
---------------------------
  Rule name: allGroup
  User category: all
  Enabled: TRUE
```

2 番目のルールは、**hbacrule-add-host** コマンドを使用してターゲットホストを追加します。

```
$ ipa hbacrule-add-host --hosts=server.example.com allGroup
  Rule name: allGroup
  User category: all
  Enabled: TRUE
  Successful hosts/hostgroups:
    member host: server.example.com
-------------------------
Number of members added 1
-------------------------
```

例22.2 サービスへの単一ユーザーのコントロールの追加

もう1つのアクセス制御方法は、ユーザーがターゲットホストにアクセスできるサービスを指定することです。

まず、すべてのユーザーがすべてのマシンにアクセスできるようにするには、ホストとターゲットの両方としてすべてのホストを追加する必要があります。これは、カテゴリーのオプションを使用して行うことができます。

```
$ ipa hbacrule-add --hostcat=all sshd-jsmith
```

アクセス制御ルールは特定のユーザーに適用されるため、**hbacrule-add-user** コマンドを使用してルールに追加されます。

```
$ ipa hbacrule-add-user --users=jsmith sshd-jsmith
```

次に、サービスはアクセス制御ルールに追加されます。(この **hbacsvc-add** コマンドを使用して、このサービスはすでにアクセス制御システムに追加されているはずです。) これは、ユーザーがマシンへの接続に使用できるサービスです。

```
$ ipa hbacrule-add-service --hbacsvcs=sshd sshd-jsmith
```

例22.3 サービスグループのルールへの追加

1つのサービスをルールに追加できますが、サービスグループ全体を追加することもできます。1つのサービスと同様に、この **hbacrule-add-service** コマンドはグループ名を指定する **--hbacsvcgroups** オプションとともにのみ使用されます。

```
$ ipa hbacrule-add-service --hbacsvcgroups=login loginRule
```

表22.1 ホストベースのアクセス制御コマンドおよびオプション

コマンド	説明	引数	ソースまたはターゲットエントリー

コマンド	説明	引数	ソースまたはターゲットエントリー
hbacrule-add	ホストベースのアクセス制御ルールを新たに追加します。	• --usercat=all: ルールをすべてのユーザーに適用します。 • --hostcat=all。すべてのホストを許可されたターゲットサーバーとして設定します。 • --servicecat=all。設定済みのサービスをすべて許可されたターゲットサービスとして設定します。 • **ruleName**。新しいルールに必要な一意の識別子です。	
hbacrule-add-host	ターゲットホストをアクセス制御ルールに追加します。ターゲットホストは、ドメイン内 **の** 他のサーバーおよびユーザーからアクセスできます。	• --hosts。個々のサーバーまたはコマンド区切りのサーバーを、許可されたターゲットサーバーとして追加します。 • --hostgroups。ルールにホストグループを追加し、ホストグループ内のすべてのホストは、許可されているターゲットサーバーです。 • **RuleName**: ターゲットサーバーを追加するルールです。	ターゲット

コマンド	説明	引数	ソースまたはターゲットエントリー
hbacrule-add-service	ルールにサービスタイプを追加します。	• --hbacsvcs: 個々のサービス種別またはサービス種別のコンマ区切りリストを、許可されたターゲットサービスとして追加します。 • --hbacsvcgroups: サービスグループをルールに追加し、サービスグループ内のすべてのサービスが、許可されるターゲットサービスになります。 • **RuleName**: ターゲットサービスを追加するルールです。	ターゲット
hbacrule-add-user	アクセス制御ルールにユーザーを追加します。その後、ユーザーはドメイン内の許可されたターゲットホストまたはサービスにアクセスできます。	• --users。個々のユーザーまたはコマンドで区切られたユーザーのリストをルールに追加します。 • --groups。ルールにユーザーグループを追加します。グループ内のすべてのユーザー。 • **RuleName**: ユーザーを追加するルールです。	ソース

コマンド	説明	引数	ソースまたはターゲットエントリー
hbacrule-disable | hbacrule-enable	ホストベースのアクセス制御ルールを無効にするか、または有効にします。ルールは、動作を評価する必要がある場合にルールを無効にすることができます (トラブルシューティングまたは新しいルールをテストする場合)。	**RuleName**: 無効にまたは有効にするルールです。	

22.4. ホストベースのアクセス制御ルールのテスト

ホストベースのアクセス制御を効果的に実装することは、すべてのホストが適切に設定され、アクセスをユーザーとサービスに適切に適用する必要があるため、複雑になります。

この **hbactest** コマンドは、異なるホストベースのアクセス制御シナリオをテストして、ルールが期待どおりに機能するようにできます。

注記

この **hbactest** コマンドは、信頼された Active Directory ユーザーでは機能しません。Active Directory のユーザー/グループの関連付けは、ユーザーがログインし、これらのデータは IdM LDAP ディレクトリーに保存されないため、動的に決定されます。この場合、**hbactest** コマンドは、アクセス制御ルールが適用される仕組みをチェックするためにグループメンバーシップを解決できません。

22.4.1. ホストベースのアクセス制御設定の制限

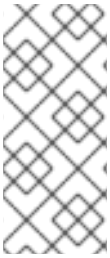
アクセス制御設定は、承認の失敗を防ぐために、実装前に必ずテストする必要があります。

ホストベースのアクセス制御ルールは、ホスト、サービス、DNS ルックアップ、およびユーザー間で多くの対話に依存します。要素の設定が間違っている場合、ルールは予期せぬ挙動で動作します。

Identity Management には、アクセス制御ルールが、定義されたシナリオでアクセスをテストすることで予想される方法で動作していることを検証するテストツールが含まれています。このテストが便利な状況は複数あります。

- 新しいルールは、実装前にテストする必要があります。
- 既存のルールには問題があり、テストツールはどのルールを適切に動作させるかを特定できます。
- 既存のルールのサブセットをテストして、それらの実行内容を確認することができます。

22.4.2. ホストベースのアクセス制御 (CLI ベース) のテストシナリオ



注記

この **hbactest** コマンドは、信頼された Active Directory ユーザーでは機能しません。Active Directory のユーザー/グループの関連付けは、ユーザーがログインし、これらのデータは IdM LDAP ディレクトリーに保存されないため、動的に決定されます。この場合、**hbactest** コマンドは、アクセス制御ルールが適用される仕組みをチェックするためにグループメンバーシップを解決できません。

この **hbactest** コマンドは、非常に特殊な状況でホストベースのアクセス制御ルールを設定していました。テスト実行では、以下を定義します。

- ユーザーは、そのユーザー (**--user**) のルールパフォーマンスをテストするために操作を実行します。
- ログインクライアント Y (**--service**) の使用
- ホスト Z (**--host**) をターゲットにするには、次のコマンドを実行します。
- テストするルール (**--rules**): これが使用されていない場合は、有効なすべてのルールがテストされます。
- **オプション**: **hbactest** は一致するルール、一致しないルール、または無効なルールに関する詳細情報を返します。この詳細なルール出力は **--nodetail** を使用して無効にすることができます。そのため、テストが単に実行され、アクセスが付与されたかどうかを返します。

注記

この **hbactest** スクリプトは、実際にはターゲットホストに接続しません。代わりに、IdM データベース内のルールを使用して、SSSD クライアントが IdM サーバーに接続しているかのように、特定の状況でルールがどのように適用されるかをシミュレートします。

さらに、指定の情報と設定に基づいてシミュレーションされたテストを実行しますが、実際にはターゲットホストに対してサービス要求を試行する訳ではありません。

例22.4 すべてのアクティブなルールのテスト

最も基本的なコマンドは、すべてのアクティブなルールをチェックします。特定の接続シナリオが必要なため、ユーザー、ログインサービス、およびターゲットホストが提供され、テストツールは接続をチェックします。

```
[jsmith@server ~]$ kinit admin
[jsmith@server ~]$ ipa  hbactest --user=jsmith --host=target.example.com --service=ssh
--------------------
Access granted: True
--------------------
Matched rules: allow_all
Matched rules: sshd-jsmith
Matched rules: web-rules
Not matched rules: allGroup
```

例22.5 特定のルールのテスト

特定のルール (または複数のルール) を確認することができます。

```
[jsmith@server ~]$ kinit admin
[jsmith@server ~]$ ipa hbactest --user=jsmith --host=target.example.com --service=ssh --
rules=myrule
---------------------
Access granted: True
---------------------
  notmatched: myrule
```

例22.6 テスト固有のルールとすべてが有効化

この **--rules** オプションは、テストする特定のルールを一覧表示します。ただし、ドメインで有効なすべてのルールに対して、指定したルールをテストすると便利です。これは、**--enabled** オプションを追加することで実行できます。これには、指定されていない有効なルールと、指定したルールを含みます。

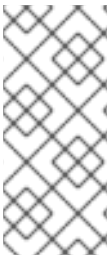
```
[jsmith@server ~]$ kinit admin
[jsmith@server ~]$ ipa hbactest --user=jsmith --host=target.example.com --service=ssh --
rules=myrule --enabled
---------------------
Access granted: True
---------------------
  matched: my-second-rule
  notmatched: my-third-rule
  matched: myrule
  matched: allow_all
```

--disabled オプションを使用すると、**無効** なルールと同様の比較を実行できます。この **--rules** オプションを使用すると、指定したルールと、無効なすべてのルールがチェックされます。この **--disabled** オプションを使用すると、無効にしたすべてのルールがチェックされます。

22.4.3. UI でのホストベースのアクセス制御ルールのテスト

「ホストベースのアクセス制御設定の制限」の詳細として、ホストベースのアクセス制御ルールの設定が間違っていると、ユーザーまたはサービスがリモートホストへの接続を試みると予測できない動作が発生する可能性があります。

ホストベースのアクセス制御のテストは、ルールがデプロイされる前に予想通りに実行されたことを確認するか、またはすでにアクティブなルールのトラブルシューティングを行うのに役立ちます。



注記

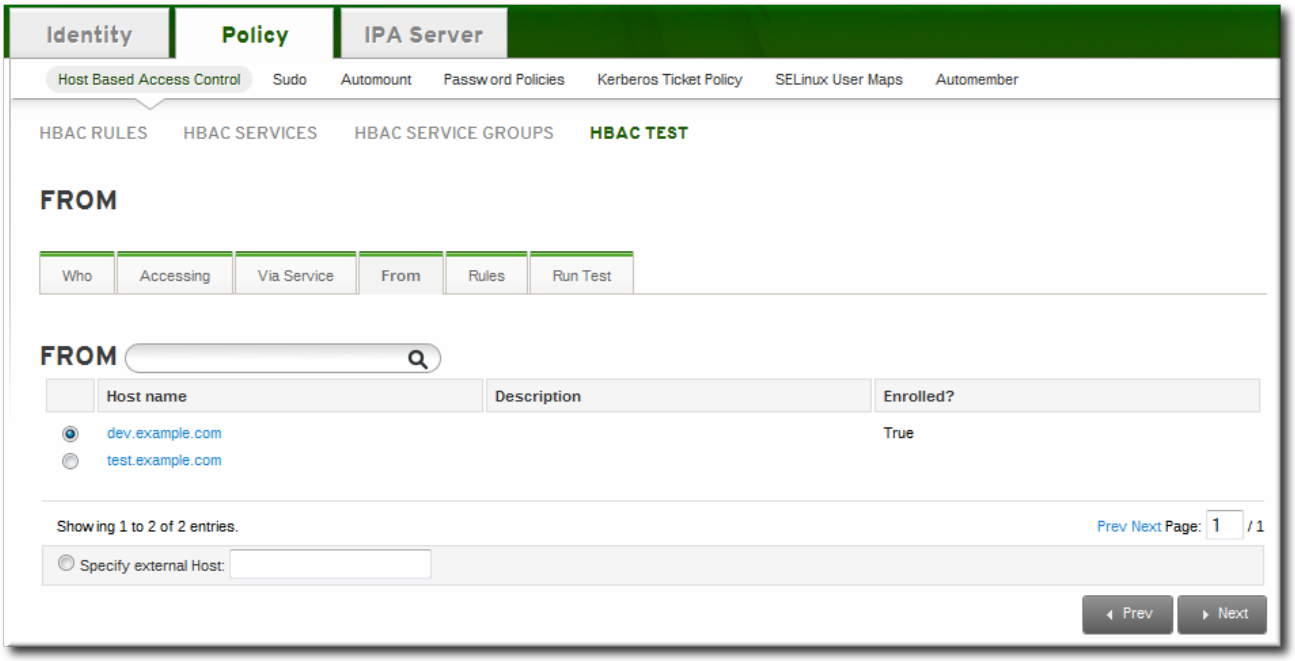
この **hbactest** コマンドは、信頼された Active Directory ユーザーでは機能しません。Active Directory のユーザー/グループの関連付けは、ユーザーがログインし、これらのデータは IdM LDAP ディレクトリーに保存されないため、動的に決定されます。この場合、**hbactest** コマンドは、アクセス制御ルールが適用される仕組みをチェックするためにグループメンバーシップを解決できません。

ホストベースのアクセス制御ルールの性質上、テストは特定の基準セットを定義し、検証する必要があります。テスト実行えは以下を定義します。

- ユーザーは、そのユーザー (**Who**) のルールパフォーマンスをテストするために操作を実行します。
- ホスト Z (**アクセス**) をターゲットにするには、次のコマンドを実行します。
- ログインクライアント Y (**Via Service**) の使用
- テストするルール。これが使用されていない場合、有効なすべてのルールはテストされます (**Rules**) になります。

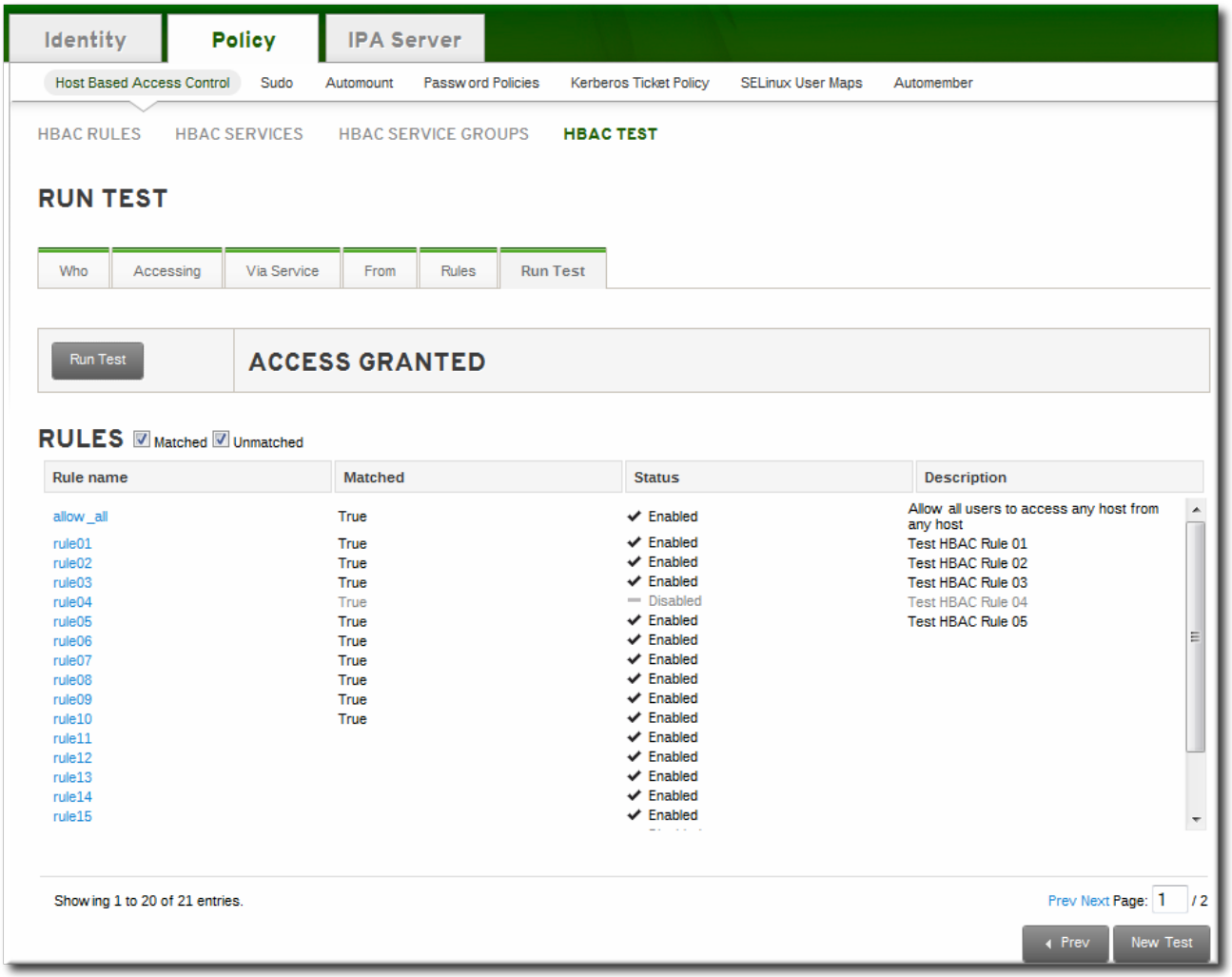
テスト環境は、**Policy** の **Host Based Access Control** タブの **HBAC TEST** ページで定義されます。設定ステップごとに一連のタブが設定されます。

図22.2 HBAC テストを設定する From タブ



環境が定義されたら、**Run Test** ページのボタンをクリックすると、テストが実行されます。その結果、アクセスがユーザーに許可または拒否されたかどうかを明確に示し、指定のパラメーターに一致するルールを経由します。

図22.3 HBAC テスト結果



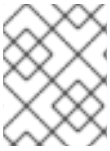
注記

一部のパラメーターを変更し、その他の結果を確認するには、テスト結果ページの下部にある **New Test** ボタンをクリックします。このボタンを選択しないと、フォームはリセットされないため、テスト設定が変更されても新しいテストは実行されません。

第23章 ポリシー: グループポリシーオブジェクトアクセス制御

グループポリシーは Microsoft Windows の機能の 1 つで、AD 環境におけるユーザーおよびコンピューターのポリシーを管理者が 1 か所で管理できるようにします。*グループポリシーオブジェクト* (GPO) は、ポリシー設定 (名前と値のペアなど) の集合で、これらはドメインコントローラー (DC) に保存され、コンピューターやユーザーなどのポリシーターゲットに適用されます。

AD 環境におけるコンピューターベースのアクセス制御の管理には、Windows *ログオン権限* に関連する GPO ポリシー設定が一般的に使用されます。SSSD は、ホストシステムおよび AD ユーザーに適用される GPO を取得できます。取得した GPO 設定に基づいて、ユーザーが特定のホストにログオンできるかどうかを判断します。したがって、SSSD が提供する GPO ベースのアクセス制御を使用すると、管理者は Red Hat Enterprise Linux と Windows クライアントの両方が AD DC に集中的に許可されるログインポリシーを定義できます。



注記

SSSD は、コンピューターベースのアクセス制御にのみ GPO の使用を許可します。現在、その他の GPO 関連のアクセス制御オプションはサポートされていません。

警告

SSSD は、サイト、ドメイン、または AD 組織ユニット (OU) 全体に適用されるルールのみを処理することに注意してください。SSSD 対応 GPO ベースのアクセス制御を特定のマシンに適用する場合は、AD ドメインで新しい OU を作成し、マシンを OU に移動してから GPO をこの OU にリンクできます。

23.1. GPO ベースのアクセス制御の設定

GPO ベースのアクセス制御は **/etc/sssd/sssd.conf** ファイルで設定できます。***ad_gpo_access_control*** オプションは、GPO ベースのアクセス制御を実行するモードを指定します。以下の値を使用できます。

ad_gpo_access_control = permissive

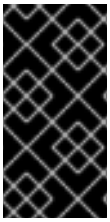
permissive の場合は、GPO ベースのアクセス制御は評価されますが、強制されません。**syslog** メッセージは、アクセスが拒否される度に復元されます。これはデフォルトの設定です。

ad_gpo_access_control = enforcing

enforcing の場合は、GPO ベースのアクセス制御は評価され、強制されます。

ad_gpo_access_control = disabled

disabled の場合は、GPO ベースのアクセス制御は評価も強制もされません。



重要

GPO ベースのアクセス制御を使用し、Enforcing モードに ***ad_gpo_access_control*** を設定する前に、***ad_gpo_access_control*** を Permissive モードに設定して、ログを調べることが推奨されます。**syslog** メッセージを見直すことで現行の GPO 設定をテスト、調節してからその後で enforcing モードに設定することができます。

GPO ベースのアクセス制御に関連する以下のパラメーターも **sssd.conf** ファイルで指定することができます。

- ***ad_gpo_map_**オプションと ***ad_gpo_default_right*** オプションは、特定の Windows ログイン権限にマッピングされる PAM サービスを設定します。
- ***ad_gpo_cache_timeout*** オプションでは、後続のアクセス制御リクエストが DC から新たに取得するのではなく、キャッシュに保存されているファイルを再利用可能な間隔を指定します。

使用できる GPO パラメーターの詳細なリストと、その説明およびデフォルト値は、sssd-ad(5) の man ページを参照してください。

第24章 ポリシー: SELINUX ユーザーマップの定義

Security-enhanced Linux (SELinux) は、システムユーザーがどのプロセス、ファイル、ディレクトリー、およびシステム設定にアクセスできるかを指定するルールを設定します。システム管理者とシステムアプリケーションの両方が、ユーザーアクセス (他のアプリケーションからのアクセスも) を制限または許可するす **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。

24.1. IDENTITY MANAGEMENT、SELINUX、およびユーザーのマッピング

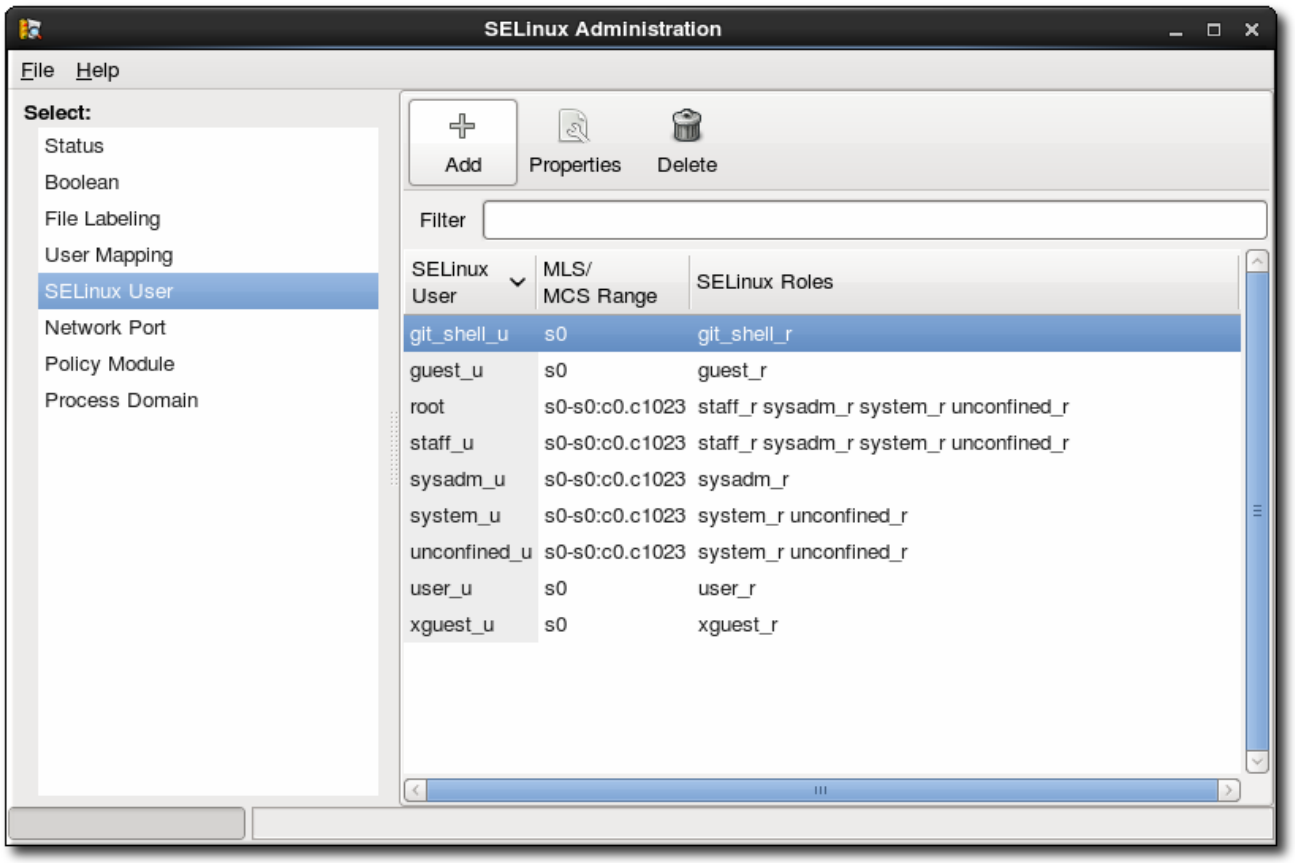
注記

Identity Management は、システムで SELinux コンテキストを作成または変更しません。代わりに、既存のコンテキストをベースとして使用し、ドメイン内の IdM ユーザーをシステム上の SELinux ユーザーにマッピングします。

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ユーザー、プロセス、アプリケーション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。これらの対話ルール (**contexts**) は、システム上の異なるオブジェクトのデータおよび動作の特性を確認し、特定のオブジェクトのセキュリティーへの影響に基づいて **ポリシー** と呼ばれるルールを設定します。これは、主に、データの重大度や評価動作を考慮せずに、ファイルの所有権とユーザーアイデンティティに関して関係する高レベルのアクセス制御とは対照的です。システム上のすべてのリソース (ユーザー、アプリケーション、ファイル、プロセス) にはコンテキストが割り当てられます。

システムユーザーが、SELinux **ロール** に関連付けられます。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 (MCS) の両方が割り当てられます。MLS/MCS コンテキストは、システムでアクセスできるプロセス、ファイル、および操作に対して、ユーザーを **制限** します。

図24.1 SELinux Manager の SELinux ユーザー



これは、Red Hat Enterprise Linux 6 Security-Enhanced Linux で詳細に説明されています。

SELinux のユーザーとポリシーは、ネットワークレベルではなく、システムレベルで機能します。つまり、SELinux ユーザーは、各システムで個別に設定されます。これは多くの状況で許容されますが、SELinux には定義されているシステムユーザーがあり、SELinux 対応のサービスは独自のポリシーを定義します。ローカルリソースにアクセスするリモートユーザーおよびシステムを処理する際に問題があります。リモートユーザーとサービスは、実際の SELinux ユーザーとロールについて多くの情報なしに、デフォルトのゲストコンテキストにシャッフルされます。

これは、Identity Management が ID ドメインをローカルの SELinux サービスにクリーンに統合する方法です。Identity Management は、IdM ユーザーを、**ホストごと** に設定した SELinux ロールに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。

- ホストが IT 環境に追加されるか、またはローカルシステムを編集しなくても、ユーザーが追加、削除、または変更されると、SELinux ユーザーは自動的に更新されます。

- SELinux ポリシーは、IdM ホストベースのアクセス制御ルールのようなドメイン全体のセキュリティ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 つの部分で構成されます。これらは、2 つの異なる関係を定義します。まず、特定のホスト (ローカルまたはターゲットシステム) の SELinux ユーザーのマップを定義します。次に、SELinux ユーザーと IdM

ユーザーのマップを定義します。

この組み合わせにより、管理者はアクセスするホストによって、同一の IdM ユーザーに異なる SELinux ユーザーを設定することが可能になります。

SELinux ユーザーマップは、System Security Services Daemon (SSSD) および **pam_selinux** モジュールと機能します。リモートユーザーがマシンにログインを試みると、SSSD はその IdM ID プロバイダーをチェックして、SELinux マップを含むユーザー情報を収集します。すると PAM モジュールはこのユーザーを処理し、適切な SELinux ユーザーコンテキストを割り当てます。

SELinux 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。これらは、IdM ドメインのすべてのホストに設定されている SELinux ユーザーです。デフォルトでは、共通の SELinux ユーザーを 5 つ定義しています。

- 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

IdM サーバー設定で、各 SELinux ユーザーはユーザー名とその MLS/MCS 範囲である **SELinux_username:MLS[:MCS]** の両方で設定されています。この形式は、マップを設定する際に SELinux ユーザーを識別するために使用されます。

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。

ホストベースのアクセス制御ルールを使用すると、追加のセキュリティー層を使用できます。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。(「22章 *ポリシー: 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。

24.2. SELINUX ユーザーマップの順序とデフォルト値の設定

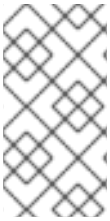
SELinux ユーザーマップは、名前が示すように、SELinux ユーザーと IdM ユーザー間の関連付けを作成します。関連が確立される前に、IdM サーバーは、管理するシステムの基本的な SELinux ユーザー設定を認識する必要があります。

利用可能な **システム** の SELinux ユーザーマップは、IdM サーバー設定の一部です。これは、SELinux ユーザーの最も限定的な一覧です。SELinux ユーザーエントリには、以下の形式が使われます。

SELinux_username:MLS[:MCS]

個別のユーザーエントリーは、ドル記号 ($) で区切ります。

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 マップ一覧すべてのユーザーの 1 人) がマッピングされていない IdM ユーザーエントリーを使用するよう設定します。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。



注記

この設定は、利用可能なシステム SELinux ユーザーのマップ順序を定義します。これは、IdM ユーザーの SELinux ポリシーを定義しません。「SELinux ユーザーおよび IdM ユーザーのマッピング」のように、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。

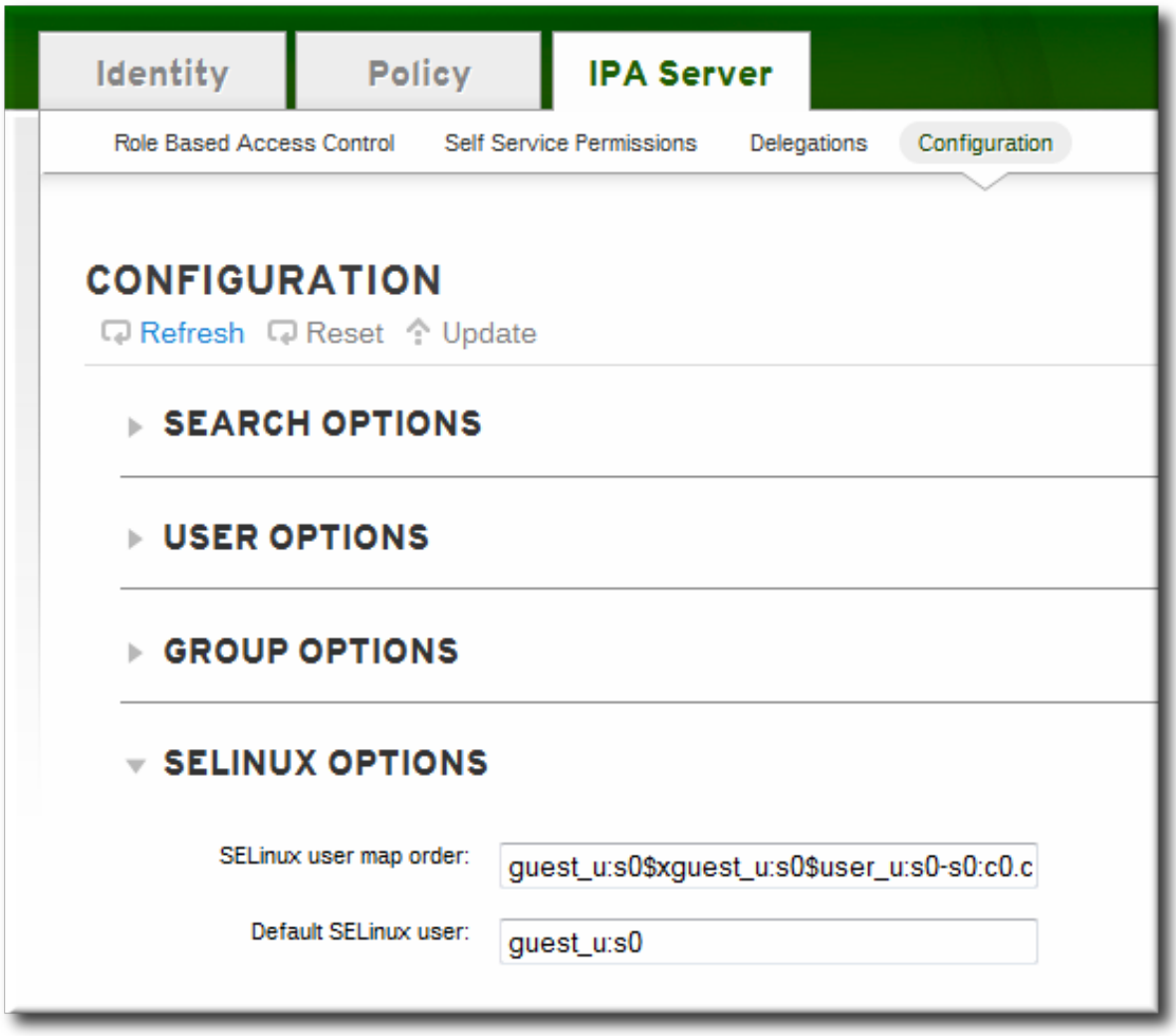
24.2.1. Web UI での設定

1. トップメニューで **IPA Server** メインタブをクリックし、**Configuration** サブタブをクリックします。
2. **SELINUX OPTIONS** まで、サーバー設定エリア一覧でスクロールダウンします。
3. SELinux のユーザー設定を設定します。

   編集できるエリアは、SELinux ユーザーの優先順位付きリストと、マッピングされていない IdM ユーザーに使用するデフォルトの SELinux ユーザーの一覧です。

   SELinux ユーザーマップの順序 は、ローカルの Linux システムで定義されている SELinux ユーザーの一覧を提供します。これは、マッピングルールの設定に利用できます。これは、優先が低い順です。各 SELinux ユーザーの形式は、**SELinux_user:MLS** です。

   Default SELinux user フィールドは、**マッピングされていない** IdM ユーザーに使用する SELinux ユーザーを設定します。



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。

24.2.2. コマンドラインでの設定

SELinux マッピングルールを作成する前に、マッピングできる SELinux ユーザーの定義済みリストと汎用リストが必要です。これは、IdM サーバー設定で設定されます。

```
[jsmith@server ~]$ ipa config-show
...
SELinux user map order: guest_u:s0$xguest_u:s0$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 のユーザー設定は、**config-mod** コマンドを使用して編集できます。

例24.1 SELinux ユーザーの一覧

SELinux ユーザーの完全なリストは、**--ipaselinuxusermaporder** オプションで渡されます。この一覧は、最も制限のあるユーザーから優先順位を設定します。

SELinux ユーザーエントリには、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。

以下に例を示します。

```
[jsmith@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一覧に含めないと、編集操作は失敗します。同様に、デフォルトを編集する際は、SELinux マップ一覧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一覧を先に更新する必要があります。

例24.2 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 ユーザーは、マッピングされていない IdM ユーザーエントリーに使用するフォールバックユーザーを設定します。デフォルトでは、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 **unconfined_u** になります。

このデフォルトユーザーは、**--ipaselinuxusermapdefault** で変更できます。たとえば、以下のようになります。

```
[jsmith@server ~]$ ipa config-mod --ipaselinuxusermapdefault="guest_u:s0"
```

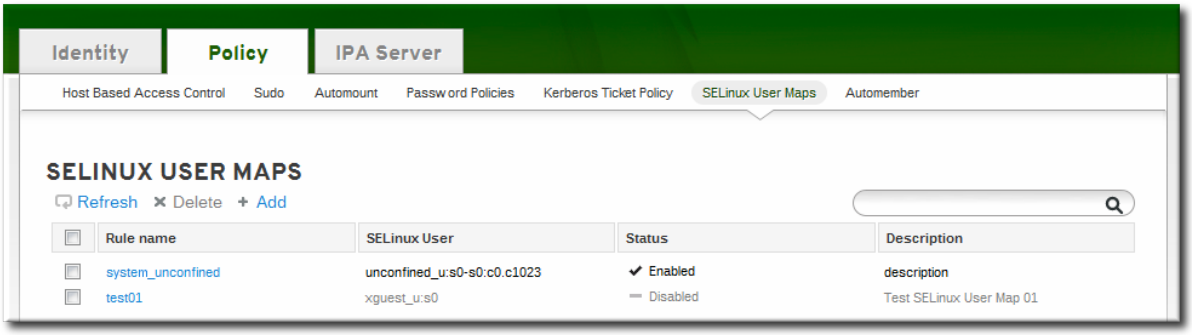
24.3. SELINUX ユーザーおよび IDM ユーザーのマッピング

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 つの部分で構成されています。この IdM ユーザー/ホストのペアは、以下のいずれかの方法で定義できます。明示的なホスト上の明示的なユーザーまたはホスト (ユーザーおよびグループ) に設定するか、ホストベースのアクセス制御ルールを使って定義できます。

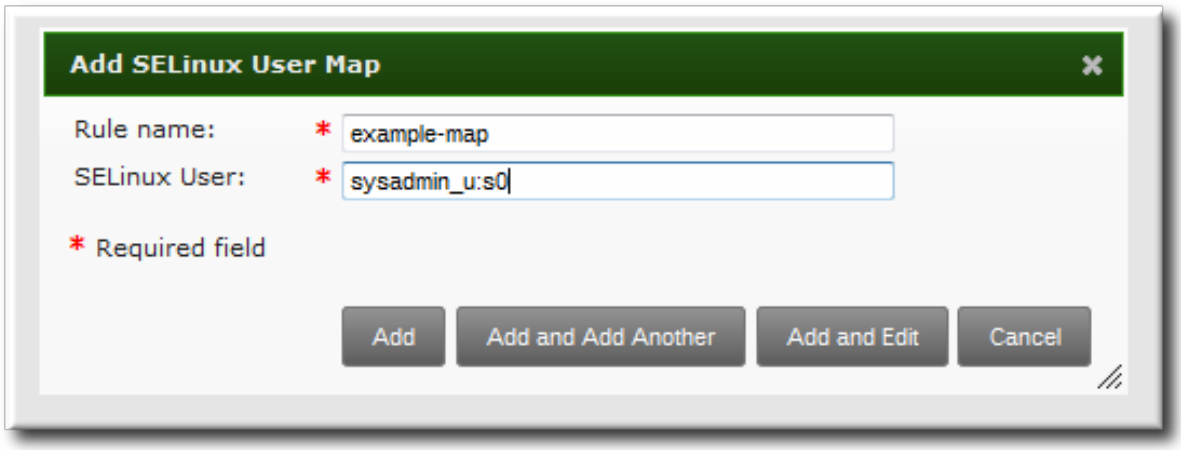
24.3.1. Web UI での設定

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. マップの名前と、**IdM サーバー設定に表示されているように** SELinux ユーザーを入力します。SELinux ユーザーの形式は、**SELinux_username:MLS[:MCS]** です。

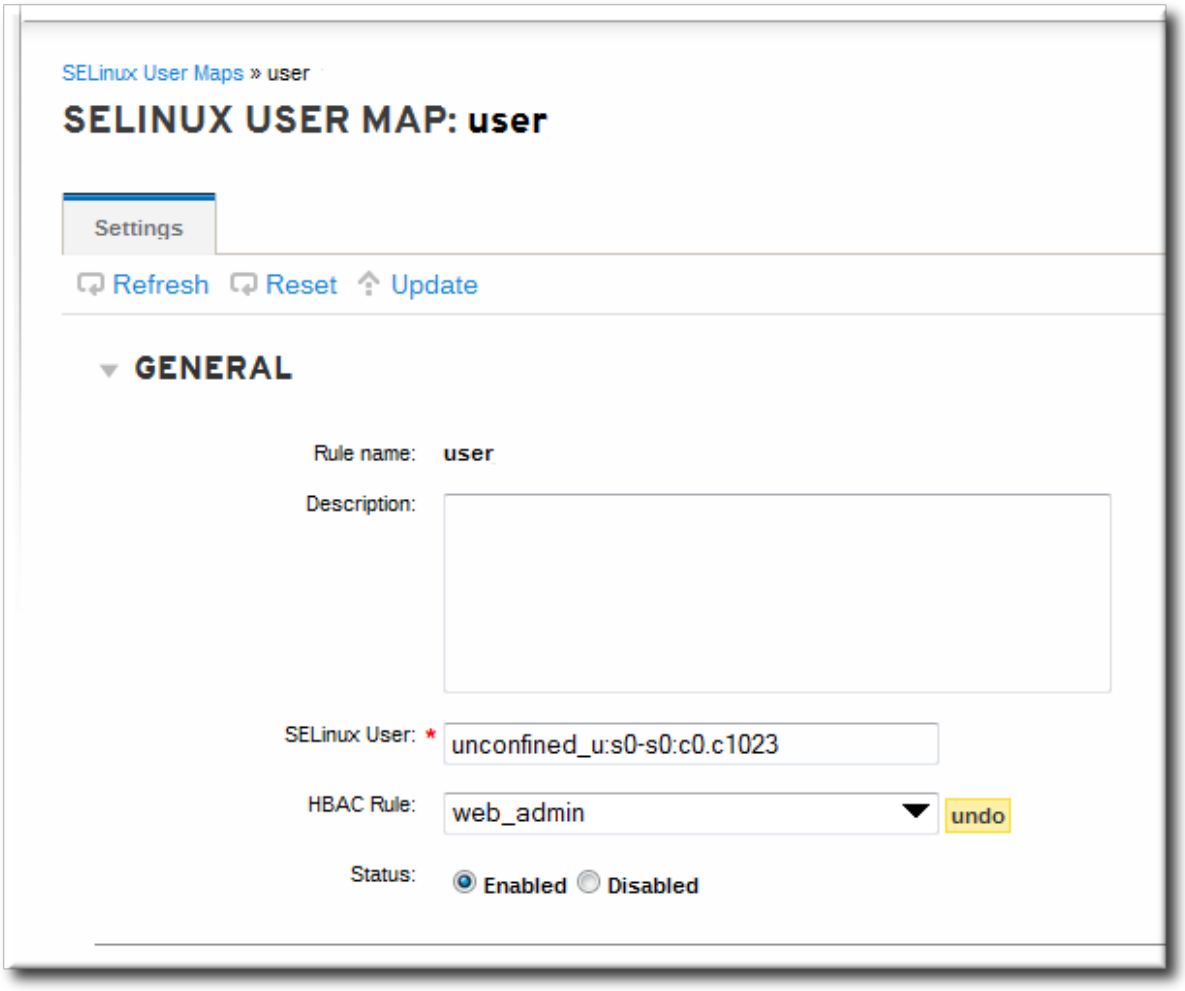


4. **Add and Edit** をクリックして、IdM ユーザー情報を追加します。

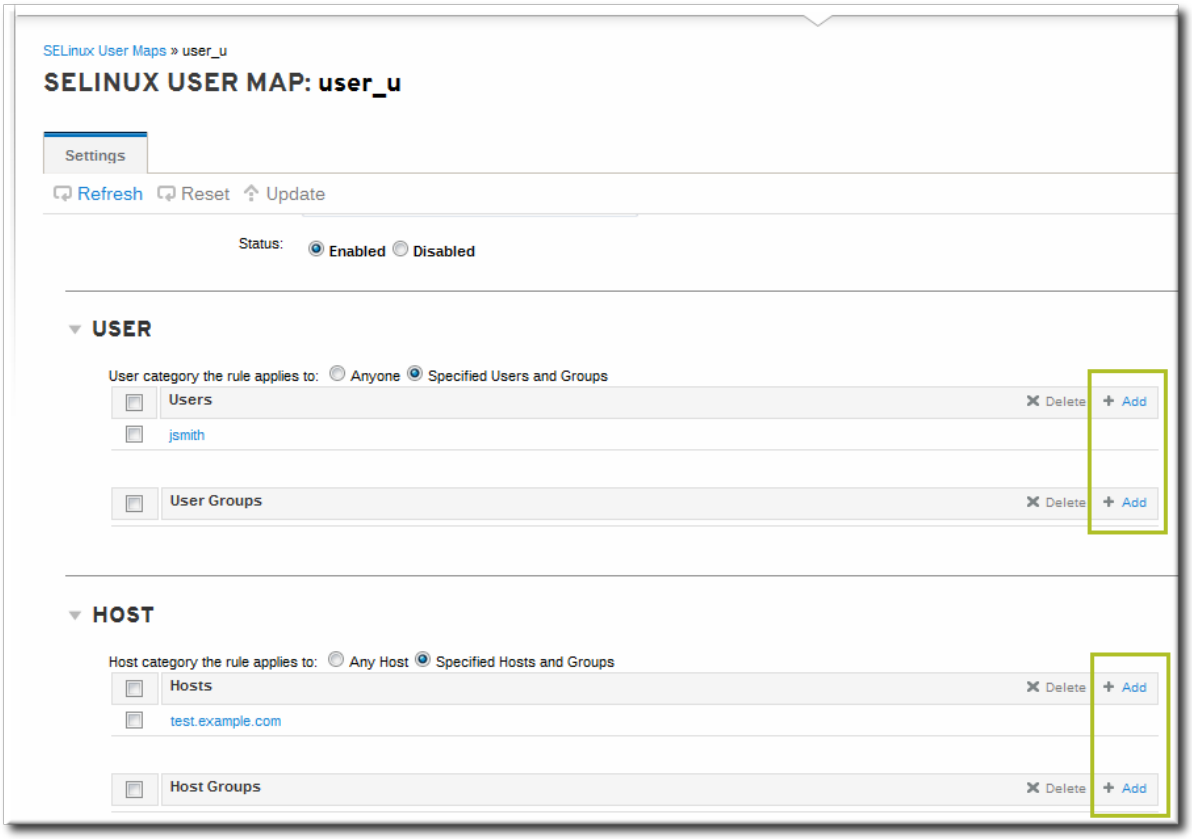
5. ホストベースのアクセス制御ルールを設定するには、設定の **General** エリアでドロップダウンメニューからルールを選択します。ホストベースのアクセス制御ルールを使用すると、リモートユーザーがターゲットマシンにアクセスする際に使用するホストでアクセス制御が導入されます。**割り当て可能なホストベースのアクセス制御ルールは、1つのみです。**

注記

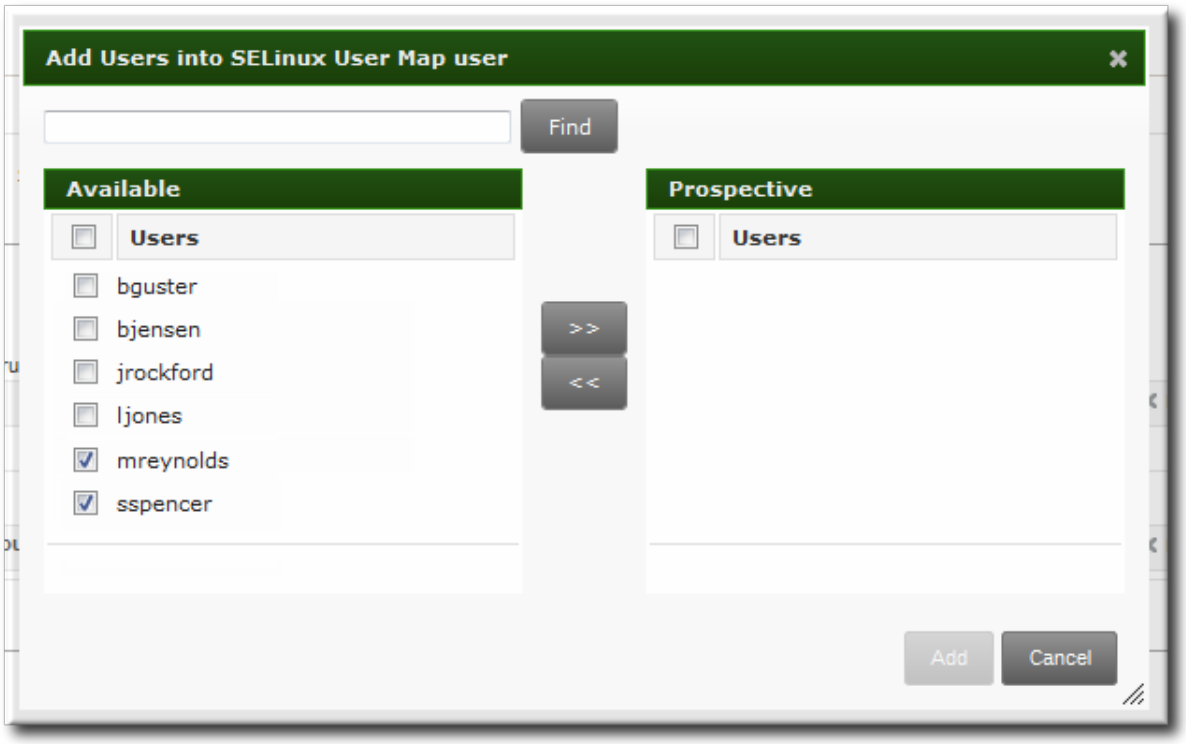
ホストベースのアクセス制御ルールには、サービスだけでなく、ユーザーとホストも含める必要があります。

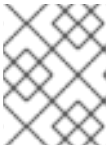


別の方法では、**Users** と **Hosts** のエリアでスクロールダウンし、**Add** をクリックしてユーザー、ユーザーグループ、ホスト、もしくはホストグループを SELinux マップに割り当てます。



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

ホストベースのアクセス制御ルールを指定するか、ユーザーと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。

24.3.2. 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24.3 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_username:MLS[:MCS]** です。

SELinux マッピングを有効にするには、ユーザーとホストの両方 (または適切なグループ) を指定する必要があります。ユーザー、ホスト、またはグループはコンマ区切りの一覧で指定できます。

```
[jsmith@server ~]$ ipa selinuxusermap-add --users=jsmith,bjensen,jrockford --
hosts=server.example.com,test.example.com --selinuxuser="xguest_u:s0" selinux1
```

例24.4 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。

```
[jsmith@server ~]$ ipa selinuxusermap-add --hbacrule=webserver --selinuxuser="xguest_u:s0"
selinux1
```

ホストベースのアクセス制御ルールは、「22章 *ポリシー: 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24.5 ユーザーを SELinuxマッピングに追加する

すべてのユーザーとホストは作成時にマップに追加できますが、ルールの作成後にユーザーとホストを追加することもできます。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。

```
[jsmith@server ~]$ ipa selinuxusermap-add-user --users=jsmith selinux1
```

ルールは 1 つしかないため、別のコマンドを使用してホストベースのアクセス制御ルールを追加する必要はありません。**selinuxusermap-mod** コマンドを **--hbacrule** オプションとともに使用する場合は、ホストベースのアクセス制御ルールを追加するか、以前のアクセス制御ルールを上書きします。

例24.6 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。たとえば、以下のようになります。

```
[jsmith@server ~]$ ipa selinuxusermap-remove-user --users=jsmith selinux1
```

第25章 ポリシー: 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

Identity Management ドメイン内のポリシーおよび設定の多くは、**グループ** に基づいています。自動マウントの sudo ルールからアクセス制御への設定はグループに対して定義され、それらの設定はグループメンバーに適用されます。

グループメンバーシップの管理は、ユーザーとホストを管理する上で重要な要素となります。**automember グループ** を作成すると、新規エントリーが追加されてからすぐに、指定したグループにユーザーとホストを自動的に追加するルールを定義します。

25.1. AUTOMEMBERSHIP について

ポリシー、アイデンティティー、およびセキュリティーをの最も重要なタスクの 1 つは、Identity Management でグループメンバーシップを管理することです。グループは、ほとんどのポリシー設定の中核となります。

デフォルトでは、ホストは作成時にグループに属しません。ユーザーは catchall **ipausers** グループに追加されます。カスタムグループが設定され、すべてのポリシー設定が行われても、ユーザーとホストはグループに参加するまでこれらのポリシーを利用することはできません。当然、これは手動で実行できますが、グループメンバーシップが自動的に割り当てられる場合は効率的で一貫性が高まります。

これは、automembership **グループ**で行われます。

Automembership は基本的に、特定の基準に基づいて少なくともある程度エントリーを整理する自動グローバルエントリーフィルターです。この場合、automember ルールはフィルターの指定方法になります。

たとえば、IT および組織環境内のアイデンティティーを分類するための反復可能な方法が多数あります。

- すべてのホストまたはすべてのユーザーを単一のグローバルグループに追加します。
- 従業員タイプ、ID 番号、マネージャー、または物理的な場所に基づいて社員を特定のグループに追加します。
- ホストの IP アドレスまたはサブネットに基づいたホストの分割。

Automembers は、これらのエントリーを事前分類する方法を提供します。これにより、異なるサブネット上の異なるユーザータイプまたはマシンに異なる sudo ルールを付与したり、ユーザーごとに異なる自動マウント設定など、設定したい実際の動作の設定が容易になります。

注記

Automembership は、**新規** ユーザーまたはホストにのみ適用されます。既存のユーザーまたはグループの設定を変更しても、グループメンバーシップの変更は発生しません。

Automembership は、既存のユーザーグループまたはホストグループに設定したターゲットです。ポリシーとして **自動メンバールール** が作成されます。これは、実際のグループエントリーへの参加エントリーで、指定のグループが自動グループメンバーシップに使用されることを示します。

ルールが作成された場合、グループがターゲットとして特定されると、次のステップは **automember 条件** を定義することとなります。Condition は、グループメンバーを識別するために使用される正規表現フィルターです。条件は包含または排他的にすることができます。つまり、これらの条件に基づいて一致するエントリーを追加または無視することができます。

1 つのルールに複数の条件がある可能性があります。ユーザーエントリーまたはホストエントリーは、複数のルールに一致し、複数のグループに追加できます。

Automembership は、ユーザーおよびホストのエントリーを作成する際に、それらをグループに追加することによって信頼性の高い順序を課す手段です。

automember グループを効果的に使用する鍵は、アクセス制御ポリシー、sudo ルール、ホスト/サービス管理ルール、ホストグループ、およびユーザーグループ全体 (Identity Management 全体の構造の計画) を計画することです。

構造が開始されたら、いくつかの構造が明確になります。

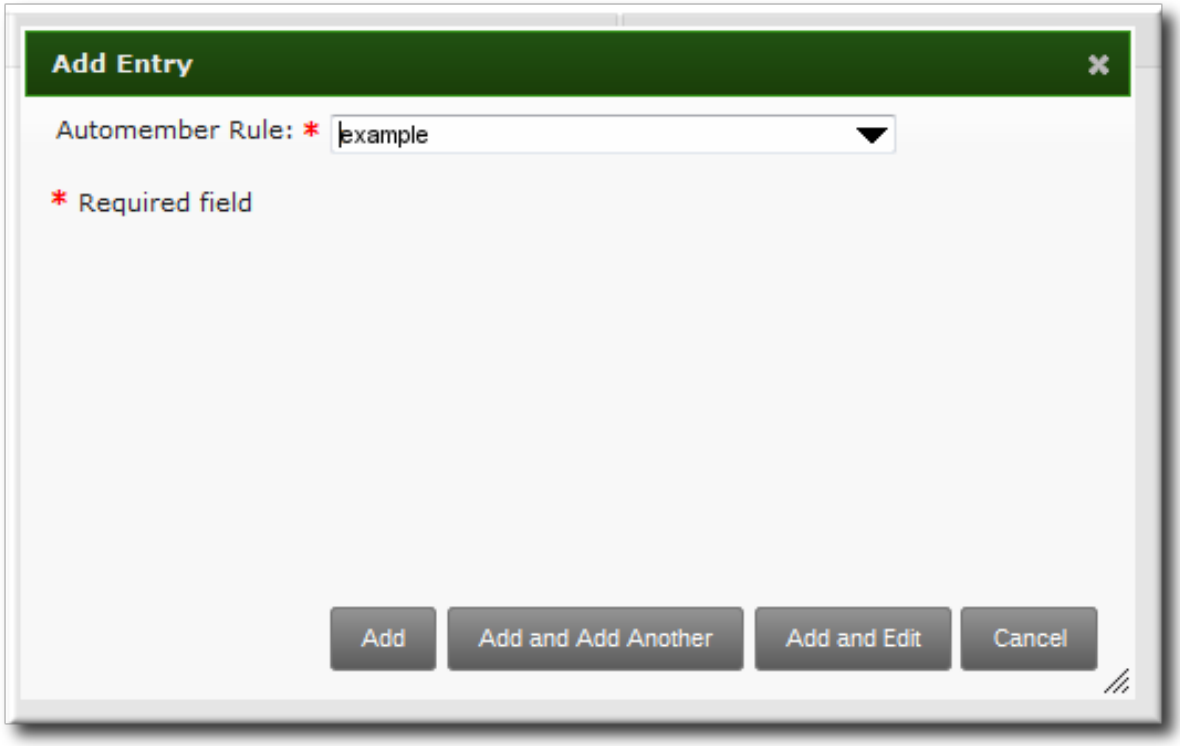
- Identity Management で使用されるグループ
- 指定した機能を実行するために、さまざまなタイプのユーザーおよびホストが属する必要のある特定のグループ
- 適切なグループにユーザーおよびホストをフィルターするために使用できる属性の記述

25.2. AUTOMEMBERSHIP RULES (基本手順) の定義

25.2.1. Web UI での操作

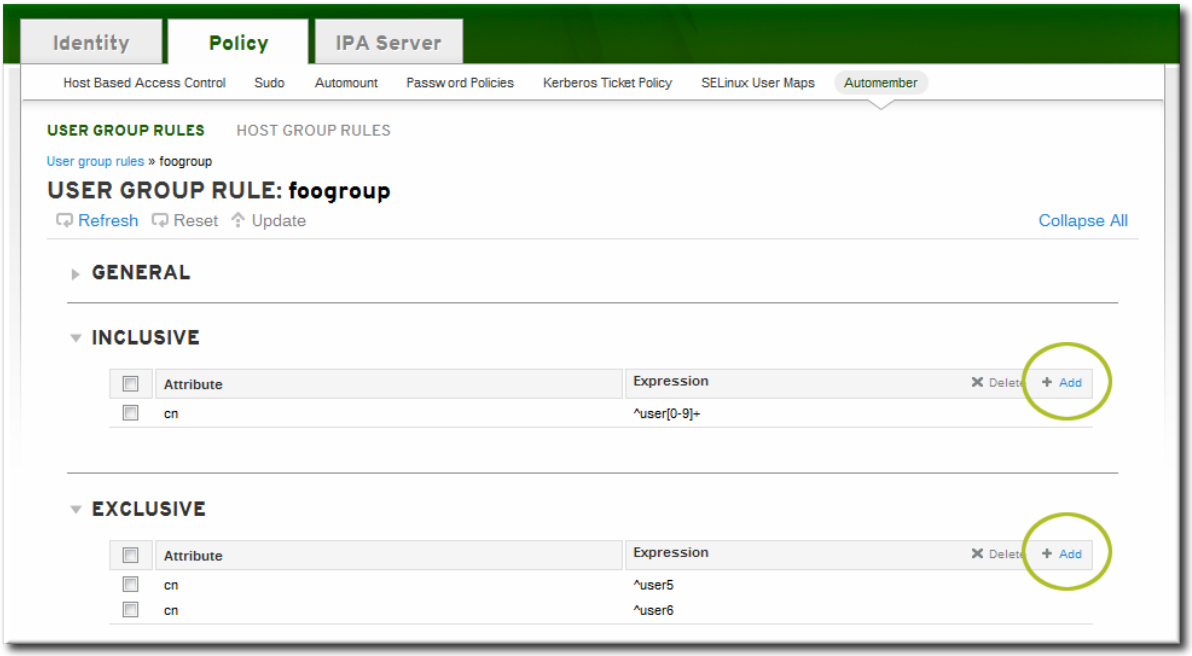
1. ユーザーグループ (「ユーザーグループの作成」) またはホストグループ (「Web UI でホストグループの作成」) を作成します。
2. **Policy** タブを開き、**Automembers** サブタブを選択します。
3. **Automembers** エリアの上部で、**USER GROUP RULES** または **HOST GROUP RULES** のいずれかを作成する自動グループのタイプを選択します。

4. ドロップダウンメニューで、automember ルールを作成するグループを選択します。



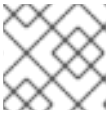
5. **追加および編集** ボタンをクリックします。

6. ルールの編集ページで、条件のタイプで **+ Add** をクリックしてエントリーを特定します。



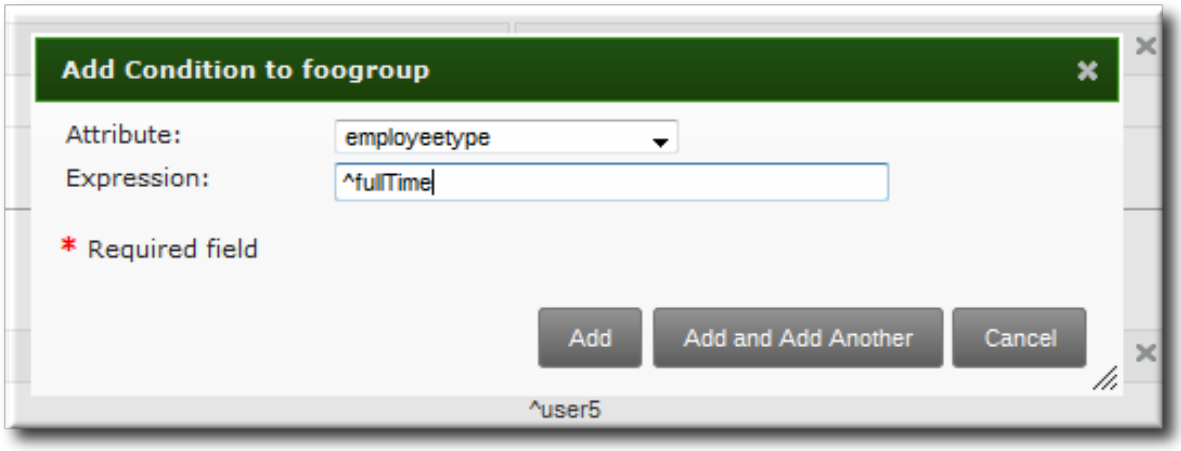
7. 検索のベースとして使用する属性を選択し、属性値と一致するために使用する正規表現を設定します。

   条件は、グループに **含める** グループから明示的に **除外** するエントリーを検索します。条件の形式は、Perl と互換性のある正規表現 (PCRE) です。PCRE パターンの詳細は、pcresyntax(3) の man ページを参照してください。



注記

除外条件は最初に評価され、包含条件よりも優先されます。



8. **Add and Add Another** をクリックして別の条件を追加します。1 つのルールに複数の包含および除外条件を指定することができます。すべての条件が設定されている場合は、**追加** ボタンをクリックして最後の条件を保存し、ダイアログウィンドウを閉じます。

25.2.2. CLI からの操作

automember ルールを定義するのに使用されるコマンドは 2 つあります。

- グループを automember グループとして対象とするコマンド **automember-add**
- グループメンバーを識別するための正規表現条件を追加するコマンド **automember-add-condition**

たとえば、以下のようになります。

1. ユーザーグループ (「コマンドラインの使用」) またはホストグループ (「コマンドラインでのホストグループの作成」) を作成します。

2. グループの automember ルールエントリーを作成します。**--type** を指定して、ターゲットグループがユーザーグループ (**group**) またはホストグループ (**hostgroup**) であるかを特定します。このコマンドの形式は以下のとおりです。

   ```
   ipa automember-add --type=group|hostgroup groupName
   ```

   以下に例を示します。

   ```
   [jsmith@server ~]$ ipa automember-add --type=group exampleGroup
   ```

3. ルールの条件を作成します。複数のパターンを設定するには、**--inclusive-regex|--exclusive-regex** オプションのパターンのコンマ区切りリストを指定するか、コマンドを複数回実行します。

   このコマンドの形式は以下のとおりです。

   ```
   ipa automember-add-condition --type=group|hostgroup --key=attribute --inclusive-regex=regex | --exclusive-regex=regex groupName
   ```

   automember ルールと同様に、条件はグループのタイプ (**--type**) とターゲットグループ (**groupName**) を指定する必要があります。

   条件は、属性 (キー) と属性値のパターンも指定する必要があります。**--key** は、条件の重点と

なる属性名です。次に、一致する値を識別する正規表現パターンがあります。一致するエントリーは含まれるか (**--inclusive-regex**) またはグループから除外 (**--exclusive-regex**) されるかのいずれかになります。除外ルールが優先されます。

たとえば、Barbara Jensen がマネージャーの社員すべてを含めて、一時社員を除外するには、以下に従います。

```
[jsmith@server ~]$ ipa automember-add-condition --type=group --key=manager --inclusive-regex=^uid=bjensen$ exampleGroup
[jsmith@server ~]$ ipa automember-add-condition --type=group --key=employeetype --exclusive-regex=^temp exampleGroup
```

ヒント

正規表現は文字列の任意の部分に一致できます。キャレット (^) を使用すると、開始時に一致する必要があることを意味します。ドル記号 ($) を使用すると、最後に一致する必要があることを意味します。^ および $ でパターンをラップした場合は、文字列全体が一致する必要があります。

Perl と互換性のある正規表現 (PCRE) パターンの詳細は、pcresyntax(3) の man ページを参照してください。

ルールの条件を削除するには、キーと正規表現の両方の完全条件情報を渡します。

```
[jsmith@server ~]$ ipa automember-remove-condition --key=fqdn --type=hostgroup --inclusive-regex=^web[1-9]+\.example\.com webservers
```

ルール全体を削除するには、**automember-del** コマンドを実行します。

25.3. AUTOMEMBER グループの使用例

注記

これらの例は CLI を使用して示しています。同じ設定は Web UI で実行できます。

デフォルトグループの作成に関する注意事項

一般的な環境要件の 1 つに、ユーザーまたはホストを追加するデフォルトグループがあります。これには、いくつかの方法があります。

- すべてのエントリーを 1 つのグローバルグループに追加できます。追加先の他のグループに関係なく、すべてのエントリーを単一のグローバルグループに追加できます。
- エントリーは、特定の automember グループに追加できます。新しいエントリーが autogroup に一致しない場合は、デフォルトまたはフォールバックグループに追加されます。

これらのストラテジーは相互排他的です。エントリーがグローバルグループと一致する場合は、automember グループと一致するため、フォールバックグループに追加されません。

25.3.1. 全ユーザー/ホストルールの設定

すべてのユーザーまたはすべてのホストを 1 つのグループに追加するには、すべてのエントリーに含まれる属性 (***cn*** または ***fqdn*** など) に含まれる正規表現を使用します。

すべてのエントリーに一致する正規表現は、単に **.*** を使用します 。たとえば、すべてのホストを同じホストグループに追加するには、次のコマンドを実行します。

```
[jsmith@server ~]$ ipa automember-add-condition --type=hostgroup allhosts --inclusive-regex=.* --key=fqdn
-------------------------------
Added condition(s) to "allhosts"
-------------------------------
  Automember Rule: allhosts
  Inclusive Regex: fqdn=.*
---------------------------
Number of conditions added 1
---------------------------
```

その後に追加したすべてのホストが、自動的に **allhosts** グループに追加されます。

```
[jsmith@server ~]$ ipa host-add test.example.com
---------------------------
Added host "test.example.com"
---------------------------
  Host name: test.example.com
  Principal name: host/test.example.com@EXAMPLE.COM
  Password: False
  Keytab: False
  Managed by: test.example.com

[jsmith@server ~]$ ipa hostgroup-show allhosts
  Host-group: allhosts
  Description: Default hostgroup
  Member hosts: test.example.com
```

PCRE パターンの詳細は、pcresyntax(3) の man ページを参照してください。

25.3.2. デフォルトの自動メンバーグループの定義

デフォルトグループを設定する特別なコマンド **automember-default-group-set** があります。これにより、automember ルールと同様にグループ名 (**--default-group**) およびグループタイプ (**--type**) が設定されますが、一致する条件はありません。定義上、デフォルトのグループメンバーは一致しないエントリーです。

たとえば、以下のようになります。

```
[jsmith@server ~]$ ipa automember-default-group-set --default-group=ipaclients --type=hostgroup
[jsmith@server ~]$ ipa automember-default-group-set --default-group=ipausers --type=group
```

デフォルトのグループルールは、**automember-default-group-remove** コマンドを使用して削除できます。グループタイプにはデフォルトグループが 1 つしかないため、グループ名ではなく、グループタイプのみを指定する必要があります。

```
[jsmith@server ~]$ ipa automember-default-group-remove --type=hostgroup
```

25.3.3. Windows ユーザーによる自動メンバーグループの使用

ユーザーが IdM で作成されると、そのユーザーは、自動的に **ipausers** グループにメンバーとして追加されます (automember グループとは別に、すべての新規ユーザーのデフォルトグループになります)。ただし、Windows ユーザーが Active Directory から同期すると、そのユーザーは自動的に **ipausers** グループに追加されません。

Identity Management で作成したユーザーと同様に、automember グループを使用して新しい Windows ユーザーを **ipausers** グループに追加できます。すべての Windows ユーザーが **ntUser** オブジェクトクラスとともに追加されます。そのオブジェクトクラスは包含フィルターとして使用でき、automember グループに追加する新しい Windows ユーザーを特定できます。

まず、**ipausers** グループを automember グループとして定義します。

```
[jsmith@server ~]$ ipa automember-add --type=group ipausers
```

次に、**ntUser** オブジェクトクラスを条件として使用し、ユーザーを追加します。

```
[jsmith@server ~]$ ipa automember-add-condition ipausers --key=objectclass --type=group --
inclusive-regex=ntUser
```

第26章 ポリシー: PAM サービスのドメイン制限

一部の環境では、異なる PAM アプリケーションが、異なる SSSD ドメインセットにアクセスする必要があります。**pam_ldap** などのレガシー PAM モジュールは、個別の設定ファイルを PAM モジュールのパラメーターとして使用できました。本章では、SSSD に類似した機能を説明します。

ユースケースの 1 つは、外部ユーザーが FTP サーバーへの認証を行える環境です。このサーバーは、権限のない別のユーザーとして実行されます。このユーザーは、内部の企業アカウントとは別に、選択した SSSD ドメインに対してのみ認証できます。この機能を使うと、管理者は FTP ユーザーが FTP PAM 設定ファイルに指定されている特定のドメインのみに認証できるようにすることができます。

以下のオプションは、PAM モジュールおよび SSSD で、選択したドメインへのアクセスを安全な方法で制限できます。

pam_trusted_users (for sssd.conf)

このオプションは、SSSD デーモンが信頼する数値の UID またはユーザー名の一覧を受け入れます。デフォルト値は特殊なキーワード **all** ですべてのユーザーが信頼されることを意味します。これは、すべてのユーザーが任意のドメインにアクセスできる現在の動作と並行して行います。

pam_public_domains (for sssd.conf)

このオプションは、信頼できないユーザーであってもアクセス可能な SSSD ドメインのコンマ区切りリストを受け入れます。**all** および **none** の 2 つの特別なキーワードも使用できます。管理者が信頼できないドメインと信頼されていないドメイン間で区別を開始する場合、信頼できないクライアントがアクセスできるドメインを手動で指定するのに必要な値は **none** です。

ドメイン (個々の PAM モジュール設定用)

このオプションは、PAM サービスが認証できるように制限されるドメインの一覧を受け入れます。この設定は **/etc/sssd/sssd.conf** ファイル内の **domains=** オプションと対話します。これは、SSSD がクエリーの順序でドメインの一覧を指定します。PAM モジュール設定はこの一覧に追加できませんが、短いリストを指定して制限できます。

例26.1 PAM モジュール設定のサンプル

設定ファイルの一般的な **/etc/pam.d/** 設定行の形式は以下のとおりです。

```
module-type control-flag module-path arguments
```

この例では、テストモジュールの設定例を示しています。ドメインアクセスを制限する引数は、各行の末尾に追加されます。テストモジュールは **openldap** ドメインのみに制限され、**pam_env** モジュールはすべてのユーザーに環境変数を設定/未設定にすることができます。

```
$ cat /etc/pam.d/sss_test
auth     required   pam_sss.so domains=openldap
account  required   pam_sss.so domains=openldap
session  required   pam_sss.so domains=openldap
password required   pam_sss.so domains=openldap
```

関連するスニペットは、PAM 設定の他に、**/etc/sssd/sssd.conf** のようになります。

```
[sssd]
domains = ipa, openldap # the list can be restricted by specific PAM module configuration
```

```
[pam]
pam_public_domains = ipa # all users are allowed to access the ipa domain
pam_trusted_users = root, sss_test # root and sss_test are allowed to run PAM
```

第27章 設定: IDM ユーザーのアクセス制御の定義

アクセス制御は、マシンとサービスからエントリーまでの特定のリソースにアクセスできるユーザーや、実行できる操作の種類を定義するセキュリティーシステムです。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。

27.1. 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。

27.1.1. アクセス制御の概念に関する簡単な概要

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 インスタンスにも保管される LDAP エントリーとして保存される) へのアクセスが許可されている IdM ユーザー (backend 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。

- **操作が実行できるユーザー**。これは、何かを実行するパーミッションを付与されるエンティティーです。これは、アクターです。これはユーザーが誰か (バインド情報に基づいて) 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。

- **アクセス可能なもの**。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。これは、アクセス制御ルールの **ターゲット** です。

- **実行できる操作のタイプ**。最後に、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。制限されるのは、パスワードや Kerberos キーなどの重要な属性のみです。匿名ユーザーは、**sudo** ルールやホストベースのアクセス制御など、セキュリティー関連の設定は読み取ることができません。

  付与できる権限は、エントリーの**変更**に必要な追加、削除、書き込みパーミッションです。

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。

27.1.2. 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**。これは、ユーザーが自分のパーソナルエントリーで実行可能な操作を定義します。アクセス制御タイプは、エントリー内での属性への書き込みパーミッションのみを許可します。エントリー自体の追加もしくは削除操作は許可されません。

- **委任ルール**。委任ルールでは、特定のユーザーグループが別のユーザーグループ内のユーザーの特定属性に関して書き込み (編集) 操作を許可されます。セルフサービスルールのように、この形式のアクセス制御は特定の属性値の編集に制限されており、エントリー全体を追加したり削除する権限や特定されていない属性に対する制御を付与するものではありません。

- **ロールベースのアクセス制御**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。

27.2. 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。

デフォルトでは、セルフサービスルールが 3 つ存在します。

- 個人エントリーの一般的な属性 (名、姓、電話番号、アドレスなど) を編集するルール。

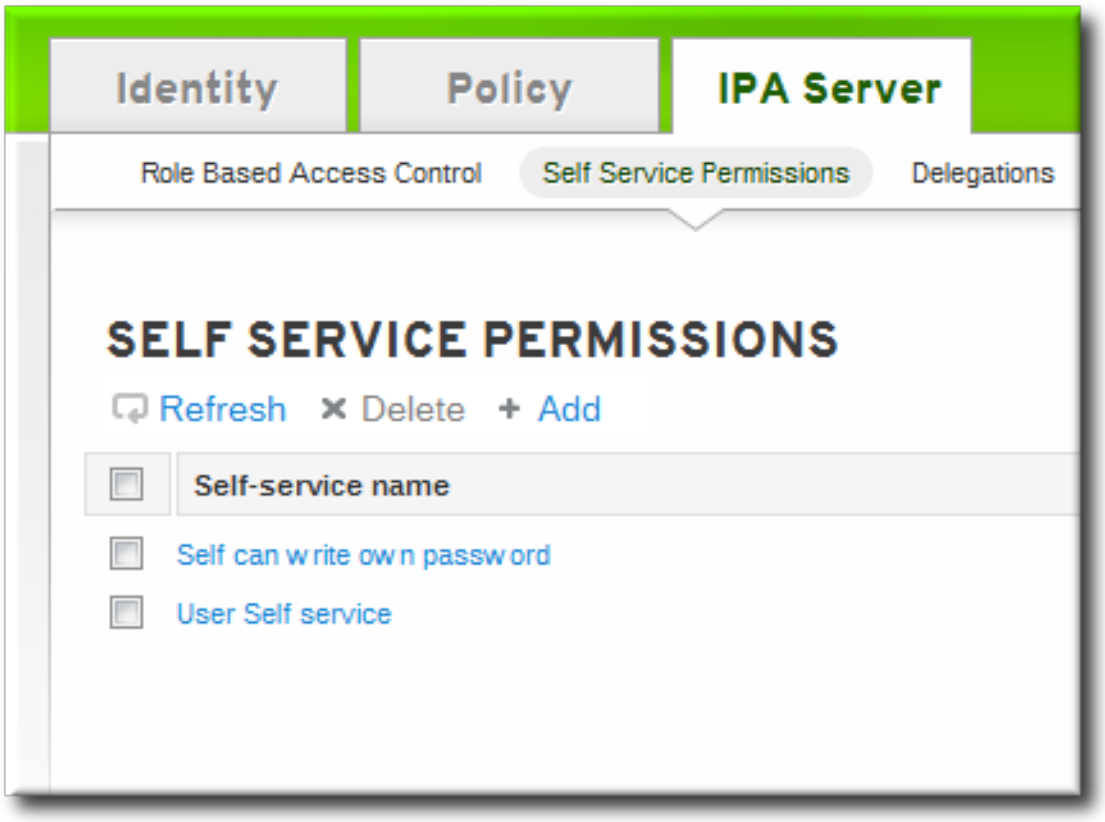
- 2 つの Samba パスワード、Kerberos パスワード、および一般的なユーザーパスワードなど、個人パスワードを編集するルール。

- 個人 SSH キーを管理するルール。

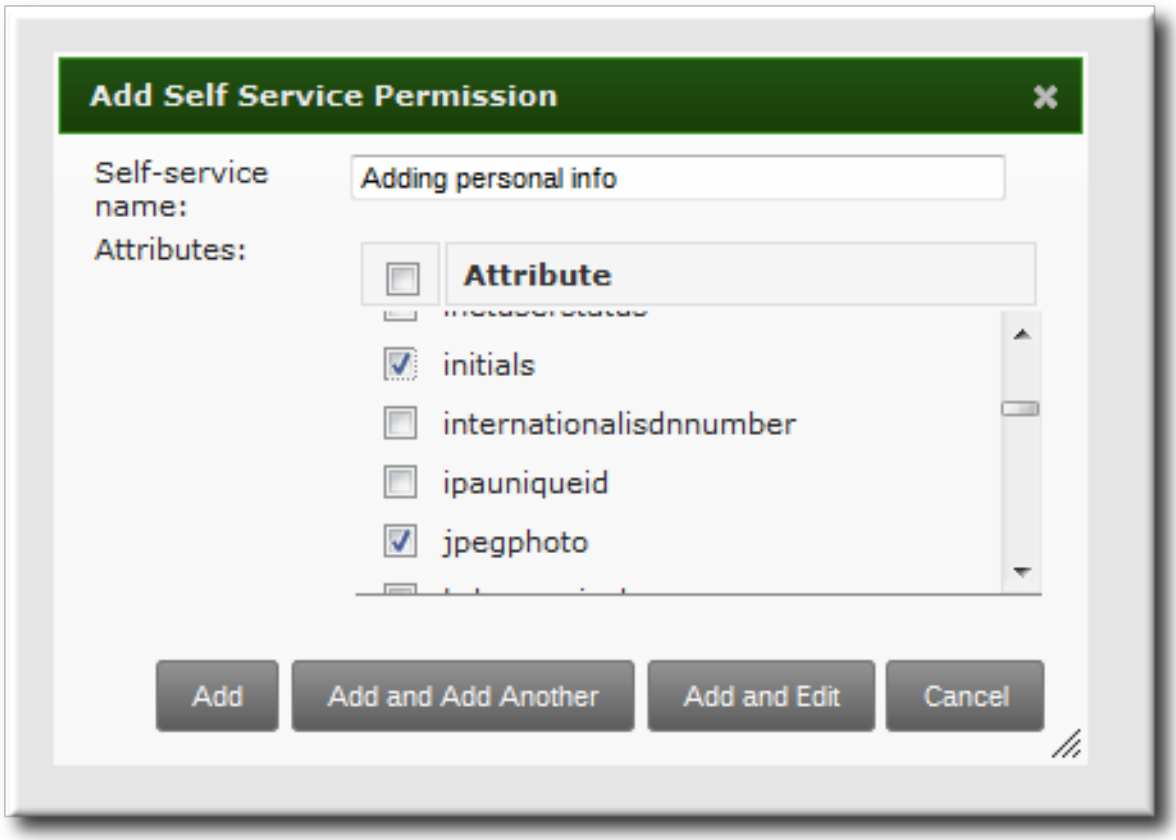
27.2.1. Web UI でのセルフサービスルールの作成

1. トップメニューで **IPA Server** タブを開き、**Self Service Permissions** サブタブを選択します。

2. セルフサービス ACI の一覧の上部にある **Add** リンクをクリックします。



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。



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。

27.2.2. コマンドライン でのセルフサービスルールの作成

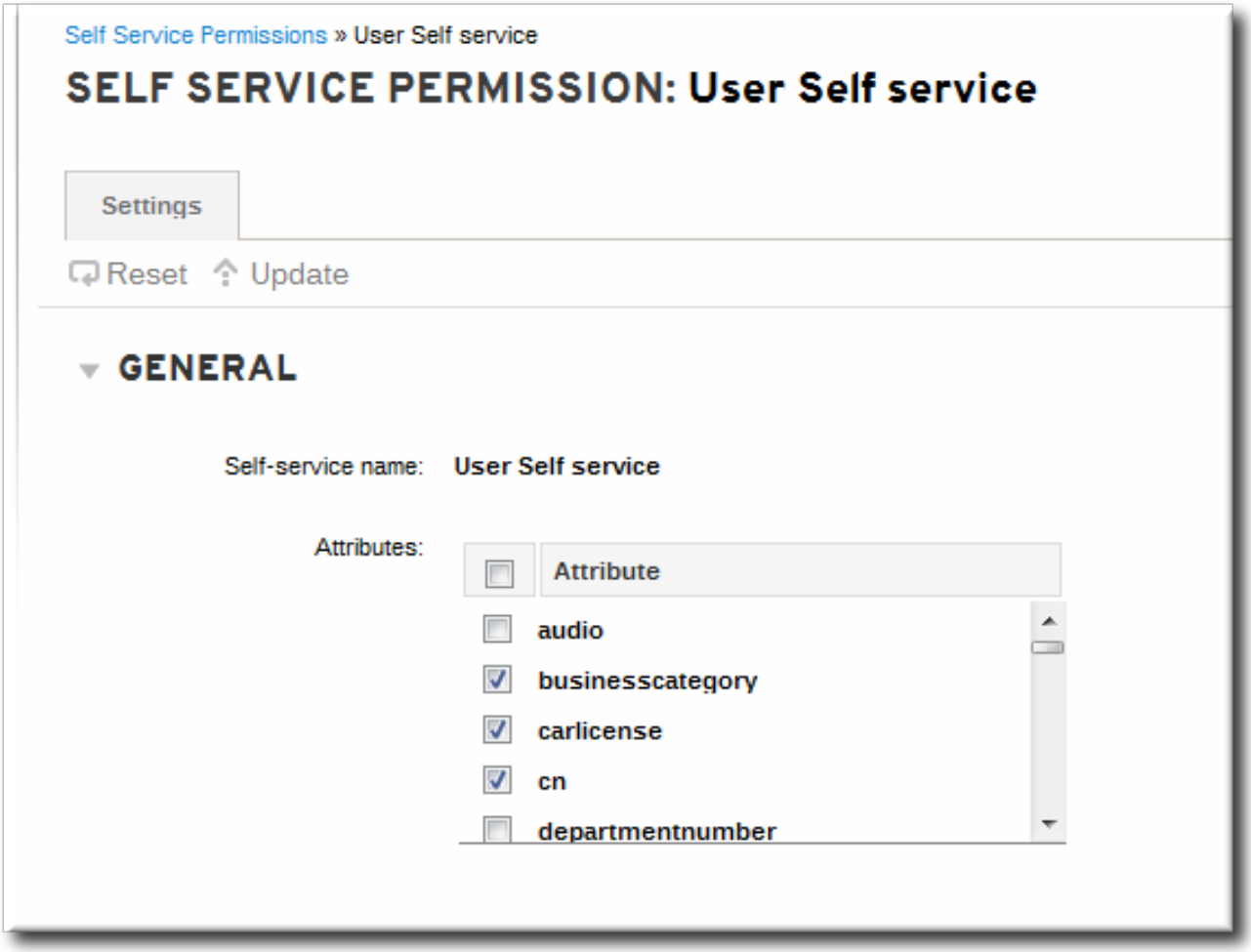
selfservice-add コマンドを使用すると、新しいセルフサービスルールを追加できます。ACI が書き込み、追加、または削除パーミッションを付与するかどうかを設定する **--permissions**、この ACI が付与する属性の完全なリストを付与する **--attrs** の 2 つのオプションがあります。

```
$ ipa selfservice-add "Users can manage their own name details" --permissions=write --
attrs=givenname,displayname,title,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
```

27.2.3. セルフサービスルールの編集

ウェブ UI のセルフサービスエントリーでは、ACI に含まれている属性の一覧のみが編集可能な要素です。チェックボックスは選択または選択解除できます。

図27.1 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。

```
$ ipa selfservice-mod "Users can manage their own name details" --
```

```
attrs=givenname,displayname,title,initials,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

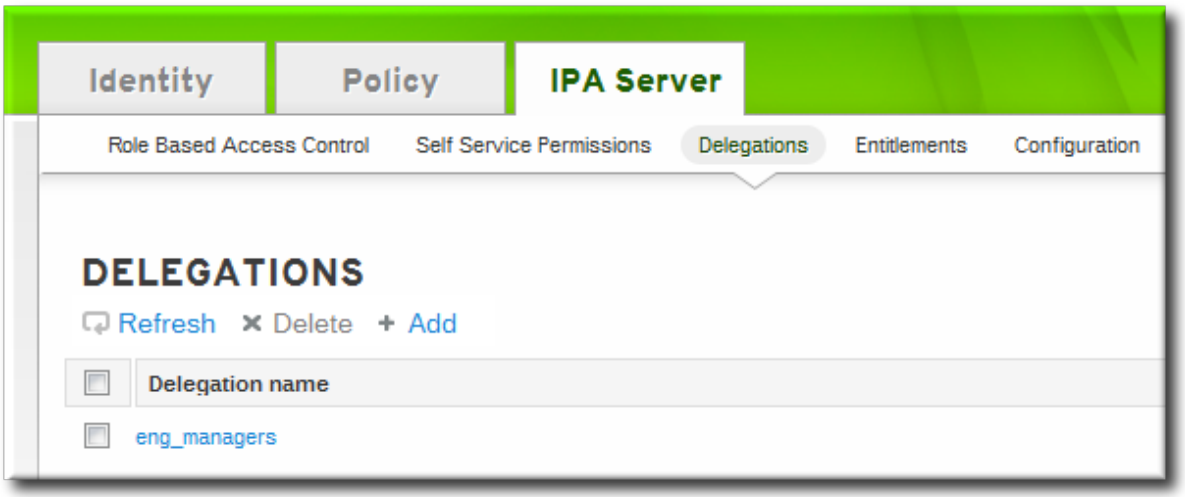
セルフサービスルールを修正する際は、既存の属性も含め、すべての属性を含めるようにしてください。

27.3. 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。

27.3.1. Web UI でのユーザーグループへのアクセス委任

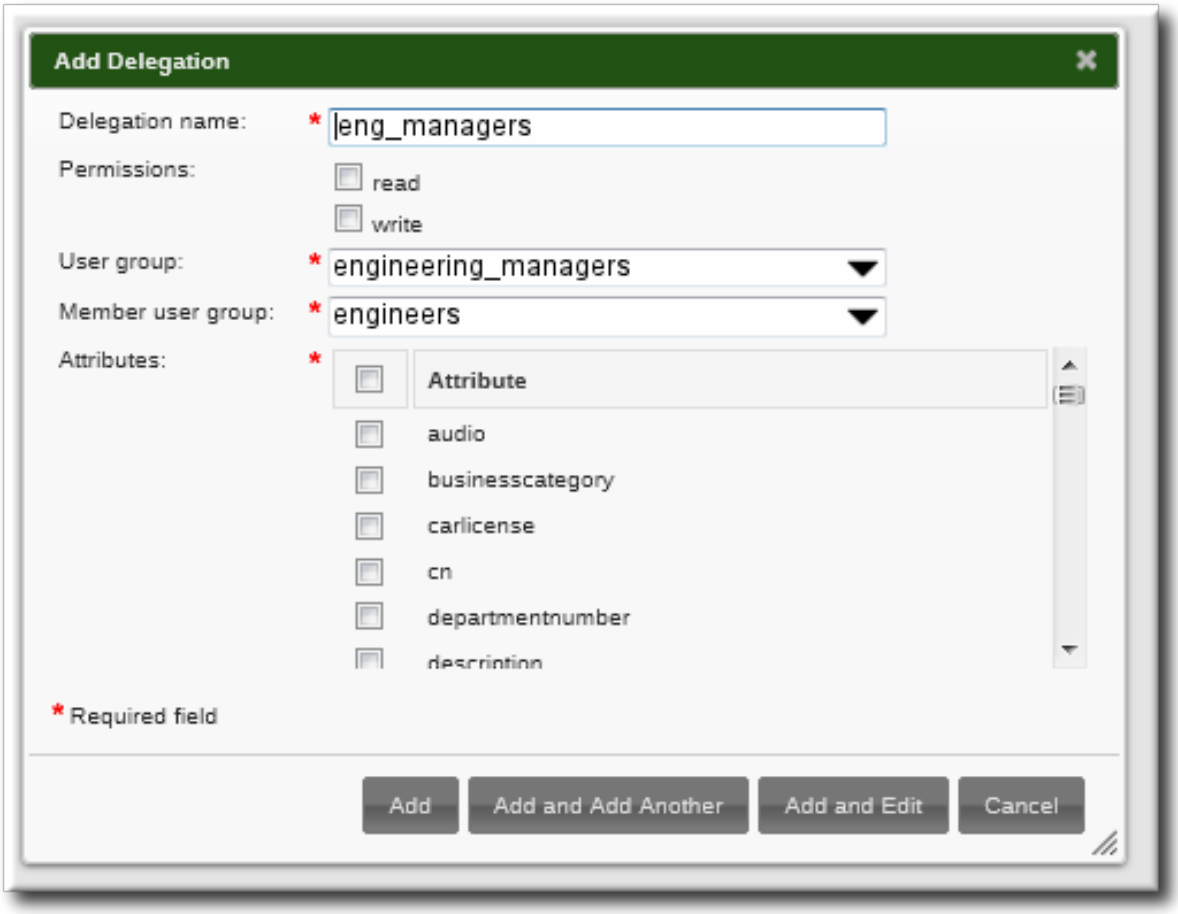
1. トップメニューで **IPA Server** タブを開き、**Delegations** サブタブを選択します。
2. 委譲 ACI 一覧の上部にある **Add** リンクをクリックします。



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。



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。

27.3.2. 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編集できる属性です。

たとえば、以下のようになります。

```
$ ipa delegation-add "basic manager attrs" --attrs=manager,title,employeetype,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。

```
$ ipa delegation-mod "basic manager attrs" --
attrs=manager,title,employeetype,employeenumber,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。

27.4. 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います。エントリーの追加や削除および大幅な変更の可能性があります。

ロールベースのアクセス制御には、以下の 3 つの部分があります。

- **パーミッション**。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。

- ロールで利用可能な **特権**。特権は基本的にパーミッションのグループです。パーミッションはロールに直接適用されません。権限が特権に追加され、特権によってアクセス制御ルールのセットの一貫性と完全な情報が作成されます。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。

- **ロール**。これは、特権に定義されているアクションを実行できる IdM ユーザーの一覧です。

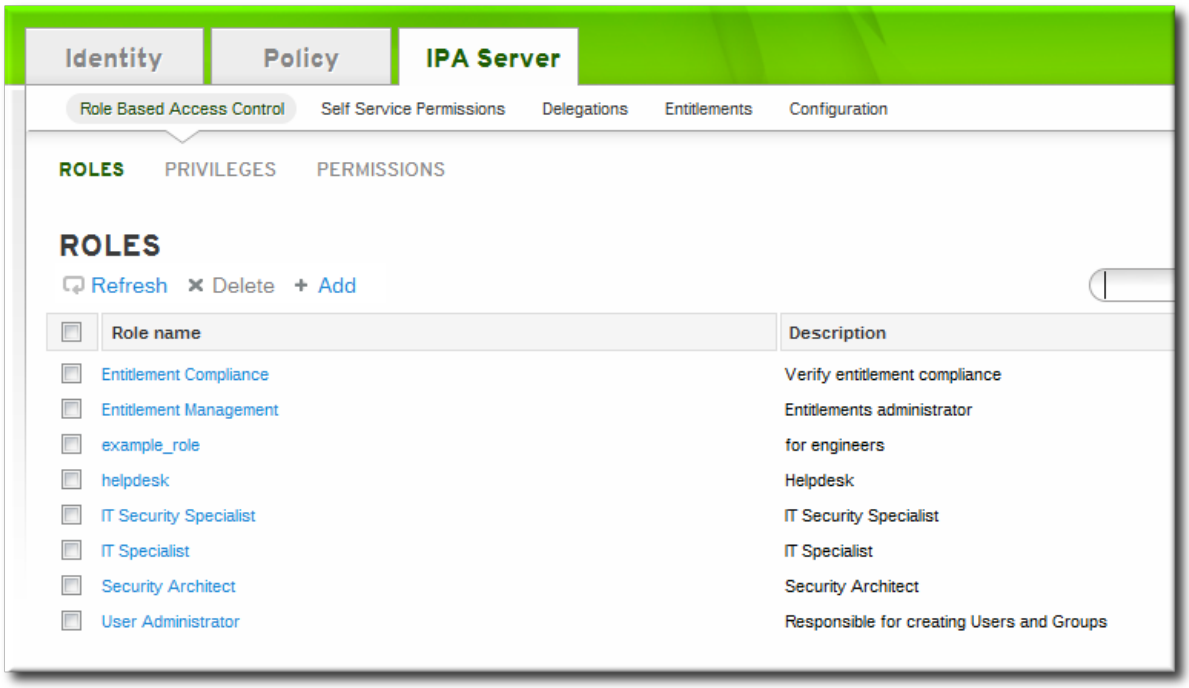
完全に新しいパーミッションを作成したり、既存または新規のパーミッションをベースにして新たな権限を作成したりすることができます。

27.4.1. ロールの作成

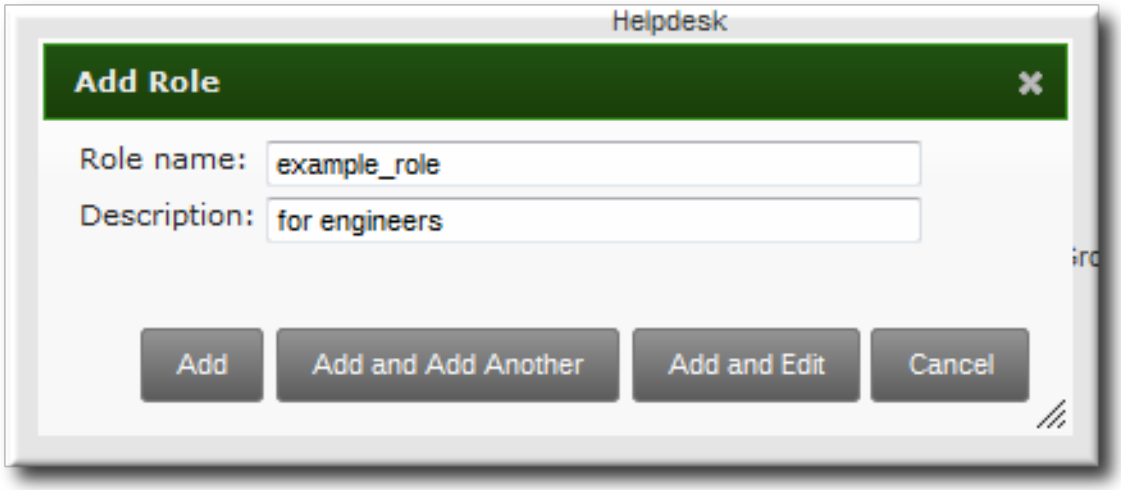
27.4.1.1. Web UI でのロールの作成

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 ACI 一覧の上部にある **Add** リンクをクリックします。

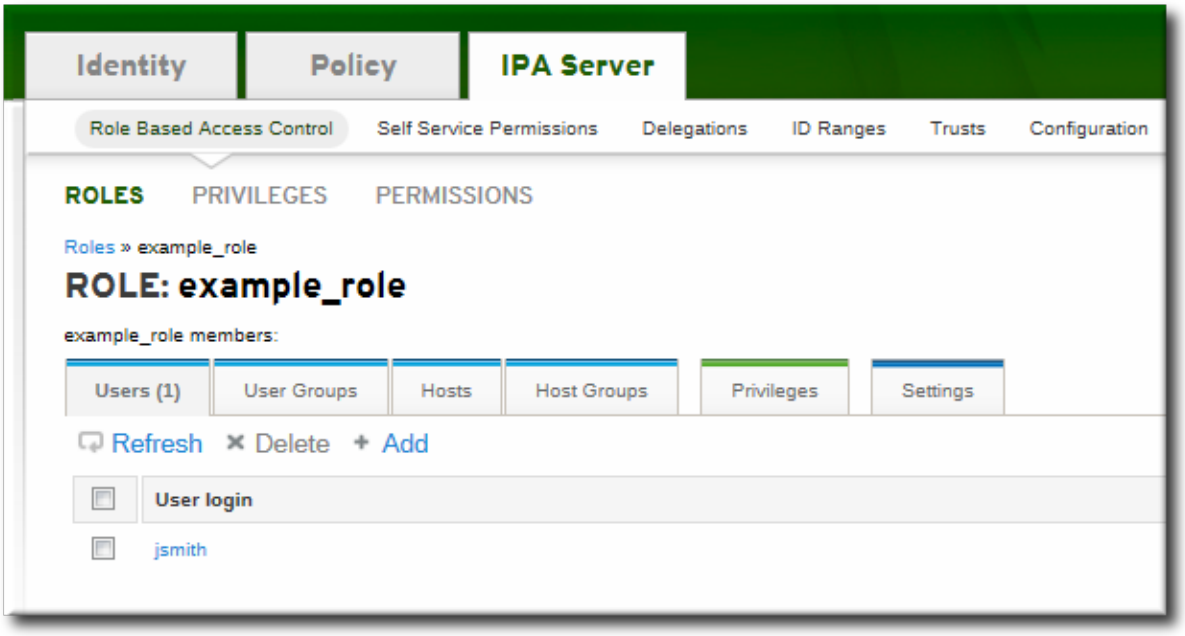


3. ロール名と説明を入力します。

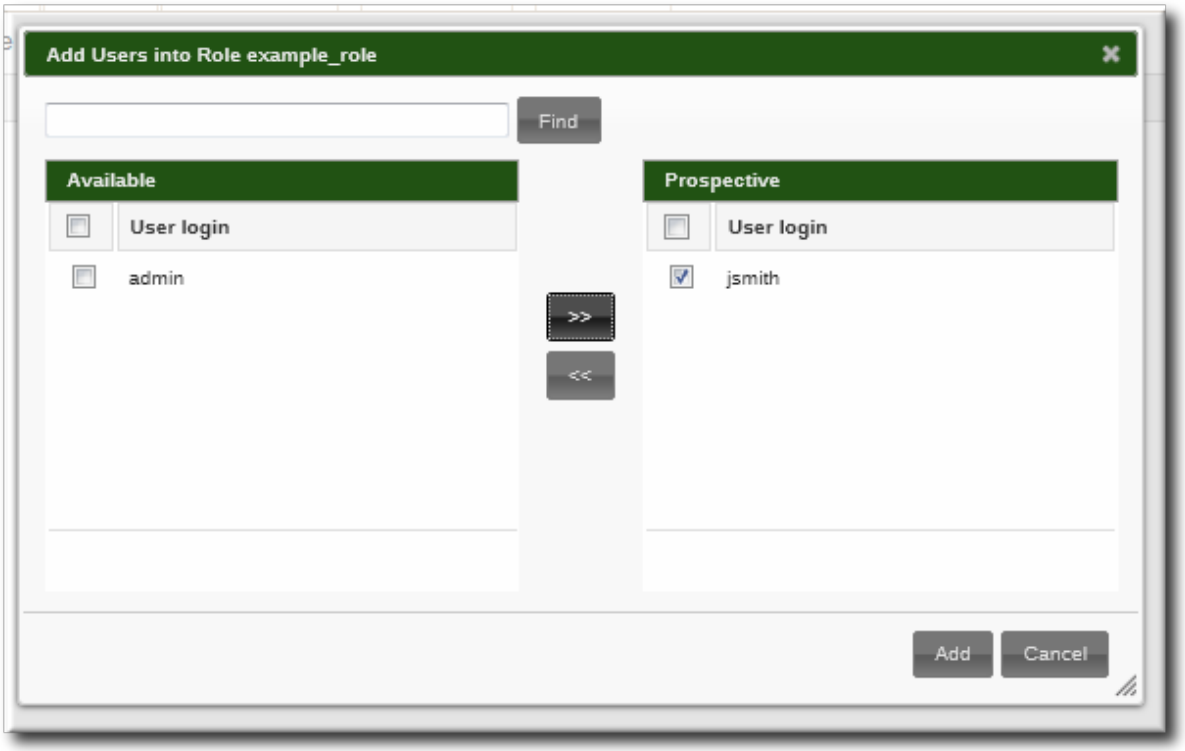


4. **Add and Edit** ボタンをクリックして新規ロールを保存し、設定ページに移動します。

5. **Users** タブの上部で、グループ追加の場合は **Users Groups** タブで、**Add** リンクをクリックします。

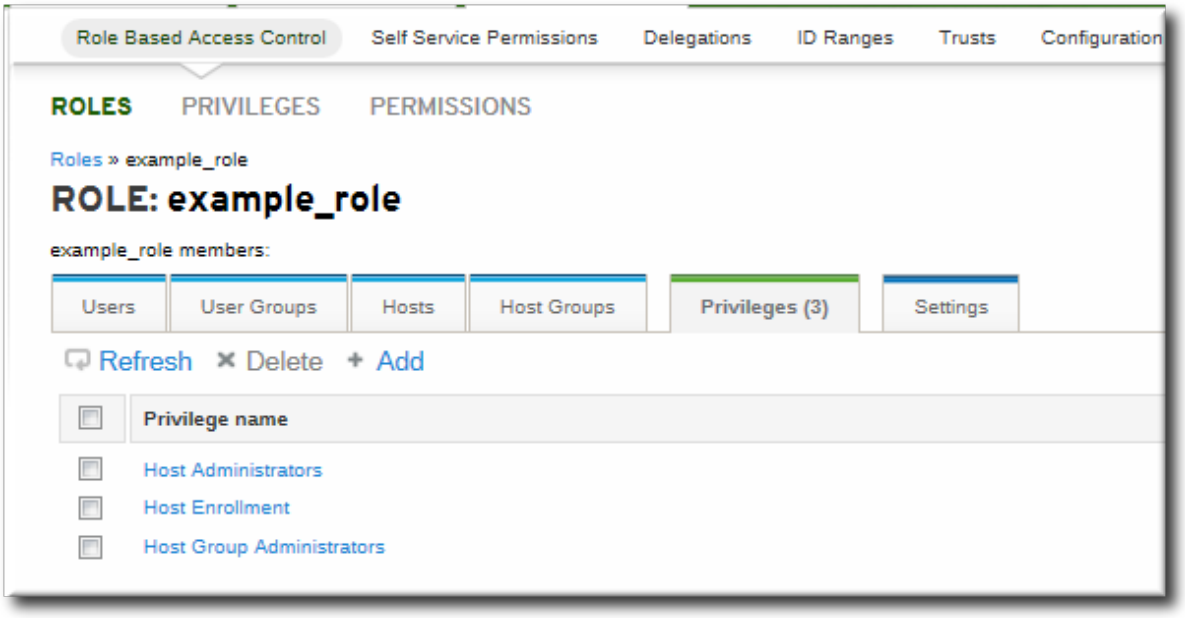


6. 左側のユーザーを選択し、**>>** ボタンを使用して、割り当てられたボックスに移動します。

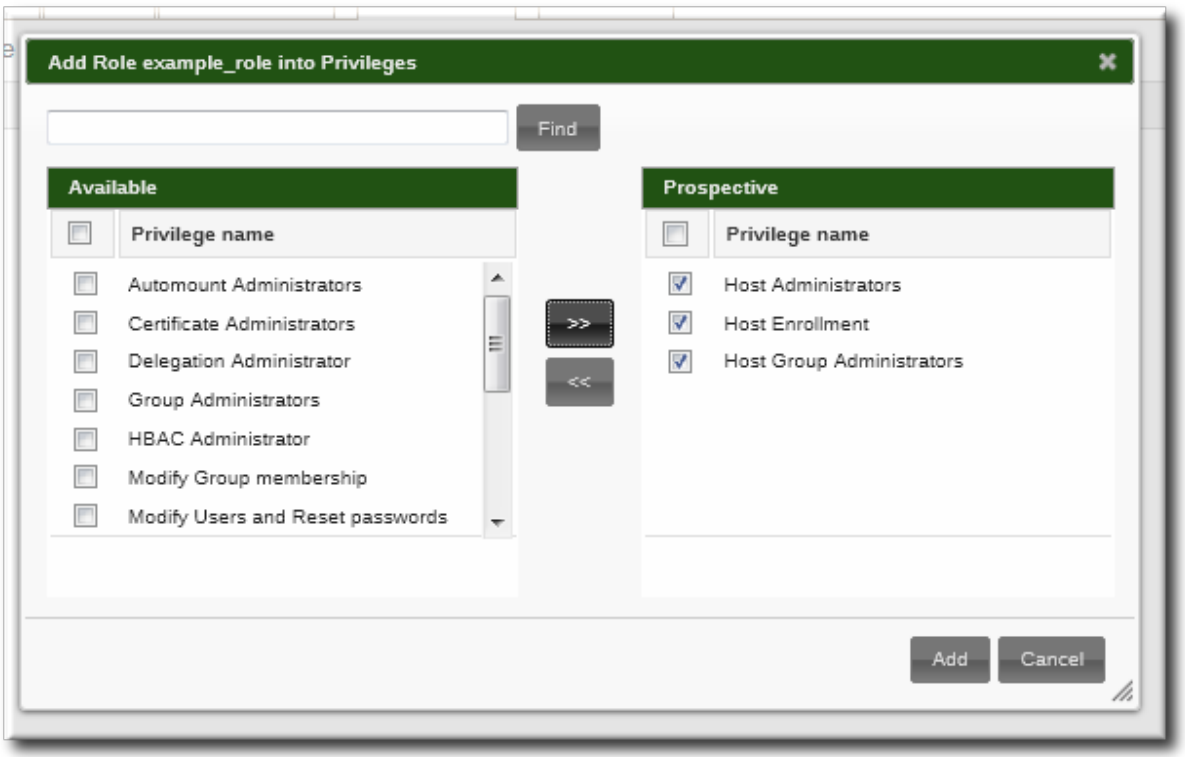


7. ロール設定ページで **Privileges** タブを開きます。

8. 特権一覧の上部にある **Add** リンクをクリックして、新しい権限を追加します。



9. 左側の権限を選択し、**>>** ボタンを使用して割り当てられたボックスに移動します。



10. **Add** ボタンをクリックして保存します。

27.4.1.2. 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
-------------------------
```

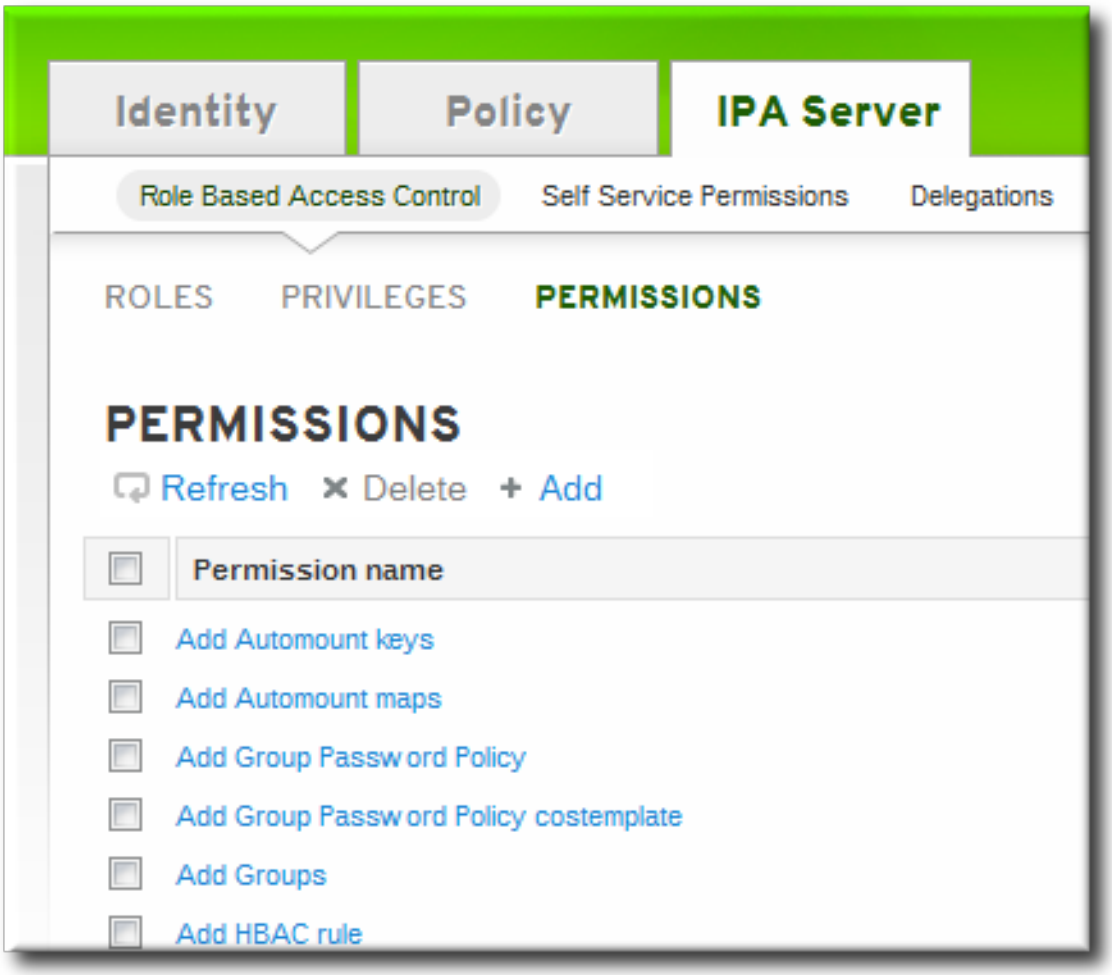
27.4.2. 新規パーミッションの作成

27.4.2.1. 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。

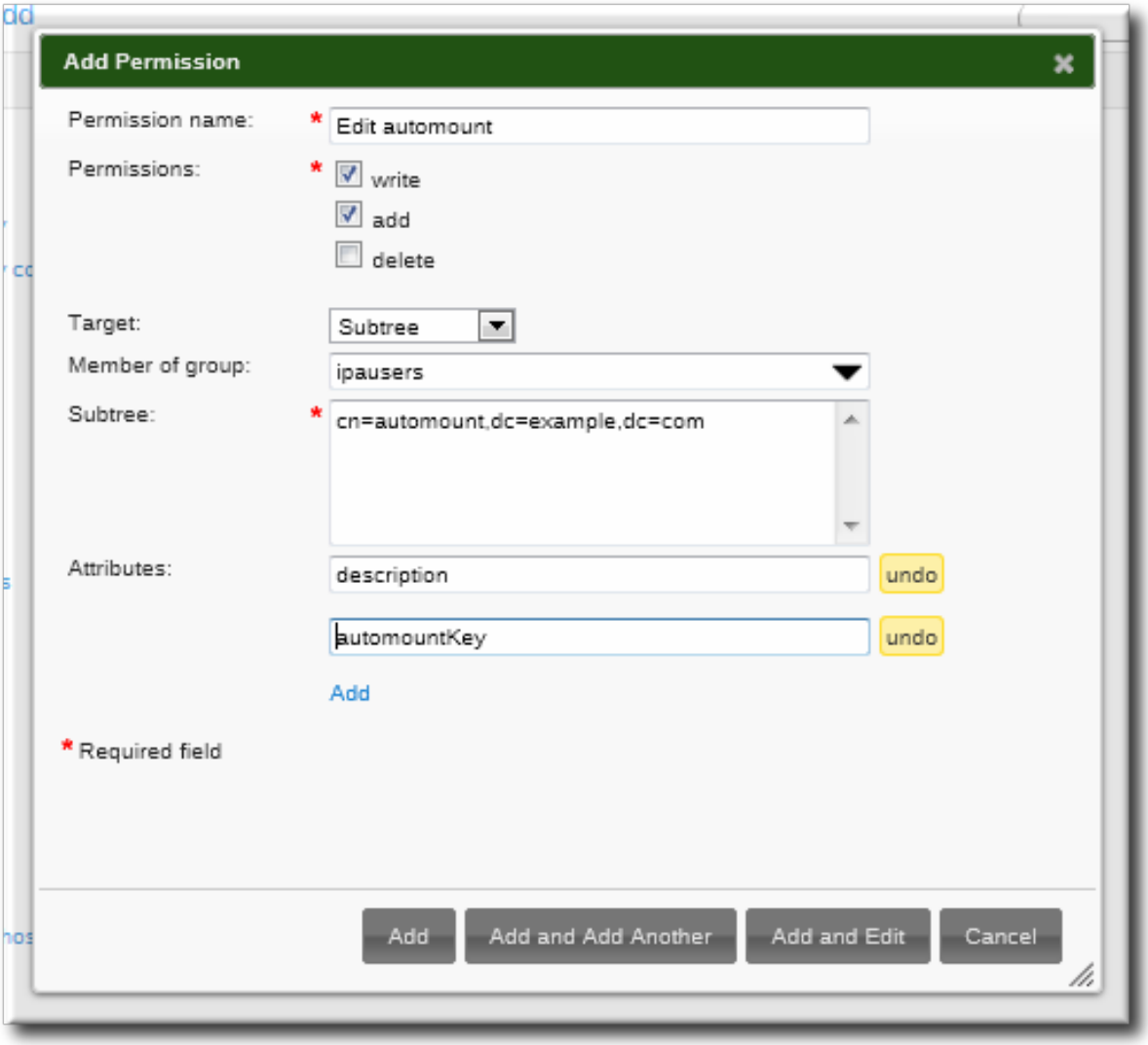
2. **Permissions** タスクリンクを選択します。

3. パーミッションの一覧の上部にある **Add** リンクをクリックします。



4. 新規パーミッションの名前を入力します。

5. このパーミッションで許可される操作の横にあるチェックボックスを選択します。



6. **Target** ドロップダウンメニューからターゲットエントリーを識別するのに使用する方法を選択します。以下の 4 つの方法があります。

   - **type** は、ユーザー、ホスト、またはサービスといったエントリータイプを検索し、そのエントリータイプに対して可能なすべての属性のリストを提供します。この ACI からアクセス可能な属性が一覧から選択されます。

   - **Filter** は、パーミッションが適用されるエントリーを特定する LDAP フィルターを使用します。

   - **サブツリー** は、指定された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。一致するエントリー内のすべての属性を変更できます。

   - **ターゲットグループ** はユーザーグループを指定し、そのグループ内のすべてのユーザーエントリーは ACI 経由で利用できます。一致するエントリー内のすべての属性を変更できます。

7. 選択したタイプに応じて、ターゲットエントリーを特定するために必要な情報を入力します。

8. **Filter**、**Subtree**、および **Target group** ターゲットの場合は、**Add** リンクをクリックして、パーミッションに含まれる属性を追加します。1 つの属性が一度に追加されます。複数の属性を追加するには、再度 **Add** をクリックして別のフィールドを追加します。

   パーミッションに属性が設定されていない場合、デフォルトではすべての属性が除外されます。

9. **追加** ボタンをクリックしてパーミッションを保存します。

27.4.2.2. コマンドライン での新規パーミッションの作成

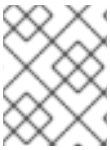
この **permission-add** コマンドを使用して、新しいパーミッションが追加されます。すべてのパーミッションには、パーミッションが付与される属性のリスト (**--attr**)、許可されるアクション (**--permissions**) のリスト、および ACI のターゲットエントリーが必要です。ターゲットエントリーを識別する方法は 4 つあります。

- **--type** は、ユーザー、ホスト、またはサービスなどのエントリータイプを検索し、そのエントリータイプに使用できるすべての属性の一覧を表示します。
- **--filter** は、パーミッションが適用されるエントリーを特定する LDAP フィルターを使用します。
- **--subtree** は、指定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。
- **--targetgroup** はユーザーグループを指定し、そのグループ内のすべてのユーザーエントリーは ACI 経由で利用できます。

例27.1 フィルターを使用したパーミッションの追加

フィルターは有効な LDAP フィルターにすることができます。

```
$ ipa permission-add "manage Windows groups" --filter="(!(objectclass=posixgroup))" --
permissions=write --attrs=description
```



注記

この **permission-add** コマンドは、指定の LDAP フィルターを検証しません。パーミッションを設定する前に、フィルターが想定された結果を返すことを確認します。

例27.2 サブツリーのパーミッションの追加

サブツリーフィルターに必要なのは、ディレクトリー内の DN です。IdM は簡素化された平坦なディレクトリーツリー構造を使用しているので、これを使って、自動マウントの場所のような、他の設定のコンテナーや親エントリーである、一定タイプのエントリーをターゲットにすることができます。

```
$ ipa permission-add "manage automount locations" --
subtree="ldap://ldap.example.com:389/cn=automount,dc=example,dc=com" --permissions=write -
-attrs=automountmapname,automountkey,automountInformation
```

例27.3 オブジェクトタイプに基づいたパーミッションの追加

パーミッションを形成するために使用できるオブジェクトタイプは 7 つあります。

- user
- グループ

- host
- サービス
- hostgroup
- netgroup
- dnsrecord

各タイプには、カンマ区切りリストの独自の許可される属性セットがあります。

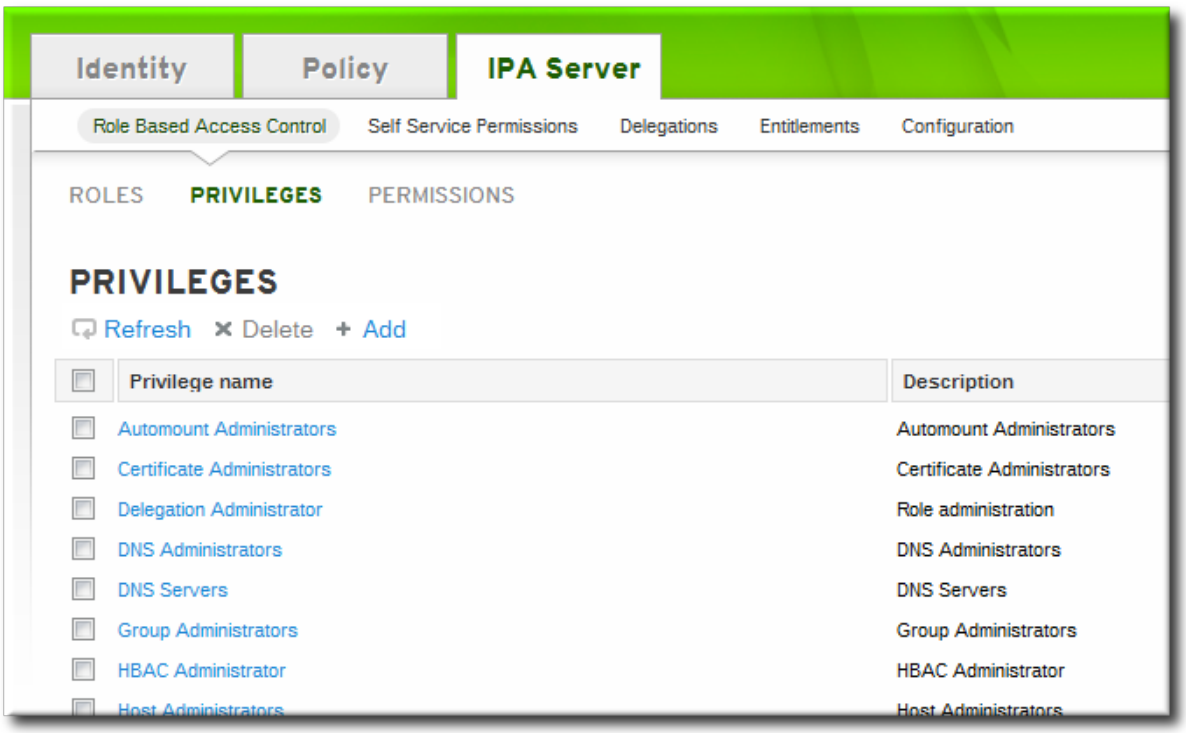
```
$ ipa permission-add "manage service" --permissions=all --type=service --
attrs=krbprincipalkey,krbprincipalname,managedby
```

属性 (**--attrs**) が存在し、指定のオブジェクトタイプの属性を許可する必要があります。そうでない場合、パーミッション操作はスキーマ構文エラーで失敗します。

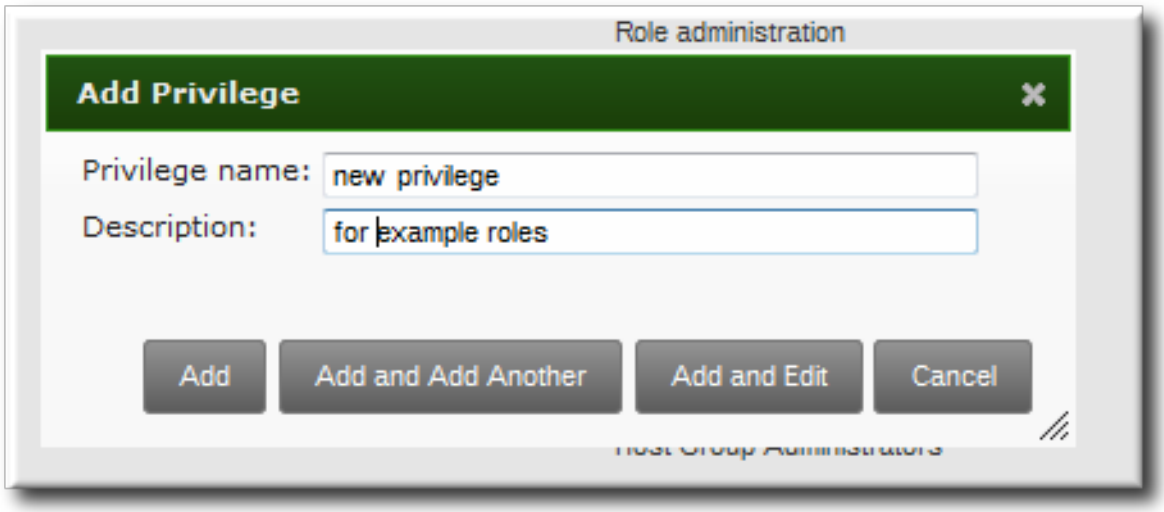
27.4.3. 新規権限の作成

27.4.3.1. Web UI での新規権限の作成

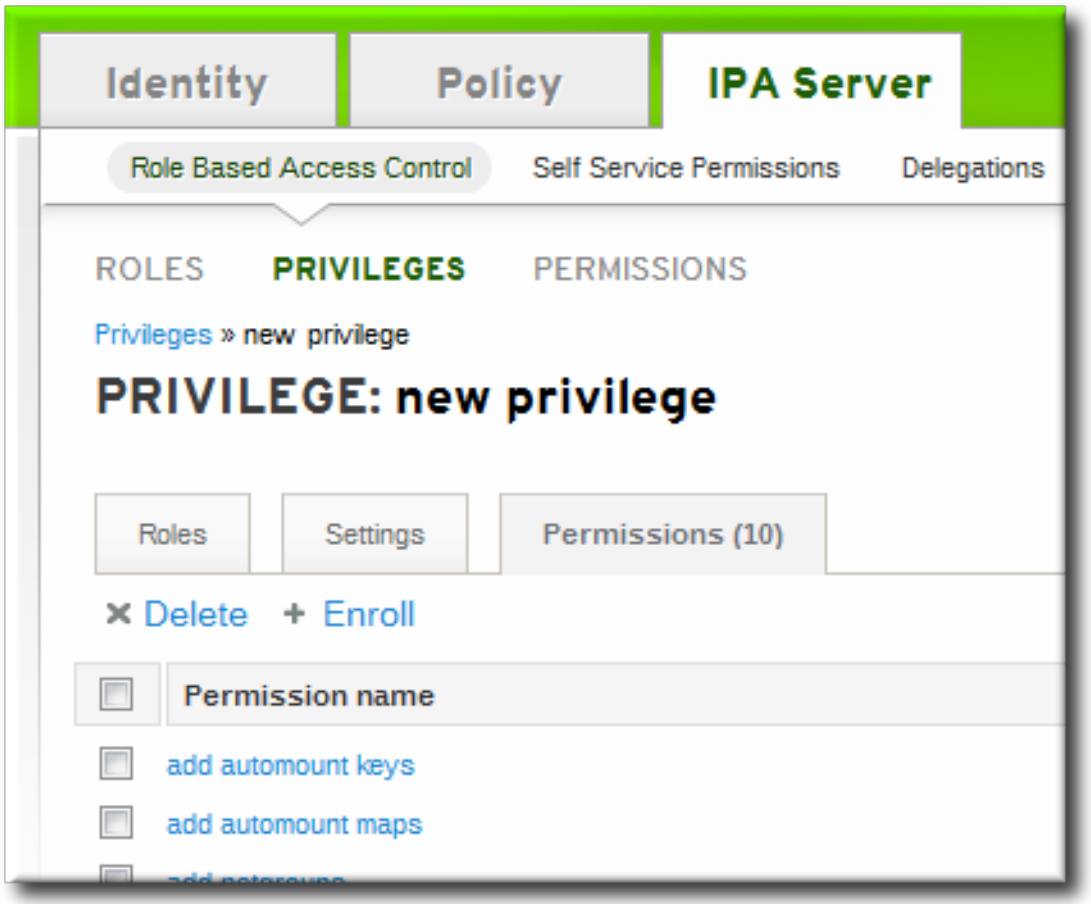
1. トップメニューで **IPA Server** タブを開き、**Role Based Access Control** サブタブを選択します。
2. **Privileges** タスクリンクを選択します。
3. 特権一覧の上部にある **Add** リンクをクリックします。



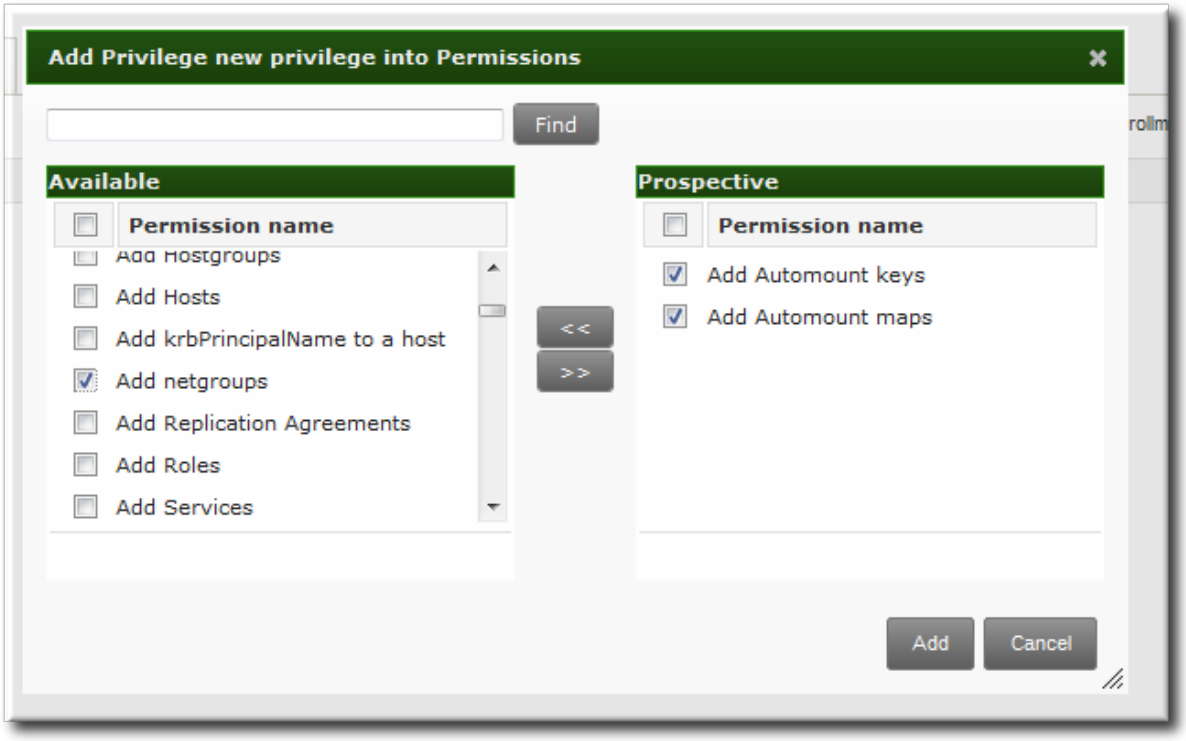
4. 権限の名前と説明を入力します。



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. パーミッション一覧の上部にある **Add** リンクをクリックして、権限を付与します。



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、右矢印ボタン (**>>**) をクリックし、パーミッションを選択項目のボックスに移動します。



9. **追加** ボタンをクリックします。

27.4.3.2. 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。

```
$ 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。

```
$ ipa privilege-add-permission "managing filesystems" --permissions="managing
automount","managing ftp services"
```

第28章 設定: IDM サーバーおよびレプリカの設定

IdM サーバーおよびバックエンドサービスは、ほとんどの環境で適用可能なデフォルト設定で設定されます。

特定の状況でセキュリティーやパフォーマンスを改善するために、IdM サーバーの設定を調整できる設定エリアがあります。

本章では、IdM サーバーが使用するファイルおよびログ、IdM サーバー設定自体を更新する手順など、IdM 設定に関する情報を説明します。

28.1. IDENTITY MANAGEMENT ファイルおよびログ

Identity Management は、異なる Linux サービスを単一の管理コンテキストに統合する統合フレームワークです。ただし、Kerberos、DNS、389 Directory Server、Dogtag Certificate System などの基盤となる技術は、独自の設定ファイルとログファイルを保持します。Identity Management は、独自の設定ファイルおよびツールを使用して、これらの要素を直接管理します。

本セクションでは、IdM が使用するディレクトリー、ファイル、およびログを説明します。IdM で使用される特定サーバーの設定ファイルまたはログの詳細は、製品ドキュメントを参照してください。

28.1.1. IdM サーバー設定ファイルおよびディレクトリーのリファレンス

表28.1 IdM サーバー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
サーバー設定	
/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。
/etc/ipa/server.conf	IdM のオプション設定ファイルこれはデフォルトでは存在しませんが、IdM サーバーの起動時にカスタム設定を読み込むことができます。
/etc/ipa/cli.conf	IdM コマンドラインツールのオプション設定ファイル。これはデフォルトでは存在しませんが、**ipa** の使用時にカスタム設定を適用するために作成できます。
/etc/ipa/ca.crt	IdM サーバーの CA が発行する CA 証明書。
~/.ipa/	ユーザーが IdM コマンドを初めて実行したときに、システムユーザーのホームディレクトリーのローカルシステムに作成された、ユーザー固有の IdM ディレクトリー。
IdM ログ	

ディレクトリーまたはファイル	説明
~/.ipa/log/cli.log	XML-RPC 呼び出しで返されるエラーと IdM コマンドユーティリティーの応答に関するログファイルです。これは、IdM ユーザーとは異なる名前を持つツールを実行する**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。
/var/log/ipaclient-install.log	クライアントサービスのインストールログ。
/var/log/ipaserver-install.log	IdM サーバーのインストールログ。
/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
システムサービス	
/etc/rc.d/init.d/ipa/	IdM サーバーの init スクリプト。
Web UI	
/etc/ipa/html/	IdM Web UI が使用する HTML ファイルのメイン設定ディレクトリーにあるシンボリックリンクディレクトリー。
/etc/httpd/conf.d/ipa.conf /etc/httpd/conf.d/ipa-rewrite.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/ipa.keytab	Web UI サービス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メインディレクトリー。
/usr/share/ipa/ipa-rewrite.conf /usr/share/ipa/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
/usr/share/ipa/updates/	Identity Management の更新されたファイル、スキーマ、およびその他の要素が含まれます。
/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。

ディレクトリーまたはファイル	説明
/usr/share/ipa/ipaclient/	Firefox の自動設定機能にアクセスし、IdM Kerberos レルムで機能するように Firefox ブラウザーを設定するのに使用する JavaScript ファイルが含まれています。
/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 が使用するすべてのスクリプトが含まれます。
/var/log/httpd/	Apache Web サーバーのログファイル。
Kerberos	
/etc/krb5.conf	Kerberos サービスの設定ファイル
SSSD	
/usr/share/sssd/sssd.api.d/sssd-ipa.conf	SSSD が使用する IdM サーバー、IdM Directory Server、およびその他の IdM サービスの特定に使用される設定ファイル。
/var/log/sssd/	SSSD のログファイル。
389 ディレクトリーサーバー	
/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スキーマ、設定、およびデータベースファイルすべて。
/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
Dogtag Certificate System	
/etc/pki-ca/	IdM CA インスタンスのメインディレクトリー。
/var/lib/pki-ca/conf/CS.cfg	IdM CA インスタンスの主な設定ファイル。
/var/lib/dirsrv/slapd-PKI-IPA/	IdM CA が使用する Directory Server インスタンスに関連付けられたスキーマ、設定、およびデータベースファイルすべて。
/var/log/dirsrv/slapd-PKI-IPA/	IdM CA 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

ディレクトリーまたはファイル	説明
キャッシュファイル	
/var/cache/ipa/	IdM サーバーおよび IdM Kerberos パスワードデーモンのキャッシュファイル。
システムバックアップ	
/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。
/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。一般的には、これは SSSD 認証サービスの **sssd.conf** ファイルです。

28.1.2. IdM ドメインサービスとログローテーション

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。ファイルのサイズ、ログローテーションの間隔、およびログファイルが保持される期間など、ログロテーション設定は、389 Directory Server 設定を編集して、すべて設定できます。これは、Red Hat Directory Server Administrator's Guide に記載されています。

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。

- ネームド (DNS)
- httpd (Apache)
- tomcat6
- sssd
- krb5kdc (Kerberos ドメインコントローラー)

これらのポリシーのほとんどは、ローテーションスケジュール (週) およびログのアーカイブ (4 週間で 4 週間) の **logrotate** デフォルトを使用します。

ログローテーション後にサービスを再起動する個々のポリシーは、欠落しているログファイルが受け入れ可能で、圧縮設定になります。

例28.1 デフォルトの httpd ログローテーションファイル

```
[root@server ~]# cat /etc/logrotate.d/httpd
/var/log/httpd/*log {
    missingok
    notifempty
```

```
    sharedscripts
    delaycompress
    postrotate
        /sbin/service httpd reload > /dev/null 2>/dev/null || true
    endscript
}
```

圧縮設定やログファイルのサイズなど、その他の潜在的なログ設定は、グローバルの **logrotate** 設定または個々のポリシーのいずれかで編集できます。この **logrotate** 設定は、logrotate の man ページで説明されています。

警告

2 つのポリシーは特殊な ***create*** ルールを設定します。すべてサービスでは、以前のログと同じ名前、デフォルトの所有者、デフォルトのパーミッションで新しいログファイル作成します。**named** および **tomcat6** ログの場合、**create** は明示的なパーミッションとユーザー/グループの所有権で設定されます。

```
[root@server ~]# cat /etc/logrotate.d/named
/var/named/data/named.run {
    missingok
    create 0644 named named
    postrotate
        /sbin/service named reload  2> /dev/null > /dev/null || true
    endscript
}
```

ログファイルを所有するユーザーおよびグループは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。ログ移動ポリシーまたはファイルの所有権を変更すると、IdM ドメインサービスが失敗したり、起動できなくなる可能性があります。

28.1.3. default.conf およびコンテキスト設定ファイル

レルム情報、LDAP 設定、CA 設定など、一部のグローバルデフォルトは **default.conf** ファイルに保存されます。この設定ファイルは、IdM クライアントおよびサーバーが起動し、**ipa** コマンドが操作を実行する際に情報を提供するたびに参照されます。

default.conf ファイルのパラメーターは単なる **attribute=value** のペアです。属性は大文字と小文字を区別せず、順序を区別しません。

```
[global]
basedn=dc=example,dc=com
realm=EXAMPLE.COM
domain=example.com
xmlrpc_uri=https://server.example.com/ipa/xml
ldap_uri=ldapi://%2fvar%2frun%2fslapd-EXAMPLE-COM.socket
```

```
enable_ra=True
ra_plugin=dogtag
mode=production
```

設定属性を追加したり、グローバル値をオーバーライドする場合、ユーザーは **コンテキスト** 設定ファイルを追加することができます。IdM サーバーの開始時または **ipa** コマンドの実行時に、異なるオプションを作成する場合ために **server.conf** および **cli.conf** ファイルを別々に作成できます。IdM サーバーは、まず **server.conf** および **cli.conf** ファイルを確認してから **default.conf** ファイルを確認します。

/etc/ipa ディレクトリー内の設定ファイルは、システムのすべてのユーザーに適用されます。ユーザーは、ローカル IdM ディレクトリー ~/**.ipa**/ に **default.conf**、**server.conf**、または **cli.conf** ファイルを作成して、個別のオーバーライドを設定できます。このオプションのファイルは、ローカルの IdM サービス **default.conf** とマージされ、使用されます。

28.1.4. IdM サーバーログの確認

Identity Management は複数の異なる Linux サービスを統合するため、それらのサービスのネイティブログを使用して、それらのサービスの追跡とデバッグを行います。

その他のサービス (Apache、389 Directory Server、および Dogtag Certificate System) はすべて詳細なログとログレベルを持ちます。リターンコード、ログ形式、およびログレベルの詳細は、特定のサーバーのドキュメントを参照してください。

表28.2 IdM ログファイル

サービス	ログファイル	説明	追加情報
IdM サーバー	/var/log/ipaserver-install.log	サーバーインストールログ	
IdM サーバー	~/.ipa/log/cli.log	コマンドラインツールのログ	
IdM クライアント	/var/log/ipaclient-install.log	クライアントインストールログ	
Apache サーバー	/var/log/httpd/access_log /var/log/httpd/error_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。Web UI と XML-RPC コマンドラインインターフェースは Apache を使用するため、IdM 固有のメッセージは Apache メッセージとともにエラーログに記録されます。	Apache ログの章
Dogtag Certificate System	/var/log/pki-ca-install.log	IdM KRA のインストールログ	

サービス	ログファイル	説明	追加情報
Dogtag Certificate System	/var/log/pki-ca/debug /var/log/pki-ca/system /var/log/pki-ca/transactions /var/log/pki-ca/signedAudit	これらのログは、主に証明書操作に関連します。IdM では、これ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。	ロギングの章
389 ディレクトリーサーバー	/var/log/dirsrv/slapd-**REALM**/access /var/log/dirsrv/slapd-**REALM**/audit /var/log/dirsrv/slapd-**REALM**/errors	アクセスログとエラーログには、ドメイン Directory Server インスタンスのアクセスと操作の詳細情報が含まれます。エラーログ設定を変更して、非常に詳細な出力を提供できます。	アクセスログはバッファーされるため、サーバーはデフォルトでは 30 秒ごとにログでのみ書き込みます。 • サーバーおよびデータベースの監視 • ログエントリーの説明
389 ディレクトリーサーバー	/var/log/dirsrv/slapd-**REALM**/access /var/log/dirsrv/slapd-**REALM**/audit /var/log/dirsrv/slapd-**REALM**/errors	このディレクトリーサーバーインスタンスは、証明書情報を保存するために IdM CA により使用されます。ここでの操作データの多くは、server-replica の対話に関連します。	アクセスログはバッファーされるため、サーバーはデフォルトでは 30 秒ごとにログでのみ書き込みます。 • サーバーおよびデータベースの監視 • ログエントリーの説明
Kerberos	/var/log/krb5libs.log	これは、Kerberos 接続のプライマリーログファイルです。	この場所は **krb5.conf** ファイルで設定されるため、一部のシステムで異なる可能性があります。
Kerberos	/var/log/krb5kdc.log	これは、Kerberos KDC サーバーの主なログファイルです。	この場所は **krb5.conf** ファイルで設定されるため、一部のシステムで異なる可能性があります。

サービス	ログファイル	説明	追加情報
Kerberos	/var/log/kadmind.log	これは、Kerberos 管理システムサーバーの主なログファイルです。	この場所は **krb5.conf** ファイルで設定されるため、一部のシステムで異なる可能性があります。
DNS	/var/log/messages	DNS エラーメッセージは、他のシステムメッセージに含まれます。	DNS ロギングはデフォルトでは有効になって**いません**。DNS ロギングは、以下の **querylog** コマンドを実行して有効にします。 `/usr/sbin/rndc querylog` これにより、システムの **/var/log/messages** ファイルにログメッセージが書き込まれます。ログを無効にするには、**querylog** コマンドを実行します。

28.1.4.1. サーバーデバッグロギングの有効化

IdM サーバーのデバッグロギングが **server.conf** ファイルに設定されます。

注記

default.conf 設定ファイルを編集すると、IdM サーバーだけでなく、すべての IdM コンポーネントに影響します。

1. **server.conf** ファイルを編集するか、または作成します。

```
vim server.conf
```

2. **debug** 行を追加し、その値を true に設定します。

```
[global]
debug=True
```

3. Apache デーモンを再起動して、変更を読み込みます。

```
service httpd restart
```

28.1.4.2. コマンドライン操作のデバッグ

ipa コマンドによるコマンドライン操作は、**-v** オプションを指定してデバッグ情報を返すことができます。たとえば、以下のようになります。

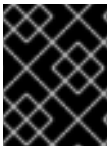
```
$ ipa -v user-show admin
ipa: INFO: trying https://ipaserver.example.com/ipa/xml
First name: John
Last name: Smythe
User login [jsmythe]:
ipa: INFO: Forwarding 'user_add' to server u'https://ipaserver.example.com/ipa/xml'
--------------------
Added user "jsmythe"
--------------------
  User login: jsmythe
  First name: John
  Last name: Smythe
  Full name: John Smythe
  Display name: John Smythe
  Initials: JS
  Home directory: /home/jsmythe
  GECOS field: John Smythe
  Login shell: /bin/sh
  Kerberos principal: jsmythe@EXAMPLE.COM
  UID: 1966800003
  GID: 1966800003
  Keytab: False
  Password: False
```

オプションを 2 回使用して (**-vv**)、XML-RPC 交換を表示します。

```
$ ipa -vv user-add

ipa: INFO: trying https://ipaserver.example.com/ipa/xml
First name: Jane
Last name: Russell
User login [jrussell]:
ipa: INFO: Forwarding 'user_add' to server u'https://ipaserver.example.com/ipa/xml'
send: u'POST /ipa/xml HTTP/1.0\r\nHost: ipaserver.example.com\r\nAccept-Language: en-
us\r\nAuthorization: negotiate
YIIFgQYJKoZIhvcSAQICAQBuggVwMIIFbKADAgEFoQMCAQ6iBwMFACAAAACjggGHYYIBgzCCAX+
gAwIBBaEZGxdSSFRTLkVORy5CT1MuUkVESEFULkNPTaI5MDegAwIBA6EwMC4bBEhUVFAbJmRl
bGwtcGUxODUwLTAxLnJodHMuZW5nLmJvcy5yZWRoYXQuY29to4IBIDCCARygAwIBEqEDAgECooI
BDgSCAQpV2YEWv03X+SENdUBfOhMFGc3Fvnd51nELV0rIB1tfGVjpNlkuQxXKSfFKVD3vyAUqkii255
T0mnXyTwayE93W1U4sOshetmG50zeU4KDmhuupzQZSCb5xB0KPU4HMDvP1UnDFJUGCk9tcqDJi
YE+lJrEcz8H+Vxvvl+nP6yIdUQKqoEuNhJaLWIiT8ieAzk8zvmDlDzpFYtInGWe9D5ko1Bb7Npu0SEpdVJ
B2gnB5vszCIdLlzHM4JUqX8p21AZV0UYA6QZOWX9OXhqHdElKcuHCN2s9FBRoFYK83gf1voS7xSFl
zZaFsEGHNmdA0qXbzREKGqUr8fmWmNvBGpDiR2ILQep09lL56JqSCA8owggPGoAMCARKiggO9BI
IDuarbB67zjmBu9Ax2K+0klSD99pNv97h9yxol8c6NGLB4CmE8Mo39rL4MMXHeOS0OCbn+TD97XVG
Lu+cgkfVcuIG4PMMBoajuSnPmIf7qDvfa8IYDFlDDnRB7I//IXtCc/Z4rBbaxk0SMIRLrsKf5wha7aWtN1Jbi
zBMQw+J0UlN8JjsWxu0Ls75hBtIDbPf3fva3vwBf7kTBChBsheewSAlck9qUglyNxAODgFVvRrXbfkw51U
o++9qHnhh+zFSWepfv7US7RYK0KxOKFd+uauY1ES+xlnMvK18ap2pcy0odBkKu1kwJDND0JXUdSY0
MxK2zb/UGWrVEf6GIsBgu122UGiHp6C+0fEu+nRrvtORY65Bgi8E1vm55fbb/9dQWNcQheL9m6QJWP
w0rgc+E5SO99ON6x3Vv2+Zk17EmbZXinPd2tDe7fJ9cS9o/z7Qjw8z8vvSzHL4GX7FKi2HJdBST3nEgO
C8PqO46UnAJcA8pf1ZkwCK9xDWH+5PSph6WnvpRqugqf/6cq+3jk3MEjCrx+JBJ8QL6AgN3oEB4kvjZp
AC+FfTkdX59VLDwfL/r0gMw3ZNk0nLLCLkiiYUMTEHZBzJw9kFbsX3LmS8qQQA6rQ2L782DYisElywf
Z/0Sax8JO/G62Zvy7BHy7SQSGIvcdAOafeNyfxaWM1vTpvSh0GrnllYfs3FhZAKnVcLYrlPTapR23uLgR
```

Mv+0c9wAbwuUDfKgOOl5vAd1j55VUyapgDEzi/URsLdVdcF4zigt4KrTByCwU2/pI6FmEPqB2tsjM2A8J
mqA+9Nt8bjaNdNwCOWE0dF50zeL9P8oodWGkbRZLk4DLIurpCW1d6IyTBhPQ5qZqHJWeoGiFa5y9
4zBpp27goMPmE0BskXT0JQmveYflOeKEMSzyiWPL2mwi7KEMtfgCpwTIGP2LRE/QxNvPGkwFfO+P
DjZGVw+APKkMKqcIVXxhtJA/2NmBrO1pZIIJ9R+41sR/QoACcXIUXJnhrTwwR1viKCB5Tec87gN+e0Cf
0g+fmZuXNRscwJfhYQJYwJqdYzGtZW+h8jDWqa2EPcDwIQwyFAgXNQ/aMvh1yNTECpLEgrMhYmF
AUDLQzI2BDnfbDftIs0rXjSC0oZn/Uaoqdr4F5syOrYAxH47bS6MW8CxyylreH8nT2qQXjenakLFHcNjt4
M1nOc/igzNSeZ28qW9WSr4bCdkH+ra3BVpT/AF0WHWkxGF4vWr/iNHCjq8fLF+DsAEx0Zs696Rg0fW
Zy079A\r\nUser-Agent: xmlrpclib.py/1.0.1 (by www.pythonware.com)\r\nContent-Type:
text/xml\r\nContent-Length: 1240\r\n\r\n'
send: "<?xml version='1.0' encoding='UTF-8'?
>\n<methodCall>\n<methodName>user_add</methodName>\n<params>\n<param>\n<value>
<array><data>\n<value><string>jrussell</string></value>\n</data></array>
</value>\n</param>\n<param>\n<value><struct>\n<member>\n<name>all</name>\n<value>
<boolean>0</boolean></value>\n</member>\n<member>\n<name>displayname</name>\n<value>
<string>Jane Russell</string></value>\n</member>\n<member>\n<name>cn</name>\n<value>
<string>Jane Russell</string>
</value>\n</member>\n<member>\n<name>noprivate</name>\n<value><boolean>0</boolean>
</value>\n</member>\n<member>\n<name>uidnumber</name>\n<value><int>999</int>
</value>\n</member>\n<member>\n<name>raw</name>\n<value><boolean>0</boolean>
</value>\n</member>\n<member>\n<name>version</name>\n<value><string>2.11</string>
</value>\n</member>\n<member>\n<name>gecos</name>\n<value><string>Jane Russell</string>
</value>\n</member>\n<member>\n<name>sn</name>\n<value><string>Russell</string>
</value>\n</member>\n<member>\n<name>krbprincipalname</name>\n<value>
<string>jrussell@EXAMPLE.COM</string>
</value>\n</member>\n<member>\n<name>givenname</name>\n<value><string>Jane</string>
</value>\n</member>\n<member>\n<name>initials</name>\n<value><string>JR</string>
</value>\n</member>\n</struct></value>\n</param>\n</params>\n</methodCall>\n"
reply: 'HTTP/1.1 200 OK\r\n'
header: Date: Thu, 15 Sep 2011 00:50:39 GMT
header: Server: Apache/2.2.15 (Red Hat)
header: WWW-Authenticate: Negotiate
YIGZBgkqhkiG9xIBAgICAG+BiTCBhqADAgEFoQMCAQ+iejB4oAMCARKicQRvVl5x6Zt9PbWNzvPEW
kdu+3PTCq/ZVKjGHM+1zDBz81GL/f+/Pr75zTuveLYn9de0C3k27vz96fn2HQsy9qVH7sfqn0RWGQWzl
+kDkuD6bJ/Dp/mpJvicW5gSkCSH6/UCNuE4I0xqwabLIz8MM/5o
header: Connection: close
header: Content-Type: text/xml; charset=utf-8
body: "<?xml version='1.0' encoding='UTF-8'?
>\n<methodResponse>\n<params>\n<param>\n<value>
<struct>\n<member>\n<name>result</name>\n<value>
<struct>\n<member>\n<name>dn</name>\n<value>
<string>uid=jrussell,cn=users,cn=accounts,dc=example,dc=com</string>
</value>\n</member>\n<member>\n<name>has_keytab</name>\n<value><boolean>0</boolean>
</value>\n</member>\n<member>\n<name>displayname</name>\n<value><array><data>\n<value>
<string>Jane Russell</string></value>\n</data></array>
</value>\n</member>\n<member>\n<name>uid</name>\n<value><array><data>\n<value>
<string>jrussell</string></value>\n</data></array>
</value>\n</member>\n<member>\n<name>objectclass</name>\n<value><array><data>\n<value>
<string>top</string></value>\n<value><string>person</string></value>\n<value>
<string>organizationalperson</string></value>\n<value><string>inetorgperson</string>
</value>\n<value><string>inetuser</string></value>\n<value><string>posixaccount</string>
</value>\n<value><string>krbprincipalaux</string></value>\n<value>
<string>krbticketpolicyaux</string></value>\n<"
body: 'value><string>ipaobject</string></value>\n</data></array>
</value>\n</member>\n<member>\n<name>loginshell</name>\n<value><array><data>\n<value>
<string>/bin/sh</string></value>\n</data></array>
</value>\n</member>\n<member>\n<name>uidnumber</name>\n<value><array><data>\n<value>

```
<string>1966800004</string></value>\n</data></array>
</value>\n</member>\n<member>\n<name>initials</name>\n<value><array><data>\n<value>
<string>JR</string></value>\n</data></array>
</value>\n</member>\n<member>\n<name>gidnumber</name>\n<value><array><data>\n<value>
<string>1966800004</string></value>\n</data></array>
</value>\n</member>\n<member>\n<name>gecos</name>\n<value><array><data>\n<value>
<string>Jane Russell</string></value>\n</data></array>
</value>\n</member>\n<member>\n<name>sn</name>\n<value><array><data>\n<value>
<string>Russell</string></value>\n</data></array>
</value>\n</member>\n<member>\n<name>homedirectory</name>\n<value><array>
<data>\n<value><string>/home/jrussell</string></value>\n</data></array>
</value>\n</member>\n<member>\n<name>has_password</name>\n<value><boolean>0</'
body: 'boolean></value>\n</member>\n<member>\n<name>krbprincipalname</name>\n<value>
<array><data>\n<value><string>jrussell@EXAMPLE.COM</string></value>\n</data></array>
</value>\n</member>\n<member>\n<name>givenname</name>\n<value><array><data>\n<value>
<string>Jane</string></value>\n</data></array>
</value>\n</member>\n<member>\n<name>cn</name>\n<value><array><data>\n<value>
<string>Jane Russell</string></value>\n</data></array>
</value>\n</member>\n<member>\n<name>ipauniqueid</name>\n<value><array><data>\n<value>
<string>bba27e6e-df34-11e0-a5f4-001143d2c060</string></value>\n</data></array>
</value>\n</member>\n</struct>
</value>\n</member>\n<member>\n<name>value</name>\n<value><string>jrussell</string>
</value>\n</member>\n<member>\n<name>summary</name>\n<value><string>Added user
"jrussell"</string></value>\n</member>\n</struct>
</value>\n</param>\n</params>\n</methodResponse>\n'
----------------------
Added user "jrussell"
----------------------
  User login: jrussell
  First name: Jane
  Last name: Russell
  Full name: Jane Russell
  Display name: Jane Russell
  Initials: JR
  Home directory: /home/jrussell
  GECOS field: Jane Russell
  Login shell: /bin/sh
  Kerberos principal: jrussell@EXAMPLE.COM
  UID: 1966800004
  GID: 1966800004
  Keytab: False
  Password: False
```



重要

-v と **-vv** オプションは **グローバル** オプションであり、**ipa** の実行時にサブコマンドの前に使用する必要があります。

28.2. 証明書と認証局の管理

ほぼすべての IdM トポロジーには、IdM ドメイン内のサーバー/レプリケーション、ホスト、ユーザー、およびサービスの証明書を管理する統合 Dogtag Certificate System が含まれます。

Dogtag Certificate System の設定自体に変更が加えられると、ドメインおよび物理マシンが変更される際に変更が必要になる場合があります。

注記

IdM 環境での複数の認証局(CA)署名証明書の使用は、Red Hat Enterprise Linux 6 ではサポートされません。この設定に対応するには、IdM システムを Red Hat Enterprise Linux 7 にアップグレードします。

28.2.1. 外部 CA が発行する CA 証明書の更新

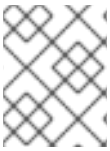
ホスト証明書およびユーザー証明書 (内部 IdM サービスで使用されるサブシステムおよびサーバー証明書を含む) などの IdM サーバーが発行するすべての証明書は、**certmonger** ユーティリティーにより追跡され、有効期限が近いときに自動的に更新されます。

CA 証明書自体の例外が 1 つあります。この証明書は有効期限が切れても自動的に更新されません。

警告

CA 証明書の有効期限が切れる前に、常に CA 証明書を更新するようにしてください。CA 証明書の有効期限は独自に監視する必要があることに注意してください。IdM は、Red Hat Enterprise Linux 6 で自動的に有効期限を監視しません。

CA 証明書は、発行元 CA を使用して更新してから、証明書データベース (**NSS データベース**とも呼ばれます) で手動で更新する必要があります。これは、**certutil** NSS セキュリティーユーティリティーを使用して行います。[8]



注記

IdM Web UI または IdM コマンドラインユーティリティーを使用して CA 証明書を更新することはできません。

証明書の更新には、いくつかの要件があります。

- 証明書を発行した外部 CA は更新を許可する必要があります。
- CA の秘密鍵は変更しないでください。
- 新しい証明書には、元の証明書と同じサブジェクト名を指定する必要があります。

新しい証明書を取得するには、元の CSR (証明書署名要求) が必要です。これは、以下の 3 つの場所のいずれかで確認できます。

- 外部 CA のコピーが依然としてある可能性があります。
- 最初にインストールした IdM サーバーの **/root/ipa.csr** ファイルで、
- 最初にインストールした IdM サーバーの **/etc/pki-ca/CS.cfg** ファイルの **ca.signing.certreq セクション**。これは PEM 形式に変換する必要があります。

また、NSS データベースで CA のニックネームを知っている必要があります。通常、これは **<REALM> IPA CA** です。ここでは **EXAMPLE.COM IPA CA** を使用します。Apache データベースに対してクエリーを実行して、現在のニックネームを見つけるには、以下のコマンドを実行します。

```
# certutil -L -d /etc/httpd/alias
```

28.2.1.1. 更新手順

他の証明書がまだ有効である期間に更新を行う必要があります。独自のサブシステム証明書を更新するには、CA が稼働している必要があります。CA 証明書の有効期限が切れてから CA 証明書を更新しようとすると、有効期間が CA サブシステム証明書の有効期限を過ぎても IdM サーバーは機能しません。

証明書の更新
外部 CA に CSR を付与し、新しい証明書を発行します。作成された証明書が **/root/ipa.crt** ファイルに保存されることを前提とします。また、**/root/external-ca.pem** ファイルに PEM 形式の外部 CA 証明書チェーンが含まれていることを前提としています。更新の管理には、指定した IdM CA で更新を行う必要があります。最初にインストールされた IdM サーバーを特定する方法として、**subsystem.select** の値が **New** であるかどうかを確認することができます。

```
# grep subsystem.select /etc/pki-ca/CS.cfg
subsystem.select=New
```

別の方法として、**getcert list** のコマンドの出力で **renew_ca_cert** 保存後コマンドを検索する方法があります。

```
Number of certificates and requests being tracked: 8.
Request ID '20131125153455':
        status: MONITORING
        stuck: no
        key pair storage: type=NSSDB,location='/var/lib/pki-ca/alias',nickname='auditSigningCert cert-
pki-ca',token='NSS Certificate DB',pin='455536908955'
        certificate: type=NSSDB,location='/var/lib/pki-ca/alias',nickname='auditSigningCert cert-pki-
ca',token='NSS Certificate DB'
        CA: dogtag-ipa-renew-agent
        issuer: CN=Certificate Authority,O=EXAMPLE.COM
        subject: CN=CA Audit,O=EXAMPLE.COM
        expires: 2015-11-15 15:34:12 UTC
        pre-save command: /usr/lib64/ipa/certmonger/stop_pkicad
 post-save command: /usr/lib64/ipa/certmonger/renew_ca_cert "auditSigningCert cert-pki-ca"
 track: yes
        auto-renew: yes
...
```

最初にインストールした IdM サーバーに新しい CA 証明書をインストールする

1. 証明書を更新するには、CA をシャットダウンする必要があります。

   ```
   # service ipa stop
   ```

2. CA 証明書の NSS データベースを更新します。

   ```
   # certutil -A -d /var/lib/pki-ca/alias -n 'caSigningCert cert-pki-ca' -t CT,C,C -a -i /root/ipa.crt
   ```

3. **/etc/pki-ca/CS.cfg** の **ca.signing.cert** の値を置き換えます。これは、証明書の base64 値です。これを取得するには、BEGIN/END ブロックを **ipa.crt** から削除し、1 つの行に圧縮します。

4. Apache NSS データベースを更新します。

```
# certutil -A -d /etc/httpd/alias -n 'EXAMPLE.COM IPA CA'  -t CT,C,C -a -i /root/ipa.crt
```

5. LDAP サーバーインスタンスを更新します。

```
# certutil -A -d /etc/dirsrv/slapd-EXAMPLE-COM -n 'EXAMPLE.COM IPA CA' -t CT,C,C -a -i
/root/ipa.crt
# certutil -A -d /etc/dirsrv/slapd-PKI-IPA -n 'EXAMPLE.COM IPA CA' -t CT,C,C -a -i
/root/ipa.crt
```

6. ファイルシステムの CA 証明書を更新します。

```
# cp /root/ipa.crt /etc/ipa/ca.crt
# cat /root/ipa.crt /root/external-ca.pem >/etc/httpd/alias/cacert.asc
# cp /etc/httpd/alias/cacert.asc /usr/share/ipa/html/ca.crt
```

7. 共有システムデータベースを更新します。

```
# certutil -A -d /etc/pki/nssdb -n 'IPA CA' -t CT,C,C -a -i /root/ipa.crt
```

8. サービスを再起動します。

```
# service ipa start
```

9. LDAP で CA 証明書を更新します。まず、証明書を DER 形式に変換します。

```
# openssl x509 -outform DER -in /root/ipa.crt  -out /tmp/ipa.der
```

10. 証明書を LDAP に追加します。

```
# kinit admin
# ldapmodify -Y GSSAPI
SASL/GSSAPI authentication started
SASL username: admin@EXAMPLE.COM
SASL SSF: 56
SASL data security layer installed.
dn: cn=CAcert,cn=ipa,cn=etc,dc=example,dc=com
changetype: modify
replace: cacertificate;binary
cacertificate;binary:<file:///tmp/ipa.der
```

CA を使用する他の IdM サーバーに新しい CA 証明書をインストールする

1. 更新された証明書をマシンにコピーし、サービスを停止します。ファイルは **/root/ipa.crt** と想定します。

```
# service ipa stop
```

2. Apache NSS データベースを更新します。

```
# certutil -A -d /var/lib/pki-ca/alias -n 'caSigningCert cert-pki-ca' -t CT,C,C -a -i /root/ipa.crt
```

3. **/etc/pki-ca/CS.cfg** の **ca.signing.cert** の値を置き換えます。これは、証明書の base64 値です。これを取得するには、BEGIN/END ブロックを **ipa.crt** から削除し、1 つの行に圧縮します。

4. Apache NSS データベースを更新します。

```
# certutil -A -d /etc/httpd/alias -n 'EXAMPLE.COM IPA CA'  -t CT,C,C -a -i /root/ipa.crt
```

5. LDAP サーバーインスタンスを更新します。

```
# certutil -A -d /etc/dirsrv/slapd-EXAMPLE-COM -n 'EXAMPLE.COM IPA CA' -t CT,C,C -a -i
/root/ipa.crt
# certutil -A -d /etc/dirsrv/slapd-PKI-IPA -n 'EXAMPLE.COM IPA CA' -t CT,C,C -a -i
/root/ipa.crt
```

6. ファイルシステムの CA 証明書を更新します。

```
# cp /root/ipa.crt /etc/ipa/ca.crt
# cat /root/ipa.crt /root/external-ca.pem >/etc/httpd/alias/cacert.asc
# cp /etc/httpd/alias/cacert.asc /usr/share/ipa/html/ca.crt
```

7. 共有システムデータベースを更新します。

```
# certutil -A -d /etc/pki/nssdb -n 'IPA CA' -t CT,C,C -a -i /root/ipa.crt
```

8. サービスを再起動します。

```
# service ipa start
```

CA を使用しない他の IdM マスターに新しい CA 証明書をインストールする

1. 更新された証明書をマシンにコピーし、サービスを停止します。ファイルは **/root/ipa.crt** と想定します。

```
# service ipa stop
```

2. Apache NSS データベースを更新します。

```
# certutil -A -d /etc/httpd/alias -n 'EXAMPLE.COM IPA CA'  -t CT,C,C -a -i /root/ipa.crt
```

3. LDAP サーバーインスタンスを更新します。

```
# certutil -A -d /etc/dirsrv/slapd-EXAMPLE-COM -n 'EXAMPLE.COM IPA CA' -t CT,C,C -a -i
/root/ipa.crt
# certutil -A -d /etc/dirsrv/slapd-PKI-IPA -n 'EXAMPLE.COM IPA CA' -t CT,C,C -a -i
/root/ipa.crt
```

4. ファイルシステムの CA 証明書を更新します。

```
# cp /root/ipa.crt /etc/ipa/ca.crt
# cat /root/ipa.crt /root/external-ca.pem >/etc/httpd/alias/cacert.asc
# cp /etc/httpd/alias/cacert.asc /usr/share/ipa/html/ca.crt
```

5. 共有システムデータベースを更新します。

```
# certutil -A -d /etc/pki/nssdb -n 'IPA CA' -t CT,C,C -a -i /root/ipa.crt
```

6. サービスを再起動します。

```
# service ipa start
```

すべての IdM クライアントマシンに新しい CA 証明書をインストールする

更新された IdM CA 証明書を取得します。ファイルは **/tmp/ipa.crt** と想定します。

```
# certutil -A -d /etc/pki/nssdb -n 'IPA CA' -t CT,C,C -a -i /tmp/ipa.crt
# cp /tmp/ipa.crt /etc/ipa/ca.crt
```

28.2.2. IdM CA が発行する CA 証明書の更新

ホスト証明書およびユーザー証明書 (内部 IdM サービスで使用されるサブシステムおよびサーバー証明書を含む) などの IdM サーバーが発行するすべての証明書は、**certmonger** ユーティリティーにより追跡され、有効期限が近いときに自動的に更新されます。

CA 証明書自体の例外が 1 つあります。この証明書は有効期限が切れても自動的に更新されません。

警告

CA 証明書の有効期限が切れる前に、常に CA 証明書を更新するようにしてください。CA 証明書の有効期限は独自に監視する必要があることに注意してください。IdM は、Red Hat Enterprise Linux 6 で自動的に有効期限を監視しません。

28.2.2.1. 更新手順

他の証明書がまだ有効である期間に更新を行う必要があります。独自のサブシステム証明書を更新するには、CA が稼働している必要があります。CA 証明書の有効期限が切れてから CA 証明書を更新しようとすると、有効期間が CA サブシステム証明書の有効期限を過ぎても IdM サーバーは機能しません。

IdM CA の署名証明書を更新し、最初にインストールした IdM サーバーに新しい CA 証明書のインストールする

1. IPA が停止していることを確認します。

```
# ipactl status
# ipactl stop
```

2. **ntpd** が実行していないことを確認します。

```
# service ntpd status
# service ntpd stop
```

3. Directory Server を起動し、これが実行していることを確認します。

```
# service dirsrv start
# service dirsrv status
```

4. Dogtag CA を起動し、これが実行されていることを確認します。

```
# service pki-cad start
# service pki-cad status
```

5. **dogtag-ipa-renew-agent-submit** コマンドを入力して、certmonger ヘルパー経由で Dogtag CA 署名証明書を直接更新します。

```
# /usr/libexec/certmonger/dogtag-ipa-renew-agent-submit -D 1 -T caCACert | tail -n 1 | xargs
/usr/libexec/certmonger/dogtag-ipa-renew-agent-submit -d /etc/httpd/alias -n ipaCert -p
/etc/httpd/alias/pwdfile.txt -v -S
```

6. CA 証明書の NSS データベースを更新します。

```
# certutil -A -d /var/lib/pki-ca/alias -n 'caSigningCert cert-pki-ca' -t CT,C,C -a -i /root/ipa.crt
```

7. **/etc/pki-ca/CS.cfg** の **ca.signing.cert** の値を置き換えます。これは、証明書の base64 値です。これを取得するには、BEGIN/END ブロックを **ipa.crt** から削除し、1 つの行に圧縮します。

8. Apache NSS データベースを更新します。

```
# certutil -A -d /etc/httpd/alias -n 'EXAMPLE.COM IPA CA'  -t CT,C,C -a -i /root/ipa.crt
```

9. LDAP サーバーインスタンスを更新します。

```
# certutil -A -d /etc/dirsrv/slapd-EXAMPLE-COM -n 'EXAMPLE.COM IPA CA' -t CT,C,C -a -i
/root/ipa.crt
  # certutil -A -d /etc/dirsrv/slapd-PKI-IPA -n 'EXAMPLE.COM IPA CA' -t CT,C,C -a -i
/root/ipa.crt
```

10. ファイルシステムの CA 証明書を更新します。

```
# cp /root/ipa.crt /etc/ipa/ca.crt
  # cat /root/ipa.crt /root/external-ca.pem >/etc/httpd/alias/cacert.asc
  # cp /etc/httpd/alias/cacert.asc /usr/share/ipa/html/ca.crt
```

11. 共有システムデータベースを更新します。

```
# certutil -A -d /etc/pki/nssdb -n 'IPA CA' -t CT,C,C -a -i /root/ipa.crt
```

12. サービスを再起動します。

```
# ipactl start
```

13. LDAP で CA 証明書を更新します。まず、証明書を DER 形式に変換します。

```
# openssl x509 -outform DER -in /root/ipa.crt  -out /tmp/ipa.der
```

14. 証明書を LDAP に追加します。

```
# kinit admin
  # ldapmodify -Y GSSAPI
  SASL/GSSAPI authentication started
  SASL username: admin@EXAMPLE.COM
  SASL SSF: 56
  SASL data security layer installed.
  dn: cn=CAcert,cn=ipa,cn=etc,dc=example,dc=com
  changetype: modify
  replace: cacertificate;binary
  cacertificate;binary:<file:///tmp/ipa.der
```

15. **ipa-getcert list** を使用して、certmonger が追跡するすべての要求の一覧を表示します。

```
# ipa-getcert list
```

16. この出力で、サブシステム証明書のいずれかがすでに期限切れであることが示されたら、それぞれの証明書に対して個別に **ipa-getcert resubmit** を使用して証明書を更新します。詳細は、ナレッジベースソリューション「Dealing with expiring IDM CA certificates on Red Hat Enterprise Linux 6 and 7」を参照してください。

CA を使用する他の IdM サーバーに新しい CA 証明書をインストールする

1. 更新された証明書をマシンにコピーし、サービスを停止します。ファイルは **/root/ipa.crt** と想定します。

```
# service ipa stop
```

2. Apache NSS データベースを更新します。

```
# certutil -A -d /var/lib/pki-ca/alias -n 'caSigningCert cert-pki-ca' -t CT,C,C -a -i /root/ipa.crt
```

3. **/etc/pki-ca/CS.cfg** の **ca.signing.cert** の値を置き換えます。これは、証明書の base64 値です。これを取得するには、BEGIN/END ブロックを **ipa.crt** から削除し、1 つの行に圧縮します。

4. Apache NSS データベースを更新します。

```
# certutil -A -d /etc/httpd/alias -n 'EXAMPLE.COM IPA CA'  -t CT,C,C -a -i /root/ipa.crt
```

5. LDAP サーバーインスタンスを更新します。

```
# certutil -A -d /etc/dirsrv/slapd-EXAMPLE-COM -n 'EXAMPLE.COM IPA CA' -t CT,C,C -a -i
/root/ipa.crt
  # certutil -A -d /etc/dirsrv/slapd-PKI-IPA -n 'EXAMPLE.COM IPA CA' -t CT,C,C -a -i
```

```
/root/ipa.crt
```

6. ファイルシステムの CA 証明書を更新します。

```
# cp /root/ipa.crt /etc/ipa/ca.crt
  # cat /root/ipa.crt /root/external-ca.pem >/etc/httpd/alias/cacert.asc
  # cp /etc/httpd/alias/cacert.asc /usr/share/ipa/html/ca.crt
```

7. 共有システムデータベースを更新します。

```
# certutil -A -d /etc/pki/nssdb -n 'IPA CA' -t CT,C,C -a -i /root/ipa.crt
```

8. サービスを再起動します。

```
# service ipa start
```

CA を使用しない他の IdM マスターに新しい CA 証明書をインストールする

1. 更新された証明書をマシンにコピーし、サービスを停止します。ファイルは **/root/ipa.crt** と想定します。

```
# service ipa stop
```

2. Apache NSS データベースを更新します。

```
# certutil -A -d /etc/httpd/alias -n 'EXAMPLE.COM IPA CA'  -t CT,C,C -a -i /root/ipa.crt
```

3. LDAP サーバーインスタンスを更新します。

```
# certutil -A -d /etc/dirsrv/slapd-EXAMPLE-COM -n 'EXAMPLE.COM IPA CA' -t CT,C,C -a -i
/root/ipa.crt
  # certutil -A -d /etc/dirsrv/slapd-PKI-IPA -n 'EXAMPLE.COM IPA CA' -t CT,C,C -a -i
/root/ipa.crt
```

4. ファイルシステムの CA 証明書を更新します。

```
# cp /root/ipa.crt /etc/ipa/ca.crt
  # cat /root/ipa.crt /root/external-ca.pem >/etc/httpd/alias/cacert.asc
  # cp /etc/httpd/alias/cacert.asc /usr/share/ipa/html/ca.crt
```

5. 共有システムデータベースを更新します。

```
# certutil -A -d /etc/pki/nssdb -n 'IPA CA' -t CT,C,C -a -i /root/ipa.crt
```

6. サービスを再起動します。

```
# service ipa start
```

すべての IdM クライアントマシンに新しい CA 証明書をインストールする

更新された IdM CA 証明書を取得します。ファイルは **/tmp/ipa.crt** と想定します。

```
# certutil -A -d /etc/pki/nssdb -n 'IPA CA' -t CT,C,C -a -i /tmp/ipa.crt
  # cp /tmp/ipa.crt /etc/ipa/ca.crt
```

28.2.3. 代替認証局の設定

IdM は、サーバーのインストールプロセス時に Dogtag Certificate System 認証局 (CA) を作成します。外部 CA を使用するには、必要なサーバー証明書を作成してから、IdM サーバー証明書を必要とする 389 Directory Server および HTTP サーバーにインポートすることができます。

ヒント

に従って、CA 証明書の ASCII コピーを **/usr/share/ipa/html/ca.crt** として保存します。これにより、ユーザーはブラウザーの設定時に正しい証明書をダウンロードできます。

1. **ipa-server-certinstall** コマンドを使用して、証明書をインストールします。

   ```
   # /usr/sbin/ipa-server-certinstall -d /path/to/pkcs12.p12
   ```

2. Firefox でブラウザーの自動設定を使用し続けるには、**/usr/share/ipa/html/configure.jar** ファイルを再生成します。

   a. ディレクトリーを作成し、そのディレクトリーに新しいセキュリティーデータベースを作成します。

   ```
   # mkdir /tmp/signdb

   # certutil -N -d /tmp/signdb
   ```

   b. 署名証明書の PKCS #12 ファイルをそのディレクトリーにインポートします。

   ```
   # pk12util -i /path/to/pkcs12.p12 -d /tmp/signdb
   ```

   c. 一時的な署名ディレクトリーを作成し、IdM JavaScript ファイルをそのディレクトリーにコピーします。

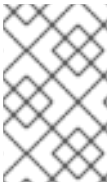
   ```
   # mkdir /tmp/sign
   # cp /usr/share/ipa/html/preferences.html /tmp/sign
   ```

   d. オブジェクト署名証明書を使用して JavaScript ファイルに署名し、**configure.jar** ファイルを再生成します。

   ```
   # signtool -d /tmp/signdb -k Signing_cert_nickname -Z /usr/share/ipa/html/configure.jar -e
   .html /tmp/sign
   ```

28.2.4. CRL を生成するサーバーの変更

マスター CA は権威 CA です。これにはルート CA 署名キーがあり、トポロジー内の他のサーバーおよびレプリカに配布される CRL を生成します。通常、最初にインストールされた IdM サーバーは、PKI 階層でマスター CA を所有します。後続のレプリカデータベースはすべて、**ipa-replica-install** の実行時にそのマスターデータベースから直接クローン (またはコピー) さます。



注記

マスターサーバーを置き換える唯一の理由は、マスターサーバーがオフラインの場合に限られます。CRL を発行し、最終的に証明書チェックを検証することができるルート CA が必要です。

「IdM サーバーおよびレプリカの概要」の説明にあるように、すべてのサーバーとレプリカが連携してデータを共有します。この配置は **サーバートポロジー** です。

サーバー (**ipa-server-install** で作成したもの) は、認証局サービスをホストするためにほぼ常に作成されます。[9]これらは **元** の CA サービスです。レプリカが (**ipa-replica-install**) 作成されるとき、これは既存サーバーの設定に基づきます。レプリカは CA サービスをホス **できます** が、これは必須ではありません。

作成後、サーバーとレプリカはサーバートポロジーのピアと等しくなります。これらはすべて読み取り/書き込みデータマスターであり、マルチマスターレプリケーションを介して情報を相互に複製します。CA をホストするサーバーおよびレプリカも、トポロジーのピアと同等です。IdM クライアントに証明書と鍵をすべて発行し、それぞれで情報を複製できます。

サーバーとレプリカの違いは、CRL を発行する IdM インスタンスのみです。

最初のサーバーがインストールされると、CRL を発行するよう設定されます。CA 設定ファイル **/var/lib/pki-ca/conf/CS.cfg** では、CRL 生成が有効にされています。

```
ca.crl.issuingPointId.enableCRLCache=true
ca.crl.issuingPointId.enableCRLUpdates=true
ca.listenToCloneModifications=false
```

すべてのレプリカは、マスター CA を CRL 情報のソースとして参照し、CRL 設定を無効にします。

```
ca.crl.issuingPointId.enableCRLUpdates=false
```

CRL を発行する IdM トポロジーにあるインスタンスが 1 つ必要です。元のサーバーがオフラインまたは使用を停止する場合は、レプリカをオフラインにするか、または使用停止するように設定する必要があります。レプリカをマスターサーバーに **プロモート** すると、設定が変更され、CRL を発行し、ルート CA として機能できるようになります。

CRL 生成をサーバーからレプリカに移動するには、**まず元のマスター CA を廃止します。**

1. マスター CA サーバーであるサーバーインスタンスを特定します。CRL の生成 **および** 更新操作は、いずれも同じ CA サーバーによって処理されます。したがって、マスター CA は、**certmonger** で追跡されている **renew_ca_cert** 証明書を利用して識別できます。

```
[root@server ~]# 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"
```

2. **元のマスター CA で**、元の CA 証明書の追跡を無効にします。

```
[root@server ~]# getcert stop-tracking -d /var/lib/pki-ca/alias -n "auditSigningCert cert-pki-ca"
Request "20131127184547" removed.
[root@server ~]# getcert stop-tracking -d /var/lib/pki-ca/alias -n "ocspSigningCert cert-pki-ca"
Request "20131127184548" removed.
[root@server ~]# getcert stop-tracking -d /var/lib/pki-ca/alias -n "subsystemCert cert-pki-ca"
```

```
Request "20131127184549" removed.
[root@server ~]# getcert stop-tracking -d /etc/httpd/alias -n ipaCert
Request "20131127184550" removed.
```

3. 元のマスター CA を再設定し、新規マスター CA から更新された証明書を取得します。

   a. 更新ヘルパーを **certmonger** 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@server ~]# cp /usr/share/ipa/ca_renewal /var/lib/certmonger/cas/ca_renewal
   [root@server ~]# chmod 0600 /var/lib/certmonger/cas/ca_renewal
   ```

   b. SELinux 設定を更新します。

   ```
   [root@server ~]# /sbin/restorecon  /var/lib/certmonger/cas/ca_renewal
   ```

   c. **certmonger** を再起動します。

   ```
   [root@server ~]# 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。これは CA 設定で出力されます。

   ```
   [root@server ~]# 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@server ~]# 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@server ~]# 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 "20131127184743" added.
   [root@server ~]# 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 "20131127184744" added.
   [root@server ~]# 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 "20131127184745" added.
   [root@server ~]# 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 "20131127184746" added.
```

4. 元のマスター CA で CRL 生成を停止します。

   a. CA サービスを停止します。

   ```
   [root@server ~]# service pki-cad stop
   ```

   b. CA 設定ファイルを開きます。

   ```
   [root@server ~]# vim /var/lib/pki-ca/conf/CS.cfg
   ```

   c. ***ca.crl.MasterCRL.enableCRLCache*** および ***ca.crl.MasterCRL.enableCRLUpdates*** パラメーターの値を **false** に変更して CRL 生成を無効にします。

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

   d. CA サービスを起動します。

   ```
   [root@server ~]# service pki-cad start
   ```

5. CRL 要求を新規マスターにリダイレクトするように Apache を設定します。

   a. CA プロキシー設定を開きます。

   ```
   [root@server ~]# vim /etc/httpd/conf.d/ipa-pki-proxy.conf
   ```

   b. 最後の行で **RewriteRule** のコメントを解除します。

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

   c. Apache を再起動します。

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

次に、レプリカを新しいマスターとして設定します。

1. CA の証明書の追跡を停止して、更新設定を変更します。クローンとして、CA はマスターから更新された証明書を取得するよう設定されています。マスター CA として、更新した証明書を発行します。

```
[root@server ~]# getcert stop-tracking -d /var/lib/pki-ca/alias -n "auditSigningCert cert-pki-ca"
Request "20131127163822" removed.
[root@server ~]# getcert stop-tracking -d /var/lib/pki-ca/alias -n "ocspSigningCert cert-pki-ca"
Request "20131127163823" removed.
[root@server ~]# getcert stop-tracking -d /var/lib/pki-ca/alias -n "subsystemCert cert-pki-ca"
```

```
Request "20131127163824" removed.
[root@server ~]# getcert stop-tracking -d /etc/httpd/alias -n ipaCert
Request "20131127164042" removed.
```

2. CA 証明書データベースの PIN を取得します。

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

3. 更新エージェントプロファイルを使用して **certmonger** で追跡される証明書を設定します。

```
[root@server ~]# getcert start-tracking -c dogtag-ipa-renew-agent -d /var/lib/pki-ca/alias -n
"auditSigningCert cert-pki-ca" -B /usr/lib64/ipa/certmonger/stop_pkicad -C
'/usr/lib64/ipa/certmonger/renew_ca_cert "auditSigningCert cert-pki-ca"' -P database_pin
New tracking request "20131127185430" added.
[root@server ~]# getcert start-tracking -c dogtag-ipa-renew-agent -d /var/lib/pki-ca/alias -n
"ocspSigningCert cert-pki-ca" -B /usr/lib64/ipa/certmonger/stop_pkicad -C
'/usr/lib64/ipa/certmonger/renew_ca_cert "ocspSigningCert cert-pki-ca"' -P database_pin
New tracking request "20131127185431" added.
[root@server ~]# getcert start-tracking -c dogtag-ipa-renew-agent -d /var/lib/pki-ca/alias -n
"subsystemCert cert-pki-ca" -B /usr/lib64/ipa/certmonger/stop_pkicad -C
'/usr/lib64/ipa/certmonger/renew_ca_cert "subsystemCert cert-pki-ca"' -P database_pin
New tracking request "20131127185432" added.
[root@server ~]# getcert start-tracking -c dogtag-ipa-renew-agent -d /etc/httpd/alias -n
ipaCert -C /usr/lib64/ipa/certmonger/renew_ra_cert -p /etc/httpd/alias/pwdfile.txt
New tracking request "20131127185433" added.
```

4. 新規マスター CA が CRL を生成するように設定します。

   a. CA サービスを停止します。

   ```
   [root@server ~]# service pki-cad stop
   ```

   b. CA 設定ファイルを開きます。

   ```
   [root@server ~]# vim /var/lib/pki-ca/conf/CS.cfg
   ```

   c. ***ca.crl.MasterCRL.enableCRLCache*** および ***ca.crl.MasterCRL.enableCRLUpdates*** パラメーターの値を **true** に変更して、CRL 生成を有効にします。

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

   d. CA サービスを起動します。

   ```
   [root@server ~]# service pki-cad start
   ```

5. Apache を設定してリダイレクト CRL 要求を無効にします。クローンとして、すべての CRL 要求は元のマスターにルーティングされました。新規マスターとして、このインスタンスは CRL リクエストに応答します。

   a. CA プロキシー設定を開きます。

```
[root@server ~]# vim /etc/httpd/conf.d/ipa-pki-proxy.conf
```

b. 最後の行で **RewriteRule** 引数をコメントアウトします。

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

c. Apache を再起動します。

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

28.2.5. OCSP 応答の設定

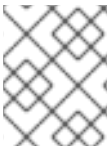
証明書は有効期間で作成されます。つまり、証明書は期限切れになり、有効ではなくなります。有効期限は証明書自体に含まれるため、クライアントは常に証明書の有効期間を確認して、証明書がまだ有効かどうかを確認します。

ただし、有効期間が切れる前に証明書を取り消すこともできますが、この情報は証明書に含まれていません。CA は、**証明書失効リスト** (CRL)を公開します。これには、その CA によって発行されたすべての証明書の完全なリストが含まれ、その後に取り消されます。クライアントは CRL を確認して、有効期間内の証明書が取り消され、無効であるかどうかを確認できます。

有効性チェックは、**OCSP レスポンダー** にリクエストを送信するオンライン証明書ステータスプロトコル (OCSP) を使用して実行されます。IdM サーバーと統合されている各 CA は、内部の OCSP レスポンダーを使用し、有効性チェックを実行するクライアントは、IdM CA の内部 OCSP レスポンダーを確認できます。

IdM CA が発行するすべての証明書は、証明書に OCSP レスポンダーサービス URL を追加します。以下に例を示します。

```
http://ipaserver.example.com:9180/ca/ocsp
```



注記

IdM OCSP レスポンダーを利用できるようにするには、ファイアウォールで 9180 ポートを開く必要があります。

28.2.5.1. SELinux での OSCP レスポンダーの使用

クライアントは、Identity Management OCSP レスポンダーを使用して証明書の有効性を確認するか、CRL を取得することができます。クライアントはさまざまなサービスにすることができますが、最も頻繁に Apache サーバーと、CRL および OCSP 操作を処理する mod_revocator モジュールです。

Identity Management CA には、ポート 9180 経由でリッスンする OCSP レスポンダーがあります。これは CRL の取得に使用できるポートです。このポートはデフォルトで SELinux ポリシーで保護され、承認されていないアクセスを防ぐことができます。Apache サーバーが OCSP ポートへの接続を試みると、SELinux によるアクセスが拒否される可能性があります。

Identity Management OCSP レスポンダーに接続できるようにするため、ローカルマシンの Apache サーバーには 9180 ポートへのアクセスが付与される必要があります。これを回避するには、以下の 2 つの方法があります。

- SELinux ポリシーを編集して、mod_revocator モジュールを使用して Apache サーバーをポート 9180 に接続できるようにします。

```
semodule -i revoker.pp
```

- mod_revocator 接続の SELinux エラーログに基づいてアクセスを許可する新しい SELinux ポリシーを生成します。

```
audit2allow -a -M revoker
```

28.2.5.2. CRL 更新間隔の変更

CRL ファイルは、4 時間ごとに Dogtag Certificate System CA によって自動的に生成されます。この間隔は、Dogtag Certificate System 設定を編集して変更できます。

1. 認証局サーバーを停止します。

```
[root@server ~]# service pki-ca stop
```

2. **CS.cfg** ファイルを開きます。

```
[root@server ~]# vim /var/lib/pki-ca/conf/CS.cfg
```

3. ***ca.crl.MasterCRL.autoUpdateInterval*** を新しい間隔設定に変更します。

4. CA サーバーを再起動します。

```
[root@server ~]# service pki-ca start
```

28.2.5.3. OCSP レスポンダーの場所の変更

各 IdM サーバーは、独自の CRL を生成します。同様に、各 IdM サーバーは、発行する証明書の独自の OCSP レスポンダー URL とともに、独自の OCSP レスポンダーを使用します。

DNS CNAME は IdM クライアントが使用でき、そこから適切な IdM サーバーの OCSP レスポンダーにリダイレクトされます。

1. 証明書プロファイルを開きます。

```
[root@server ~]# vim /var/lib/pki-ca/profiles/ca/caIPAserviceCert.cfg
```

2. ***policyset.serverCertSet.9.default.params.crlDistPointsPointName_0*** パラメーターを DNS CNAME ホスト名に変更します。

3. CA サーバーを再起動します。

```
service pki-ca restart
```

この変更は、すべての IdM サーバーで行う必要があり、***crlDistPointsPointName_0*** パラメーターが同じホスト名に設定されます。

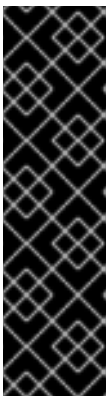
28.3. 匿名バインドの無効化

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することができます。

1. ***nsslapd-allow-anonymous-access*** 属性を **rootdse** に変更します。

```
ldapmodify -x -D "cn=Directory Manager" -w secret -h server.example.com -p 389

Enter LDAP Password:
dn: cn=config
changetype: modify
replace: nsslapd-allow-anonymous-access
nsslapd-allow-anonymous-access: rootdse
```



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。

```
service dirsrv restart
```

28.4. ドメイン DNS 設定の変更

28.4.1. マルチキューサーバーの DNS エントリーの設定

サーバーマシンによっては、複数のネットワークインターフェースカード (NIC) をサポートする場合があります。マルチホームマシンには、通常、複数の IP があり、すべて同じホスト名に割り当てられます。localhost を除く利用可能なすべてのインターフェースをリッスンするため、ほとんどの場合 IdM では問題なく機能します。サーバーを NIC 経由で利用できるようにするには、DNS ゾーンファイルを編集し、各 IP アドレスのエントリーを追加します。以下に例を示します。

```
ipaserver  IN A  192.168.1.100
ipaserver  IN A  192.168.1.101
ipaserver  IN A  192.168.1.102
```

28.4.2. ネームサーバーの追加設定

設定の完了時に、**/etc/resolv.conf** で設定済みのネームサーバーの一覧には、IdM サーバーのみが含まれます。ローカル **named** サービスがクラッシュした場合、IdM サーバーは実行できず、ドメイン全体の DNS サービスが利用できなくなりました。

その他の DNS サーバーは、IdM サーバーの **/etc/resolv.conf** ファイルに手動で追加する必要があります。

```
[root@server ~]# vim /etc/resolv.conf

search example.com

; the IdM server
nameserver 127.0.0.1

; backup DNS servers
nameserver 198.51.100.0
nameserver 192.0.2.0
```

注記

/etc/resolv.conf ファイルには、3 つのサーバーのデフォルト制限が設定されています。

/etc/resolv.conf ファイルの設定に関するその他の情報は **resolv.conf** man ページにあります。

28.4.3. IdM サーバーおよびレプリカの負荷分散の変更

「IdM サーバーおよびレプリカの概要」に示すように、ドメイン内の IdM サーバーおよびレプリカがインスタンス間の負荷を自動的に共有し、パフォーマンスを維持します。負荷分散は、まず SRV エントリーのサーバーまたはレプリカの **優先順位** で定義され、その後同じ優先順位を持つサーバー/レプリカに対するインスタンスの **重み** で定義されます。クライアントは優先度が最も高いサーバー/レプリカに問い合わせ、ダウン作業を行います。

負荷分散は、サーバー、レプリカ、およびクライアントによって自動的に行われます。負荷分散に使用される設定は、サーバーまたはレプリカに指定される優先順位と重みを変更することで変更できます。

(すべてのレプリカは最初に同じ優先度で作成されます。)

たとえば、これにより server1 が server 2 よりも優先度が高くなります。つまり、最初に接続されます。

```
$ ipa dnsrecord-add server.example.com _ldap._tcp --srv-rec="0 100 389 server1.example.com."

$ ipa dnsrecord-add server.example.com _ldap._tcp --srv-rec="1 100 389 server2.example.com."
```

SRV レコードの詳細は、RFC 2782 を参照してください。

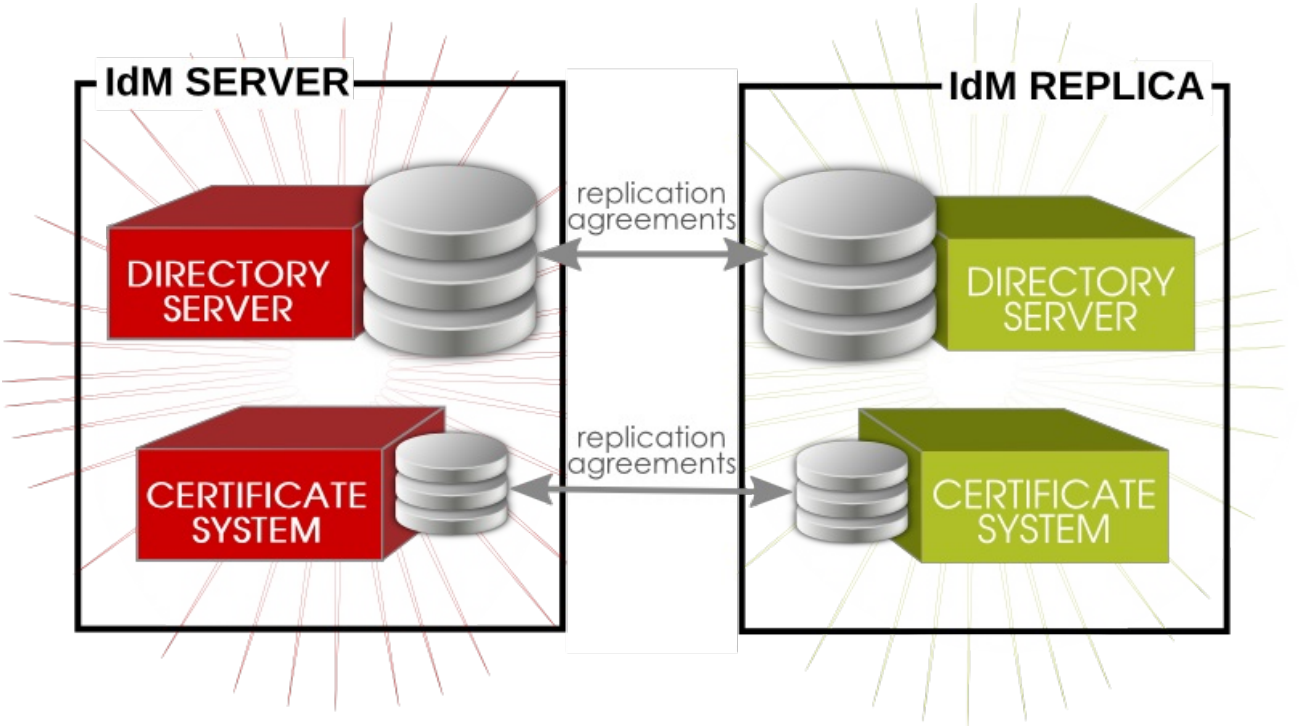
28.5. IDM サーバー間のレプリカ合意の管理

マルチマスターレプリケーションを使用して、IdM サーバーとレプリカ間で情報が共有されます。つまり、サーバーとレプリカがすべて更新を受け取るため、データマスターになります。ドメイン情報は、**レプリケーション** を使用してサーバーとレプリカ間でコピーされます。

レプリカがドメインに追加されると、レプリカと、そのレプリカが基になるサーバーとの間に相互レプリカ合意が自動的に作成されます。他のレプリカとサーバー間で追加のレプリカ合意を作成したり、**ipa-replica-manage** コマンドを使用してレプリカ合意の設定を変更したりできます。

レプリカを作成すると、レプリカのインストールスクリプトは 2 つのレプリカ合意を作成します。1 つはマスターサーバーからレプリカへのレプリカ合意と、レプリカからマスターサーバーへのレプリカ合意の 1 つです。

図28.1 サーバーとレプリカの合意



ドメインにレプリカとサーバーを追加すると、他のサーバーおよびレプリカにレプリカ合意を持つレプリカやサーバーがあり、相互にレプリカ合意があるわけではありません。たとえば、最初の IdM サーバーはサーバー A であり、管理者は Replica B を作成し、インストールスクリプトは Server A => レプリカ B レプリカ合意と Replica B => Server A レプリカ合意を作成します。次に、管理者はサーバー A に基づいて Replica C を作成します。インストールスクリプトは Server A =&gt、Replica C =&gt、Replica C => Server A レプリカ合意を作成します。レプリカ B とレプリカ C はいずれもサーバー A とレプリカ合意を持ちますが、サーバー A と合意はありません。データの可用性、一貫性、フェイルオーバーの耐障害性、およびパフォーマンスについては、最終的にサーバー A のレプリケーションによってデータが相互に複製されますが、レプリカ B とレプリカ C の間にペアのレプリカ合意を作成することが有益です。

28.5.1. レプリカ合意の一覧表示

この **ipa-replica-manage** コマンドは、**list** コマンドを使用して、レプリケーショントポロジー内のすべてのサーバーおよびレプリカを一覧表示できます。

```
[root@server ~]# ipa-replica-manage list
srv1.example.com
srv2.example.com
srv3.example.com
srv4.example.com
```

サーバー/レプリケーション一覧を取得すると、サーバーのレプリカ合意を一覧表示できます。これらは、指定したサーバーが更新を送信する他のサーバー/レプリケーションです。

```
[root@server ~]# ipa-replica-manage list srv1.example.com
srv2.example.com
srv3.example.com
```

28.5.2. レプリカ合意の作成と削除

レプリカ合意は、あるサーバーを別のサーバーに **接続** して作成されます。

```
ipa-replica-manage connect server1 server2
```

サーバーを 1 つだけ指定すると、ローカルホストと指定されたサーバー間にレプリカ合意が作成されます。

以下に例を示します。

```
[root@server ~]# ipa-replica-manage connect srv2.example.com srv4.example.com
```

レプリケーションは標準の LDAP で行われ、SSL を有効にするには、ローカルホストの CA 証明書 (または指定された **サーバー1**) を追加します。次に、CA 証明書がリモートサーバーの証明書データベースにインストールされ、TLS/SSL 接続を有効にします。たとえば、以下のようになります。

```
[root@server ~]# ipa-replica-manage connect --cacert=/etc/ipa/ca.crt srv2.example.com
srv4.example.com
```

特定のサーバー/レプリカ間のレプリカ合意を削除するには、以下の **disconnect** コマンドを使用します。

```
[root@server ~]# ipa-replica-manage disconnect srv2.example.com srv4.example.com
```

この **disconnect** コマンドを使用すると、レプリカ合意が 1 つ削除されますが、サーバー/レプリケーションインスタンスの両方がレプリケーショントポロジー全体に残されます。IdM レプリケーショントポロジーからサーバーを完全に削除し、そのすべてのデータ (および機能的には IdM ドメインからサーバーとして削除) を削除するには、次の **del** コマンドを実行します。

```
[root@server ~]# ipa-replica-manage del srv2.example.com
```

28.5.3. レプリケーションの強制

サーバーとレプリカ間のレプリケーションはスケジュールで行われます。レプリケーションは頻繁に実行されますが、レプリケーション操作を手動で開始する必要がある場合があります。たとえば、サーバーをメンテナンスのためにオフラインにした場合は、キューに置かれたレプリケーションをすべてフラッシュしてから、変更ログから変更を破棄してから停止する必要があります。

レプリケーションの更新を手動で開始するには、**force-sync** コマンドを使用します。更新を受け取るサーバーはローカルサーバーです。更新を送信するサーバーは **--from** オプションで指定されます。

```
[root@server ~]# ipa-replica-manage force-sync --from srv1.example.com
```

28.5.4. IdM サーバーの初期化

レプリカが最初に作成されると、マスターサーバーのデータベースが、レプリカデータベースにコピーされます。このプロセスは **初期化** と呼ばれます。サーバー/レプリケーションが長期間オフラインの場合や、データベースに何らかの破損がある場合、新しいデータセットでサーバーを再初期化できます。

これは、**re-initialize** コマンドを使用して行います。初期化されるターゲットサーバーはローカルホストです。ローカルデータベースを初期化するためにデータをプルするサーバーまたはレプリカを **--from** オプションで指定します。

```
[root@server ~]# ipa-replica-manage re-initialize --from srv1.example.com
```

28.5.5. レプリケーションの競合の解決

IdM ドメインデータと、証明書および鍵データの両方の変更は、IdM サーバーとレプリカ (および同様のパスで、IdM サーバーと Active Directory サーバー間) 間で複製されます。

レプリケーション操作は継続的に実行されますが、1 台の IdM サーバーで変更を同時に実行でき、別の IdM サーバーの同じエントリーに別の変更が加えられる可能性があります。レプリケーションがこれらのエントリーを処理し始めると、変更が照合されます。これは **レプリケーションの競合** です。

すべてのディレクトリー変更操作には、サーバー固有の **変更状態番号** (CSN)が割り当てられ、レプリケーション中にそれらの変更が伝播される方法を追跡します。CSN には変更のタイムスタンプも含まれます。レプリケーションの競合がある場合、タイムスタンプを確認して最後の変更が優先されます。

最新の変更を受け入れるだけでは、属性値との競合の解決に有効です。ただし、一部の操作ではこのメソッドが不安定ですが、ディレクトリーツリーに影響します。modrdn、DN の変更、親および子エントリーの追加/削除などの一部の操作では、競合を解決する前に管理者のレビューが必要になります。

注記

レプリケーションの競合は、LDAP データベースの entries ディレクトリーを編集して解決されます。

レプリケーションの競合がある場合は、両方のエントリーがディレクトリーに追加され、***nsds5ReplConflict*** 属性が割り当てられます。これにより、競合のあるエントリーの検索が容易になります。

```
ldapsearch -x -D "cn=directory manager" -w password -b "dc=example,dc=com"
"nsds5ReplConflict=*" \* nsds5ReplConflict
```

28.5.5.1. ネーミングの競合の解決

同じ DN を持つ IdM ドメインに 2 つのエントリーを追加すると、両方のエントリーがディレクトリーに追加されますが、この ***nsuniqueid*** 属性を命名属性として使用するよう変更されます。以下に例を示します。

```
nsuniqueid=0a950601-435311e0-86a2f5bd-
3cd26022+uid=jsmith,cn=users,cn=accounts,dc=example,dc=com
```

これらのエントリーは、IdM CLI で検索および表示することができますが、競合が解決され、DN が更新されるまで編集または削除できません。

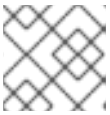
競合を解決するには、以下を実行します。

1. 別の naming 属性を使用してエントリーの名前を変更し、古い RDN を維持します。以下に例を示します。

```
ldapmodify -x -D "cn=directory manager" -w secret -h ipaserver.example.com -p 389
 dn: nsuniqueid=66446001-
1dd211b2+uid=jsmith,cn=users,cn=accounts,dc=example,dc=com
 changetype: modrdn
 newrdn: cn=TempValue
 deleteoldrdn: 0
```

2. naming 属性の古い RDN 値と競合マーカー属性を削除します。以下に例を示します。

```
ldapmodify -x -D "cn=directory manager" -w secret -h ipaserver.example.com -p 389
 dn: cn=TempValue,cn=users,cn=accounts,dc=example,dc=com
 changetype: modify
 delete: uid
 uid: jsmith
 -
 delete: nsds5ReplConflict
 -
```



注記

一意の識別子属性 ***nsuniqueid*** は削除できません。

3. エントリーの名前を目的の属性と値のペアに変更します。たとえば、以下のようになります。

```
ldapmodify -x -D "cn=directory manager" -w secret -h ipaserver.example.com -p 389
 dn: cn=TempValue,dc=example,dc=com
 changetype: modrdn
 newrdn: uid=jsmith
 deleteoldrdn: 1
```

deleteoldrdn 属性の値を **1** に設定して、一時属性と値のペア **cn=TempValue** を削除します。この属性を保持するには、***deleteoldrdn*** 属性の値を **0** に設定します。

28.5.5.2. 孤立エントリーの競合の解決

削除操作が複製され、コンシューマーサーバーが、削除されるエントリーに子エントリーがあることを検出すると、競合解決の手順により、ディレクトリーに孤立したエントリーが存在しないように、**glue** エントリーが作成されます。

同様に、追加操作が複製され、コンシューマーサーバーが親エントリーを検出できない場合は、競合解決の手順により、新しいエントリーが孤立エントリーではないように、親を表す glue エントリーが作成されます。

glue エントリーは、オブジェクトクラス **glue** および **extensibleObject** を含む一時エントリーです。チャンネルエントリーは、複数の方法で作成できます。

- 競合解決手続で、一致する一意の識別子を持つ削除されたエントリーが見つかった場合、glue エントリーは、そのエントリーの再生であり、**glue** のオブジェクトクラスと ***nsds5ReplConflict*** の属性が追加されます。

このような場合は、glue エントリーを変更して **glue** オブジェクトクラスと ***nsds5ReplConflict*** 属性を削除して、エントリーを通常のエントリーとして維持するか、その子エントリーを削除します。

- サーバーは **glue** および **extensibleObject** オブジェクトクラスを使用して最小のエントリーを作成します。

このような場合は、エントリーを変更して意味のあるエントリーに変換するか、またはそのすべての子エントリーを削除します。

28.6. レプリカの削除

レプリカを削除または **降格** すると、サーバー/レプリカトポロジーから IdM レプリカが削除され、IdM 要求を処理しなくなり、IdM ドメインからホストマシンも削除されます。

1. IdM サーバーで、IdM ツールを実行する前に Kerberos チケットを取得します。

```
[root@replica ~]# kinit admin
```

2. IdM ドメインに設定されたレプリカ合意の一覧を表示します。

```
[root@replica ~]# ipa-replica-manage list
Directory Manager password:

ipaserver.example.com: master
ipaserver2.example.com: master
replica.example.com: master
replica2.example.com: master
```

3. トポロジーからレプリカを削除するには、レプリカと IdM ドメイン内の他のサーバーとの間のすべての合意と、ドメイン設定のレプリカに関するすべてのデータを削除する必要があります。

```
[root@replica ~]# ipa-replica-manage del replica.example.com
```

4. **レプリカが独自の CA で設定されている場合** は、**ipa-csreplica-manage** コマンドを使用して、レプリカの証明書データベース間のレプリカ合意をすべて削除します。

   これは、レプリカ自体が Dogtag Certificate System CA で設定されている場合に必要です。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。

```
[root@replica ~]# ipa-csreplica-manage del replica.example.com
```

5. レプリカで、レプリカパッケージをアンインストールします。

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

28.7. サーバーまたはレプリカホストシステムの名前変更

IdM サーバーまたはレプリカマシンのホスト名を変更することはできません。Kerberos キーおよび証明書管理では、ホスト名の変更を許可するには複雑過ぎます。

サーバーまたはレプリカの名前を変更する必要がある場合は、インスタンスの置き換えが容易になります。

1. 新しいホスト名または IP アドレスを使用して、CA を使用して新規レプリカを作成します。この操作は、「4章*IdM レプリカの設定*」に説明があります。
2. 元の IdM サーバーインスタンスを停止します。

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

3. 他のサーバー/レプリケーションおよびクライアントがすべて以前と同じように機能していることを確認します。
4. 「7章*IdM サーバーおよびレプリカのアンインストール*」に従って、IdM サーバーをアンインストールします。

---

[8] **certutil** の詳細は、Mozilla NSS developer ドキュメントを参照してください。

[9] 唯一の例外は、インストール時に CA なしのインストールでシステム証明書を手動で読み込まれる場合です。それ以外の場合は、Dogtag Certificate System インスタンスがインストールされ、設定されています。

第29章 LDAP ディレクトリーから IDM への移行

認証およびアイデンティティー検索のためにインフラストラクチャーが LDAP サーバーをデプロイしていた場合は、パスワードを含むユーザーデータを新しい Identity Management インスタンスに移行できます。ユーザーやパスワードデータを損失することはありません。

Identity Management には、ディレクトリーデータの移動に役立つ移行ツールがあり、クライアントへの更新が最小限に抑えられます。ただし、移行プロセスでは、単純なデプロイメントシナリオ (LDAP 名前空間 1 つにつき IdM 名前空間 1 つ) とします。複数の名前空間やカスタムスキーマを含むより複雑な環境については、Red Hat サポートサービスにお問い合わせください。

Identity Management には、ディレクトリーデータの移動に役立つ移行ツールがあり、クライアントへの更新が最小限に抑えられます。ただし、移行プロセスでは、単純なデプロイメントシナリオ (LDAP 名前空間 1 つにつき IdM 名前空間 1 つ) とします。

29.1. 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。

移行の重要な部分は、データ移行ではなく、クライアントが Identity Management を使用するようどのように設定するかを決定するものです。インフラストラクチャーのクライアントごとに、どのサービス (Kerberos、SSSD など) を使用して最終的なIdM デプロイメントで使用可能なサービスがどれかを決定する必要があります。

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。

29.1.1. 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

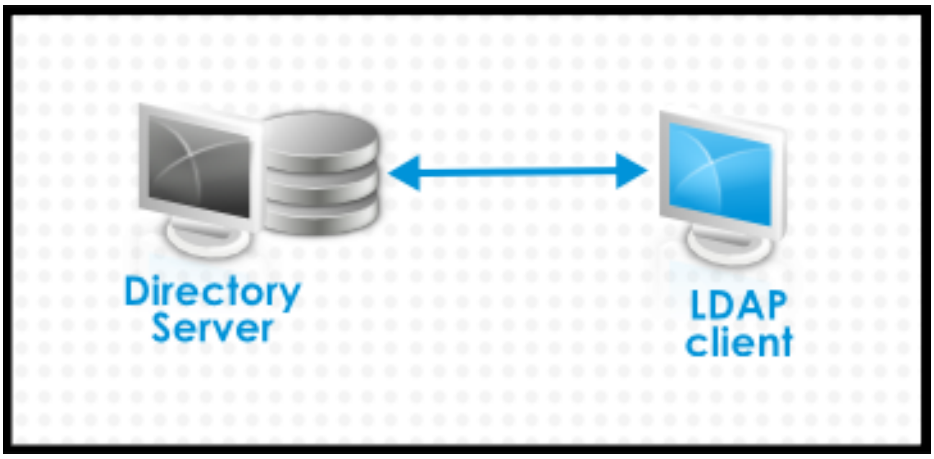
異なるクライアント設定は **相互に排他的とはなりません。**ほとんどの環境でクライアントが IdM ドメインへの接続に使用する方法はクライアントによって異なっています。管理者は各クライアント別に最適となるシナリオを決定しなければなりません。

29.1.1.1. 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29.1 基本的な LDAP ディレクトリーとクライアント設定



Linux および Unix のクライアントは PAM_LDAP と NSS_LDAP ライブラリーを使って LDAP サービスに直接接続を行います。こ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。)

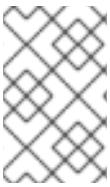
LDAP ディレクトリーと IdM サーバーの間には特にスキーマサポートとディレクトリーツリーに構造的な違いがあります。(これらの相違点の詳細は、「IdM v.LDAP: より集約的なサービスタイプ」 を参照してください。 こうした違いはデータ (特にエントリー名に影響するディレクトリーツリー) には影響する可能性がありますが **クライアントの設定** にはほとんど影響しないため、Identity Management にクライアントを移行させる上では実際にはほとんど影響がありません。

29.1.1.2. 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 『Red Hat Enterprise Linux Deployment 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。

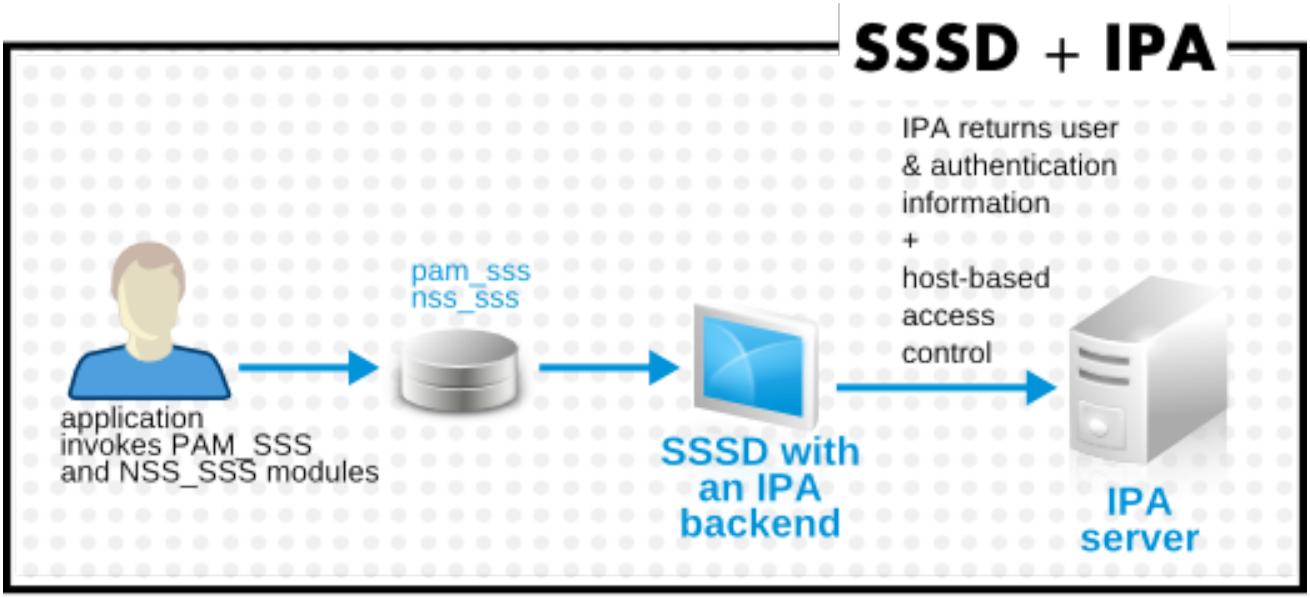
SSSD は、そのバックエンド機能すべてに Identity Management を使用できます。LDAP ID の汎用プロバイダーや Kerberos 認証とは異なり、多岐に渡る Identity Management の機能性をすべて利用することができます。たとえば、SSSD では 日常的な運用時に Identity Management でセキュリティー機能やホストベースのアクセス制御ルールを有効化させることができます。



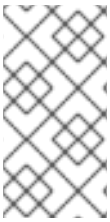
ヒント

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29.2 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。

注記

このクライアント設定は、最新バージョンの SSSD および **ipa-client** がサポートされる Red Hat Enterprise Linux 15 以降でのみサポートされます。「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。

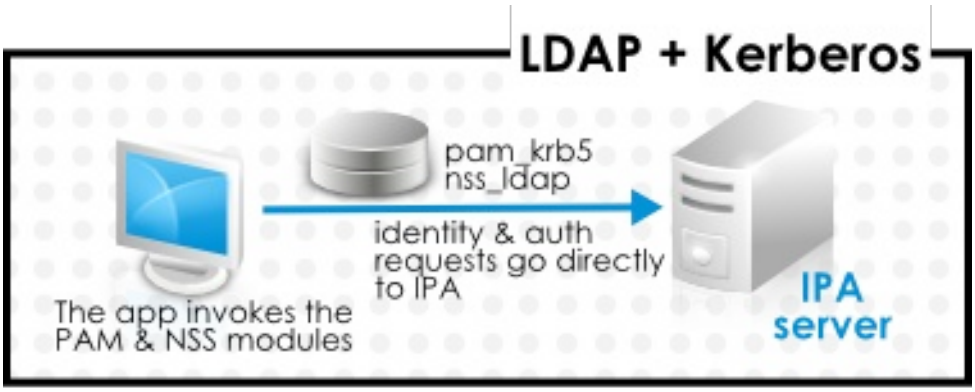
29.1.1.3. 推奨設定以外で対応している設定

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。

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 (15) 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29.3 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、『Red Hat Enterprise Linux デプロイメントガイド』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。共通する構成要素が最低限となるようなメンテナンス環境や IT インフラストラクチャーなどの場合には LDAP を ID と認証の両方に使用する必要があるかもしれません (**nss_ldap** と **pam_ldap**)。ただし、通常、クライアントに可能な最も安全な設定 (SSSD、Kerberos、LDAP、および Kerberos) を使用することが推奨されます。

29.1.2. 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。これは、Identity Management でユーザーを作成する場合です。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[10]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。

29.1.2.1. 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。(これは、IdM で LDAP ディレクトリーアカウントを再作成するだけで行うこともできます。これにより、適切なキーを持つアカウントが自動的に作成されます。) 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。

29.1.2.2. 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。

29.1.2.3. 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ので、ユーザーは単純にクライアントサービスにログインし、通常通りに動作したということしかわかりません。

29.1.2.4. 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 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。

29.1.2.5. 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:
```

29.1.3. 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ので、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。

29.1.3.1. 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。

- SunONE Directory Server
- Apache Directory Server
- OpenLDAP

LDAP サーバーから Identity Management への移行は Red Hat Directory Server 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った移行には**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。

注記

Microsoft Active Directory の場合、移行用スクリプトを使った移行には**対応していません**。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。

29.1.3.2. 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、IdM Directory Server がサポートする LDAP ディレクトリーのハッシュとして保存されます。
- 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。カスタム属性は Identity Management に移行されません。

29.1.3.3. 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** コマンドを使って移行プロセスを進めます。

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。

29.1.3.4. 移行順序

Identity Management への移行は大きく分けて 4 ステップになります。ただし、サーバーを先に移行するのかクライアントを先に移行するのかによってこの順序は若干異なります。

クライアントを先に移行する場合は SSSD を使ってクライアント設定を変更し、IdM サーバーを設定します。

1. SSSD をディプロイします。
2. クライアントが現在の LDAP サーバーに接続し IdM にフェールオーバーするよう再設定を行います。
3. IdM サーバーをインストールします。
4. IdM **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. **任意。**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。

29.2. 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実行されます。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
```

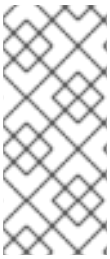
コマンドによるデータの識別およびエクスポート **migrate-ds** 方法をカスタマイズできます。元のディレクトリーツリーがユニークな構造である場合や、エントリー内のエントリーや属性を移行から除外すべき場合に便利です。

29.2.1. 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)。管理者が別のユーザーまたはグループのサブツリーの RDN を付与できるオプションは 2 つあります。

- **--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** サブツリーは **--user-container=ou=Employees** を使用して移行できます。ただし、**ou=Employees** はベース DN の直接子ではないため、**ou=Employees,ou=People,dc=example,dc=com** は、そのオプションを使用して移行することはできません。

たとえば、以下のようになります。

```
[root@ipaserver ~]# ipa migrate-ds --user-container=ou=employees --group-
container="ou=employee groups" ldap://ldap.example.com:389
```

3 つ目のオプションにより、管理者は移行用にベース DN を設定できます。**--base-dn**このオプションを使用すると、コンテナーのサブツリーのターゲットを変更することができます。たとえば、以下のようになります。

```
[root@ipaserver ~]# ipa migrate-ds --user-container=ou=employees --base-
dn="ou=people,dc=example,dc=com" ldap://ldap.example.com:389
```

今回のリリースより、**ou=Employees** ユーザーサブツリーは、ユーザー関連のすべてのサブツリーを移行せずに、大規模な **ou=People** サブツリー内から移行できるようになりました。

29.2.2. 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、**migrate-ds** スクリプトは、**person** オブジェクトクラスと指定のユーザーおよびグループサブツリー内のすべてのグループエントリーをとともに、すべてのユーザーエントリーをエクスポートします。

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,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,bjensen ldap://ldap.example.com:389
```

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,bjensen,mreynolds ldap://ldap.example.com:389
```

29.2.3. 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って無視させることができます。

- **--user-ignore-objectclass**
- **--user-ignore-attribute**
- **--group-ignore-objectclass**
- **--group-ignore-attribute**

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。

29.2.4. 使用するスキーマの設定

デフォルトでは、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。この schema オプションは、代わりに RFC2307 スキーマを使用するようにリセットできます。

```
[root@ipaserver ~]# ipa migrate-ds --schema=RFC2307 ldap://ldap.example.com:389
```

29.3. シナリオ 1: 移行の一部として SSSD を使用する

重要

この例は一般的な移行手順のため、あらゆる環境に対応するわけではありません。

実際に LDAP 環境の移行に入る前に、LDAP のテスト環境を設定して移行プロセスを検証することを強く推奨します。

1. SSSD を設定します。SSSD を使用すると、必要な Kerberos 鍵とサーバー証明書をクライアントに配信できます。

   a. すべてのクライアントマシンに SSSD をインストールします。

   ```
   # yum install sssd
   ```

   b. SSSD で LDAP アイデンティティープロバイダーを、すべての機能 (認証、ID ルックアップ、アクセス、およびパスワードの変更) に既存の Directory Server を使用するように設定します。これにより、すべてのクライアントが既存のディレクトリーサービスで適切に動作するようになります。

2. カスタム LDAP ディレクトリースキーマを含む Identity Management のインストール[11]既存の LDAP ディレクトリーとは異なるマシン上にある。

3. IdM サーバーが移行を許可できるようにします。

   ```
   # ipa config-mod --enable-migration=TRUE
   ```

4. compat プラグインを無効にします。

   ```
   # ipa-compat-manage disable
   ```

5. IdM Directory Server インスタンスを再起動します。

   ```
   # service dirsrv restart
   ```

6. 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。

   ```
   # 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 「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。

   情報のエクスポートが完了すると、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。

7. compat プラグインを再度有効にします。

   ```
   # ipa-compat-manage enable
   ```

8. IdM Directory Server インスタンスを再起動します。

   ```
   # service dirsrv restart
   ```

9. SSSD が LDAP バックエンドから Identity Management バックエンドにインストールされたクライアントを移行し、クライアントとして IdM 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。たとえば、以下のようになります。

```
# ipa-client-install --enable-dns-updates
```

10. ユーザーが SSSD バックエンドおよび Identity Management バックエンドを使用してマシンにログインしている。これにより、ユーザーに必要な Kerberos キーが生成されます。

    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。

    ```
    $ ldapsearch -LL -x -D 'cn=Directory Manager' -w secret -b 'ou=people,dc=example,dc=com'
    '(&(!(krbprincipalkey=*))(userpassword=*))' uid
    ```

    注記

    フィルターの前後に引用符を付けてシェルで解釈されないようにします。

11. ユーザーが移行したら、必要に応じて SSSD 以外のクライアントが IdM ドメインを使用するように設定します。

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。

29.4. シナリオ 2: LDAP サーバーを直接 IDENTITY MANAGEMENT に移行する

重要

この例は一般的な移行手順のため、あらゆる環境に対応するわけではありません。

実際に LDAP 環境の移行に入る前に、LDAP のテスト環境を設定して移行プロセスを検証することを強く推奨します。

1. カスタム LDAP ディレクトリースキーマを含む IdM サーバーのインストール[12]既存の LDAP ディレクトリーとは異なるマシン上にある。

2. compat プラグインを無効にします。

    ```
    # ipa-compat-manage disable
    ```

3. IdM Directory Server インスタンスを再起動します。

    ```
    # service dirsrv restart
    ```

4. IdM サーバーが移行を許可できるようにします。

    ```
    # 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。

```
# 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 「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。

情報のエクスポートが完了すると、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。

6. compat プラグインを再度有効にします。

```
# ipa-compat-manage enable
```

7. IdM Directory Server インスタンスを再起動します。

```
# service dirsrv restart
```

8. LDAP ディレクトリー、NIS、またはローカルファイルに接続する代わりに、PAM_LDAP および NSS_LDAP を使用して IdM に接続するようにクライアント設定を更新します。

9. **任意。**SSSD を設定します。SSSD を使用すると、「パスワード移行のプランニング」で説明されているように、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. すべてのクライアントマシンに SSSD をインストールします。

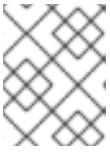
   ```
   # yum install sssd
   ```

   b. 以下のコマンド **ipa-client-install** を実行して、ID および Kerberos 認証に IdM サーバーを使用するように、SSSD および関連サービスを設定します。

10. SSSD がクライアントで利用できない場合は、SSSD クライアントまたは移行 Web ページを使用して IdM にログインするように指示します。どちらの方法でも、ユーザーパスワードが Identity Management に自動的に移行されます。

```
https://ipaserver.example.com/ipa/migration
```

11. **任意。**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。



注記

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。

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。

---

[10] 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。

[11] Identity Management では、カスタムのユーザーおよびグループスキーマのサポートは制限されています。

[12] Identity Management では、カスタムのユーザーおよびグループスキーマのサポートは制限されています。

付録A IDENTITY MANAGEMENT のトラブルシューティング

A.1. インストールの問題

A.1.1. サーバーのインストール

サーバーインストールログは、**/var/log/ipaserver-install.log** に存在します。サーバー用の IdM ログと、IdM 関連のサービスの両方が「IdM サーバーログの確認」で説明されています。

A.1.1.1. IPA コマンドの実行時に GSS 障害

インストール直後に、**ipa-*** コマンドを実行する際に Kerberos に問題が発生する可能性があります。たとえば、以下のようになります。

```
ipa: ERROR: Kerberos error: ('Unspecified GSS failure.  Minor code may provide more information', 851968)/('Decrypt integrity check failed', -1765328353)
```

これには 2 つの原因があります。

- DNS が正しく設定されていません。
- Active Directory は、IdM サーバーと同じドメインにあります。

A.1.1.2. named デーモンの起動失敗

IdM サーバーが DNS を管理して正常に設定されているが、**named** サービスが起動できない場合は、パッケージの競合があることを示すことができます。**named** 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 **/var/log/messages** ファイルでないか確認します。

```
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
```

これは通常、bind-chroot パッケージがインストールされ、**named** サービスが起動しないことを意味します。この問題を解決するには、bind-chroot パッケージを削除して、IdM サーバーを再起動します。

```
[root@server ~]# yum remove bind-chroot

# ipactl restart
```

A.1.2. レプリカのインストール

A.1.2.1. 証明書システムのセットアップに失敗しました。

手順 3 でレプリカのインストールに失敗する (**[3/11]: 証明書サーバーインスタンスの設定**) 場合、通常は必要なポートが利用できないことを意味します。CA のデバッグログ **/var/log/pki-ca/debug** を確認して検証できます。これは、特定のエントリーが見つからないことを示すエラーメッセージを表示する可能性があります。たとえば、以下のようになります。

```
[04/Feb/2016:22:29:03][http-9445-Processor25]: DatabasePanel comparetAndWaitEntries ou=people,o=ipaca not found, let's wait
```

レプリカをアンインストールする唯一の解決方法は次のとおりです。

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

レプリカをアンインストールしたら、レプリカでポート 7389 が利用可能であることを確認し、レプリカのインストールを再試行します。

A.1.2.2. レプリカの起動時に、389 Directory Server ログには SASL、GSS-API、および Kerberos エラーがあります。

レプリカが起動すると、389 Directory Server ログに一連の SASL バインドエラーが記録され、認証情報を見つけられないため、GSS-API 接続が失敗したと報告されます。

```
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)) ...
```

レプリカは **/tmp/krb5cc_496** の認証情報キャッシュを探しており (496 は 389 Directory Server のユーザー ID)、これを見つけられません。

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」というメッセージが表示される場合もあります。

```
set_krb5_creds - Could not get initial credentials for principal [ldap/ replica1.example.com] in keytab
[WRFILE:/etc/dirsrv/ds.keytab]: -1765328324 (Generic error)
```

このエラーは、389 Directory Server インスタンスが Kerberos 認証情報キャッシュを読み込む方法とタイミングの両方に関連します。

389 Directory Server 自体は複数の異なる認証メカニズムをサポートしますが、Identity Management は Kerberos 接続に GSS-API のみを使用します。Identity Management の 389 Directory Server 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 レプリカが停止した場合など、389 Directory Server プロセスが終了すると、認証情報キャッシュが破棄されます。

また、389 Directory Server は KDC のプリンシパル情報のバックエンドストレージとして使用されます。

レプリカが再起動すると、KDC の情報を提供し、KDC サーバーを起動するため、389 Directory Server インスタンスが最初に起動します。この開始順序は、GSS-API および Kerberos 接続エラーの原因となります。

389 Directory Server は GSS-API 接続を開こうとしますが、認証情報キャッシュがなく、KDC が起動していないため、GSS 接続は失敗します。同様に、ホストの認証情報の取得を試みると失敗します。

これらのエラーは一時的なものです。389 Directory Server は、KDC の起動後に GSS-API 接続を再使用し、認証情報キャッシュを持ちます。389 Directory Server のログは **bind resumed** メッセージを記録します。

これらの起動時の GSS-API 接続の失敗は、接続が正常に確立されていれば無視できます。

A.1.2.3. 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際に、証明書エラーが連続してインストールが失敗する可能性があり、DNS フォワードレコードおよび逆引きレコードが一致しないというエラーが発生する可能性があります。

-

```
ipa: DEBUG: approved_usage = SSLServer intended_usage = SSLServer
ipa: DEBUG: cert valid True for "CN=ipa-server2.example.com,O=EXAMPLE.COM"
ipa: DEBUG: handshake complete, peer = 192.168.17.37:9444
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 ipa-server2.example.com. does not match the reverse address ipa-
server2.example.org
```

IdM ドメインのすべてのサーバーとレプリカのホスト名は、DNS フォワード (A) および逆引き (PTR) レコードの両方に完全に解決できる必要があります。前方レコードと逆引きレコードの両方が、認証および証明書関連の操作時にチェックされます。レコードのホスト名が一致しない場合は、証明書エラーと DNS エラーの両方が返されます。

この問題は、複数のホスト名が単一の PTR レコードに使用される場合に発生する可能性があります。これは DNS 標準で許可されますが、サービスの設定を試みる際に IdM レプリカの作成時に問題が発生します。

レプリカホストのプライマリーホスト名が PTR ルックアップに対して返された唯一のホスト名で、複製または追加のホスト名を削除します。

DNS A および PTR レコードの確認については、「DNS レコード」を参照してください。

A.1.3. クライアントインストール

ipa-client-install を使用して設定されたクライアントの場合、クライアントのインストールログは **/var/log/ipaclient-install.log** にあります。サーバーおよびクライアントと IdM 関連のサービス両方の IdM ログは、「IdM サーバーログの確認」で説明されています。

これらは、一部の問題およびクライアントインストールの問題に対する回避策です。

A.1.3.1. クライアントは、外部 DNS を使用する際に逆引きホスト名を解決できません。

IdM はドメインサービスの一部として独自の DNS サーバーをホストできますが、外部 DNS ネームサーバーを使用することもできます。ただし、逆引き DNS にはいくつかの制限があるため、外部 DNS がクライアントの **/etc/resolv.conf** ファイルに一覧表示されている場合や、Active Directory などの SRV レコードのあるネットワークに他のリソースがある場合に、逆引き参照に問題が発生することがあります。

問題として、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ことです。

この方法の 1 つは、Kerberos データベースで IdM サーバーを検索する際にエラーを示しています。

```
Jun 30 11:11:48 server1 krb5kdc[1279](info): AS_REQ (4 etypes {18 17 16 23}) 192.168.60.135:
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}) 192.168.60.135:
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}) 192.168.60.135:
UNKNOWN_SERVER: authtime 0,  admin EXAMPLE COM for
HTTP/server1.wrong.example.com@EXAMPLE.COM, Server not found in Kerberos database
```

この問題を回避する方法は複数あります。

- **/etc/resolv.conf** ファイルを編集し、外部 DNS ネームサーバーの参照を削除します。
- 各 IdM サーバーに逆引き参照レコードを追加します。
- IdM クライアントまたはドメインにサブネットを付与し、そのサブネットのすべての要求を転送します。

A.1.3.2. クライアントは DNS ゾーンに追加されません。

クライアントが IdM DNS サーバーによって制御されていないサブネットにある場合、**nsupdate** コマンドは、**ipa-client-install** の実行時にクライアントを DNS ゾーンに追加できなくなる可能性があります。

IdM が DNS ドメインを管理する場合は、「DNS レコードエントリーの管理」の説明に従って、クライアントのゾーンエントリーを手動で追加します。たとえば、以下のようになります。

```
[jsmith@ipaserver ~]$ kinit admin
[jsmith@ipaserver ~]$ ipa dnsrecord-add ipaclient.example.com www --a-rec 1.2.3.4
```

DNS ドメインが IdM 以外で管理されている場合は、リソースレコードをゾーン設定に手動で追加できます。Red Hat Enterprise Linux の DNS の詳細は、デプロイメントガイドの DNS の章を参照してください。

A.1.4. IdM クライアントのアンインストール

Red Hat Enterprise Linux クライアントでは、**ipa-client-install** ユーティリティーを使用してクライアントをアンインストールし、IdM ドメインから削除できます。クライアントを削除するには、 **--uninstall** オプションを使用します。

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

注記

ipa-join コマンドに uninstall オプションがあります。これは、アンインストールプロセスの一部 **ipa-client-install --uninstall** として呼び出されます。ただし、**ipa-join** オプションはドメインからクライアントを削除しますが、実際にはクライアントをアンインストールしたり、IdM 関連の設定をすべて適切に削除したりしません。IdM クライアントのアンインストールには、**ipa-join -u** を実行しないでください。クライアントを完全にアンインストールする唯一の方法は、**ipa-client-install --uninstall** 使用することです。

A.2. UI 接続の問題

認証のネゴシエートが機能しない場合は、認証プロセスの詳細ロギングをオンにし、問題を診断します。

1. すべてのブラウザーウィンドウを閉じます。
2. ターミナルで、Firefox の新しいログレベルを設定します。

```
export NSPR_LOG_MODULES=negotiateauth:5
export NSPR_LOG_FILE=/tmp/moz.log
```

-

これにより、詳細なロギングが可能になり、すべての情報が **/tmp/moz.log** に記録されます。

3. 同じターミナルウィンドウからブラウザーを再起動します。

一般的なエラーメッセージおよび回避策の一部は、表A.1「UI エラーログメッセージ」にあります。

表A.1 UI エラーログメッセージ

エラーログメッセージ	説明および修正
-1208550944[90039d0]: entering nsNegotiateAuth::GetNextToken() -1208550944[90039d0]: gss_init_sec_context() failed: Miscellaneous failure No credentials cache found	Kerberos チケットはありません。**kinit** を実行します。
-1208994096[8d683d8]: entering nsAuthGSSAPI::GetNextToken() -1208994096[8d683d8]: gss_init_sec_context() failed: Miscellaneous failure Server not found in Kerberos database	Kerberos チケットを正常に取得しても、UI に対して認証できない場合に発生する可能性があります。これは、Kerberos 設定に問題があることを示しています。最初に確認する場所は、**/etc/krb5.conf** ファイルの **[domain_realm]** セクションです。IdM Kerberos ドメインエントリーが正しく、Firefox ネゴシエーションパラメーターの設定と一致していることを確認します。以下に例を示します。 .example.com = EXAMPLE.COM example.com = EXAMPLE.COM
ログファイルには含まれません。	認証のネゴシエートに必要な HTTP ヘッダーを削除するプロキシーの背後にある可能性があります。代わりに HTTPS を使用してサーバーに接続し、要求を変更せずに渡すことを可能にします。次に、ログファイルを再度確認します。

A.3. IDM サーバーの問題

A.3.1. レプリカの起動時に、389 Directory Server ログには SASL、GSS-API、および Kerberos エラーがあります。

レプリカが起動すると、389 Directory Server ログに一連の SASL バインドエラーが記録され、認証情報を見つけられないため、GSS-API 接続が失敗したと報告されます。

```
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)) ...
```

レプリカは **/tmp/krb5cc_496** の認証情報キャッシュを探しており (496 は 389 Directory Server のユーザー ID)、これを見つけられません。

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」というメッセージが表示される場合もあります。

```
set_krb5_creds - Could not get initial credentials for principal [ldap/ replica1.example.com] in keytab [WRFILE:/etc/dirsrv/ds.keytab]: -1765328324 (Generic error)
```

このエラーは、389 Directory Server インスタンスが Kerberos 認証情報キャッシュを読み込む方法とタイミングの両方に関連します。

389 Directory Server 自体は複数の異なる認証メカニズムをサポートしますが、Identity Management は Kerberos 接続に GSS-API のみを使用します。Identity Management の 389 Directory Server 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 レプリカが停止した場合など、389 Directory Server プロセスが終了すると、認証情報キャッシュが破棄されます。

また、389 Directory Server は KDC のプリンシパル情報のバックエンドストレージとして使用されます。

レプリカが再起動すると、KDC の情報を提供し、KDC サーバーを起動するため、389 Directory Server インスタンスが最初に起動します。この開始順序は、GSS-API および Kerberos 接続エラーの原因となります。

389 Directory Server は GSS-API 接続を開こうとしますが、認証情報キャッシュがなく、KDC が起動していないため、GSS 接続は失敗します。同様に、ホストの認証情報の取得を試みると失敗します。

これらのエラーは一時的なものです。389 Directory Server は、KDC の起動後に GSS-API 接続を再使用し、認証情報キャッシュを持ちます。389 Directory Server のログは **bind resumed** メッセージを記録します。

これらの起動時の GSS-API 接続の失敗は、接続が正常に確立されていれば無視できます。

A.4. ホストの問題

A.4.1. 証明書が検出されない/識別番号が検出されないエラー

IdM 情報は、証明書情報とは別の LDAP ディレクトリーに保存され、これら 2 つの LDAP データベースは別々にレプリケートされます。レプリカ合意があるディレクトリーに対して破損していても、別のディレクトリーで機能することがあります。これにより、クライアントの管理で問題が発生する可能性があります。

具体的には、2 つの CA データベース間のレプリカ合意が破損している場合は、サーバーが有効な IdM クライアントの証明書情報を見つけられなくなり、証明書エラーが発生する可能性があります。

```
Certificate operation cannot be completed: EXCEPTION (Certificate serial number 0x2d not found)
```

たとえば、IdM サーバーとレプリカは、IdM データベース間で機能レプリカ合意を持ちますが、CA データベース間のレプリカ合意は破損しています。サーバーでホストが作成されると、ホストエントリーはレプリカに複製されますが、そのホストの証明書は複製されません。レプリカはクライアントを認識しますが、レプリカの証明書のコピーがないため、そのクライアントの管理操作は失敗します。

A.4.2. クライアント接続の問題のデバッグ

クライアント接続の問題はすぐに行われます。つまり、ユーザーはマシンにログインしたり、ユーザーおよびグループの情報にアクセス試行すると失敗する可能性があります (例: **getent passwd admin**)。

IdM の認証は、SSSD デーモンで管理されます。これは『Red Hat Enterprise Linux デプロイメントガイド』に記載されています。クライアント認証に問題がある場合は、SSSD 情報を確認します。

まず、**/var/log/sssd/** で SSSD ログを確認します。**sssd_example.com.log** など、DNS ドメインには、以下のような特定のログファイルがあります。デフォルトのロギングレベルでログに十分な情報がない場合には、ログレベルを増やします。

ログレベルを増やすには、以下を実行します。

1. **sssd.conf** ファイルを開きます。

```
vim /etc/sssd/sssd.conf
```

2. **[domain/example.com]** セクションで、**debug_level** を設定します。

```
debug_level = 9
```

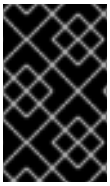
3. **sssd** デーモンを再起動します。

```
service sssd restart
```

4. デバッグメッセージの **/var/log/sssd/sssd_example.com.log** ファイルを確認します。

A.5. KERBEROS エラー

Kerberos エラーは、**kinit** または同様のクライアントを使用してレルムへの接続を試みると頻繁に行われます。Kerberos に関する情報は、まず Kerberos の man ページ、ヘルプファイル、その他のリソースを確認します。



重要

Identity Management には、Kerberos ポリシーの管理に使用する独自のコマンドラインツールがあります。IdM Kerberos 設定の管理には、**kadmin** または **kadmin.local** は **使わないで** ください。

Kerberos エラーログ情報を検索する場所は複数あります。

- **kinit** の問題またはその他の Kerberos サーバーの問題は、**/var/log/krb5kdc.log** の KDC ログインを参照してください。
- IdM 固有のエラーは、**/var/log/httpd/error_log** を参照してください。

サーバー用の IdM ログと、IdM 関連のサービスの両方が「IdM サーバーログの確認」で説明されています。

A.5.1. GSS-API の使用時に SSH で接続する場合の問題

DNS 設定に誤った逆引き DNS エントリーがある場合は、SSH を使用して IdM リソースにログインできない可能性があります。SSH がセキュリティーメソッドとして GSS-API を使用してリソースへの接続を試みると、GSS-API は最初に DNS レコードをチェックします。レコードが正しくないため、SSH がリソースを見つけることができません。

SSH 設定で逆引き DNS ルックアップを無効にすることができます。SSH は逆引き DNS レコードを使用するのではなく、指定のユーザー名を GSS-API に直接渡します。

SSH で逆引き DNS ルックアップを無効にするには、***GSSAPITrustDNS*** ディレクティブを追加または編集し、値を **no** に設定します。

```
# vim /etc/ssh/ssh_config

GSSAPITrustDNS no
```

A.5.2. キータブの変更後に NFS サーバーへの接続に問題があります。

NFS エクスポートのマウントを試みるクライアントは、有効なプリンシパルキーと秘密鍵が NFS サーバーとクライアントホストの両方に存在する必要があります。クライアント自体は NFS キータブにアクセスできないはずです。NFS 接続のチケットは KDC からクライアントに渡されます。

更新されたキータブのエクスポートに失敗すると、分離が困難な問題が発生する可能性があります。たとえば、既存のサービス接続は引き続き機能しますが、新しい接続は実行できません。

A.6. SELINUX ログインの問題

SELinux は、リモートユーザーに対してのみ動作しますが、ローカルアカウントを持つユーザーには対応しません。

リモートユーザーがログインしたら、IdM サーバーに対して認証を行うと、PAM SElinux モジュールは **/etc/selinux/policy_name/logins/login** にそのユーザーのファイルを作成します。

そのファイルが存在しない場合は、SSSD が、アイデンティティープロバイダーの 1 つとして IdM サーバーを使用するように適切に設定されていないことを意味します。これは、SELinux マッピングが機能するために必要です。SSSD の設定は、Red Hat 6 デプロイメントガイド を参照してください。

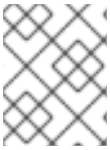
リモートユーザーが誤った SELinux コンテキストが付与されている場合は、PAM スタックで **pam_selinux** モジュールが正しく設定されないことがあります。これは、SELinux 情報を読み取り、ユーザーコンテキストを設定するモジュールです。モジュールがないと、SELinux マップが処理されず、ユーザーはシステムのデフォルトコンテキストを定義します。

付録B CERTMONGER を使った作業

マシンの認証管理には、マシン証明書の管理が含まれます。クライアントでは、IdM は **certmonger** サービスで証明書ライフサイクルを管理します。これは、IdM が提供する認証局 (CA) と連携します。

certmonger デーモンとそのコマンドラインクライアントを使用すると、公開鍵と秘密鍵のペア生成や証明書リクエストの作成、CA に対する署名のリクエスト提出といった処理を簡素化することができます。証明書の管理の一環として、**certmonger** デーモンは証明書の有効期限を監視し、期限が切れそうになった証明書を更新することができます。**certmonger** が監視する証明書はファイルで追跡しており、このファイルは設定可能なディレクトリー内に保存します。デフォルトの場所は **/var/lib/certmonger/requests** です。

certmonger は IdM **getcert** コマンドを使用して、すべての証明書を管理します。「例: 異なる CA 設定を使用したインストール」で説明しているように、IdM サーバーは、さまざまな種類の認証局を使用するように設定できます。最も一般的な (および推奨) 設定は、完全な CA サーバーを使用することにありますが、より限定的な自己署名 CA を使用することもできます。IdM バックエンドと通信するために **certmonger** で使用される **getcert** コマンドは、使用する CA の種別によって異なります。この **ipa-getcert** コマンドは完全な CA で使用されますが、**selfsign-getcert** コマンドは自己署名の CA で使用されます。



注記

一般的なセキュリティー上の問題により、自己署名証明書は通常実稼働環境で使用されませんが、開発およびテストに使用することができます。

B.1. CERTMONGER で証明書の要求

IdM CA で、**certmonger** は **ipa-getcert** コマンドを 使用します。

証明書および鍵は、プレーンテキストファイル (**.pem**) または NSS データベース (証明書のニックネームで識別される) でローカルに保存されます。証明書をリクエストする際には、リクエストで証明書の保存場所とニックネームを特定します。たとえば、以下のようになります。

```
# ipa-getcert request -d /etc/pki/nssdb -n Server-Cert
```

この **/etc/pki/nssdb** ファイルはグローバル NSS データベースで、 **Server-Cert** はこの証明書のニックネームです。証明書のニックネームはこのデータベース内で一意のものである必要があります。

IdM サービスと使用するように証明書を要求する場合は、サービスプリンシパルの指定に **-K** オプションが必要になります。そうでない場合、**certmonger** は証明書がホスト用であると仮定します。この **-N** オプションは、証明書サブジェクト DN を指定し、サブジェクトベース DN が IdM サーバーのベース DN と一致するか、要求が拒否される必要があります。

```
$ ipa-getcert request -d /etc/httpd/alias -n Server-Cert -K HTTP/client1.example.com -N
'CN=client1.example.com,O=EXAMPLE.COM'
```

例B.1 サービスにおける certmonger の使用

```
$ ipa-getcert request -r -f /etc/httpd/conf/ssl.crt/server.crt -k /etc/httpd/conf/ssl.key/server.key -N
CN=`hostname --fqdn` -D `hostname` -U id-kp-serverAuth
```

このオプションは、自己署名証明書 (**selfsign-getcert**) と最終証明書の希望の設定を使用しているかどうかや、その他の設定によって異なります。例B.1「サービスにおける certmonger の使用」で、これらは一般的なオプションです。

- この **-r** オプションは、鍵ペアがすでに存在する場合に証明書を自動的に更新します。これは、デフォルトで使用されます。
- **-f** オプションは、指定したファイルに証明書を保存します。
- **-k** は、鍵を特定ファイルに保存するか、鍵のファイルが存在する場合は、ファイル内の鍵を使用します。
- **-N** オプションはサブジェクト名を指定します。
- この **-D** オプションは、DNS ドメイン名を指定します。
- この **-U** オプションは、拡張キー使用フラグを設定します。

B.2. NSS データベースでの証明書の保存

デフォルトでは、**certmonger** はプレーンテキストファイルを使用して鍵と証明書を保存します。ただし、これらの鍵と証明書は NSS データベースに格納することもできます。これは、**-d** オプションを使用してセキュリティーデータベースの場所と **-n** を設定し、データベースの証明書に使用される証明書のニックネームを指定します。これらのオプションは **-f** および **-k** オプションで指定される PEM ファイルの代わりに使用されます。

たとえば、以下のようになります。

```
# ipa-getcert request -d /export/alias -n ServerCert ...
```

B.3. CERTMONGER を使った証明書の追跡

certmonger は、証明書ライフサイクル全体を管理できます。要求の生成とともに、**certmonger** は証明書を追跡し、有効期間の終了時に証明書を自動的に更新することができます。

これは、**getcert** とともに **start-tracking** コマンドでコマンドを使用して行います。この **-I** オプションは、NSS データベース (**-d** および **-n**) または PEM ファイル (**-f** および **-k**) 内のいずれかでキーおよび証明書ファイルへのポインターと共に追跡エントリーを作成します。この **-r** オプションは、証明書の更新を **certmonger** に指示します。

```
# ipa-getcert start-tracking -I cert1-tracker -d /export/alias -n ServerCert -r
```

ヒント

-r オプションは、例B.1「サービスにおける certmonger の使用」において、**request** コマンドで渡すことができます。この場合、要求された証明書は **certmonger** によって自動的に追跡および更新されます。その後、手動で追跡を設定する必要はありません。

証明書は、**stop-tracking** コマンドを使用して、**certmonger** で追跡 **解除** できます。

索引

シンボル

A

B

C

D

W

付録C 改訂履歴

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 6.7-4 Mon Apr 10 2017 Aneta Šteflová Petrová
スマートカードのアップデート

改訂 6.7-3 Wed Mar 8 2017 Aneta Šteflová Petrová
6.9 GA 公開用バージョン

改訂 6.7-2 Wed May 4 2016 Marc Muehlfeld
6.8 GA 公開用ドキュメントの準備

改訂 6.7-1 Thu Feb 18 2016 Aneta Petrová
信頼および sudo の章に対するマイナーな更新により、外部 CA が発行する CA 証明書を更新する警告が追加されました。

改訂 6.7-0 Tue Jul 14 2015 Tomáš Čapek
6.7 GA リリース向けバージョン

改訂 6.6-2 Tue Mar 31 2015 Tomáš Čapek
Kerberized NFS サーバーおよびクライアントの設定に関するセクションが改善されました。

改訂 6.6-1 Fri Dec 19 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 6.6-0 Fri Oct 10 2014 Tomáš Čapek
6.6 GA リリース向けバージョン

改訂 6.5-5 July 9, 2014 Ella Deon Ballard
バグが修正されました。

改訂 6.5-4 February 3, 2014 Ella Deon Ballard
バグが修正されました。

改訂 6.5-1 November 20, 2013 Ella Deon Ballard
バグが修正されました。

改訂 6.4-3 August 20, 2013 Ella Deon Lackey
バグが修正され、一部の章を再編成しました。

改訂 6.4-1 March 1, 2013 Ella Deon Lackey
信頼の追加。

改訂 6.3-1 October 18, 2012 Ella Deon Lackey
sudo 設定例、グループ同期情報、CRL 生成セクションを削除しました。

改訂 6.2-8 December 16, 2011 Ella Deon Lackey
更新された sudoers_debug の例移行コマンドの例が修正されました。

改訂 6.2-7 December 6, 2011 Ella Deon Lackey
Red Hat Enterprise Linux 6.2 の GA リリース