Red Hat Enterprise Linux 9

Identity Management でのパフォーマンスの調整

Red Hat Enterprise Linux 9 のパフォーマンスを向上させるために Identity Management サービスの調整

Last Updated: 2022-08-15

Red Hat Enterprise Linux 9 Identity Management でのパフォーマンスの調整

Red Hat Enterprise Linux 9 のパフォーマンスを向上させるために Identity Management サービスの調整

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

このドキュメントコレクションでは、Red Hat Enterprise Linux 9 の Identity Management で一般的なパフォーマンス設定を調整する方法を説明します。

目次

多様性を受け入れるオープンソースの強化

Red Hat では、コード、ドキュメント、Web プロパティーにおける配慮に欠ける用語の置き換えに取り組んでいます。まずは、マスター (master)、スレーブ (slave)、ブラックリスト (blacklist)、ホワイトリスト (whitelist) の 4 つの用語の置き換えから始めます。この取り組みは膨大な作業を要するため、今後の複数のリリースで段階的に用語の置き換えを実施して参ります。詳細は、弊社 の CTO、Chris Wright のメッセージ を参照してください。

Identity Management では、以下のような用語の置き換えが含まれます。

- **ブラックリスト**から**ブロックリスト**
- **ホワイトリスト**から**許可リスト**
- **スレーブ**から**セカンダリー**
- 単語 **マスター** は、コンテキストに応じて、より正確な言語に置き換えられます。
  - **マスター**から**IdM サーバー**
  - **CA 更新マスター**から**CA 更新サーバー**
  - **CRL マスター**から**CRL パブリッシャーサーバー**
  - **マルチマスター**から**マルチサプライヤー**

RED HAT ドキュメントへのフィードバック (英語のみ)

ご意見ご要望をお聞かせください。ドキュメントの改善点はございませんか。

- 特定の部分についての簡単なコメントをお寄せいただく場合は、以下をご確認ください。
    1. ドキュメントの表示が **Multi-page HTML** 形式になっていて、ドキュメントの右上隅に **Feedback** ボタンがあることを確認してください。
    2. マウスカーソルで、コメントを追加する部分を強調表示します。
    3. そのテキストの下に表示される **Add Feedback** ポップアップをクリックします。
    4. 表示される手順に従ってください。
- Bugzilla を介してフィードバックを送信するには、新しいチケットを作成します。
    1. Bugzilla の Web サイトに移動します。
    2. Component で **Documentation** を選択します。
    3. **Description** フィールドに、ドキュメントの改善に関するご意見を記入してください。ドキュメントの該当部分へのリンクも記入してください。
    4. **Submit Bug** をクリックします。

第1章 IDM のチューニングにおける重要な考慮事項

Identity Management のコンポーネントサービスは、ほとんどのデプロイメントに最適な方法で機能するように調整されています。システム管理者は、使用環境の要求に合わせて IdM サービスのパフォーマンスを調整できます。

重要な留意事項

- 各 IdM デプロイメントは、ハードウェア、ソフトウェア、ネットワーク、データ、ワークロードなどの多くの要因の固有の組み合わせです。ある環境に影響を与える調整は、別の環境に悪影響を与える可能性があります。

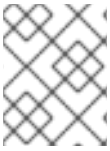
- パフォーマンスの調整は、反復的な実験的プロセスです。Red Hat では、一度に 1 つの変数のみを調整し、環境への影響を監視することを推奨しています。ある変数で目的の結果が得られたら、以前の調整のパフォーマンスを監視しながら、次の変数を調整します。

第2章 ハードウェア推奨事項

ハードウェアでは、RAM の容量を適切に確保することが最も重要になります。システムに十分な RAM があるようにしてください。一般的な RAM の要件は次のとおりです。

- 10,000 ユーザーおよび 100 グループには、最低 4 GB の RAM と 4 GB のスワップ領域を割り当てます。
- 100,000 ユーザーおよび 50,000 グループには、最低 16 GB の RAM と 4 GB のスワップ領域を割り当てます。

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。通常、RAM を追加すると、キャッシュにより大きなデプロイメントのパフォーマンスが向上します。



注記

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

第3章 IDM のフェイルオーバー、負荷分散、および高可用性

Identity Management (IdM) には、IdM クライアント用のフェイルオーバーメカニズムと、IdM サーバーの負荷分散機能および高可用性機能が組み込まれています。

3.1. クライアント側のフェイルオーバー機能

- デフォルトでは、IdM クライアントの **SSSD** サービスは、DNS からのサービス (SRV) リソースレコードを使用して、接続する最善の IdM サーバーを自動的に判断するように設定されています。この挙動は、**/etc/sssd/sssd.conf** ファイルの **ipa_server** パラメーターの **_srv_** オプションで制御されます。

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

[domain/example.com]
id_provider = ipa
ipa_server = _srv_, server.example.com
...
```

  IdM サーバーがオフラインになると、IdM クライアントの SSSD サービスは、自動的に検出された別の IdM サーバーに接続します。

- パフォーマンス上の理由から DNS 検索を回避する場合は、**ipa_server** パラメーターから **_srv_** エントリーを削除し、クライアントが接続する IdM サーバーを優先的に指定します。

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

[domain/example.com]
id_provider = ipa
ipa_server = server1.example.com, server2.example.com
...
```

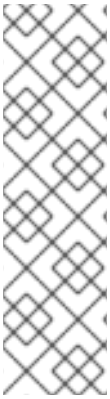
3.2. サーバー側の負荷分散とサービスの可用性

複数の IdM レプリカをインストールすることで、IdM で負荷分散と高可用性を実現できます。

- 地理的に分散しているネットワークがある場合は、データセンターごとに 複数の IdM レプリカを設定することで、IdM クライアントと、最も近くにあるアクセス可能なサーバーとの間のパスを短縮できます。

- Red Hat は、最大 60 のレプリカを持つ環境に対応します。

- IdM レプリカメカニズムにより、アクティブ/アクティブのサービスの可用性が提供されます。つまり、すべての IdM レプリカのサービスは、同時にすぐに利用できます。



注記

Red Hat では、IdM とその他の負荷分散または高可用性 (HA) ソフトウェアを組み合わせないことを推奨しています。

サードパーティーの高可用性ソリューションの多くは、アクティブ/パッシブのシナリオを想定し、IdM の可用性に対して不要なサービスの中断を発生させます。その他のソリューションでは、仮想 IP を使用するか、クラスターサービスごとに 1 つのホスト名を使用します。これらのすべての方法は、通常、IdM ソリューションが提供するサービスの可用性のタイプとは連携しません。また、Kerberos との統合が非常に悪いため、デプロイメントの全体的なセキュリティーと安定性が低下します。

第4章 レプリカトポロジーの最適化

堅牢なレプリカトポロジーにより、ワークロードが分散し、レプリカの遅延が低減します。以下のガイドラインに従って、レプリカトポロジーのレイアウトを最適化します。

4.1. 適切なレプリカ数の決定

各データセンターに少なくとも 2 つのレプリカを設定 (必須要件ではありません)

データセンターは、たとえば、本社または地理的な位置 (領域) に置かれます。

クライアントにサービスを提供するために十分な数のサーバーを設定

1 台の Identity Management (IdM) サーバーで 2000 ~ 3000 台のクライアントにサービスを提供できます。ここでは、クライアントがサーバーに対して 1 日に複数回クエリーする (毎分ではありません) ことを想定しています。より頻繁なクエリーが予想される場合は、より多くのサーバーを計画してください。

十分な数の認証局 (CA) レプリカを設定します。

CA ロールがインストールされているレプリカのみが、証明書データを複製できます。IdM CA を使用する場合は、環境に、証明書のレプリカ合意がある CA レプリカが 2 つ以上あることを確認します。

1 つの IdM ドメインに最大 60 台のレプリカを設定

Red Hat は、最大 60 のレプリカを持つ環境に対応します。

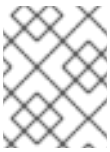
4.2. トポロジー内でレプリカの接続

1 台のレプリカを少なくとも 2 つのレプリカに接続

追加のレプリカ合意を設定すると、初期レプリカと最初にインストールしたサーバーとの間だけでなく、他のレプリカ間でも情報が複製されます。

レプリカを、その他のレプリカ (最大 4 つ) に接続 (必須要件ではありません)

サーバーごとに多数のレプリカ合意を行っても、大きな利点はありません。受信レプリカは、一度に 1 つのレプリカによってのみ更新でき、その間、その他のレプリカ合意はアイドル状態になります。通常、レプリカごとに 4 つ以上のレプリカ合意があると、リソースが無駄になります。



注記

この推奨事項は、証明書のレプリケーションとドメインのレプリケーションの両方に適用されます。

1 台のレプリカに対するレプリケーション合意が 4 つに制限される点について、2 つの例外があります。

- 特定のレプリカがオンラインでないか、応答していない場合はフェールオーバーパスが必要。
- 大規模デプロイメントでは、特定のノード間に追加の直接リンクが必要。

レプリケーション合意を多数構成すると、全体のパフォーマンスに影響を及ぼす場合があります。トポロジー内の複数のレプリカ合意が更新を送信すると、特定のレプリカは、受信更新と送信更新の間で changelog データベースファイルに対して競合が多くなる可能性があります。

レプリカごとにレプリカ合意を使用する場合は、レプリケーションの問題およびレイテンシーが発生しないようにしてください。ただし、距離が長く、中間ノードの数が多いと、レイテンシーの問題が発生する場合があることに注意してください。

データセンター内のレプリカを互いに接続

これにより、データセンター間のドメインレプリケーションが保証されます。

各データセンターを少なくとも 2 つの他のデータセンターに接続

これにより、データセンター間のドメインレプリケーションが保証されます。

少なくとも一対のレプリカ合意を使用してデータセンターを接続

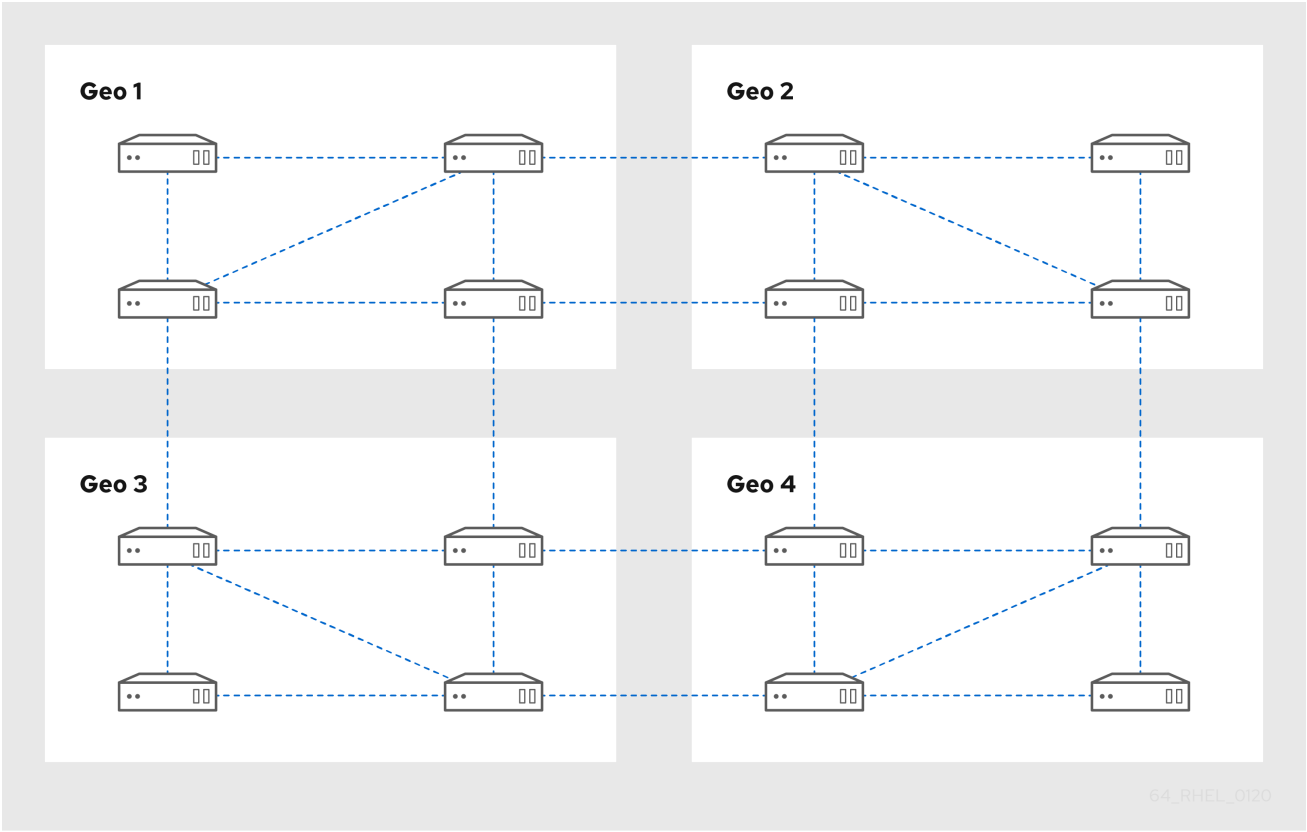
データセンター A および B に、A1 への B1 までのレプリカ合意がある場合は、A2 から B2 へのレプリカ合意があれば、いずれかのサーバーがダウンしても、2 つのデータセンター間でレプリケーションを続行できます。

4.3. レプリカトポロジーの例

以下の図は、信頼できるトポロジーを作成するガイドラインに基づく Identity Management (IdM) トポロジーの例を示しています。

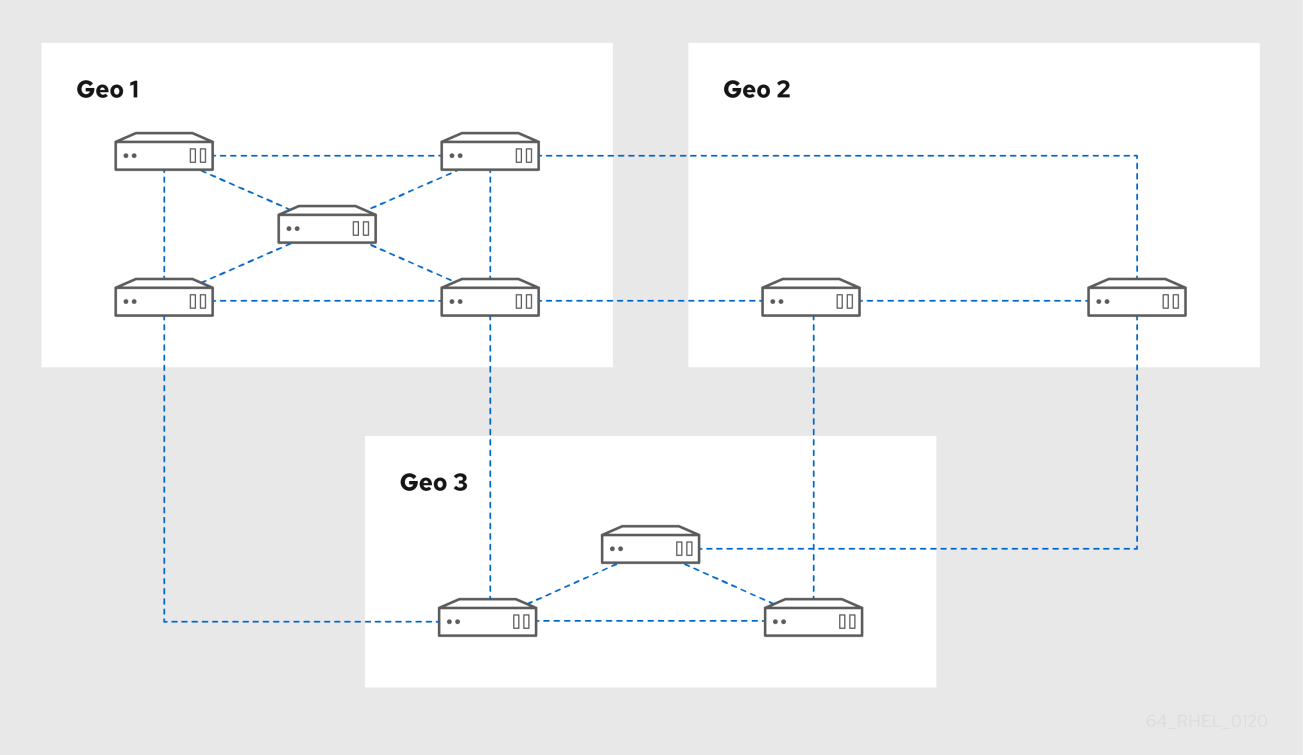
レプリカトポロジー例 1 には 4 つのデータセンターがあり、各データセンターに 4 つのサーバーがあります。このサーバーは、レプリカ合意に接続しています。

図4.1 レプリカトポロジーの例 1



レプリカトポロジー例 2には、所有するサーバー数が異なる 3 つのデータセンターが表示されます。このサーバーは、レプリカ合意に接続しています。

図4.2 レプリカトポロジーの例 2



4.4. 関連情報

- レプリカトポロジーの計画
- レプリケーショントポロジーの管理

第5章 検索サイズおよび時間制限の調整

IdM ユーザーの一覧を要求するなど、一部のクエリーでは、エントリー数が大量に返される場合があります。この検索操作を調整して、**ipa user-find** などの **ipa *-find** コマンドの実行時や、Web UI で対応する一覧を表示する際に、全体的なサーバーのパフォーマンスを向上できます。

Search size limit

クライアントの CLI または IdM Web UI にアクセスするブラウザーからサーバーに送信されるリクエストで返される最大エントリー数を定義します。
デフォルト - 100 エントリー

Search time limit

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

この値が **-1** に設定されていると、IdM は、検索時に制限を適用しません。

重要

検索のサイズや時間制限を高く設定しすぎると、サーバーのパフォーマンスに影響を及ぼすことがあります。

5.1. コマンドラインで検索サイズおよび時間制限の調整

以下の手順では、コマンドラインで検索サイズと時間制限を調整する方法について説明します。

- システム全体
- 特定のエントリーの場合

手順

1. 現在の検索時間およびサイズ制限を CLI で表示するには、**ipa config-show** コマンドを使用します。

```
$ ipa config-show

Search time limit: 2
Search size limit: 100
```

2. すべてのクエリーに対して **グローバルに** 制限を調整するには、**ipa config-mod** コマンドを使用して、**--searchrecordslimit** および **--searchtimelimit** のオプションを追加します。以下に例を示します。

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

3. 特定のクエリに対してのみ **一時的に** 制限を調整するには、コマンドに **--sizelimit** または **--timelimit** オプションを追加してください。以下に例を示します。

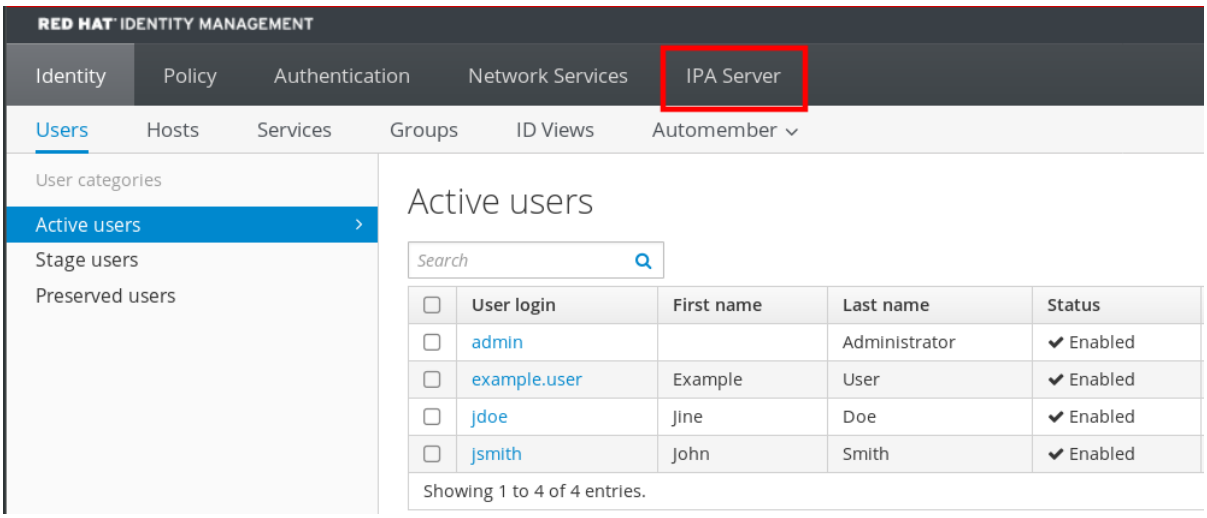
```
$ ipa user-find --sizelimit=200 --timelimit=120
```

5.2. WEB UI で検索サイズおよび時間制限の調整

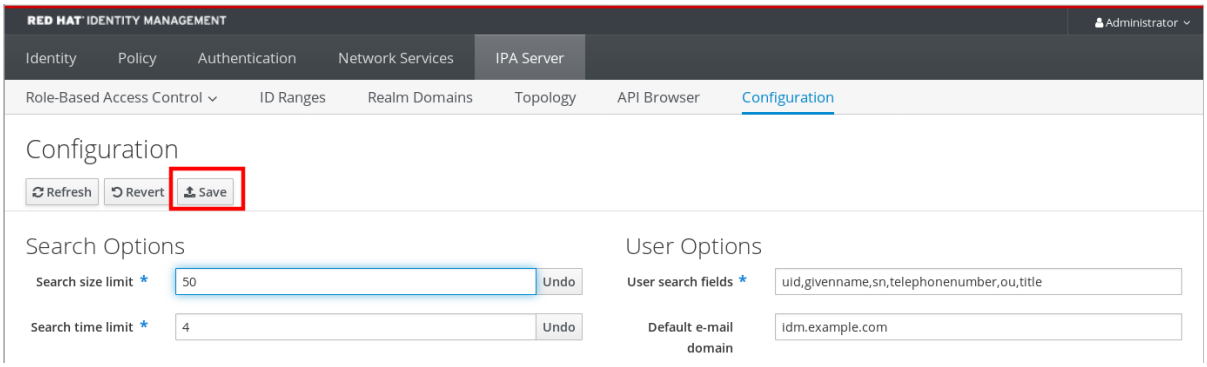
以下の手順では、IdM Web UI でグローバル検索のサイズと時間制限を調整する方法について説明します。

手順

1. IdM Web UI にログインします。
2. **IPA Server** をクリックします。



3. **IPA Server** タブで、**Configuration** をクリックします。
4. **Search Options** エリアに必要な値を設定します。
   デフォルト値は以下の通りです。
   - 検索サイズの制限 - 100 エントリー
   - 検索時間の制限 - 2 秒
5. ページ上部にある **Save** をクリックします。



第6章 ADJUSTING IDM DIRECTORY SERVER PERFORMANCE

Directory Server のリソースと動作を制御する LDAP 属性を調整することで、Identity Management のデータベースのパフォーマンスを調整できます。

Directory Server による **データのキャッシュ** 方法を調整するには、以下の手順を参照してください。

- エントリーキャッシュサイズの調整
- データベースのインデックスキャッシュサイズの調整
- エントリーおよびデータベースキャッシュの自動サイズ設定の再有効化
- DN キャッシュサイズの調整
- 正規化された DN キャッシュサイズの調整

Directory Server の **リソース制限** を調整するには、以下の手順を参照してください。

- メッセージの最大サイズの調整
- ファイルディスクリプターの最大数の調整
- 接続バックログサイズの調整
- データベースロックの最大数の調整

パフォーマンスに最も影響を与える **タイムアウト** を調整するには、以下の手順を参照してください。

- 入出力ブロックのタイムアウト調整
- アイドル接続のタイムアウトの調整
- レプリケーションリリースのタイムアウトの調整

LDIF ファイルからカスタム Directory Server 設定を使用して IdM サーバーまたはレプリカをインストールするには、次の手順を参照してください。

- LDIF ファイルからのカスタムデータベース設定を使用した IdM サーバーまたはレプリカのインストール

6.1. エントリーキャッシュサイズの調整

重要

Red Hat は、パフォーマンスを最適化するために、内蔵キャッシュの自動サイズ設定機能を使用することを推奨します。オートチューニングした値から意図的に逸脱する必要がある場合に限り、この値を変更してください。

nsslapd-cachememsize 属性は、エントリーキャッシュに使用できるメモリー領域のサイズ (バイト) を指定します。この属性は、ディレクトリーサーバーが使用する物理 RAM の量を制御するうえで最も重要な値の 1 つです。

エントリーキャッシュサイズが小さすぎると、**/var/log/dirsrv/slapd-INSTANCE-NAME/errors** ログファイルの Directory Server エラーログに以下のエラーが表示される場合があります。

-

> REASON: entry too large (**83886080** bytes) for the import buffer size (**67108864** bytes). **Try increasing nsslapd-cachememsize.**

Red Hat では、エントリーキャッシュとデータベースのインデックスエントリーキャッシュをメモリーに収めることを推奨しています。

デフォルト値	**209715200** (200 MiB)
有効範囲	**500000 - 18446744073709551615** (500 kB - (2^{64}-1))
エントリー DN の場所	**cn=database-name,cn=ldbm database,cn=plugins,cn=config**

前提条件

- LDAP Directory Manager のパスワード

手順

1. 自動キャッシュチューニングを無効にします。

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://server.example.com backend config set --cache-autosize=0
```

2. データベースの接尾辞と、対応するバックエンドを表示します。

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://server.example.com backend suffix list
cn=changelog (changelog)
dc=example,dc=com (userroot)
o=ipaca (ipaca)
```

このコマンドにより、各接尾辞の横にバックエンドデータベースが表示されます。次の手順では、接尾辞のデータベース名を使用します。

3. データベースのエントリーキャッシュサイズを設定します。この例では、userrooot データベースのエントリーキャッシュを 2 ギガバイトに設定します。

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://server.example.com backend suffix set --cache-memsize=2147483648 userroot
```

4. Directory Server を再起動します。

```
[root@server ~]# systemctl restart dirsrv.target
```

5. IdM ディレクトリーサーバーのパフォーマンスを監視します。望ましい方法で変更が行われない場合は、この手順を繰り返して **cache-memsize** を別の値に調整するか、キャッシュの自動サイズ設定を再度有効にします。

検証手順

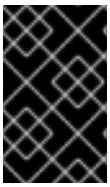
- **nsslapd-cachememsize** 属性の値を表示し、目的の値に設定されていることを確認します。

```
[root@server ~]# ldapsearch -D "cn=directory manager" -w DirectoryManagerPassword
-b "cn=userroot,cn=ldbm database,cn=plugins,cn=config" | grep nsslapd-
cachememsize
nsslapd-cachememsize: 2147483648
```

関連情報

- Directory Server 11 ドキュメントの nsslapd-cachememsize
- エントリーおよびデータベースキャッシュの自動サイズ設定の再有効化

6.2. データベースのインデックスキャッシュサイズの調整



重要

Red Hat は、パフォーマンスを最適化するために、内蔵キャッシュの自動サイズ設定機能を使用することを推奨します。オートチューニングした値から意図的に逸脱する必要がある場合に限り、この値を変更してください。

nsslapd-dbcachesize 属性は、データベースインデックスが使用するメモリーの容量を制御します。このキャッシュサイズは、エントリーキャッシュサイズよりも Directory Server のパフォーマンスに影響を及ぼしません。ただし、エントリーキャッシュサイズの設定後に利用可能な RAM がある場合は、データベースキャッシュに割り当てられるメモリーの量を増やすことが推奨されます。

データベースキャッシュは、RAM が 1.5GB に制限されています。これを超える値を設定するとパフォーマンスが向上しないためです。

デフォルト値	**10000000** (10 MB)
有効範囲	**500000 - 1610611911** (500 kB - 1.5GB)
エントリー DN の場所	**cn=config,cn=ldbm database,cn=plugins,cn=config**

前提条件

- LDAP Directory Manager のパスワード

手順

1. 自動キャッシュチューニングを無効にし、データベースのキャッシュサイズを設定します。この例では、データベースキャッシュを 256 メガバイトに設定します。

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://server.example.com
backend config set --cache-autosize=0 --dbcachesize=268435456
```

2. Directory Server を再起動します。

```
[root@server ~]# systemctl restart dirsrv.target
```

3. IdM ディレクトリーサーバーのパフォーマンスを監視します。望ましい方法で変更が行われない場合は、この手順を繰り返して **dbcachesize** を別の値に調整するか、キャッシュの自動サイズ設定を再度有効にします。

検証手順

- **nsslapd-dbcachesize** 属性の値を表示し、目的の値に設定されていることを確認します。

```
[root@server ~]# ldapsearch -D "cn=directory manager" -w DirectoryManagerPassword
-b "cn=config,cn=ldbm database,cn=plugins,cn=config" | grep nsslapd-dbcachesize
nsslapd-dbcachesize: 2147483648
```

関連情報

- Directory Server 11 ドキュメント の nsslapd-dbcachesize
- エントリーおよびデータベースキャッシュの自動サイズ設定の再有効化

6.3. データベースとエントリーキャッシュの自動サイズ設定の再有効化

重要

Red Hat は、パフォーマンスを最適化するために、内蔵キャッシュの自動サイズ設定機能を使用することを推奨します。Red Hat では、キャッシュサイズを手動で設定することは推奨されていません。

デフォルトでは、IdM Directory Server は、データベースキャッシュおよびエントリーキャッシュに最適なサイズを自動的に判断します。自動サイズ設定では、空き RAM の一部が確保され、インスタンスの起動時に、サーバーのハードウェアリソースに基づいて両方のキャッシュのサイズが最適化されます。

この手順を使用して、カスタムデータベースキャッシュとエントリーキャッシュの値を元に戻し、キャッシュの自動サイズ設定機能をデフォルト値に復元します。

nsslapd-cache-autosize	この設定では、データベースおよびエントリーキャッシュの自動サイズ設定に割り当てる空き RAM の量を制御します。**0** に設定すると、自動サイズ設定が無効になります。
デフォルト値	**10** (空き RAM の 10%)
有効範囲	**0 - 100**
エントリー DN の場所	**cn=config,cn=ldbm database,cn=plugins,cn=config**

nsslapd-cache-autosize-split	この値は、データベースキャッシュに使用される **nsslapd-cache-autosize** により決定される空きメモリーの割合を設定します。残りの割合はエントリーキャッシュに使用されます。

デフォルト値	**25** (データベースキャッシュの場合は 25%、エントリーキャッシュの場合は 60%)
有効範囲	**0 - 100**
エントリー DN の場所	**cn=config,cn=ldbm database,cn=plugins,cn=config**

前提条件

- データベースとエントリーキャッシュのオートチューニングを以前に無効にしている。

手順

1. Directory Server を停止します。

```
[root@server ~]# systemctl stop dirsrv.target
```

2. **/etc/dirsrv/slapd-instance_name/dse.ldif** のバックアップを作成してから、修正を行ってください。

```
[root@server ~]# *cp /etc/dirsrv/slapd-instance_name/dse.ldif \
    /etc/dirsrv/slapd-instance_name/dse.ldif.bak.$(date "+%F_%H-%M-%S")
```

3. **/etc/dirsrv/slapd-instance_name/dse.ldif** ファイルを編集します。

   a. データベースおよびエントリーキャッシュに使用する空きシステム RAM の割合を設定し、デフォルトの空き RAM の 10% に戻します。

```
nsslapd-cache-autosize: 10
```

   b. データベースキャッシュの空きシステム RAM から使用されている割合を、デフォルトの 25% に設定します。

```
nsslapd-cache-autosize-split: 25
```

4. 変更を **/etc/dirsrv/slapd-instance_name/dse.ldif** に保存します。

5. Directory Server を起動します。

```
[root@server ~]# systemctl start dirsrv.target
```

検証手順

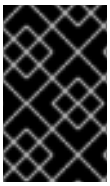
- **nsslapd-cache-autosize** 属性および **nsslapd-cache-autosize-split** 属性の値を表示し、必要な値に設定されていることを確認します。

```
[root@server ~]# ldapsearch -D "cn=directory manager" -w DirectoryManagerPassword
-b "cn=config,cn=ldbm database,cn=plugins,cn=config" | grep nsslapd-cache-autosize
nsslapd-cache-autosize: *10
nsslapd-cache-autosize-split: 25
```

関連情報

- Directory Server 11 ドキュメントの nsslapd-cache-autosize

6.4. DN キャッシュサイズの調整



重要

Red Hat は、パフォーマンスを最適化するために、内蔵キャッシュの自動サイズ設定機能を使用することを推奨します。オートチューニングした値から意図的に逸脱する必要がある場合に限り、この値を変更してください。

nsslapd-dncachememsize 属性は、識別名 (DN) キャッシュに使用できるメモリー領域のサイズ (バイト) を指定します。DN キャッシュはデータベースのエントリーキャッシュと似ていますが、そのテーブルにはエントリー ID とエントリー DN のみが保存されるため、**rename** 操作や**moddn** 操作での検索が速くなります。

デフォルト値	**10485760** (10 MB)
有効範囲	**500000 - 18446744073709551615** (500 kB - (2^{64}-1))
エントリー DN の場所	**cn=database-name,cn=ldbm database,cn=plugins,cn=config**

前提条件

- LDAP Directory Manager のパスワード

手順

1. **(任意)** データベースの接尾辞と、対応するデータベース名を表示します。

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://server.example.com backend suffix list
dc=example,dc=com (userroot)
```

このコマンドにより、各接尾辞の横にバックエンドデータベースが表示されます。次の手順では、接尾辞のデータベース名を使用します。

2. データベースの DN キャッシュサイズを設定します。この例では、DN キャッシュを 20 メガバイトに設定します。

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://server.example.com backend suffix set --dncache-memsize=20971520 userroot
```

3. Directory Server を再起動します。

```
[root@server ~]# systemctl restart dirsrv.target
```

4. IdM ディレクトリーサーバーのパフォーマンスを監視します。適切な変更が行われない場合は、この手順を繰り返して **dncache-memsize** を別の値に調整するか、デフォルトの 10MB に戻します。

検証手順

- **nsslapd-dncachememsize** 属性の新しい値を表示し、目的の値に設定されていることを確認します。

```
[root@server ~]# ldapsearch -D "cn=directory manager" -w DirectoryManagerPassword
-b "cn=userroot,cn=ldbm database,cn=plugins,cn=config" | grep nsslapd-
dncachememsize
nsslapd-dncachememsize: 20971520
```

関連情報

- Directory Server 11 ドキュメントの nsslapd-dncachememsize

6.5. 正規化された DN キャッシュサイズの調整

重要

Red Hat は、パフォーマンスを最適化するために、内蔵キャッシュの自動サイズ設定機能を使用することを推奨します。オートチューニングした値から意図的に逸脱する必要がある場合に限り、この値を変更してください。

nsslapd-ndn-cache-max-size 属性は、正規化識別名 (NDN) を保存するキャッシュのサイズ (バイト) を制御します。この値を上げると、頻繁に使用される DN がメモリー内に保持されます。

デフォルト値	**20971520** (20 MB)
有効範囲	**0 - 2147483647**
エントリー DN の場所	**cn=config**

前提条件

- LDAP Directory Manager のパスワード

手順

1. NDN キャッシュが有効になっていることを確認します。

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://server.example.com config
get nsslapd-ndn-cache-enabled
Enter password for cn=Directory Manager on ldap://server.example.com:
nsslapd-ndn-cache-enabled: on
```

キャッシュが **off** の場合は、次のコマンドを使用して有効にします。

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://server.example.com config
```

```
replace nsslapd-ndn-cache-enabled=on
Enter password for cn=Directory Manager on ldap://server.example.com:
Successfully replaced "nsslapd-ndn-cache-enabled"
```

2. **nsslapd-ndn-cache-max-size** パラメーターの現在値を取得し、メモを取ってから、調整を行ってください (復元が必要な場合)。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。

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://server.example.com config get nsslapd-ndn-cache-max-size
Enter password for cn=Directory Manager on ldap://server.example.com:
nsslapd-ndn-cache-max-size: 20971520
```

3. **nsslapd-ndn-cache-max-size** 属性の値を変更します。この例では、数値を **41943040** (40MB) に上げています。

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://server.example.com config replace nsslapd-ndn-cache-max-size=41943040
```

4. IdM ディレクトリーサーバーのパフォーマンスを監視します。望ましい方法で変更が行われない場合は、この手順を繰り返して**nsslapd-ndn-cache-max-size** を別の値に調整するか、キャッシュの自動サイズ設定を再度有効にします。

検証手順

- **nsslapd-ndn-cache-max-size** 属性の新しい値を表示し、目的の値に設定されていることを確認します。

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://server.example.com config get nsslapd-ndn-cache-max-size
Enter password for cn=Directory Manager on ldap://server.example.com:
nsslapd-ndn-cache-max-size: 41943040
```

関連情報

- Directory Server 11 ドキュメントの nsslapd-ndn-cache-max-size

6.6. メッセージの最大サイズの調整

nsslapd-maxbersize 属性は、受信メッセージまたは LDAP リクエストに許可される最大サイズ (バイト) を設定します。リクエストのサイズを制限することで、一部の種類のサービス拒否攻撃を防ぎます。

メッセージの最大サイズが小さすぎると、**/var/log/dirsrv/slapd-INSTANCE-NAME/errors** の Directory Server エラーログに以下のエラーが表示される場合があります。

```
Incoming BER Element was too long, max allowable is 2097152 bytes. Change the nsslapd-maxbersize attribute in cn=config to increase.
```

この制限は、LDAP 要求の合計サイズに適用されます。たとえば、リクエストでエントリーを追加する場合で、リクエストのエントリーが設定値またはデフォルトよりも大きい場合は、追加のリクエストが拒否されます。ただし、この制限はレプリケーションプロセスには適用されません。この属性を変更する前に注意してください。

デフォルト値	**209715200** (20 MB)
有効範囲	**0 - 2147483647**
エントリー DN の場所	**cn=config**

前提条件

- LDAP Directory Manager のパスワード

手順

1. **nsslapd-maxbersize** パラメーターの現在値を取得し、メモを取ってから、調整を行ってください (復元が必要な場合)。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。

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://server.example.com config
get nsslapd-maxbersize
Enter password for cn=Directory Manager on ldap://server.example.com:
nsslapd-maxbersize: 209715200
```

2. **nsslapd-maxbersize** 属性の値を変更します。この例では、数値を **419430400** に上げています。

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://server.example.com config
replace nsslapd-maxbersize=419430400
```

3. Directory Manager として認証して、設定を変更します。

```
Enter password for cn=Directory Manager on ldap://server.example.com:
Successfully replaced "nsslapd-maxbersize"
```

4. IdM ディレクトリーサーバーのパフォーマンスを監視します。適切な変更が行われない場合は、この手順を繰り返して**nsslapd-maxbersize** を別の値に調整するか、**209715200** の既定値に戻します。

検証手順

- **nsslapd-maxbersize** 属性の値を表示し、目的の値に設定されていることを確認します。

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://server.example.com config
get nsslapd-maxbersize
Enter password for cn=Directory Manager on ldap://server.example.com:
nsslapd-maxbersize: 419430400
```

関連情報

- Directory Server 11 ドキュメントの nsslapd-maxbersize (最大メッセージサイズ)

6.7. ファイルディスクリプターの最大数の調整

nsslapd-maxdescriptors 属性は、Directory Server が使用するプラットフォーム依存のファイル記述子の最大数を設定します。ファイル記述子は、クライアント接続、ログファイル、ソケット、およびその他のリソースに使用されます。

nsslapd-maxdescriptors の値を、オペレーティングシステムが **ns-slapd** プロセスで使用できるファイルディスクリプターの合計よりも大きく設定すると、Directory Server は、オペレーティングシステムに許容可能な最大値を問い合わせてから、その値を使用します。

デフォルト値	**4096** 記述子
有効範囲	**1 - 65535**
エントリー DN の場所	**cn=config**

前提条件

- LDAP Directory Manager のパスワード

手順

1. **nsslapd-maxdescriptors** パラメーターの現在値を取得し、メモを取ってから、調整を行ってください (復元が必要な場合)。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。

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://server.example.com config get nsslapd-maxdescriptors
Enter password for cn=Directory Manager on ldap://server.example.com:
nsslapd-maxdescriptors: 4096
```

2. **nsslapd-maxdescriptors** 属性の値を変更します。この例では、数値を **8192** に上げています。

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://server.example.com config replace nsslapd-maxdescriptors=8192
```

3. Directory Manager として認証して、設定を変更します。

```
Enter password for cn=Directory Manager on ldap://server.example.com:
Successfully replaced "nsslapd-maxdescriptors"
```

4. IdM ディレクトリーサーバーのパフォーマンスを監視します。適切な変更が行われない場合は、この手順を繰り返して**nsslapd-maxdescriptors** を別の値に調整するか、**4096** の既定値に戻します。

検証手順

- **nsslapd-maxdescriptors** 属性の値を表示し、目的の値に設定されていることを確認します。

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://server.example.com config get nsslapd-maxdescriptors
Enter password for cn=Directory Manager on ldap://server.example.com:
nsslapd-maxdescriptors: 8192
```

関連情報

- Directory Server 11 ドキュメントの nsslapd-maxdescriptors (最大ファイル記述子)

6.8. 接続バックログサイズの調整

listen サービスは、着信接続の受信に使用できるソケット数を設定します。**nsslapd-listen-backlog-size** は、接続を拒否するまでの **sockfd** ソケットのキューの最大長を設定します。

大量の接続を処理する場合は、**nsslapd-listen-backlog-size** の値を大きくすることを検討してください。

デフォルト値	**128** キュースロット
有効範囲	**0 - 9223372036854775807**
エントリー DN の場所	**cn=config**

前提条件

- LDAP Directory Manager のパスワード

手順

1. **nsslapd-listen-backlog-size** パラメーターの現在値を取得し、メモを取ってから、調整を行ってください (復元が必要な場合)。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。

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://server.example.com config get nsslapd-listen-backlog-size
Enter password for cn=Directory Manager on ldap://server.example.com:
nsslapd-listen-backlog-size: 128
```

2. **nsslapd-listen-backlog-size** 属性の値を変更します。この例では、数値を **192** に上げています。

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://server.example.com config replace nsslapd-listen-backlog-size=192
```

3. Directory Manager として認証して、設定を変更します。

```
Enter password for cn=Directory Manager on ldap://server.example.com:
Successfully replaced "nsslapd-listen-backlog-size"
```

検証手順

- **nsslapd-listen-backlog-size** 属性の値を表示し、目的の値に設定されていることを確認します。

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://server.example.com config get nsslapd-listen-backlog-size
Enter password for cn=Directory Manager on ldap://server.example.com:
```

```
nsslapd-listen-backlog-size: 192
```

関連情報

- Directory Server 11 のドキュメントの nsslapd-listen-backlog-size

6.9. データベースロックの最大数の調整

ロックメカニズムは、Directory Server プロセスが同時に実行できるコピー数を制御し、**nsslapd-db-locks** パラメーターはロックの最大数を設定します。

/var/log/dirsrv/slapd-instance_name/errors ログファイルに以下のエラーメッセージが表示された場合は、ロックの上限を引き上げます。

```
libdb: Lock table is out of available locks
```

デフォルト値	**50000** ロック
有効範囲	**0 - 2147483647**
エントリー DN の場所	**cn=bdb,cn=config,cn=ldbm database,cn=plugins,cn=config**

前提条件

- LDAP Directory Manager のパスワード

手順

1. **nsslapd-db-locks** パラメーターの現在値を取得し、メモを取ってから、調整を行ってください (復元が必要な場合)。

```
[root@server ~]# ldapsearch -D "cn=directory manager" -w DirectoryManagerPassword -b "cn=bdb,cn=config,cn=ldbm database,cn=plugins,cn=config" | grep nsslapd-db-locks
nsslapd-db-locks: 50000
```

2. **locks** の値を変更します。この例では、値を **100000** ロック の倍にしています。

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://server.example.com backend config set --locks=100000
```

3. Directory Manager として認証して、設定を変更します。

```
Enter password for cn=Directory Manager on ldap://server.example.com:
Successfully updated database configuration
```

4. Directory Server を再起動します。

```
[root@server ~]# systemctl restart dirsrv.target
```

検証手順

- **nsslapd-db-locks** 属性の値を表示し、目的の値に設定されていることを確認します。

```
[root@server ~]# ldapsearch -D "cn=directory manager" -w DirectoryManagerPassword
-b "cn=bdb,cn=config,cn=ldbm database,cn=plugins,cn=config" | grep nsslapd-db-
locks
nsslapd-db-locks: 100000
```

関連情報

- Directory Server 11 ドキュメントの nsslapd-db-locks

6.10. 入出力ブロックのタイムアウト調整

nsslapd-ioblocktimeout 属性は、停止している LDAP クライアントへの接続を閉じるまでの時間をミリ秒単位で設定します。読み取り操作または書き込み操作で I/O の進捗がなかった場合、LDAP クライアントは停止していると見なされます。

接続をより早く解放するには、**nsslapd-ioblocktimeout** アトリビュートの値を下げる必要があります。

デフォルト値	**10000** ミリ秒
有効範囲	**0 - 2147483647**
エントリー DN の場所	**cn=config**

前提条件

- LDAP Directory Manager のパスワード

手順

1. **nsslapd-ioblocktimeout** パラメーターの現在値を取得し、メモを取ってから、調整を行ってください (復元が必要な場合)。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。

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://server.example.com config
get nsslapd-ioblocktimeout
Enter password for cn=Directory Manager on ldap://server.example.com:
nsslapd-ioblocktimeout: 10000
```

2. **nsslapd-ioblocktimeout** 属性の値を変更します。この例では、値を **8000** に下げています。

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://server.example.com config
replace nsslapd-ioblocktimeout=8000
```

3. Directory Manager として認証して、設定を変更します。

```
Enter password for cn=Directory Manager on ldap://server.example.com:
Successfully replaced "nsslapd-ioblocktimeout"
```

4. IdM ディレクトリーサーバーのパフォーマンスを監視します。適切な変更が行われない場合は、この手順を繰り返し **nsslapd-ioblocktimeout** を別の値に調整するか、**10000** の既定値に戻します。

検証手順

- **nsslapd-ioblocktimeout** 属性の値を表示し、目的の値に設定されていることを確認します。

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://server.example.com config get nsslapd-ioblocktimeout
Enter password for cn=Directory Manager on ldap://server.example.com:
nsslapd-idletimeout: 8000
```

関連情報

- Directory Server 11 ドキュメントの nsslapd-ioblocktimeout (IO ブロックタイムアウト)

6.11. アイドル接続のタイムアウトの調整

nsslapd-idletimeout 属性は、アイドル状態の LDAP クライアント接続が IdM サーバーにより閉じられるまでの秒数を設定します。**0** に設定すると、サーバーはアイドル状態の接続を閉じなくなります。

Red Hat では、古い接続は閉じても、アクティブな接続は早めに閉じないように、この値を調整することを推奨しています。

デフォルト値	**3600** 秒 (1 時間)
有効範囲	**0 - 2147483647**
エントリー DN の場所	**cn=config**

前提条件

- LDAP Directory Manager のパスワード

手順

1. **nsslapd-idletimeout** パラメーターの現在値を取得し、メモを取ってから、調整を行ってください (復元が必要な場合)。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。

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://server.example.com config get nsslapd-idletimeout
Enter password for cn=Directory Manager on ldap://server.example.com:
nsslapd-idletimeout: 3600
```

2. **nsslapd-idletimeout** 属性の値を変更します。この例では、値を **1800** (30 分) に下げています。

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://server.example.com config replace nsslapd-idletimeout=1800
```

3. Directory Manager として認証して、設定を変更します。

```
Enter password for cn=Directory Manager on ldap://server.example.com:
Successfully replaced "nsslapd-idletimeout"
```

4. IdM ディレクトリーサーバーのパフォーマンスを監視します。適切な変更が行われない場合は、この手順を繰り返して**nsslapd-idletimeout** を別の値に調整するか、**3600** の既定値に戻します。

検証手順

- **nsslapd-idletimeout** 属性の値を表示し、目的の値に設定されていることを確認します。

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://server.example.com config get nsslapd-idletimeout
Enter password for cn=Directory Manager on ldap://server.example.com:
nsslapd-idletimeout: 3600
```

関連情報

- Directory Server 11 ドキュメントの nsslapd-idletimeout(Default Idle Timeout)

6.12. レプリケーションリリースのタイムアウトの調整

IdM レプリカは、別のレプリカとのレプリケーションセッション中に排他的にロックされます。一部の環境では、大規模な更新やネットワークの輻輳によりレプリカが長期間ロックされると、レプリカの待ち時間が長くなります。

repl-release-timeout パラメーターを調整することで、決められた時間が経過した後にレプリカを解放できます。Red Hat では、**30** と **120** の間で設定することを推奨しています。

- この値を設定しすぎると、レプリカが常に互いに再取得し合い、レプリカがより大きな更新を送信できなくなります。
- タイムアウトを長くすると、トラフィックの多い状況を改善できます。この状況では、サーバーが長時間レプリカに排他的にアクセスするのが最善ですが、**120** 秒を超えるとレプリカの速度が低下します。

デフォルト値	**60** 秒
有効範囲	**0 - 2147483647**
推奨範囲	**30 - 120**

前提条件

- LDAP Directory Manager のパスワード

手順

1. データベースの接尾辞と、対応するバックエンドを表示します。

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://server.example.com
backend suffix list
cn=changelog (changelog)
dc=example,dc=com (userroot)
o=ipaca (ipaca)
```

このコマンドは、バックエンドデータベースの接尾辞の横に名前を表示します。次の手順で接尾辞の名前を使用します。

2. メインユーザールートデータベースの **repl-release-timeout** 属性の値を変更します。この例では、値を **90** 秒に増加させています。

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://server.example.com
replication set --suffix="dc=example,dc=com" --repl-release-timeout=90
```

3. Directory Manager として認証して、設定を変更します。

```
Enter password for cn=Directory Manager on ldap://server.example.com:
Successfully replaced "repl-release-timeout"
```

4. (必要に応じて) IdM 環境が IdM 認証局 (CA) を使用する場合は、CA データベースの **repl-release-timeout** 属性の値を変更できます。この例では、値を **90** 秒に増加させています。

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://server.example.com replication
set --suffix="o=ipaca" --repl-release-timeout=90
Enter password for cn=Directory Manager on ldap://server.example.com:
Successfully replaced "repl-release-timeout"
```

5. Directory Server を再起動します。

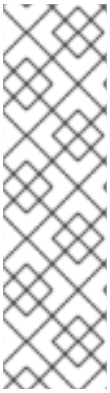
```
[root@server ~]# systemctl restart dirsrv.target
```

6. IdM ディレクトリーサーバーのパフォーマンスを監視します。適切な変更が行われない場合は、この手順を繰り返して**repl-release-timeout** を別の値に調整するか、**60** 秒の既定値に戻します。

検証手順

- **nsds5ReplicaReleaseTimeout** 属性の値を表示し、目的の値に設定されていることを確認します。

```
[root@server ~]# ldapsearch -D "cn=directory manager" -w DirectoryManagerPassword
-b "cn=replica,cn=dc\3Dexample\2Cdc\3Dcom,cn=mapping tree,cn=config" | grep
nsds5ReplicaReleaseTimeout
nsds5ReplicaReleaseTimeout: 90
```



注記

この例の接尾辞の識別名は **dc=example,dc=com** ですが、**ldapsearch** では、等号 (**=**) とコンマ (**,**) をエスケープする必要があります。

接尾辞 DN を、以下のエスケープ文字を使用して **cn=dc\3Dexample\2Cdc\3Dcom** に変換します。

- **=** を **\3D**に
- **,** を **\2C**に

関連情報

- Directory Server 11 ドキュメントの nsDS5ReplicaReleaseTimeout

6.13. LDIF ファイルからのカスタムデータベース設定を使用した IDM サーバーまたはレプリカのインストール

Directory Server データベースのカスタム設定を使用して、IdM サーバーおよび IdM レプリカをインストールできます。以下の手順は、データベース設定で LDAP データ交換形式 (LDIF) ファイルを作成する方法と、その設定を IdM サーバーおよびレプリカインストールコマンドに渡す方法を示しています。

前提条件

- IdM 環境のパフォーマンスを向上させるカスタムの Directory Server 設定を行っている。IdM Directory Server パフォーマンスの調整 を参照してください。

手順

1. カスタムデータベース設定で LDIF 形式のテキストファイルを作成します。LDAP 属性の変更はダッシュ (-) で区切ります。この例では、idle タイムアウトおよび最大ファイルディスクリプターにデフォルト以外の値を設定します。

```
dn: cn=config
changetype: modify
replace: nsslapd-idletimeout
nsslapd-idletimeout=1800
-
replace: nsslapd-maxdescriptors
nsslapd-maxdescriptors=8192
```

2. **--dirsrv-config-file** パラメーターを使用して、LDIF ファイルをインストールスクリプトに渡します。

   a. IdM サーバーをインストールするには、次のコマンドを実行します。

```
# ipa-server-install --dirsrv-config-file filename.ldif
```

   b. IdM レプリカをインストールするには、次のコマンドを実行します。

```
# ipa-replica-install --dirsrv-config-file filename.ldif
```

関連情報

- **ipa-server-install** コマンドおよび **ipa-replica-install** コマンドのオプション

6.14. 関連情報

- Directory Server 11 パフォーマンスチューニングガイド

第7章 KDC のパフォーマンスの調整

次のセクションでは、ユーザー、ホスト、およびサービスの認証を担当する Kerberos Key Distribution Center (KDC) のパフォーマンスを調整する方法を説明します。

7.1. KDC リッスンキューの長さの調整

/var/kerberos/krb5kdc/kdc.conf ファイルの **[kdcdefaults]** セクションで **kdc_tcp_listen_backlog** オプションを設定することにより、KDC デーモンのリッスンキューの長さのサイズを調整できます。**5** のデフォルト値は、大量の Kerberos トラフィックが発生する IdM デプロイメントでは低すぎる可能性がありますが、この値を高く設定しすぎるとパフォーマンスが低下します。

デフォルト値	**5**
有効範囲	**1 - 10**

手順

1. テキストエディターで **/var/kerberos/krb5kdc/kdc.conf** を開きます。
2. TCP リッスンバックログを **7** などの目的の値に設定します。

```
[kdcdefaults]
 ...
kdc_tcp_listen_backlog = 7
```

3. **/var/kerberos/krb5kdc/kdc.conf** ファイルを保存して閉じます。
4. KDC を再起動して、新しい設定を読み込みます。

7.2. レルムごとの KDC の動作を制御するオプション

各 Kerberos レルムのユーザーアカウントのロックおよびロック解除を追跡するため、認証に成功および失敗するたびに、KDC がデータベースに書き込みます。**/etc/krb5.conf** ファイルの **[dbmodules]** セクションで以下のオプションを調整することで、KDC が情報を書き込む頻度を最小限にとどめることで、パフォーマンスを改善できる場合があります。

disable_last_success

true に設定すると、事前認証を必要とするプリンシパルエントリーの **Last successful authentication** フィールドへの KDC 更新を抑制します。

デフォルト値	**false**
有効範囲	**true** または **false**

disable_lockout

true に設定すると、事前認証を必要とするプリンシパルエントリーの **Last failed authentication** および **Failed password attempts** フィールドへの KDC 更新を抑制します。このフラグを設定するとパフォーマンスが向上しますが、アカウントのロックアウトを無効にすることはセキュリティー上のリスクと見なされる場合があります。

デフォルト値	**false**
有効範囲	**true** または **false**

関連情報

- レルムごとの KDC 設定の調整

7.3. レルムごとの KDC 設定の調整

この手順では、Kerberos レルムごとに KDC の動作を調整します。

手順

1. テキストエディターで **/etc/krb5.conf** を開きます。
2. **[dbmodules]** セクションと、各 Kerberos レルムで、オプションと必要な値を指定します。この例では、**EXAMPLE.COM** Kerberos レルムの **disable_last_success** 変数を設定します。

```
[dbmodules]
    EXAMPLE.COM = {
        disable_last_success = true
    }
```

3. **/etc/krb5.conf** を保存して閉じます。
4. KDC を再起動して、新しい設定を読み込みます。

関連情報

- レルムごとの KDC の動作を制御するオプション

7.4. KRB5KDC プロセス数の調整

この手順では、Key Distribution Center (KDC) が着信接続の処理を開始するプロセスの数を手動で調整する方法を説明します。

デフォルトでは、IdM インストーラーは CPU コアの数を検出し、その値を **/etc/sysconfig/krb5kdc** ファイルに入力します。たとえば、ファイルには次のエントリーが含まれている場合があります。

```
KRB5KDC_ARGS='-w 2'
[...]
```

この例では、**KRB5KDC_ARGS** パラメーターを **-w 2** に設定すると、KDC は 2 つの別個のプロセスを開始して、メインプロセスからの着信接続を処理します。特に、要件に基づいて仮想 CPU の数を簡単に追加または削除できる仮想環境では、この値を調整することが推奨されます。ポート 88 の TCP/IP キューが増え続けてパフォーマンスの問題が発生したり、IdM サーバーが応答しなくなったりするのを防ぐには、**KRB5KDC_ARGS** パラメーターを手動で高い値に設定して、より多くのプロセスをシミュレートします。

手順

1. **/etc/sysconfig/krb5kdc** ファイルをテキストエディターで開きます。
2. **KRB5KDC_ARGS** パラメーターの値を指定します。この例では、プロセスの数を 10 に設定しています。

```
KRB5KDC_ARGS='-w 10'
[...]
```

3. **/etc/sysconfig/krb5kdc** ファイルを保存して閉じます。
4. systemd 設定をリロードします。

```
# systemctl daemon-reload
```

5. **krb5kdc** サービスを再起動します。

```
# systemctl restart krb5kdc.service
```

7.5. 関連情報

- MIT Kerberos ドキュメント - kdc.conf

第8章 大規模な IDM-AD 信頼デプロイメントのための SSSD パフォーマンスの調整

ユーザーおよびグループ情報の取得は、特に AD (System Security Services Daemon) ドメイン、つまり大規模な Active Directory (AD) ドメインへの信頼を持つ IdM デプロイメントでは、データ集中型の操作です。SSSDがアイデンティティプロバイダーから取得する情報とその期間を調整することで、このパフォーマンスを向上させることができます。

8.1. 大規模な IDM-AD 信頼デプロイメント向けの IDM サーバーでの SSSD の調整

この手順では、IdM サーバーで SSSD サービスの設定に調整オプションを適用して、大規模な AD 環境から情報を取得する際の応答時間を改善します。

前提条件

- **/etc/sssd/sssd.conf** 設定ファイルを編集するには、**root** のパーミッションが必要です。

手順

1. テキストエディターで **/etc/sssd/sssd.conf** 設定ファイルを開きます。
2. Active Directory ドメインの **[ドメイン]** に次のオプションを追加します。AD ドメインのドメインセクションがない場合は、作成します。

```
[domain/ad.example.com]
ignore_group_members = true
subdomain_inherit = ignore_group_members
...
```

3. サーバー上の **/etc/sssd/sssd.conf** ファイルを保存して閉じます。
4. SSSD サービスを再起動して、設定の変更を読み込みます。

```
[root@client ~]# systemctl restart sssd
```

関連情報

- IdM サーバーおよびクライアントで、大規模な IdM-AD 信頼デプロイメント用に SSSD を調整するオプション

8.2. IDM サーバーでの IPA-EXTDOM プラグインの設定タイムアウトの調整

IdM クライアントは Active Directory (AD) からユーザーとグループに関する情報を直接受信できないため、IdM サーバーは **ipa-extdom** プラグインを使用して AD ユーザーとグループに関する情報を受信し、その情報は要求元のクライアントに転送されます。

ipa-extdom プラグインは、AD ユーザーのデータに要求を SSSD に送信します。情報が SSSD キャッシュにない場合、SSSD は AD ドメインコントローラー (DC) にデータを要求します。config タイムアウト値を調整できます。これは、プラグインが接続をキャンセルして発信者にタイムアウトエラーを返す前に、**ipa-extdom** プラグインが SSSD からの応答を待機する時間を定義します。デフォルト値は 10000 ミリ秒 (10 秒) です。

次の例では、設定タイムアウトを 20 秒 (20000 ミリ秒) に調整します。

警告

設定タイムアウトを調整するときは注意してください。

- 設定する値が小さすぎる (例: 500 ミリ秒) と、SSSD に応答するのに十分な時間がない可能性があり、要求は常にタイムアウトを返します。
- 設定する値が大きすぎる (例: 30000 ミリ秒 (30 秒)) と、1 つの要求が、この期間、SSSD への接続をブロックする可能性があります。一度に SSSD に接続できるのは 1 つのスレッドであるため、プラグインからの他のリクエストはすべて待機する必要があります。
- IdM クライアントから送信された要求が多い場合、IdM サーバー上の Directory Server 用に設定された使用可能なすべてのワーカーをブロックできます。その結果、サーバーはしばらくの間、どのような種類の要求にも応答できない可能性があります。

以下の状況で設定のタイムアウトを変更します。

- AD ユーザーおよびグループに関する情報を要求する際に、独自の検索タイムアウトが発生する前に IdM クライアントが頻繁にタイムアウトエラーを受け取ると、設定のタイムアウト値が **小さすぎ** ます。
- IdM サーバーで Directory Server がロックされていることが多く、**pstack** ユーティリティーは、この時点で多数またはすべてのワーカースレッドが **ipa-extdom** 要求を処理していることを報告する場合は、値が **大きすぎ** ます。

前提条件

- LDAP Directory Manager のパスワード

手順

- 次のコマンドを使用して、設定タイムアウトを 20000 ミリ秒に調整します。

```
# ldapmodify -D "cn=directory manager" -W
dn: cn=ipa_extdom_extop,cn=plugins,cn=config
changetype: modify
replace: ipaExtdomMaxNssTimeout
ipaExtdomMaxNssTimeout: 20000
```

8.3. IDM サーバー上の IPA-EXTDOM プラグインの最大バッファーサイズの調整

IdM クライアントは Active Directory (AD) からユーザーとグループに関する情報を直接受信できないため、IdM サーバーは **ipa-extdom** プラグインを使用して AD ユーザーとグループに関する情報を受信し、その情報は要求元のクライアントに転送されます。

ipa-extdom プラグインの最大バッファーサイズを調整できます。これにより、SSSD が受信するデータを保存できるバッファーのサイズが調整されます。バッファーが小さすぎると、SSSD は **ERANGE** エラーを返し、プラグインはより大きなバッファーで要求を再試行します。デフォルトのバッファーサイズは 134217728 バイト (128 MB) です。

次の例では、最大バッファーサイズを 256 MB (268435456 バイト) に調整します。

前提条件

- LDAP Directory Manager のパスワード

手順

- 次のコマンドを使用して、最大バッファーサイズを 268435456 バイトに設定します。

```
# ldapmodify -D "cn=directory manager" -W
dn: cn=ipa_extdom_extop,cn=plugins,cn=config
changetype: modify
replace: ipaExtdomMaxNssBufSize
ipaExtdomMaxNssBufSize: 268435456
```

8.4. IDM サーバーの IPA-EXTDOM プラグインのインスタンスの最大数の調整

IdM クライアントは、Active Directory(AD)から直接ユーザーおよびグループに関する情報を受け取ることができないため、IdM サーバーは **ipa-extdom** プラグインを使用して AD ユーザーおよびグループに関する情報を受信し、この情報を要求元のクライアントに転送します。

デフォルトでは、**ipa-extdom** プラグインは、IdM クライアントからの要求を処理するために LDAP ワーカースレッドの最大 80% を使用するように設定されています。IdM クライアントの SSSD サービスが、AD 信頼ユーザーおよびグループに関する情報を大量に要求している場合、LDAP サービスはほとんどの LDAP スレッドを使用する場合、この操作は LDAP サービスを停止できます。このような問題が発生した場合は、AD ドメインの SSSD ログファイル(**/var/log/sssd/sssd__your-ad-domain-name.com_.log**)にも同様のエラーが表示されます。

```
(2022-05-22  5:00:13): [be[ad.example.com]] [ipa_s2n_get_user_done] (0x0040): s2n exop request failed.
(2022-05-22  5:00:13): [be[ad.example.com]] [ipa_s2n_get_user_done] (0x0040): s2n exop request failed.
(2022-05-22  5:00:13): [be[ad.example.com]] [ipa_s2n_exop_done] (0x0040): ldap_extended_operation result: Server is busy(51), Too many extdom instances running.
```

ipaExtdomMaxInstances オプションの値を設定することで、**ipa-extdom** インスタンスの最大数を調整できます。これは、0 以上の整数でワーカースレッドの合計数より小さくなければなりません。

前提条件

- LDAP Directory Manager のパスワード

手順

1. ワーカースレッドの合計数を取得します。

```
# ldapsearch -xLLLD cn=directory\ manager -W -b cn=config -s base nsslapd-threadnumber
Enter LDAP Password:
dn: cn=config
nsslapd-threadnumber: 16
```

これは、**ipaExtdomMaxInstances** の現在の値が 13 であることを意味します。

2. インスタンスの最大数を調整します。この例では、値を 14 に変更します。

```
# ldapmodify -D "cn=directory manager" -W
dn: cn=ipa_extdom_extop,cn=plugins,cn=config
changetype: modify
replace: ipaExtdomMaxInstances
ipaExtdomMaxInstances: 14
```

3. IdM ディレクトリーサーバーのパフォーマンスを監視し、改善しなかった場合には、この手順を繰り返して **ipaExtdomMaxInstances** 変数の値を調整します。

8.5. 大規模な IDM-AD 信頼デプロイメント向けの IDM クライアントでの SSSD の調整

この手順では、IdM クライアントで SSSD サービス設定に調整オプションを適用し、大規模な AD 環境から情報を取得する際の応答時間を改善します。

前提条件

- **/etc/sssd/sssd.conf** 設定ファイルを編集するには、**root** のパーミッションが必要です。

手順

1. キャッシュされていない 1 回のログインにかかる秒数を判定します。
   a. IdM クライアント **client.example.com** の SSSD キャッシュを消去します。

```
[root@client ~]# sss_cache -E
```

   b. **time** で AD ユーザーとしてログインするのにかかる時間を測定します。この例では、IdM クライアント**client.example.com**から、**ad.example.com** AD ドメインのユーザー**アドユーザー** と同じホストにログインします。

```
[root@client ~]# time ssh ad-user@ad.example.com@client.example.com
```

   c. 早急にパスワードを入力してください。

```
Password:
Last login: Sat Jan 23 06:29:54 2021 from 10.0.2.15
[ad-user@ad.example.com@client ~]$
```

   d. できるだけ早くログアウトして、経過時間を表示します。この例では、キャッシュされていないログインは 1 回で約 **9** 秒かかります。

```
[ad-user@ad.example.com@client /]$ exit
logout
```

```
Connection to client.example.com closed.

real 0m8.755s
user    0m0.017s
sys     0m0.013s
```

2. テキストエディターで **/etc/sssd/sssd.conf** 設定ファイルを開きます。
3. Active Directory ドメインの **[ドメイン]** に次のオプションを追加します。**pam_id_timeout** オプションおよび **krb5_auth_timeout** オプションを、キャッシュされていないログインにかかる秒数に設定します。AD ドメインのドメインセクションがない場合は、作成します。

```
[domain/example.com/ad.example.com]
krb5_auth_timeout = 9
ldap_deref_threshold = 0
...
```

4. **[pam]** セクションに以下のオプションを追加します。

```
[pam]
pam_id_timeout = 9
```

5. サーバー上の **/etc/sssd/sssd.conf** ファイルを保存して閉じます。
6. SSSD サービスを再起動して、設定の変更を読み込みます。

```
[root@client ~]# systemctl restart sssd
```

関連情報

- IdM サーバーおよびクライアントで、大規模な IdM-AD 信頼デプロイメント用に SSSD を調整するオプション

8.6. TMPFS への SSSD キャッシュのマウント

SSSD (System Security Services Daemon) は、LDAP オブジェクトをキャッシュに常に書き込みます。この内部 SSSD トランザクションはデータをディスクに書き込みますが、これは RAM (Random-Access Memory) からの読み書きに比べてはるかに遅くなります。

このパフォーマンスを向上させるには、RAM に SSSD キャッシュをマウントします。

留意事項

- SSSD キャッシュが RAM にある場合、システムの再起動後もキャッシュされた情報は持続しません。
- IdM サーバーの SSSD インスタンスは、同じホストの Directory Server との接続を失うことがないため、この変更を IdM サーバーで実行すると安全です。
- IdM クライアントでこの調整を実行しても、IdM サーバーへの接続が失われると、接続を再確立するまで、システムの再起動後にユーザーは認証できなくなります。

前提条件

- **/etc/fstab** 設定ファイルを変更するには、**root** のパーミッションが必要です。

手順

1. **tmpfs** 一時ファイルを作成します。

   a. RHEL 8.6 以降では、SSSD ユーザーが **config.ldb** ファイルを所有していることを確認します。

   ```
   # ls -al /var/lib/sss/db/config.ldb
   -rw-------. 1 sssd sssd 1286144 Jun  8 16:41 /var/lib/sss/db/config.ldb
   ```

   この場合は、**/etc/fstab** ファイルに以下のエントリーを 1 行として追加します。

   ```
   tmpfs /var/lib/sss/db/ tmpfs
   size=300M,mode=0700,uid=sssd,gid=sssd,rootcontext=system_u:object_r:sssd_var_lib_
   t:s0 0 0
   ```

   b. 8.6 未満の RHEL 8 バージョンでは、**config.ldb** ファイルが **root** ユーザーによって所有されます。

   ```
   # ls -al /var/lib/sss/db/config.ldb
   -rw-------. 1 root root 1286144 Jun  8 14:15 /var/lib/sss/db/config.ldb
   ```

   この場合は、**/etc/fstab** ファイルに以下のエントリーを 1 行として追加します。

   ```
   tmpfs /var/lib/sss/db/ tmpfs
   size=300M,mode=0700,rootcontext=system_u:object_r:sssd_var_lib_t:s0 0 0
   ```

   この例では、300MB のキャッシュを作成します。IdM ディレクトリーおよび AD ディレクトリーのサイズに応じて **size** パラメーターを調整します。推定 10,000 の LDAP エントリーごとに 100MB です。

2. 新しい SSSD キャッシュディレクトリーをマウントします。

   ```
   [root@host ~]# mount /var/lib/sss/db/
   ```

3. この設定変更を反映するには、SSSD を再起動します。

   ```
   [root@host ~]# systemctl restart sssd
   ```

8.7. 大規模な IDM-AD 信頼デプロイメント用に IDM サーバーとクライアントを調整するための SSSD.CONF のオプション

/etc/sssd/sssd.conf 設定ファイルで次のオプションを使用して、IdM-AD の信頼が大規模にデプロイされている場合に、IdM サーバーおよびクライアントで SSSD のパフォーマンスを調整できます。

8.7.1. IdM サーバーのチューニングオプション

ignore_group_members

ユーザーの認証および承認を行う際には、グループに属するすべてのユーザーではなく、そのユーザーがどのグループに属するかを把握することが重要です。**ignore_group_members** を **true** に設

定すると、SSSD はメンバーではなくグループオブジェクト自体の情報のみを取得するため、パフォーマンスが大幅に向上します。

注記

id user@ad-domain.com は、依然として正しいグループ一覧を返しますが、**getent group ad-group@ad-domain.com** は空の一覧を返します。

デフォルト値	**false**
推奨値	**true**

注記

デプロイメントで compat ツリーを持つ IdM サーバーがある場合は、このオプションを **true** に設定しないでください。

subdomain_inherit

subdomain_inherit オプションを使用すると、**ignore_group_members** 設定を信頼できる AD ドメインの設定に適用できます。**subdomain_inherit** オプションの設定は、メイン (IdM) ドメインおよび AD サブドメインの両方に適用されます。

デフォルト値	**none**
推奨値	**subdomain_inherit = ignore_group_members**

8.7.2. IdM クライアントのチューニングオプション

pam_id_timeout

このパラメーターは、ID ルックアップ中に ID プロバイダーへの過剰な往復を回避するために、PAM セッションの結果がキャッシュされる期間を制御します。複雑なグループメンバーシップが IdM サーバー、および IdM クライアント側で取り込まれている環境では、**5** 秒の省略時値では足りない場合があります。Red Hat では、**pam_id_timeout** に、キャッシュされていない 1 回のログインにかかる秒数を設定することを推奨しています。

デフォルト値	**5**
推奨値	**キャッシュされていない 1 回のログインにかかる秒数**

krb5_auth_timeout

krb5_auth_timeout の値を増大すると、ユーザーが多数のグループのメンバーである環境で複雑なグループ情報を処理する時間が長くなります。Red Hat では、この値を 1 回のキャッシュされていないログインにかかる秒数に設定することを推奨しています。

デフォルト値	**6**

推奨値	**キャッシュされていない1 回のログインにかかる秒数**

ldap_deref_threshold

間接参照ルックアップは、1 回の LDAP コールですべてのグループメンバーをフェッチする手段です。**ldap_deref_threshold** 値は、間接参照検索を開始するために内部キャッシュに存在しないグループメンバーの数を指定します。欠落しているメンバーが少ないと、個別に検索されます。大規模な環境では、dereference lookups の使用に時間がかかり、パフォーマンスが低下する場合があります。間接参照ルックアップを無効にするには、この項目を **0** に設定します。

デフォルト値	**10**
推奨値	**0**

8.8. 関連情報

- 大規模な IdM-AD 信頼デプロイメント向けの SSSD のパフォーマンスチューニング