



## Red Hat Enterprise Linux 9

# Identity Management を使用した障害復旧の実行

サーバーまたはデータ損失後の IdM の回復



# Red Hat Enterprise Linux 9 Identity Management を使用した障害復旧の実行

---

サーバーまたはデータ損失後の IdM の回復

## 法律上の通知

Copyright © 2024 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

## 概要

ハードウェア障害などによるサーバーとデータの損失シナリオは、IT 環境において最も高いリスクです。Red Hat Identity Management (IdM) 環境でこのようなイベントが発生した場合、回復プロセスは、問題のタイプ、IdM トポロジ、およびそのような状況を軽減するために取られたアクションによって異なります。たとえば、IdM レプリケーショントポロジで単一および複数のサーバーを回復でき、IdM バックアップとスナップショットを使用してデータを回復できます。リカバリー中またはリカバリー後に、DNS サーバーや Kerberos 設定などのクライアント設定の調整が必要になる場合があります。

## 目次

多様性を受け入れるオープンソースの強化 .....	3
RED HAT ドキュメントへのフィードバック (英語のみ) .....	4
第1章 IDM の障害シナリオ .....	5
第2章 レプリケーションを使用した1台のサーバーの復旧 .....	6
2.1. CA 更新サーバーの損失からの復旧 .....	6
2.2. 通常のレプリカ損失からの回復 .....	7
第3章 レプリケーションを使用した複数のサーバーの復旧 .....	9
3.1. CA なしのデプロイメントで複数のサーバーが失われた状態からの回復 .....	9
3.2. CA 更新サーバーが無効化された場合の複数のサーバー損失からの復旧 .....	9
3.3. CA 更新サーバーおよびその他のサーバーの損失からの復旧 .....	9
3.4. すべての CA レプリカの失われた状態からの復旧 .....	9
3.5. インフラストラクチャー全体の損失からの復旧 .....	10
第4章 仮想マシンスナップショットによるデータ損失からの復旧 .....	11
4.1. 仮想マシンのスナップショットのみからの復旧 .....	11
4.2. 部分的に機能する環境間の仮想マシンのスナップショットからの復旧 .....	12
4.3. 仮想マシンのスナップショットからの復元による新規 IDM 環境の確立 .....	14
第5章 IDM バックアップを使用したデータ損失からの復旧 .....	17
5.1. IDM バックアップから復元するタイミング .....	17
5.2. IDM バックアップから復元する際の注意点 .....	17
5.3. バックアップからの IDM サーバーの復元 .....	18
5.4. 暗号化されたバックアップからの復元 .....	21
第6章 ANSIBLE PLAYBOOK を使用した IDM サーバーの復元 .....	23
6.1. IDM 管理用の ANSIBLE コントロールノードの準備 .....	23
6.2. ANSIBLE を使用したサーバーに保存されているバックアップからの IDM サーバーの復元 .....	25
6.3. ANSIBLE を使用した ANSIBLE コントローラーに保存されているバックアップから IDM サーバーの復元 .....	26
6.4. ANSIBLE を使用した IDM サーバーのバックアップの ANSIBLE コントローラーへのコピー .....	27
6.5. ANSIBLE を使用した IDM サーバーのバックアップの ANSIBLE コントローラーから IDM サーバーへのコピー .....	29
6.6. ANSIBLE を使用した IDM サーバーからのバックアップの削除 .....	31
第7章 データ損失の管理 .....	33
7.1. 分離されたデータ損失への応答 .....	33
7.2. すべてのサーバー間の制限されたデータ損失への対応 .....	34
7.3. すべてのサーバー間の未定義のデータ損失への応答 .....	34
第8章 復旧時の IDM クライアントの調整 .....	36



## 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、[Red Hat CTO である Chris Wright のメッセージ](#) を参照してください。

## RED HAT ドキュメントへのフィードバック (英語のみ)

Red Hat ドキュメントに関するご意見やご感想をお寄せください。また、改善点があればお知らせください。

### Jira からのフィードバック送信 (アカウントが必要)

1. [Jira](#) の Web サイトにログインします。
2. 上部のナビゲーションバーで **Create** をクリックします。
3. **Summary** フィールドにわかりやすいタイトルを入力します。
4. **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追加してください。
5. ダイアログの下部にある **Create** をクリックします。



## 第1章 IDM の障害シナリオ

障害シナリオには、主に **サーバー損失** と **データ損失** の2種類があります。

表1.1 サーバー損失とデータ損失の比較

障害タイプ	考えられる原因	応答方法
<b>サーバー損失</b> - IdM デプロイメントからサーバーが1台以上なくなる	<ul style="list-style-type: none"><li>● ハードウェアの誤作動</li></ul>	<ul style="list-style-type: none"><li>● <a href="#">レプリケーションを使用した1台のサーバーの復旧</a></li></ul>
<b>データ損失</b> - サーバーで IdM データが突然修正され、変更が他のサーバーに伝播している。	<ul style="list-style-type: none"><li>● ユーザーの過失によるデータの削除</li><li>● ソフトウェアバグによるデータの変更</li></ul>	<ul style="list-style-type: none"><li>● <a href="#">仮想マシンスナップショットによるデータ損失からの復旧</a></li><li>● <a href="#">IdM バックアップを使用したデータ損失からの復旧</a></li><li>● <a href="#">データ損失の管理</a></li></ul>

## 第2章 レプリケーションを使用した1台のサーバーの復旧

1台のサーバーで深刻な中断や損失が発生した場合は、複数のレプリカを使用すると、レプリカを置き換えて、以前の冗長性レベルを迅速に復元できます。

IdM トポロジーに統合認証局 (CA) が含まれている場合は、CA 更新サーバーおよびその他のレプリカで、破損したレプリカを削除して置き換える手順が異なります。

### 2.1. CA 更新サーバーの損失からの復旧

認証局 (CA) 更新サーバーが失われた場合は、CA 更新サーバーロールを満たすために別の CA レプリカをプロモートしてから、代替 CA レプリカをデプロイする必要があります。

#### 前提条件

- デプロイメントで、IdM の内部認証局 (CA) を使用している。
- 環境内の別のレプリカには CA サービスがインストールされている。



#### 警告

IdM デプロイメントは、以下の場合に修復できません。

1. CA 更新サーバーが失われた場合
2. CA がインストールされている他のサーバーがない場合
3. CA ロールを持つレプリカのバックアップはありません。  
証明書データが保護されるように、CA ロールでレプリカからのバックアップを作成することが重要です。バックアップの作成および復元に関する詳細は、[IdM バックアップでデータ損失に備える](#) を参照してください。

#### 手順

1. 環境内の別のレプリカから、環境内で別の CA レプリカをプロモートして、新しい CA 更新サーバーとして機能します。[IdM CA 更新サーバーの変更およびリセット](#) を参照してください。
2. 環境内の別のレプリカから、失われた CA 更新サーバーへのレプリカ合意を削除します。[CLI を使用したトポロジーからのサーバーの削除](#) を参照してください。
3. 新しい CA レプリカをインストールして、失われた CA レプリカを置き換えます。[CA を使用して IdM レプリカのインストール](#) を参照してください。
4. DNS を更新して、レプリカトポロジーの変更を反映させます。IdM DNS を使用すると、DNS サービスレコードが自動的に更新されます。
5. IdM クライアントが IdM サーバーに到達できることを確認します。[復旧時の IdM クライアントの調整](#) を参照してください。

## 検証手順

1. IdM ユーザーとして Kerberos TGT (Ticket-Granting-Ticket) を正常に取得して、新しいレプリカで Kerberos サーバーをテストします。

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

[root@server ~]# klist
Ticket cache: KCM:0
Default principal: admin@EXAMPLE.COM

Valid starting    Expires          Service principal
10/31/2019 15:51:37  11/01/2019 15:51:02  HTTP/server.example.com@EXAMPLE.COM
10/31/2019 15:51:08  11/01/2019 15:51:02  krbtgt/EXAMPLE.COM@EXAMPLE.COM
```

2. ユーザー情報を取得して、Directory Server および SSSD 設定をテストします。

```
[root@server ~]# ipa user-show admin
User login: admin
Last name: Administrator
Home directory: /home/admin
Login shell: /bin/bash
Principal alias: admin@EXAMPLE.COM
UID: 1965200000
GID: 1965200000
Account disabled: False
Password: True
Member of groups: admins, trust admins
Kerberos keys available: True
```

3. **ipa cert-show** コマンドを使用して CA 設定をテストします。

```
[root@server ~]# ipa cert-show 1
Issuing CA: ipa
Certificate: MII EjgCC AuqgAwIB AgIjoSIP...
Subject: CN=Certificate Authority,O=EXAMPLE.COM
Issuer: CN=Certificate Authority,O=EXAMPLE.COM
Not Before: Thu Oct 31 19:43:29 2019 UTC
Not After: Mon Oct 31 19:43:29 2039 UTC
Serial number: 1
Serial number (hex): 0x1
Revoked: False
```

## 関連情報

- [IdM CA 更新サーバーの使用](#)

## 2.2. 通常のレプリカ損失からの回復

認証局 (CA) 更新サーバーではないレプリカを置き換えるには、トポロジーから失われたレプリカを削除し、その場所に新しいレプリカをインストールします。

## 前提条件

- CA 更新サーバーが適切に動作している。CA 更新サーバーが失われた場合は、[CA 更新サーバーの損失からの復旧](#) を参照してください。

## 手順

1. 失われたサーバーにレプリカ合意を削除します。[IdM サーバーのアンインストール](#) を参照してください。
2. 必要なサービス (CA、KRA、DNS) で新規レプリカをデプロイします。[IdM レプリカのインストール](#) を参照してください。
3. DNS を更新して、レプリカトポロジーの変更を反映させます。IdM DNS を使用すると、DNS サービスレコードが自動的に更新されます。
4. IdM クライアントが IdM サーバーに到達できることを確認します。[復旧時の IdM クライアントの調整](#) を参照してください。

## 検証手順

1. IdM ユーザーとして Kerberos TGT (Ticket-Granting-Ticket) を正常に取得して、新しいレプリカで Kerberos サーバーをテストします。

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

```
[root@newreplica ~]# klist
Ticket cache: KCM:0
Default principal: admin@EXAMPLE.COM
```

```
Valid starting    Expires          Service principal
10/31/2019 15:51:37 11/01/2019 15:51:02 HTTP/server.example.com@EXAMPLE.COM
10/31/2019 15:51:08 11/01/2019 15:51:02 krbtgt/EXAMPLE.COM@EXAMPLE.COM
```

2. ユーザー情報を取得して、新しいレプリカで Directory Server および SSSD 設定をテストします。

```
[root@newreplica ~]# ipa user-show admin
User login: admin
Last name: Administrator
Home directory: /home/admin
Login shell: /bin/bash
Principal alias: admin@EXAMPLE.COM
UID: 1965200000
GID: 1965200000
Account disabled: False
Password: True
Member of groups: admins, trust admins
Kerberos keys available: True
```

## 第3章 レプリケーションを使用した複数のサーバーの復旧

複数のサーバーが同時に失われた場合は、以下のいずれかのシナリオに該当することで、環境を再構築できるかどうかを判断します。

### 3.1. CA なしのデプロイメントで複数のサーバーが失われた状態からの回復

CA なしのデプロイメントでは、サーバーはすべて同等であるとみなされ、失われたレプリカを削除し、置き換えて環境を再構築できます。

#### 前提条件

- デプロイメントで、外部認証局 (CA) を使用している。

#### 手順

- [通常のレプリカ損失からの復旧](#) を参照してください。

### 3.2. CA 更新サーバーが無効化された場合の複数のサーバー損失からの復旧

CA 更新サーバーに問題がない場合は、他のサーバーを任意の順序で交換できます。

#### 前提条件

- デプロイメントで、IdM の内部認証局 (CA) を使用している。

#### 手順

- [通常のレプリカ損失からの復旧](#) を参照してください。

### 3.3. CA 更新サーバーおよびその他のサーバーの損失からの復旧

CA 更新サーバーおよび他のサーバーを失った場合は、他のレプリカを交換する前に、別の CA サーバーを CA 更新サーバーのロールに昇格させてください。

#### 前提条件

- デプロイメントで、IdM の内部認証局 (CA) を使用している。
- 少なくとも CA レプリカが無効化されている。

#### 手順

1. 別の CA レプリカをプロモートして CA 更新サーバーロールに対応します。[CA 更新サーバーの損失からの復旧](#) を参照してください。
2. 失われたその他のレプリカをすべて置き換えます。[通常のレプリカ損失からの復旧](#) を参照してください。

### 3.4. すべての CA レプリカの失われた状態からの復旧

認証局 (CA) レプリカがないと、IdM 環境は、追加のレプリカをデプロイし、そのレプリカを再ビルドする機能がありません。

#### 前提条件

- デプロイメントで、IdM の内部認証局 (CA) を使用している。

#### 手順

- この状況は、完全に失われています。

#### 関連情報

- 完全なインフラストラクチャー損失を準備するには、[仮想マシンスナップショットによるデータ損失の準備](#) を参照してください。

### 3.5. インフラストラクチャー全体の損失からの復旧

すべてのサーバーが一度に失われ、復元する仮想マシンスナップショットやデータバックアップがない場合、この状況は復旧できません。

#### 手順

- この状況は、完全に失われています。

#### 関連情報

- [仮想マシンのスナップショットによるデータ損失の準備](#)

## 第4章 仮想マシンスナップショットによるデータ損失からの復旧

データ損失イベントが発生した場合は、認証局 (CA) のレプリカの仮想マシン (VM) スナップショットを復元して、失われたデータを修復するか、そこから新しい環境をデプロイできます。

### 4.1. 仮想マシンのスナップショットのみからの復旧

災害がすべての IdM サーバーに影響し、IdM CA レプリカ仮想マシンのスナップショットのみが残っている場合は、失われたサーバーへの参照をすべて削除し、新しいレプリカをインストールすることで、デプロイメントを再作成できます。

#### 前提条件

- CA レプリカ仮想マシンのスナップショットを作成している。[仮想マシンのスナップショットによるデータ損失の準備](#) を参照してください。

#### 手順

1. CA レプリカ仮想マシンで使用するスナップショットを起動します。
2. 失われたレプリカのレプリカ合意を削除します。

```
[root@server ~]# ipa server-del lost-server1.example.com
[root@server ~]# ipa server-del lost-server2.example.com
...
```

3. 次の CA レプリカをインストールします。[IdM レプリカのインストール](#) を参照してください。
4. VM CA レプリカが CA 更新サーバーになりました。Red Hat は、環境内の別の CA レプリカをプロモートして、CA 更新サーバーとして機能させることを推奨します。[IdM CA 更新サーバーの変更およびリセット](#) を参照してください。
5. 必要なサービス (CA、DNS) で追加のレプリカをデプロイし、必要なレプリカトポロジを再作成します。[IdM レプリカのインストール](#) を参照してください。
6. DNS を更新して、新しいレプリカトポロジを反映させます。IdM DNS を使用すると、DNS サービスレコードが自動的に更新されます。
7. IdM クライアントが IdM サーバーにアクセスできることを確認します。[復旧時の IdM クライアントの調整](#) を参照してください。

#### 検証手順

1. Kerberos TGT (Ticket-Granting-Ticket) を IdM ユーザーとして正常に取得して、すべてのレプリカで Kerberos サーバーをテストします。

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

[root@server ~]# klist
Ticket cache: KCM:0
Default principal: admin@EXAMPLE.COM
```

```
Valid starting    Expires          Service principal
10/31/2019 15:51:37 11/01/2019 15:51:02 HTTP/server.example.com@EXAMPLE.COM
10/31/2019 15:51:08 11/01/2019 15:51:02 krbtgt/EXAMPLE.COM@EXAMPLE.COM
```

2. ユーザー情報を取得して、すべてのレプリカで Directory Server および SSSD 設定をテストします。

```
[root@server ~]# ipa user-show admin
User login: admin
Last name: Administrator
Home directory: /home/admin
Login shell: /bin/bash
Principal alias: admin@EXAMPLE.COM
UID: 1965200000
GID: 1965200000
Account disabled: False
Password: True
Member of groups: admins, trust admins
Kerberos keys available: True
```

3. **ipa cert-show** コマンドを使用して、すべての CA レプリカで CA サーバーをテストします。

```
[root@server ~]# ipa cert-show 1
Issuing CA: ipa
Certificate: MIIEgjCCAuqgAwIBAgIjoSIP...
Subject: CN=Certificate Authority,O=EXAMPLE.COM
Issuer: CN=Certificate Authority,O=EXAMPLE.COM
Not Before: Thu Oct 31 19:43:29 2019 UTC
Not After: Mon Oct 31 19:43:29 2039 UTC
Serial number: 1
Serial number (hex): 0x1
Revoked: False
```

## 関連情報

- [レプリカトポロジーの計画](#)

## 4.2. 部分的に機能する環境間の仮想マシンのスナップショットからの復旧

障害が複数の IdM サーバーに影響を及ぼし、その他のサーバーが適切に動作している場合は、デプロイメントを仮想マシンスナップショットでキャプチャーされた状態に復元できます。たとえば、他のレプリカが稼働の状態でもすべての認証局 (CA) レプリカが失われると、CA レプリカを環境に戻す必要があります。

このシナリオでは、失われたレプリカへの参照を削除し、スナップショットから CA レプリカを復元し、レプリケーションを確認し、新規レプリカをデプロイします。

### 前提条件

- CA レプリカ仮想マシンのスナップショットを作成している。[仮想マシンのスナップショットによるデータ損失の準備](#) を参照してください。

### 手順



1. すべてのレプリカ合意を失われたサーバーから削除します。 [IdM サーバーのアンインストール](#) を参照してください。
2. CA レプリカ仮想マシンで使用するスナップショットを起動します。
3. 復元したサーバーと失われたサーバー間のレプリカ合意を削除します。

```
[root@restored-CA-replica ~]# ipa server-del lost-server1.example.com
[root@restored-CA-replica ~]# ipa server-del lost-server2.example.com
...
```

4. 復元されたサーバーに、実稼働のサーバーとのレプリカ合意がない場合は、復元されたサーバーをその他のサーバーのいずれかに接続して、復元されたサーバーを更新します。

```
[root@restored-CA-replica ~]# ipa topologysegment-add
Suffix name: domain
Left node: restored-CA-replica.example.com
Right node: server3.example.com
Segment name [restored-CA-replica.com-to-server3.example.com]: new_segment
-----
Added segment "new_segment"
-----
Segment name: new_segment
Left node: restored-CA-replica.example.com
Right node: server3.example.com
Connectivity: both
```

5. `/var/log/dirsrv/slapd-YOUR-INSTANCE/errors` で Directory Server のエラーログを確認し、スナップショットの CA レプリカが残りの IdM サーバーと正しく同期しているかどうかを確認します。
6. データベースが古くて復元されたサーバーのレプリケーションが失敗すると、復元されたサーバーを再初期化します。

```
[root@restored-CA-replica ~]# ipa-replica-manage re-initialize --from
server2.example.com
```

7. 復元されたサーバーのデータベースが正しく同期されている場合は、 [IdM レプリカのインストール](#) に従って、必要なサービス (CA、DNS) で追加のレプリカをデプロイし、続行します。

## 検証手順

1. Kerberos TGT (Ticket-Granting-Ticket) を IdM ユーザーとして正常に取得して、すべてのレプリカで Kerberos サーバーをテストします。

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

[root@server ~]# klist
Ticket cache: KCM:0
Default principal: admin@EXAMPLE.COM

Valid starting    Expires          Service principal
10/31/2019 15:51:37 11/01/2019 15:51:02 HTTP/server.example.com@EXAMPLE.COM
10/31/2019 15:51:08 11/01/2019 15:51:02 krbtgt/EXAMPLE.COM@EXAMPLE.COM
```

- 
2. ユーザー情報を取得して、すべてのレプリカで Directory Server および SSSD 設定をテストします。

```
[root@server ~]# ipa user-show admin
User login: admin
Last name: Administrator
Home directory: /home/admin
Login shell: /bin/bash
Principal alias: admin@EXAMPLE.COM
UID: 1965200000
GID: 1965200000
Account disabled: False
Password: True
Member of groups: admins, trust admins
Kerberos keys available: True
```

- 
- 
3. **ipa cert-show** コマンドを使用して、すべての CA レプリカで CA サーバーをテストします。

```
[root@server ~]# ipa cert-show 1
Issuing CA: ipa
Certificate: MII EjjCC AuqgAwIB AgIjoSIP...
Subject: CN=Certificate Authority,O=EXAMPLE.COM
Issuer: CN=Certificate Authority,O=EXAMPLE.COM
Not Before: Thu Oct 31 19:43:29 2019 UTC
Not After: Mon Oct 31 19:43:29 2039 UTC
Serial number: 1
Serial number (hex): 0x1
Revoked: False
```

## 関連情報

- [仮想マシンのスナップショットからの復元による新規 IdM 環境の確立](#)

## 4.3. 仮想マシンのスナップショットからの復元による新規 IDM 環境の確立

復元した仮想マシンスナップショットの認証局 (CA) レプリカが他のサーバーと複製できない場合は、仮想マシンスナップショットから新しい IdM 環境を作成します。

新しい IdM 環境を確立するには、仮想マシンサーバーを分離し、そこから追加のレプリカを作成し、IdM クライアントを新しい環境に切り替えます。

## 前提条件

- CA レプリカ仮想マシンのスナップショットを作成している。[仮想マシンのスナップショットによるデータ損失の準備](#) を参照してください。

## 手順

1. CA レプリカ仮想マシンで使用するスナップショットを起動します。
2. 現在のデプロイメントの他の部分から復元されたサーバーを分離します。複製トポロジーセグメントがすべて削除されます。

- a. まず、すべての **ドメイン** レプリケーショントポロジーセグメントを表示します。

```
[root@restored-CA-replica ~]# ipa topologysegment-find
Suffix name: domain
-----
8 segments matched
-----
Segment name: new_segment
Left node: restored-CA-replica.example.com
Right node: server2.example.com
Connectivity: both
...
-----
Number of entries returned 8
-----
```

- b. 次に、復元されたサーバーに関連するすべての **ドメイン** トポロジーセグメントを削除します。

```
[root@restored-CA-replica ~]# ipa topologysegment-del
Suffix name: domain
Segment name: new_segment
-----
Deleted segment "new_segment"
-----
```

- c. 最後に、**ca** トポロジーセグメントを使用して同じアクションを実行します。

```
[root@restored-CA-replica ~]# ipa topologysegment-find
Suffix name: ca
-----
1 segments matched
-----
Segment name: ca_segment
Left node: restored-CA-replica.example.com
Right node: server4.example.com
Connectivity: both
-----
Number of entries returned 1
-----

[root@restored-CA-replica ~]# ipa topologysegment-del
Suffix name: ca
Segment name: ca_segment
-----
Deleted segment "ca_segment"
-----
```

3. デプロイメントの負荷を処理するために、復元されたサーバーから十分な数の IdM レプリカをインストールします。これで、接続されていない 2 つの IdM デプロイメントが並行して実行されるようになりました。

4. 新しい IdM レプリカへの参照をハードコーディングして、IdM クライアントが新しいデプロイメントを使用するようにします。 [復旧時の IdM クライアントの調整](#) を参照してください。
5. 以前のデプロイメントから IdM サーバーを停止し、アンインストールします。 [IdM サーバーのアンインストール](#) を参照してください。

## 検証手順

1. IdM ユーザーとして Kerberos TGT (Ticket-Granting-Ticket) を正常に取得して、すべての新しいレプリカで Kerberos サーバーをテストします。

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

[root@server ~]# klist
Ticket cache: KCM:0
Default principal: admin@EXAMPLE.COM

Valid starting    Expires          Service principal
10/31/2019 15:51:37  11/01/2019 15:51:02  HTTP/server.example.com@EXAMPLE.COM
10/31/2019 15:51:08  11/01/2019 15:51:02  krbtgt/EXAMPLE.COM@EXAMPLE.COM
```

2. ユーザー情報を取得して、新しいレプリカごとに Directory Server および SSSD の設定をテストします。

```
[root@server ~]# ipa user-show admin
User login: admin
Last name: Administrator
Home directory: /home/admin
Login shell: /bin/bash
Principal alias: admin@EXAMPLE.COM
UID: 1965200000
GID: 1965200000
Account disabled: False
Password: True
Member of groups: admins, trust admins
Kerberos keys available: True
```

3. **ipa cert-show** コマンドを使用して、新しい CA レプリカごとに CA サーバーをテストします。

```
[root@server ~]# ipa cert-show 1
Issuing CA: ipa
Certificate: MIIEGjCCAuggAwIBAgIjoSIP...
Subject: CN=Certificate Authority,O=EXAMPLE.COM
Issuer: CN=Certificate Authority,O=EXAMPLE.COM
Not Before: Thu Oct 31 19:43:29 2019 UTC
Not After: Mon Oct 31 19:43:29 2039 UTC
Serial number: 1
Serial number (hex): 0x1
Revoked: False
```

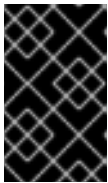
## 第5章 IDM バックアップを使用したデータ損失からの復旧

`ipa-restore` ユーティリティーを使用して、IdM サーバーを IdM バックアップでキャプチャーした以前の状態に復元できます。

### 5.1. IDM バックアップから復元するタイミング

IdM バックアップから復元すると、いくつかの障害シナリオに対応できます。

- **LDAP コンテンツに望ましくない変更が加えられた** - エントリーは変更または削除され、デプロイメント全体でそれらの変更が行われ、これらの変更を元に戻すようにします。データのみをバックアップを復元すると、IdM 設定自体に影響を与えずに LDAP エントリーが以前の状態に戻ります。
- **インフラストラクチャーの損失の合計、またはすべての CA インスタンスの損失** - 障害によりすべての認証局レプリカが損傷した場合、デプロイメントは追加のサーバーをデプロイすることで、それ自体を再構築する機能を失うようになりました。この場合は、CA レプリカのバックアップを復元し、そこから新しいレプリカを構築します。
- **分離されたサーバーのアップグレードに失敗** - オペレーティングシステムは機能し続けますが、IdM データが破損するため、IdM システムを既知の正常な状態に復元したい理由になります。Red Hat では、問題を診断してトラブルシューティングするために、テクニカルサポートの利用を推奨しています。以上の作業にすべて失敗した場合は、サーバーのフルバックアップから復元します。



#### 重要

ハードウェアまたはアップグレードの失敗で推奨されるソリューションは、失われたサーバーをレプリカから再構築することです。詳細は、[レプリケーションを使用した1台のサーバーの復旧](#) を参照してください。

### 5.2. IDM バックアップから復元する際の注意点

`ipa-backup` ユーティリティーでバックアップを作成した場合は、IdM サーバーまたは LDAP コンテンツをバックアップ実行時の状態に復元できます。

以下は、IdM バックアップからの復元時の主要な考慮事項です。

- バックアップの作成元のサーバーの設定と一致するサーバー上でのみバックアップを復元できます。サーバーには以下の項目が **必要** です。
  - 同じホスト名
  - 同じ IP アドレス
  - 同じバージョンの IdM ソフトウェア
- 多数サーバーがある中で IdM サーバーを復元すると、復元されたサーバーは、IdM の唯一の情報ソースになります。他のサーバーはすべて、復元されたサーバーをもとに再度初期化する **必要があります**。
- 最後のバックアップ後に作成されたデータはすべて失われるため、通常のシステムメンテナンスには、バックアップと復元のソリューションを使用しないでください。
- サーバーが失われた場合は、バックアップから復元するのではなく、レプリカとしてサーバー

を再インストールしてサーバーを再構築することが推奨されます。新規レプリカを作成すると、現在の作業環境のデータが保存されます。詳細は、[サーバーでのレプリケーションによる損失の準備](#) を参照してください。

- バックアップ機能および復元機能はコマンドラインからのみ管理でき、IdM Web UI では使用できません。
- `/tmp` または `/var/tmp` ディレクトリーにあるバックアップファイルからは復元できません。IdM Directory Server は `PrivateTmp` ディレクトリーを使用しており、オペレーティングシステムで一般的に利用できる `/tmp` または `/var/tmp` ディレクトリーにはアクセスできません。

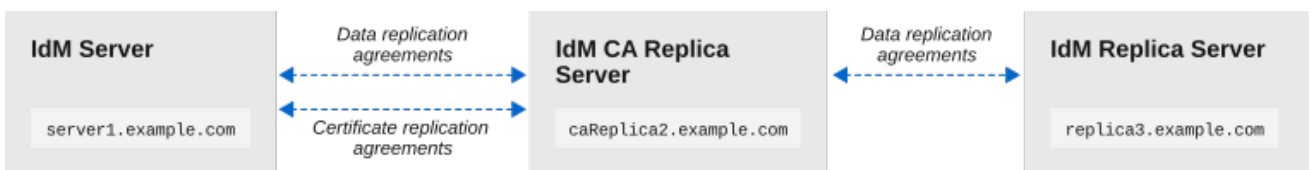
## ヒント

バックアップから復元するには、バックアップの実行時にインストールされたものと同じバージョンのソフトウェア (RPM) がターゲットホストに必要になります。このため、Red Hat は、バックアップではなく、仮想マシンのスナップショットからの復元を行うことを推奨します。詳細は [仮想マシンスナップショットによるデータ損失からの復旧](#) を参照してください。

## 5.3. バックアップからの IDM サーバーの復元

以下の手順では、IdM バックアップから IdM サーバーまたはその LDAP データを復元する方法を説明します。

図5.1 この例で使用されるレプリケーショントポロジ



157\_RHEL\_001

表5.1 この例で使用されるサーバーの命名規則

サーバーのホスト名	機能
<code>server1.example.com</code>	バックアップから復元する必要があるサーバー
<code>caReplica2.example.com</code>	<code>server1.example.com</code> ホストに接続した認証局 (CA) レプリカ。
<code>replica3.example.com</code>	<code>caReplica2.example.com</code> ホストに接続しているレプリカ。

## 前提条件

- `ipa-backup` ユーティリティを使用して IdM サーバー全体のバックアップまたはデータのみのバックアップを生成している。[IdM バックアップの作成](#) を参照してください。
- バックアップファイルが `/tmp` または `/var/tmp` ディレクトリーにない。
- 完全なサーバーバックアップからサーバーの完全な復元を実行する前に、サーバーから IdM を [アンインストール](#) し、以前と同じサーバー設定を使用して IdM を [再インストール](#) します。

## 手順

1. **ipa-restore** ユーティリティーを使用して、完全なサーバーまたはデータのためのバックアップを復元します。

- バックアップディレクトリーがデフォルトの `/var/lib/ipa/backup/` の場合は、ディレクトリーの名前のみを入力します。

```
[root@server1 ~]# ipa-restore ipa-full-2020-01-14-12-02-32
```

- バックアップディレクトリーがデフォルトの場所がない場合は、完全パスを入力します。

```
[root@server1 ~]# ipa-restore /mybackups/ipa-data-2020-02-01-05-30-00
```



## 注記

**ipa-restore** ユーティリティーは、ディレクトリーに含まれるバックアップのタイプを自動的に検出し、デフォルトで同じタイプの復元を実行します。完全なサーバーバックアップからデータのための復元を実行するには、**--data** オプションを **ipa-restore** コマンドに追加します。

```
[root@server1 ~]# ipa-restore --data ipa-full-2020-01-14-12-02-32
```

2. Directory Manager パスワードを入力します。

```
Directory Manager (existing master) password:
```

3. **Yes** を入力して、現在のデータをバックアップで上書きしていることを確認します。

```
Preparing restore from /var/lib/ipa/backup/ipa-full-2020-01-14-12-02-32 on
server1.example.com
Performing FULL restore from FULL backup
Temporary setting umask to 022
Restoring data will overwrite existing live data. Continue to restore? [no]: yes
```

4. **ipa-restore** ユーティリティーは、利用可能なすべてのサーバーでレプリケーションを無効にします。

```
Each master will individually need to be re-initialized or
re-created from this one. The replication agreements on
masters running IPA 3.1 or earlier will need to be manually
re-enabled. See the man page for details.
Disabling all replication.
Disabling replication agreement on server1.example.com to caReplica2.example.com
Disabling CA replication agreement on server1.example.com to caReplica2.example.com
Disabling replication agreement on caReplica2.example.com to server1.example.com
Disabling replication agreement on caReplica2.example.com to replica3.example.com
Disabling CA replication agreement on caReplica2.example.com to server1.example.com
Disabling replication agreement on replica3.example.com to caReplica2.example.com
```

その後、このユーティリティーは IdM サービスを停止し、バックアップを復元し、サービスを再起動します。

```

Stopping IPA services
Systemwide CA database updated.
Restoring files
Systemwide CA database updated.
Restoring from userRoot in EXAMPLE-COM
Restoring from ipaca in EXAMPLE-COM
Restarting GSS-proxy
Starting IPA services
Restarting SSSD
Restarting oddjobd
Restoring umask to 18
The ipa-restore command was successful

```

5. 復元されたサーバーに接続したすべてのレプリカを再初期化します。
  - a. **domain** 接尾辞のレプリカトポロジーセグメントのリストを表示します。復元されたサーバーに関連するトポロジーセグメントを書き留めます。

```

[root@server1 ~]# ipa topologysegment-find domain
-----
2 segments matched
-----
Segment name: server1.example.com-to-caReplica2.example.com
Left node: server1.example.com
Right node: caReplica2.example.com
Connectivity: both

Segment name: caReplica2.example.com-to-replica3.example.com
Left node: caReplica2.example.com
Right node: replica3.example.com
Connectivity: both
-----
Number of entries returned 2
-----

```

- b. 復元されたサーバーとともにすべてのトポロジーセグメントの **domain** 接尾辞を再初期化します。  
この例では、**server1** からのデータで **caReplica2** の再初期化を実行します。

```

[root@caReplica2 ~]# ipa-replica-manage re-initialize --from=server1.example.com
Update in progress, 2 seconds elapsed
Update succeeded

```

- c. 認証局データに移動し、**ca** 接尾辞のレプリケーショントポロジーセグメントのリストを表示します。

```

[root@server1 ~]# ipa topologysegment-find ca
-----
1 segment matched
-----
Segment name: server1.example.com-to-caReplica2.example.com
Left node: server1.example.com
Right node: caReplica2.example.com
Connectivity: both

```



```
-----
Number of entries returned 1
-----
```

- d. 復元されたサーバーに接続されているすべての CA レプリカを再初期化します。  
この例では、**server1** からのデータを使用して **caReplica2** の **csreplica** を再初期化します。

```
[root@caReplica2 ~]# ipa-csreplica-manage re-initialize --
from=server1.example.com
Directory Manager password:

Update in progress, 3 seconds elapsed
Update succeeded
```

6. 復元されたサーバー **server1.example.com** のデータですべてのサーバーが更新されるまで、レプリケーショントポロジを介して、後続のレプリカを再初期化します。  
この例では、**caReplica2** からのデータで、**replica3** の **domain** 接尾辞を再初期化することのみが必要になります。

```
[root@replica3 ~]# ipa-replica-manage re-initialize --from=caReplica2.example.com
Directory Manager password:

Update in progress, 3 seconds elapsed
Update succeeded
```

7. すべてのサーバーで SSSD のキャッシュをクリアし、無効なデータによる認証の問題を回避します。
  - a. SSSD サービスを停止します。

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

- b. SSSD からキャッシュされたコンテンツをすべて削除します。

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

- c. SSSD サービスを起動します。

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

- d. サーバーを再起動します。

## 関連情報

- **ipa-restore**(1) の man ページでは、復元中の複雑なレプリケーションシナリオの処理方法が詳細に説明されています。

## 5.4. 暗号化されたバックアップからの復元

この手順では、暗号化された IdM バックアップから IdM サーバーを復元します。**ipa-restore** ユーティリティーは、IdM バックアップが暗号化されているかどうかを自動的に検出し、GPG2 root キーリングを使用して復元します。

## 前提条件

- GPG 暗号化 IdM バックアップ。 [GPG2 で暗号化した IdM バックアップの作成](#) を参照してください。
- LDAP Directory Manager のパスワード
- GPG キーの作成時に使用されるパスフレーズ

## 手順

1. GPG2 キーの作成時にカスタムキーリングの場所を使用した場合は、**\$GNUPGHOME** 環境変数とそのディレクトリーに設定されていることを確認します。 [GPG2 キーの作成](#) を参照してください。

```
[root@server ~]# echo $GNUPGHOME  
/root/backup
```

2. **ipa-restore** ユーティリティーにバックアップディレクトリーの場所を指定します。

```
[root@server ~]# ipa-restore ipa-full-2020-01-13-18-30-54
```

- a. Directory Manager パスワードを入力します。

```
Directory Manager (existing master) password:
```

- b. GPG キーの作成時に使用したパスフレーズを入力します。

```
-----  
Please enter the passphrase to unlock the OpenPGP secret key: |  
"GPG User (first key) <root@example.com>" |  
2048-bit RSA key, ID BF28FFA302EF4557, |  
created 2020-01-13. |  
|  
Passphrase: <passphrase> |  
|  
<OK> <Cancel> |  
-----
```

3. 復元されたサーバーに接続されているすべてのレプリカを再初期化します。 [バックアップからの IdM サーバーの復元](#) を参照してください。

## 第6章 ANSIBLE PLAYBOOK を使用した IDM サーバーの復元

**ipabackup** Ansible ロールを使用すると、バックアップから IdM サーバーを復元し、サーバーと Ansible コントローラー間でバックアップファイルを転送できます。

本セクションでは、以下のトピックについて説明します。

- IdM 管理用の Ansible コントロールノードの準備
- Ansible を使用したサーバーに保存されているバックアップからの IdM サーバーの復元
- Ansible を使用した Ansible コントローラーに保存されているバックアップから IdM サーバーの復元
- Ansible を使用した IdM サーバーのバックアップの Ansible コントローラーへのコピー
- Ansible を使用した IdM サーバーのバックアップの Ansible コントローラーから IdM サーバーへのコピー
- Ansible を使用した IdM サーバーからのバックアップの削除

### 6.1. IDM 管理用の ANSIBLE コントロールノードの準備

Identity Management (IdM) を管理するシステム管理者は、Red Hat Ansible Engine を使用する際に以下を行うことが推奨されます。

- ホームディレクトリーに Ansible Playbook 専用のサブディレクトリー (例: `~/MyPlaybooks`) を作成します。
- `/usr/share/doc/ansible-freeipa/*` と `/usr/share/doc/rhel-system-roles/*` ディレクトリーおよびサブディレクトリーから `~/MyPlaybooks` ディレクトリーにサンプル Ansible Playbook をコピーして調整します。
- `~/MyPlaybooks` ディレクトリーにインベントリーファイルを追加します。

この方法に従うことで、すべての Playbook を 1 か所で見つけることができます。また、root 権限を呼び出さなくても Playbook を実行できます。



#### 注記

**ipaserver**、**ipareplica**、**ipacient**、**ipabackup**、**ipasmartcard\_server**、および **ipasmartcard\_client ansible-freeipa** のロールを実行するために必要なのは、管理対象ノードでの **root** 権限のみです。これらのロールには、ディレクトリーおよび **dnf** ソフトウェアパッケージマネージャーへの特権アクセスが必要です。

`~/MyPlaybooks` ディレクトリーを作成し、それを使用して Ansible Playbook を保存および実行できるように設定するには、次の手順に従います。

#### 前提条件

- 管理対象ノードに IdM サーバー (`server.idm.example.com` および `replica.idm.example.com`) をインストールしている。
- DNS およびネットワークを設定し、コントロールノードから直接管理対象ノード (`server.idm.example.com` および `replica.idm.example.com`) にログインすることができる。

- IdM **admin** のパスワードを把握している。

## 手順

1. Ansible 設定および Playbook のディレクトリーをホームディレクトリーに作成します。

```
$ mkdir ~/MyPlaybooks/
```

2. ~/MyPlaybooks/ ディレクトリーに移動します。

```
$ cd ~/MyPlaybooks
```

3. ~/MyPlaybooks/ansible.cfg ファイルを以下の内容で作成します。

```
[defaults]
inventory = /home/your_username/MyPlaybooks/inventory

[privilege_escalation]
become=True
```

4. ~/MyPlaybooks/inventory ファイルを以下の内容で作成します。

```
[ipaserver]
server.idm.example.com

[ipareplicas]
replica1.idm.example.com
replica2.idm.example.com

[ipacluster:children]
ipaserver
ipareplicas

[ipacluster:vars]
ipaadmin_password=SomeADMINpassword

[ipaclients]
ipaclient1.example.com
ipaclient2.example.com

[ipaclients:vars]
ipaadmin_password=SomeADMINpassword
```

この設定は、これらの場所にあるホストの2つのホストグループ (**eu** と **us**) を定義します。さらに、この設定は、**eu** および **us** グループのすべてのホストを含む **ipaserver** ホストグループを定義します。

5. [オプション] SSH 公開鍵および秘密鍵を作成します。テスト環境でのアクセスを簡素化するには、秘密鍵にパスワードを設定しないでください。

```
$ ssh-keygen
```

6. 各マネージドノードの IdM **admin** アカウントに SSH 公開鍵をコピーします。

```
$ ssh-copy-id admin@server.idm.example.com
$ ssh-copy-id admin@replica.idm.example.com
```

これらのコマンドを入力する場合は、IdM **admin** パスワードを入力する必要があります。

## 関連情報

- [Ansible Playbook で Identity Management サーバーのインストール](#)
- [インベントリーの構築方法](#).

## 6.2. ANSIBLE を使用したサーバーに保存されているバックアップからの IDM サーバーの復元

以下の手順では、Ansible Playbook を使用して、そのホストに保存されているバックアップから IdM サーバーを復元する方法を説明します。

### 前提条件

- 次の要件を満たすように Ansible コントロールノードを設定している。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに [ansible-freeipa](#) パッケージがインストールされている。
  - `~/MyPlaybooks/` ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して [Ansible インベントリーファイル](#) を作成している (この例の場合)。
  - この例では、`secret.yml` Ansible ボールトに `ipadmin_password` が保存されていることを前提としている。
- `ansible-freeipa` モジュールが実行されるノードであるターゲットノードは、IdM クライアント、サーバー、またはレプリカとしての IdM ドメインの一部です。
- LDAP Directory Manager のパスワードを知っている必要があります。

### 手順

1. `~/MyPlaybooks/` ディレクトリーに移動します。

```
$ cd ~/MyPlaybooks/
```

2. `/usr/share/doc/ansible-freeipa/playbooks` ディレクトリーにある `restore-server.yml` ファイルのコピーを作成します。

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/restore-server.yml restore-my-server.yml
```

3. Ansible Playbook の `restore-my-server.yml` を開いて編集します。
4. 以下の変数を設定してファイルを調整します。
  - a. `hosts` 変数を、インベントリーファイルのホストグループに設定します。この例では、`ipaserver` ホストグループに設定します。

- b. **ipabackup\_name** 変数は、復元する **ipabackup** の名前に設定します。
- c. **ipabackup\_password** 変数は LDAP Directory Manager パスワードに設定します。

```
---
- name: Playbook to restore an IPA server
  hosts: ipaserver
  become: true

  vars:
    ipabackup_name: ipa-full-2021-04-30-13-12-00
    ipabackup_password: <your_LDAP_DM_password>

  roles:
    - role: ipabackup
      state: restored
```

5. ファイルを保存します。
6. Playbook ファイルとインベントリーファイルを指定して Ansible Playbook を実行します。

```
$ ansible-playbook --vault-password-file=password_file -v -i ~/MyPlaybooks/inventory
restore-my-server.yml
```

#### 関連情報

- `/usr/share/doc/ansible-freeipa/roles/ipabackup` ディレクトリーの **README.md** ファイル
- `/usr/share/doc/ansible-freeipa/playbooks/` ディレクトリー

### 6.3. ANSIBLE を使用した ANSIBLE コントローラーに保存されているバックアップから IDM サーバーの復元

以下の手順では、Ansible Playbook を使用して、Ansible コントローラーに保存されているバックアップから IdM サーバーを復元する方法を説明します。

#### 前提条件

- 次の要件を満たすように Ansible コントロールノードを設定している。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - `~/MyPlaybooks/` ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して **Ansible インベントリーファイル** を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible ポールトに **ipadmin\_password** が保存されていることを前提としている。
- **ansible-freeipa** モジュールが実行されるノードであるターゲットノードは、IdM クライアント、サーバー、またはレプリカとしての IdM ドメインの一部です。
- LDAP Directory Manager のパスワードを知っている必要があります。

## 手順

1. `~/MyPlaybooks/` ディレクトリーに移動します。

```
$ cd ~/MyPlaybooks/
```

2. `/usr/share/doc/ansible-freeipa/playbooks` ディレクトリーにある `restore-server-from-controller.yml` ファイルのコピーを作成します。

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/restore-server-from-controller.yml restore-my-server-from-my-controller.yml
```

3. `restore-my-server-from-my-controller.yml` ファイルを開いて編集します。
4. 以下の変数を設定してファイルを調整します。
  - a. `hosts` 変数を、インベントリーファイルのホストグループに設定します。この例では、`ipaserver` ホストグループに設定します。
  - b. `ipabackup_name` 変数は、復元する `ipabackup` の名前に設定します。
  - c. `ipabackup_password` 変数は LDAP Directory Manager パスワードに設定します。

```
---
- name: Playbook to restore IPA server from controller
  hosts: ipaserver
  become: true

  vars:
    ipabackup_name: server.idm.example.com_ipa-full-2021-04-30-13-12-00
    ipabackup_password: <your_LDAP_DM_password>
    ipabackup_from_controller: yes

  roles:
    - role: ipabackup
      state: restored
```

5. ファイルを保存します。
6. Playbook ファイルとインベントリーファイルを指定して Ansible Playbook を実行します。

```
$ ansible-playbook --vault-password-file=password_file -v -i ~/MyPlaybooks/inventory restore-my-server-from-my-controller.yml
```

## 関連情報

- `/usr/share/doc/ansible-freeipa/roles/ipabackup` ディレクトリーの `README.md` ファイル
- `/usr/share/doc/ansible-freeipa/playbooks/` ディレクトリー

## 6.4. ANSIBLE を使用した IDM サーバーのバックアップの ANSIBLE コントローラーへのコピー

以下の手順では、Ansible Playbook を使用して IdM サーバーのバックアップを Ansible サーバーから Ansible コントローラーにコピーする方法を説明します。

## 前提条件

- 次の要件を満たすように Ansible コントロールノードを設定している。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - `~/MyPlaybooks/` ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して **Ansible インベントリーファイル** を作成している (この例の場合)。
  - この例では、`secret.yml` Ansible ボールトに **ipadmin\_password** が保存されていることを前提としている。
- **ansible-freeipa** モジュールが実行されるノードであるターゲットノードは、IdM クライアント、サーバー、またはレプリカとしての IdM ドメインの一部です。

## 手順

1. バックアップを保存するために、Ansible コントローラーのホームディレクトリーにサブディレクトリーを作成します。

```
$ mkdir ~/ipabackups
```

2. `~/MyPlaybooks/` ディレクトリーに移動します。

```
$ cd ~/MyPlaybooks/
```

3. `/usr/share/doc/ansible-freeipa/playbooks` ディレクトリーにある **copy-backup-from-server.yml** ファイルのコピーを作成します。

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/copy-backup-from-server.yml copy-backup-from-my-server-to-my-controller.yml
```

4. **copy-my-backup-from-my-server-to-my-controller.yml** ファイルを開いて編集します。

5. 以下の変数を設定してファイルを調整します。

- a. **hosts** 変数を、インベントリーファイルのホストグループに設定します。この例では、**ipaserver** ホストグループに設定します。
- b. **ipabackup\_name** 変数を、Ansible コントローラーにコピーする IdM サーバー上の **ipabackup** の名前に設定します。
- c. デフォルトでは、バックアップは Ansible コントローラーの現在の作業ディレクトリーに保存されます。ステップ 1 で作成したディレクトリーを指定するには、**ipabackup\_controller\_path** 変数を追加し、それを `/home/user/ipabackups` ディレクトリーに設定します。

```
---  
- name: Playbook to copy backup from IPA server  
  hosts: ipaserver
```



```

become: true
vars:
  ipabackup_name: ipa-full-2021-04-30-13-12-00
  ipabackup_to_controller: yes
  ipabackup_controller_path: /home/user/ipabackups

roles:
- role: ipabackup
state: present

```

6. ファイルを保存します。

7. Playbook ファイルとインベントリーファイルを指定して Ansible Playbook を実行します。

```
$ ansible-playbook --vault-password-file=password_file -v -i ~/MyPlaybooks/inventory copy-backup-from-my-server-to-my-controller.yml
```

### 注記

すべての IdM バックアップをコントローラーにコピーするには、Ansible Playbook の **ipabackup\_name** 変数を **all** に設定します。

```

vars:
  ipabackup_name: all
  ipabackup_to_controller: yes

```

たとえば、**/usr/share/doc/ansible-freeipa/playbooks** ディレクトリーの Ansible Playbook **copy-all-backups-from-server.yml** を参照してください。

### 検証手順

- バックアップが Ansible コントローラーの **/home/user/ipabackups** ディレクトリーにあることを確認します。

```
[user@controller ~]$ ls /home/user/ipabackups
server.idm.example.com_ipa-full-2021-04-30-13-12-00
```

### 関連情報

- **/usr/share/doc/ansible-freeipa/roles/ipabackup** ディレクトリーの **README.md** ファイル
- **/usr/share/doc/ansible-freeipa/playbooks/** ディレクトリー

## 6.5. ANSIBLE を使用した IDM サーバーのバックアップの ANSIBLE コントローラーから IDM サーバーへのコピー

以下の手順では、Ansible Playbook を使用して IdM サーバーのバックアップを Ansible コントローラーから Ansible サーバーにコピーする方法を説明します。

### 前提条件

- 次の要件を満たすように Ansible コントロールノードを設定している。

- Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - `~/MyPlaybooks/` ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して **Ansible インベントリーファイル** を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible ボールトに **ipadmin\_password** が保存されていることを前提としている。
- **ansible-freeipa** モジュールが実行されるノードであるターゲットノードは、IdM クライアント、サーバー、またはレプリカとしての IdM ドメインの一部です。

## 手順

1. `~/MyPlaybooks/` ディレクトリーに移動します。

```
$ cd ~/MyPlaybooks/
```

2. `/usr/share/doc/ansible-freeipa/playbooks` ディレクトリーにある **copy-backup-from-controller.yml** のコピーを作成します。

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/copy-backup-from-controller.yml copy-backup-from-my-controller-to-my-server.yml
```

3. **copy-my-backup-from-my-controller-to-my-server.yml** ファイルを開いて編集します。
4. 以下の変数を設定してファイルを調整します。
  - a. **hosts** 変数を、インベントリーファイルのホストグループに設定します。この例では、**ipaserver** ホストグループに設定します。
  - b. **ipabackup\_name** 変数を、IdM サーバーにコピーする Ansible コントローラー上の **ipabackup** の名前に設定します。

```
---  
- name: Playbook to copy a backup from controller to the IPA server  
  hosts: ipaserver  
  become: true  
  
  vars:  
    ipabackup_name: server.idm.example.com_ipa-full-2021-04-30-13-12-00  
    ipabackup_from_controller: yes  
  
  roles:  
    - role: ipabackup  
      state: copied
```

5. ファイルを保存します。
6. Playbook ファイルとインベントリーファイルを指定して Ansible Playbook を実行します。

```
$ ansible-playbook --vault-password-file=password_file -v -i ~/MyPlaybooks/inventory copy-backup-from-my-controller-to-my-server.yml
```

## 関連情報

- `/usr/share/doc/ansible-freeipa/roles/ipabackup` ディレクトリーの **README.md** ファイル
- `/usr/share/doc/ansible-freeipa/playbooks/` ディレクトリー

## 6.6. ANSIBLE を使用した IDM サーバーからのバックアップの削除

以下の手順では、Ansible Playbook を使用して IdM サーバーからバックアップを削除する方法を説明します。

### 前提条件

- 次の要件を満たすように Ansible コントロールノードを設定している。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - `~/MyPlaybooks/` ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して **Ansible インベントリーファイル** を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible ボールトに **ipadmin\_password** が保存されていることを前提としている。
- **ansible-freeipa** モジュールが実行されるノードであるターゲットノードは、IdM クライアント、サーバー、またはレプリカとしての IdM ドメインの一部です。

### 手順

1. `~/MyPlaybooks/` ディレクトリーに移動します。

```
$ cd ~/MyPlaybooks/
```

2. `/usr/share/doc/ansible-freeipa/playbooks` ディレクトリーにある **remove-backup-from-server.yml** ファイルのコピーを作成します。

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/remove-backup-from-server.yml remove-backup-from-my-server.yml
```

3. **remove-backup-from-my-server.yml** ファイルを開いて編集します。
4. 以下の変数を設定してファイルを調整します。
  - a. **hosts** 変数を、インベントリーファイルのホストグループに設定します。この例では、**ipaserver** ホストグループに設定します。
  - b. **ipabackup\_name** 変数を、IdM サーバーから削除する **ipabackup** の名前に設定します。

```
---
- name: Playbook to remove backup from IPA server
  hosts: ipaserver
  become: true

  vars:
```

```
ipabackup_name: ipa-full-2021-04-30-13-12-00
```

```
roles:  
- role: ipabackup  
state: absent
```

5. ファイルを保存します。
6. Playbook ファイルとインベントリーファイルを指定して Ansible Playbook を実行します。

```
$ ansible-playbook --vault-password-file=password_file -v -i ~/MyPlaybooks/inventory  
remove-backup-from-my-server.yml
```

### 注記

IdM サーバーから **すべての** IdM バックアップを削除するには、Ansible Playbook の **ipabackup\_name** 変数を **all** に設定します。

```
vars:  
  ipabackup_name: all
```

たとえば、**/usr/share/doc/ansible-freeipa/playbooks** ディレクトリーの Ansible Playbook **remove-all-backups-from-server.yml** を参照してください。

### 関連情報

- **/usr/share/doc/ansible-freeipa/roles/ipabackup** ディレクトリーの **README.md** ファイル
- **/usr/share/doc/ansible-freeipa/playbooks/** ディレクトリー

## 第7章 データ損失の管理

データ損失イベントに対する適切な応答は、影響を受けるレプリカの数と失ったデータのタイプにより異なります。

### 7.1. 分離されたデータ損失への応答

データ損失の発生時に、影響を受けるサーバーをすぐに分離することで、データの損失の複製を最小限に抑えます。次に、環境の残りの部分から置き換えられたレプリカを作成します。

#### 前提条件

- 複数のレプリカを使用した強力な IdM レプリケーショントポロジー。 [レプリケーションによるサーバーの損失の準備](#) を参照してください。

#### 手順

- データ損失の複製を制限するには、他のトポロジーのレプリカトポロジーセグメントを削除して、影響を受けたレプリカをすべて切断します。
  - デプロイメント内のすべての **ドメイン** レプリケーショントポロジーセグメントを表示します。

```
[root@server ~]# ipa topologysegment-find
Suffix name: domain
-----
8 segments matched
-----
Segment name: segment1
Left node: server.example.com
Right node: server2.example.com
Connectivity: both

...

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

- 影響を受けるサーバーに関連するすべての **ドメイン** トポロジーセグメントを削除します。

```
[root@server ~]# ipa topologysegment-del
Suffix name: domain
Segment name: segment1
-----
Deleted segment "segment1"
-----
```

- 影響を受けるサーバーに関する **ca** トポロジーセグメントを使用して、同じアクションを実行します。

```
[root@server ~]# ipa topologysegment-find
Suffix name: ca
-----
```

```

1 segments matched
-----
Segment name: ca_segment
Left node: server.example.com
Right node: server2.example.com
Connectivity: both
-----
Number of entries returned 1
-----

[root@server ~]# ipa topologysegment-del
Suffix name: ca
Segment name: ca_segment
-----
Deleted segment "ca_segment"
-----

```

2. データ損失の影響を受けるサーバーは破棄されている必要があります。置換レプリカを作成するには、[レプリケーションを使用した複数のサーバーのリカバリー](#)を参照してください。

## 7.2. すべてのサーバー間の制限されたデータ損失への対応

データ損失イベントは、すべてのサーバー間で誤って削除を実行するなど、環境内のすべてのレプリカに影響する可能性があります。データの損失が認識され、制限されている場合は、手動でデータを再度追加します。

### 前提条件

- 失われたデータを含む IdM サーバーの仮想マシンスナップショットまたは IdM バックアップ。

### 手順

1. 失われたデータを確認する必要がある場合は、別のネットワーク上の分離されたサーバーに、仮想マシンのスナップショットまたはバックアップを復元します。
2. **ipa** コマンドまたは **ldapadd** コマンドを使用して、不足している情報をデータベースに追加します。

### 関連情報

- [仮想マシンスナップショットによるデータ損失からの復旧](#)
- [IdM のバックアップおよび復元](#)

## 7.3. すべてのサーバー間の未定義のデータ損失への応答

データの損失が深刻な場合または定義されていない場合は、サーバーの仮想マシンスナップショットから新しい環境をデプロイします。

### 前提条件

- 仮想マシンスナップショットには、失われたデータが含まれます。

### 手順

1. IdM 認証局 (CA) レプリカを仮想マシンのスナップショットから既知の正常な状態に復元し、そこから新しい IdM 環境をデプロイします。[仮想マシンのスナップショットのみからの復旧](#) を参照してください。
2. **ipa** コマンドまたは **Idapadd** コマンドを使用して、スナップショットの取得後に作成されたデータを追加します。

#### 関連情報

- [仮想マシンスナップショットによるデータ損失からの復旧](#)

## 第8章 復旧時の IDM クライアントの調整

IdM サーバーが復元している間は、レプリカトポロジの変更を反映するように IdM クライアントの調整が必要になる場合があります。

### 手順

#### 1. DNS 設定を調整 します。

- a. `/etc/hosts` に IdM サーバーの参照が含まれている場合は、ハードコーディングされた IP からホスト名へのマッピングが有効になっていることを確認してください。
- b. IdM クライアントが名前解決に IdM DNS を使用している場合は、`/etc/resolv.conf` の `nameserver` のエントリーが、DNS サービスを提供する IdM レプリカを指していることを確認します。

#### 2. Kerberos 設定を調整 します。

- a. デフォルトでは、IdM クライアントは Kerberos サーバーの DNS サービスレコードを検索し、レプリカトポロジの変更に合わせて調整します。

```
[root@client ~]# grep dns_lookup_kdc /etc/krb5.conf
dns_lookup_kdc = true
```

- b. IdM クライアントが `/etc/krb5.conf` で特定の IdM サーバーを使用するようにハードコーディングされている場合は、以下を行います。

```
[root@client ~]# grep dns_lookup_kdc /etc/krb5.conf
dns_lookup_kdc = false
```

`/etc/krb5.conf` の `kdc` エントリー、`master_kdc` エントリー、および `admin_server` エントリーが適切に機能する IdM サーバーを参照することを確認します。

```
[realms]
EXAMPLE.COM = {
  kdc = functional-server.example.com:88
  master_kdc = functional-server.example.com:88
  admin_server = functional-server.example.com:749
  default_domain = example.com
  pkinit_anchors = FILE:/var/lib/ipa-client/pki/kdc-ca-bundle.pem
  pkinit_pool = FILE:/var/lib/ipa-client/pki/ca-bundle.pem
}
```

#### 3. SSSD 設定を調整 します。

- a. デフォルトでは、IdM クライアントは LDAP サーバーの DNS サービスレコードを検索し、レプリカトポロジの変更を調整します。

```
[root@client ~]# grep ipa_server /etc/sss/sss.conf
ipa_server = _srv_, functional-server.example.com
```

- b. IdM クライアントが `/etc/sss/sss.conf` で特定の IdM サーバーを使用するようにハードコーディングされている場合は、`ipa_server` エントリーが適切に動作する IdM サーバーを参照するようにしてください。

■



```
[root@client ~]# grep ipa_server /etc/sss/sss.conf  
ipa_server = functional-server.example.com
```

#### 4. SSSD のキャッシュされた情報を消去します。

- SSSD キャッシュには、失われたサーバーに関連する古い情報が含まれる場合があります。認証に一貫性がない場合は、SSSD キャッシュをパーズします。

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

### 検証手順

1. Kerberos TGT (Ticket-Granting-Ticket) を IdM ユーザーとして取得して、Kerberos 設定を確認します。

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

```
[root@client ~]# klist  
Ticket cache: KCM:0  
Default principal: admin@EXAMPLE.COM
```

```
Valid starting Expires Service principal  
10/31/2019 18:44:58 11/25/2019 18:44:55 krbtgt/EXAMPLE.COM@EXAMPLE.COM
```

2. IdM ユーザー情報を取得して、SSSD 設定を確認します。

```
[root@client ~]# id admin  
uid=1965200000(admin) gid=1965200000(admins) groups=1965200000(admins)
```