Red Hat Enterprise Linux 10

Preparing for disaster recovery with Identity Management

Mitigating the effects of server and data loss scenarios in IdM environments

Last Updated: 2026-01-09

Red Hat Enterprise Linux 10 Preparing for disaster recovery with Identity Management

Mitigating the effects of server and data loss scenarios in IdM environments

Legal Notice

Abstract

Server and data loss scenarios, for example due to a hardware failure, are the highest risks in IT environments. To mitigate the effects of these situations, Identity Management (IdM) includes features for replication, virtual machine (VM) snapshots, and data backups. Plan a resilient topology, evaluate potential failure scenarios, and implement a robust recovery strategy to protect your environment from server and data loss.

Table of Contents

PROVIDING FEEDBACK ON RED HAT DOCUMENTATION

We appreciate your feedback on our documentation. Let us know how we can improve it.

Submitting feedback through Jira (account required)

1. Log in to the Jira website.
2. Click **Create** in the top navigation bar
3. Enter a descriptive title in the **Summary** field.
4. Enter your suggestion for improvement in the **Description** field. Include links to the relevant parts of the documentation.
5. Click **Create** at the bottom of the dialogue.

CHAPTER 1. DISASTER RECOVERY TOOLS IN IDM

Identity Management (IdM) provides a set of tools designed to minimize data loss and to recover from a disaster as soon as possible. You can combine these tools to build a comprehensive disaster recovery strategy that balances real-time data redundancy with point-in-time restoration capabilities.

Replication

Replication copies database contents between IdM servers. If an IdM server fails, you can replace the lost server by creating a new replica based on one of the remaining servers.

Virtual machine (VM) snapshots

A snapshot is a view of a VM's operating system and applications on any or all available disks at a given point in time. After taking a VM snapshot, you can use it to return a VM and its IdM data to a previous state.

IdM backups

The **ipa-backup** utility creates a backup of an IdM server's configuration files and its data. You can later use a backup to restore an IdM server to a previous state.

CHAPTER 2. DISASTER SCENARIOS IN IDM

Prepare and respond to various disaster scenarios in Identity Management (IdM) systems that affect servers, data, or entire infrastructures. Identifying specific disaster scenarios and their impact on domain operations helps determine the appropriate recovery strategy.

Table 2.1. Disaster scenarios in IdM

Disaster type	Example causes	How to prepare	How to respond
Server loss: The IdM deployment loses one or several servers.	• Hardware malfunction	• Preparing for server loss with replication	• Recovering a single server with replication
Data loss: IdM data is unexpectedly modified on a server, and the change is propagated to other servers.	• A user accidentally deletes data • A software bug modifies data	• Preparing for data loss with VM snapshots • Planning for data recovery with IdM backups	• Recovering from data loss with VM snapshots • Recovering from data loss with IdM backups • Managing data loss
Total infrastructure loss: All IdM servers or Certificate Authority (CA) replicas are lost with no VM snapshots or data backups available.	• Lack of off-site backups or redundancy prevents recovery after a failure or disaster.	• Preparing for data loss with VM snapshots	This situation is a total loss.

WARNING

A total loss scenario occurs when all Certificate Authority (CA) replicas or all IdM servers are lost, and no virtual machine (VM) snapshots or backups are available for recovery. Without CA replicas, the IdM environment cannot deploy additional replicas or rebuild itself, making recovery impossible. To avoid such scenarios, ensure backups are stored off-site, maintain multiple geographically redundant CA replicas, and connect each replica to at least two others.

CHAPTER 3. PREPARING FOR SERVER LOSS WITH REPLICATION

Identity Management (IdM) replication provides high availability and protects against the loss of individual servers. In the event of a server failure, the remaining replicas maintain service continuity and serve as the data source for restoring the lost node. Proper replication topology ensures that the environment remains resilient and functional during hardware malfunctions or maintenance.

3.1. GUIDELINES FOR CONNECTING IDM REPLICAS IN A TOPOLOGY

Properly sizing and connecting your IdM replicas in a topology is critical to ensuring your infrastructure maintains high-availability, delivers optimal performance, and achieves data consistency across the entire domain.

Connect each replica to at least two other replicas

This ensures that information is replicated not just between the initial replica and the first server you installed, but between other replicas as well.

Connect a replica to a maximum of four other replicas (not a hard requirement)

A large number of replication agreements per server does not add significant benefits. A receiving replica can only be updated by one other replica at a time and meanwhile, the other replication agreements are idle. More than four replication agreements per replica typically means a waste of resources.

NOTE

This recommendation applies to both certificate replication and domain replication agreements.

There are two exceptions to the limit of four replication agreements per replica:

- You want failover paths if certain replicas are not online or responding.
- In larger deployments, you want additional direct links between specific nodes.

Configuring a high number of replication agreements can have a negative impact on overall performance: when multiple replication agreements in the topology are sending updates, certain replicas can experience a high contention on the changelog database file between incoming updates and the outgoing updates.

If you decide to use more replication agreements per replica, ensure that you do not experience replication issues and latency. However, note that large distances and high numbers of intermediate nodes can also cause latency problems.

Connect the replicas in a data center with each other

This ensures domain replication within the data center.

Connect each data center to at least two other data centers

This ensures domain replication between data centers.

Connect data centers using at least a pair of replication agreements

If data centers A and B have a replication agreement from A1 to B1, having a replication agreement from A2 to B2 ensures that if one of the servers is down, the replication can continue between the two data centers.

3.2. REPLICA TOPOLOGY EXAMPLES

These examples offer the practical application of the topology guidelines. They illustrate how you can structure your IdM deployment to build a resilient and scalable IdM infrastructure that guarantees high-availability and optimal performance across multiple data centers.

Figure 3.1. Replica topology with four data centers, each with four servers that are connected with replication agreements

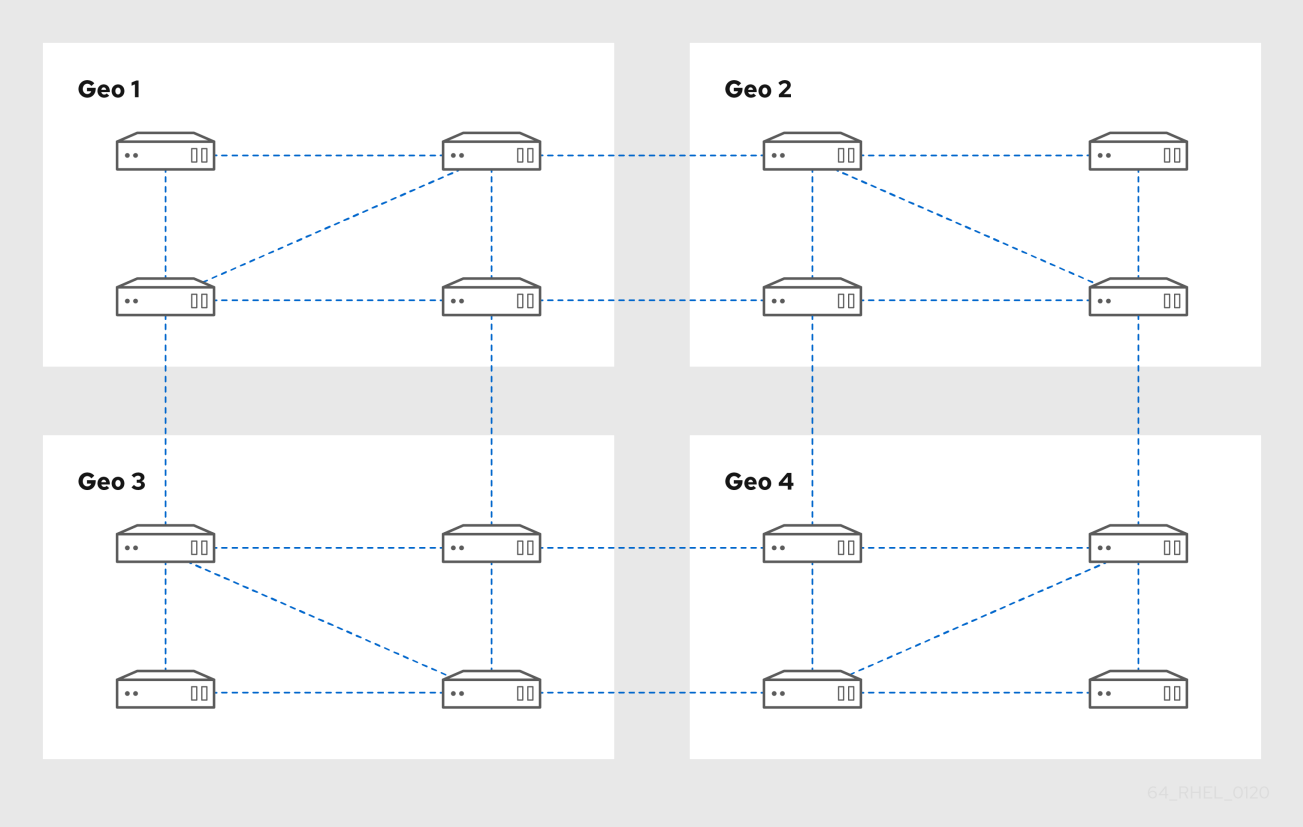
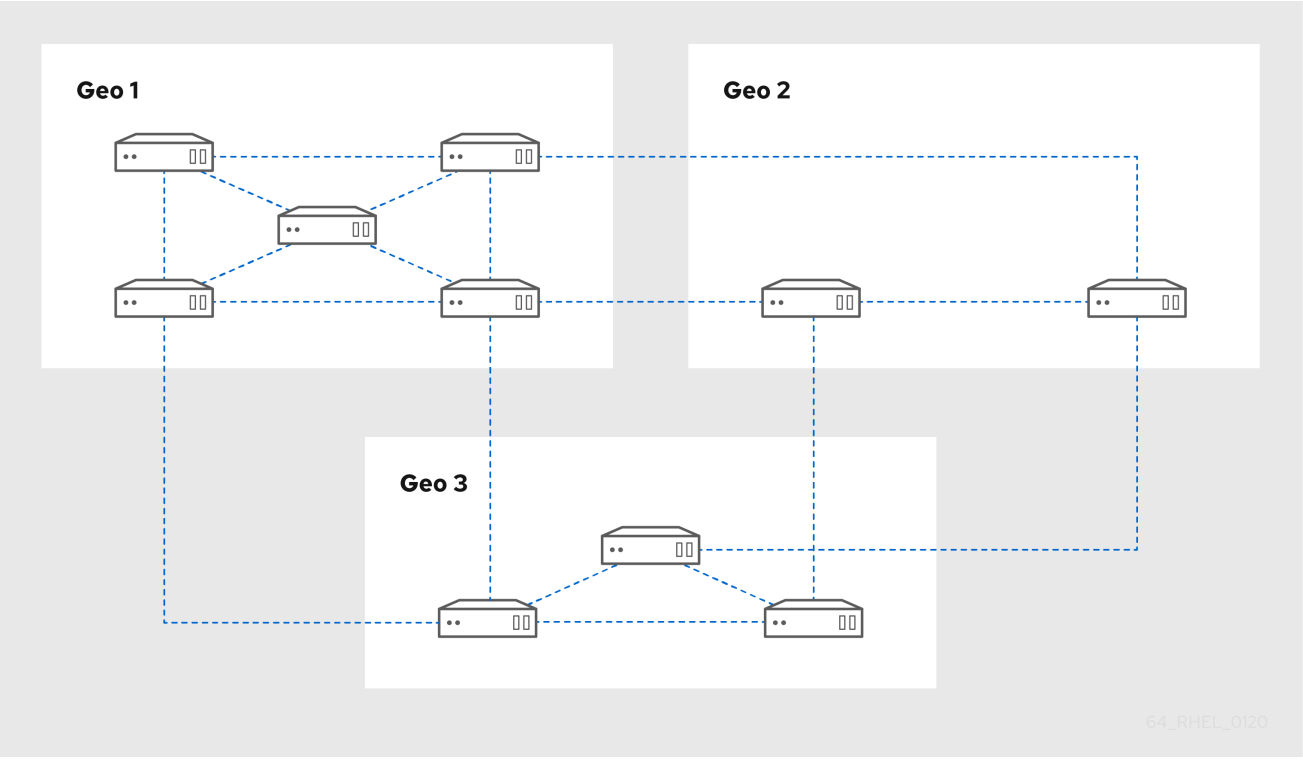


Figure 3.2. Replica topology with three data centers, each with a different number of servers that are all interconnected through replication agreements



3.3. PROTECTING IDM CA DATA

Identity Management (IdM) deployments with an integrated Certificate Authority (CA) maintain high availability through the use of multiple CA replicas. Install several CA replicas to ensure the environment remains resilient and provides the necessary data source to replace any lost nodes.

Procedure

1. Configure three or more replicas to provide CA services.
   a. To install a new replica with CA services, run **ipa-replica-install** with the **--setup-ca** option.

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

   b. To install CA services on a preexisting replica, run **ipa-ca-install**.

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

2. Create CA replication agreements between your CA replicas.

```
[root@careplica1 ~]# ipa topologysegment-add
Suffix name: ca
Left node: ca-replica1.example.com
Right node: ca-replica2.example.com
Segment name [ca-replica1.example.com-to-ca-replica2.example.com]: new_segment
----------------------------
Added segment "new_segment"
----------------------------
  Segment name: new_segment
  Left node: ca-replica1.example.com
  Right node: ca-replica2.example.com
  Connectivity: both
```

WARNING

If only one server provides CA services and it is damaged, the entire environment will be lost. If you use the IdM CA, Red Hat **strongly recommends** having three or more replicas with CA services installed, with CA replication agreements between them.

Additional resources

- Planning your CA services
- Installing an IdM replica
- Planning the replica topology

CHAPTER 4. PREPARING FOR DATA LOSS WITH VM SNAPSHOTS

Virtual machine (VM) snapshots provide a point-in-time record of the operating system software and settings, configuration, and data on an Identity Management (IdM) server. You can use a VM snapshot of an IdM Certificate Authority (CA) replica to rebuild an entire IdM deployment after a disaster.

WARNING

If your environment uses the integrated CA, a snapshot of a replica *without a CA* will not be sufficient for rebuilding a deployment, because certificate data will not be preserved.

Similarly, if your environment uses the IdM Key Recovery Authority (KRA), make sure you create snapshots of a KRA replica, or you might lose the storage key.

Red Hat recommends creating snapshots of a VM that has all of the IdM server roles installed which are in use in your deployment: CA, KRA, DNS.

Prerequisites

- A hypervisor capable of hosting RHEL VMs.

Procedure

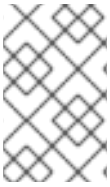
1. Configure at least one **CA replica** in the deployment to run inside a VM.
   a. If IdM DNS or KRA are used in your environment, consider installing DNS and KRA services on this replica as well.
   b. Optional: Configure this VM replica as a hidden replica.
2. Periodically shutdown this VM, take a full snapshot of it, and bring it back online so it continues to receive replication updates. If the VM is a hidden replica, IdM Clients will not be disrupted during this procedure.

Additional resources

- Which hypervisors are certified to run Red Hat Enterprise Linux?
- The hidden replica mode

CHAPTER 5. PREPARING FOR DATA LOSS WITH IDM BACKUPS

IdM provides the **ipa-backup** utility to backup IdM data, and the **ipa-restore** utility to restore servers and data from those backups. A proactive backup plan ensures that configuration and database archives are available to restore your environment to a functional state.



NOTE

Run backups as often as necessary on a *hidden replica* with all server roles installed, especially the Certificate Authority (CA) role if the environment uses the integrated IdM CA. See Installing an IdM hidden replica .

5.1. IDM BACKUP TYPES

To select the correct strategy for protecting your IdM data, you must understand the differences between a full-server backup and a data-only backup.

Full-server backup

- **Contains** all server configuration files related to IdM, and LDAP data in LDAP Data Interchange Format (LDIF) files.
- IdM services must be **offline**.
- **Suitable for** rebuilding an IdM deployment from scratch.

Data-only backup

- **Contains** LDAP data in LDIF files and the replication changelog.
- IdM services can be **online or offline**.
- **Suitable for** restoring IdM data to a state in the past.

5.2. NAMING CONVENTIONS FOR IDM BACKUP FILES

Understand how IdM automatically names backup files so you can easily locate, archive, and differentiate between full-server and data-only backups. By default, IdM stores backups as **.tar** archives in subdirectories of the **/var/lib/ipa/backup/** directory.

The archives and subdirectories follow these naming conventions:

Full-server backup

An archive named **ipa-full.tar** in a directory named **ipa-full-*<YEAR-MM-DD-HH-MM-SS>***, with the time specified in GMT time.

```
[root@server ~]# ll /var/lib/ipa/backup/ipa-full-2021-01-29-12-11-46
total 3056
-rw-r--r--. 1 root root     158 Jan 29 12:11 header
-rw-r--r--. 1 root root 3121511 Jan 29 12:11 ipa-full.tar
```

Data-only backup

An archive named **ipa-data.tar** in a directory named **ipa-data-*<YEAR-MM-DD-HH-MM-SS>***, with the time specified in GMT time.

```
[root@server ~]# ll /var/lib/ipa/backup/ipa-data-2021-01-29-12-14-23
total 1072
-rw-r--r--. 1 root root     158 Jan 29 12:14 header
-rw-r--r--. 1 root root 1090388 Jan 29 12:14 ipa-data.tar
```

NOTE

Uninstalling an IdM server does not automatically remove any backup files.

5.3. CONSIDERATIONS WHEN CREATING A BACKUP

Before you attempt to create a backup of your IdM deployment, review the prerequisites and important considerations about the **ipa-backup** utility.

The important behaviors and limitations of the **ipa-backup** command include the following:

- By default, the **ipa-backup** utility runs in offline mode, which stops all IdM services. The utility automatically restarts IdM services after the backup is finished.
- A full-server backup must **always** run with IdM services offline, but a data-only backup can be performed with services online.
- By default, the **ipa-backup** utility creates backups on the file system containing the **/var/lib/ipa/backup/** directory. Red Hat recommends creating backups regularly on a file system separate from the production filesystem used by IdM, and archiving the backups to a fixed medium, such as tape or optical storage.
- Consider performing backups on hidden replicas. IdM services can be shut down on hidden replicas without affecting IdM clients.
- The **ipa-backup** utility checks if all of the services used in your IdM cluster, such as a Certificate Authority (CA), Domain Name System (DNS), and Key Recovery Agent (KRA), are installed on the server where you are running the backup. If the server does not have all these services installed, the **ipa-backup** utility exits with a warning, because backups taken on that host would not be sufficient for a full cluster restoration.
  For example, if your IdM deployment uses an integrated Certificate Authority (CA), a backup run on a non-CA replica will not capture CA data. Red Hat recommends verifying that the replica where you perform an **ipa-backup** has all of the IdM services used in the cluster installed.

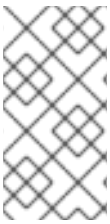
  You can bypass the IdM server role check with the **ipa-backup --disable-role-check** command, but the resulting backup will not contain all the data necessary to restore IdM fully.

5.4. ADDITIONAL RESOURCES

- Backing up and restoring IdM
- Backing up and restoring IdM servers using Ansible playbooks

CHAPTER 6. BACKING UP AND RESTORING IDM

Protect your Identity Management (IdM) deployment against system failure by creating backup and recovery plans. You can manually back up your IdM system to ensure a full restoration of your IdM setup after a data loss event.



NOTE

The IdM backup and restore features are designed to help prevent data loss. To mitigate the impact of server loss and ensure continued operation, provide alternative servers to clients. For information on establishing a replication topology see Preparing for server loss with replication.

6.1. CREATING AN IDM BACKUP

Create a full-server and data-only backup in offline and online modes using the **ipa-backup** command.

Prerequisites

- You must have **root** privileges to run the **ipa-backup** utility.

Procedure

- To create a full-server backup in offline mode, use the **ipa-backup** utility without additional options.

```
[root@server ~]# ipa-backup
Preparing backup on server.example.com
Stopping IPA services
Backing up ipaca in EXAMPLE-COM to LDIF
Backing up userRoot in EXAMPLE-COM to LDIF
Backing up EXAMPLE-COM
Backing up files
Starting IPA service
Backed up to /var/lib/ipa/backup/ipa-full-2020-01-14-11-26-06
The ipa-backup command was successful
```

- To create an offline data-only backup, specify the **--data** option.

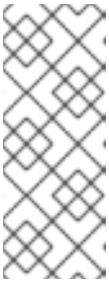
```
[root@server ~]# ipa-backup --data
```

- To create a full-server backup that includes IdM log files, use the **--logs** option.

```
[root@server ~]# ipa-backup --logs
```

- To create a data-only backup while IdM services are running, specify both **--data** and **--online** options.

```
[root@server ~]# ipa-backup --data --online
```



NOTE

If the backup fails due to insufficient space in the **/tmp** directory, use the **TMPDIR** environment variable to change the destination for temporary files created by the backup process:

```
[root@server ~]# TMPDIR=/new/location ipa-backup
```

Verification

- Ensure the backup directory contains an archive with the backup.

```
[root@server ~]# ls /var/lib/ipa/backup/ipa-full-2020-01-14-11-26-06
header  ipa-full.tar
```

Additional resources

- ipa-backup command fails to finish (Red Hat Knowledgebase)

6.2. CREATING A GPG2-ENCRYPTED IDM BACKUP

You can use GNU Privacy Guard (GPG) encryption to create an encrypted IdM backup, which protects your critical deployment data from unauthorized access and enhances overall security.

Prerequisites

- You have created a GPG2 key. See Creating a GPG2 key .

Procedure

- Create a GPG-encrypted backup by specifying the **--gpg** option.

```
[root@server ~]# ipa-backup --gpg
Preparing backup on server.example.com
Stopping IPA services
Backing up ipaca in EXAMPLE-COM to LDIF
Backing up userRoot in EXAMPLE-COM to LDIF
Backing up EXAMPLE-COM
Backing up files
Starting IPA service
Encrypting /var/lib/ipa/backup/ipa-full-2020-01-13-14-38-00/ipa-full.tar
Backed up to /var/lib/ipa/backup/ipa-full-2020-01-13-14-38-00
The ipa-backup command was successful
```

Verification

- Ensure that the backup directory contains an encrypted archive with a **.gpg** file extension.

```
[root@server ~]# ls /var/lib/ipa/backup/ipa-full-2020-01-13-14-38-00
header  ipa-full.tar.gpg
```

Additional resources

- Creating a IdM backup

6.3. CREATING A GPG2 KEY

You must create a GPG2 key to encrypt your IdM backup files.

Prerequisites

- You need **root** privileges.

Procedure

1. Install and configure the **pinentry** utility.

```
[root@server ~]# dnf install pinentry
[root@server ~]# mkdir ~/.gnupg -m 700
[root@server ~]# echo "pinentry-program /usr/bin/pinentry-curses" >> ~/.gnupg/gpg-agent.conf
```

2. Create a **key-input** file used for generating a GPG keypair with your preferred details. For example:

```
[root@server ~]# cat >key-input <<EOF
%echo Generating a standard key
Key-Type: RSA
Key-Length: 2048
Name-Real: GPG User
Name-Comment: first key
Name-Email: root@example.com
Expire-Date: 0
%commit
%echo Finished creating standard key
EOF
```

3. Optional: By default, GPG2 stores its keyring in the **~/.gnupg** file. To use a custom keyring location, set the **GNUPGHOME** environment variable to a directory that is only accessible by root.

```
[root@server ~]# export GNUPGHOME=/root/backup

[root@server ~]# mkdir -p $GNUPGHOME -m 700
```

4. Generate a new GPG2 key based on the contents of the **key-input** file.

```
[root@server ~]# gpg2 --batch --gen-key key-input
```

5. Enter a passphrase to protect the GPG2 key. You use this passphrase to access the private key for decryption.

```
┌──────────────────────────────────────────────────────┐
│ Please enter the passphrase to                       │
│ protect your new key                                 │
```

```
|                                                  |
| Passphrase: <passphrase>                         |
|                                                  |
|  <OK>                    <Cancel>                |
└──────────────────────────────────────────────────┘
```

6. Confirm the correct passphrase by entering it again.

```
┌──────────────────────────────────────────────────┐
| Please re-enter this passphrase                  |
|                                                  |
| Passphrase: <passphrase>                         |
|                                                  |
|  <OK>                    <Cancel>                |
└──────────────────────────────────────────────────┘
```

7. Verify that the new GPG2 key was created successfully.

```
gpg: keybox '/root/backup/pubring.kbx' created
gpg: Generating a standard key
gpg: /root/backup/trustdb.gpg: trustdb created
gpg: key BF28FFA302EF4557 marked as ultimately trusted
gpg: directory '/root/backup/openpgp-revocs.d' created
gpg: revocation certificate stored as '/root/backup/openpgp-
revocs.d/8F6FCF10C80359D5A05AED67BF28FFA302EF4557.rev'
gpg: Finished creating standard key
```

Verification

- List the GPG keys on the server.

```
[root@server ~]# gpg2 --list-secret-keys
gpg: checking the trustdb
gpg: marginals needed: 3  completes needed: 1  trust model: pgp
gpg: depth: 0  valid:   1  signed:   0  trust: 0-, 0q, 0n, 0m, 0f, 1u
/root/backup/pubring.kbx
------------------------
sec   rsa2048 2020-01-13 [SCEA]
      8F6FCF10C80359D5A05AED67BF28FFA302EF4557
uid           [ultimate] GPG User (first key) <root@example.com>
```

Additional resources

- GNU Privacy Guard

6.4. WHEN TO RESTORE FROM AN IDM BACKUP

Review the specific scenarios to determine when to perform an IdM restoration. Restore data only after catastrophic loss, not for simple system maintenance or server replacement.

You can respond to the following disaster scenarios by restoring from an IdM backup:

- **Undesirable changes were made to the LDAP content** Entries were modified or deleted, replication carried out those changes throughout the deployment, and you want to revert those changes. Restoring a data-only backup returns the LDAP entries to the previous state without affecting the IdM configuration itself.

- **Total Infrastructure Loss, or loss of all CA instances** If a disaster damages all Certificate Authority replicas, the deployment has lost the ability to rebuild itself by deploying additional servers. In this situation, restore a backup of a CA Replica and build new replicas from it.

- **An upgrade on an isolated server failed** The operating system remains functional, but the IdM data is corrupted, which is why you want to restore the IdM system to a known good state. Red Hat recommends working with Technical Support to diagnose and troubleshoot the issue. If those efforts fail, restore from a full-server backup.

IMPORTANT

The preferred solution for hardware or upgrade failure is to rebuild the lost server from a replica. For more information, see Recovering a single server with replication.

6.5. CONSIDERATIONS WHEN RESTORING FROM AN IDM BACKUP

Review conditions and technical constraints before restoring an IdM backup. Restoration requires that the target server precisely matches the original setup.

The following are the key considerations while restoring from an IdM backup:

- You can only restore a backup on a server that matches the configuration of the server where the backup was originally created. The server **must** have:
  - The same hostname
  - The same IP address
  - The same version of IdM software
- If one IdM server among many is restored, the restored server becomes the only source of information for IdM. All other servers **must** be re-initialized from the restored server.
- Since any data created after the last backup will be lost, do not use the backup and restore solution for normal system maintenance.
- If a server is lost, Red Hat recommends rebuilding the server by reinstalling it as a replica, instead of restoring from a backup. Creating a new replica preserves data from the current working environment. For more information, see Preparing for server loss with replication.
- The backup and restore features can only be managed from the command line and are not available in the IdM web UI.
- You cannot restore from backup files located in the **/tmp** or **/var/tmp** directories. The IdM Directory Server uses a **PrivateTmp** directory and cannot access the **/tmp** or **/var/tmp** directories commonly available to the operating system.

TIP

Restoring from a backup requires the same software (RPM) versions on the target host as were installed when the backup was performed. Due to this, Red Hat recommends restoring from a Virtual Machine snapshot rather than a backup. For more information, see Recovering from data loss with VM snapshots .

6.6. RESTORING AN IDM SERVER FROM A BACKUP

Using the **ipa-restore** utility, you can restore your IdM server or the LDAP content to the state they were in when the backup was created.

Figure 6.1. Replication topology used in this example

This example restoration scenario involves the following three servers:

- **server1.example.com**: The server that you want to restore from backup.
- **caReplica2.example.com**: A Certificate Authority (CA) replica connected to the **server1.example.com** host.
- **replica3.example.com**: A replica connected to the **caReplica2.example.com** host.

Prerequisites

- You have generated a full-server or data-only backup of the IdM server with the **ipa-backup** utility. See Creating a backup .
- Your backup files are not in the **/tmp** or **/var/tmp** directories.
- Before performing a full-server restore from a full-server backup, uninstall IdM from the server and reinstall IdM using the same server configuration as before.

Procedure

1. Use the **ipa-restore** utility to restore a full-server or data-only backup.
   - If the backup directory is in the default **/var/lib/ipa/backup/** location, enter only the name of the directory:

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

   - If the backup directory is not in the default location, enter its full path:

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

NOTE

The **ipa-restore** utility automatically detects the type of backup that the directory contains, and performs the same type of restore by default. To perform a data-only restore from a full-server backup, add the **--data** option to the **ipa-restore** command:

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

2. Enter the Directory Manager password.

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

3. Enter **yes** to confirm overwriting current data with the backup.

```
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. The **ipa-restore** utility disables replication on all servers that are available:

```
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
```

The utility then stops IdM services, restores the backup, and restarts the services:

```
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. Re-initialize all replicas connected to the restored server:

   a. List all replication topology segments for the **domain** suffix, taking note of topology segments involving the restored server

segments involving the restored server:

```
[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. Re-initialize the **domain** suffix for all topology segments with the restored server. In this example, perform a re-initialization of **caReplica2** with data from **server1**.

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

c. Moving on to Certificate Authority data, list all replication topology segments for the **ca** suffix.

```
[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. Re-initialize all CA replicas connected to the restored server. In this example, perform a **csreplica** re-initialization of **caReplica2** with data from **server1**.

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

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

6. Continue moving outward through the replication topology, re-initializing successive replicas, until all servers have been updated with the data from restored server **server1.example.com**.

In this example, we only have to re-initialize the **domain** suffix on **replica3** with the data from **caReplica2**:

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

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

7. Clear SSSD's cache on every server to avoid authentication problems due to invalid data:
   a. Stop the SSSD service:

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

   b. Remove all cached content from SSSD:

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

   c. Start the SSSD service:

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

   d. Reboot the server.

6.7. RESTORING FROM AN ENCRYPTED BACKUP

You can restore an IdM server from an encrypted IdM backup. The **ipa-restore** utility automatically detects if an IdM backup is encrypted and restores it using the GPG2 root keyring.

Prerequisites

- A GPG-encrypted IdM backup. See Creating encrypted IdM backups .
- The LDAP Directory Manager password
- The passphrase used when creating the GPG key

Procedure

1. If you used a custom keyring location when creating the GPG2 keys, verify that the **$GNUPGHOME** environment variable is set to that directory. See Creating a GPG2 key .

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

2. Provide the **ipa-restore** utility with the backup directory location.

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

   a. Enter the Directory Manager password.

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

b. Enter the passphrase you used when creating the GPG key.

```
┌──────────────────────────────────────────────────────────────────────────────┐
│ 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. Re-initialize all replicas connected to the restored server. See Restoring an IdM server from backup.

CHAPTER 7. BACKING UP AND RESTORING IDM SERVERS USING ANSIBLE PLAYBOOKS

You can use Ansible playbooks to automate IdM server backup, file transfer, and restoration. Automation simplifies disaster recovery and ensures consistency across multiple hosts in your environment.

7.1. USING ANSIBLE TO CREATE A BACKUP OF AN IDM SERVER

You can use the **ipabackup** role in an Ansible playbook to create a backup of an IdM server and store it on the IdM server.

Prerequisites

- You have configured your Ansible control node to meet the following requirements:
  - You are using Ansible version 2.15 or later.
  - You have installed the **ansible-freeipa** package.
  - The example assumes that in the ***~/MyPlaybooks/*** directory, you have created an Ansible inventory file with the fully-qualified domain name (FQDN) of the IdM server.
  - The example assumes that the **secret.yml** Ansible vault stores your **ipaadmin_password** and that you have access to a file that stores the password protecting the **secret.yml** file.
- The target node, that is the node on which the **freeipa.ansible_freeipa** module is executed, is part of the IdM domain as an IdM client, server or replica.

Procedure

1. Navigate to the **~/MyPlaybooks/** directory:

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

2. Make a copy of the **backup-server.yml** file located in the **/usr/share/ansible/collections/ansible_collections/freeipa/ansible_freeipa/playbooks** directory:

```
$ cp
/usr/share/ansible/collections/ansible_collections/freeipa/ansible_freeipa/playbooks/backup-server.yml backup-my-server.yml
```

3. Open the **backup-my-server.yml** Ansible playbook file for editing.
4. Adapt the file by setting the **hosts** variable to a host group from your inventory file. In this example, set it to the **ipaserver** host group:

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

```
roles:
- role: freeipa.ansible_freeipa.ipabackup
  state: present
```

5. Save the file.
   For details about variables and example playbooks in the FreeIPA Ansible collection, see the **/usr/share/ansible/collections/ansible_collections/freeipa/ansible_freeipa/README.md** file and the **/usr/share/ansible/collections/ansible_collections/freeipa/ansible_freeipa/playbooks/** directory on the control node.

6. Run the Ansible playbook, specifying the inventory file and the playbook file:

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

Verification

1. Log into the IdM server that you have backed up.

2. Verify that the backup is in the **/var/lib/ipa/backup** directory.

```
[root@server ~]# ls /var/lib/ipa/backup/
ipa-full-2021-04-30-13-12-00
```

7.2. USING ANSIBLE TO CREATE A BACKUP OF AN IDM SERVER ON YOUR ANSIBLE CONTROLLER

You can use the **ipabackup** role in an Ansible playbook to create a backup of an IdM server and automatically transfer it on your Ansible controller.

Prerequisites

- You have configured your Ansible control node to meet the following requirements:
  - You are using Ansible version 2.15 or later.
  - You have installed the **ansible-freeipa** package.
  - The example assumes that in the ***~/MyPlaybooks/*** directory, you have created an Ansible inventory file with the fully-qualified domain name (FQDN) of the IdM server.
  - The example assumes that the **secret.yml** Ansible vault stores your **ipaadmin_password** and that you have access to a file that stores the password protecting the **secret.yml** file.
- The target node, that is the node on which the **freeipa.ansible_freeipa** module is executed, is part of the IdM domain as an IdM client, server or replica.

Procedure

1. To store the backups, create a subdirectory in your home directory on the Ansible controller.

```
$ mkdir ~/ipabackups
```

2. Navigate to the **~/MyPlaybooks/** directory:

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

3. Make a copy of the **backup-server-to-controller.yml** file located in the **/usr/share/ansible/collections/ansible_collections/freeipa/ansible_freeipa/playbooks** directory:

```
$ cp
/usr/share/ansible/collections/ansible_collections/freeipa/ansible_freeipa/playbooks/backup-server-to-controller.yml backup-my-server-to-my-controller.yml
```

4. Open the **backup-my-server-to-my-controller.yml** file for editing.

5. Adapt the file by setting the following variables:

   a. Set the **hosts** variable to a host group from your inventory file. In this example, set it to the **ipaserver** host group.

   b. Optional: To maintain a copy of the backup on the IdM server, uncomment the following line:

```
# ipabackup_keep_on_server: true
```

6. By default, backups are stored in the present working directory of the Ansible controller. To specify the backup directory you created in Step 1, add the **ipabackup_controller_path** variable and set it to the **/home/user/ipabackups** directory.

```
---
- name: Playbook to backup IPA server to controller
  hosts: ipaserver
  become: true
  vars:
    ipabackup_to_controller: true
    # ipabackup_keep_on_server: true
    ipabackup_controller_path: /home/user/ipabackups

  roles:
  - role: freeipa.ansible_freeipa.ipabackup
    state: present
```

7. Save the file.
   For details about variables and example playbooks in the FreeIPA Ansible collection, see the **/usr/share/ansible/collections/ansible_collections/freeipa/ansible_freeipa/README.md** file and the **/usr/share/ansible/collections/ansible_collections/freeipa/ansible_freeipa/playbooks/** directory on the control node.

8. Run the Ansible playbook, specifying the inventory file and the playbook file:

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

Verification

Verification

- Verify that the backup is in the **/home/user/ipabackups** directory of your Ansible controller:

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

Your backup file name begins with the host name of the IdM server.

7.3. USING ANSIBLE TO COPY A BACKUP OF AN IDM SERVER TO YOUR ANSIBLE CONTROLLER

You can use an Ansible playbook to copy an existing backup file of an IdM server from the IdM server to your Ansible controller.

Prerequisites

- On the control node:
  - You are using Ansible version 2.15 or later.
  - You have installed the **ansible-freeipa** package.
  - The example assumes that in the ***~/MyPlaybooks/*** directory, you have created an Ansible inventory file with the fully-qualified domain name (FQDN) of the IdM server.
  - The example assumes that the **secret.yml** Ansible vault stores your **ipaadmin_password** and that you have access to a file that stores the password protecting the **secret.yml** file.
- The target node, that is the node on which the **freeipa.ansible_freeipa** module is executed, is part of the IdM domain as an IdM client, server or replica.

Procedure

1. To store the backups, create a subdirectory in your home directory on the Ansible controller.

```
$ mkdir ~/ipabackups
```

2. Navigate to the **~/MyPlaybooks/** directory:

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

3. Make a copy of the **copy-backup-from-server.yml** file located in the **/usr/share/ansible/collections/ansible_collections/freeipa/ansible_freeipa/playbooks** directory:

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

4. Open the **copy-my-backup-from-my-server-to-my-controller.yml** file for editing.
5. Adapt the file by setting the following variables:
   a. Set the **hosts** variable to a host group from your inventory file. In this example, set it to the

ipaserver host group.

b. Set the **ipabackup_name** variable to the name of the **ipabackup** on your IdM server to copy to your Ansible controller.

c. By default, backups are stored in the present working directory of the Ansible controller. To specify the directory you created in Step 1, add the **ipabackup_controller_path** variable and set it to the **/home/user/ipabackups** directory.

```
---
- 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: true
    ipabackup_controller_path: /home/user/ipabackups

  roles:
  - role: freeipa.ansible_freeipa.ipabackup
    state: present
```

6. Save the file.
For details about variables and example playbooks in the FreeIPA Ansible collection, see the **/usr/share/ansible/collections/ansible_collections/freeipa/ansible_freeipa/README.md** file and the **/usr/share/ansible/collections/ansible_collections/freeipa/ansible_freeipa/playbooks/** directory on the control node.

7. Run the Ansible playbook, specifying the inventory file and the playbook file:

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

NOTE

To copy **all** IdM backups to your controller, set the **ipabackup_name** variable in the Ansible playbook to **all**:

```
vars:
  ipabackup_name: all
  ipabackup_to_controller: true
```

For an example, see the **copy-all-backups-from-server.yml** Ansible playbook in the **/usr/share/ansible/collections/ansible_collections/freeipa/ansible_freeipa/playbooks** directory.

Verification

- Verify your backup is in the **/home/user/ipabackups** directory on your Ansible controller:

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

7.4. USING ANSIBLE TO COPY A BACKUP OF AN IDM SERVER FROM YOUR ANSIBLE CONTROLLER TO THE IDM SERVER

You can use an Ansible playbook to copy an existing backup file of an IdM server from your Ansible controller to the IdM server.

Prerequisites

- You have configured your Ansible control node to meet the following requirements:
  - You are using Ansible version 2.15 or later.
  - You have installed the **ansible-freeipa** package.
  - The example assumes that in the ***~/MyPlaybooks/*** directory, you have created an Ansible inventory file with the fully-qualified domain name (FQDN) of the IdM server.
  - The example assumes that the **secret.yml** Ansible vault stores your **ipaadmin_password** and that you have access to a file that stores the password protecting the **secret.yml** file.
- The target node, that is the node on which the **freeipa.ansible_freeipa** module is executed, is part of the IdM domain as an IdM client, server or replica.

Procedure

1. Navigate to the ~/**MyPlaybooks**/ directory:

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

2. Make a copy of the **copy-backup-from-controller.yml** file located in the **/usr/share/ansible/collections/ansible_collections/freeipa/ansible_freeipa/playbooks** directory:

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

3. Open the **copy-my-backup-from-my-controller-to-my-server.yml** file for editing.
4. Adapt the file by setting the following variables:
   a. Set the **hosts** variable to a host group from your inventory file. In this example, set it to the **ipaserver** host group.
   b. Set the **ipabackup_name** variable to the name of the **ipabackup** on your Ansible controller to copy to the IdM server.

```
---
- 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: true

roles:
- role: freeipa.ansible_freeipa.ipabackup
  state: copied
```

5. Save the file.
   For details about variables and example playbooks in the FreeIPA Ansible collection, see the **/usr/share/ansible/collections/ansible_collections/freeipa/ansible_freeipa/README.md** file and the **/usr/share/ansible/collections/ansible_collections/freeipa/ansible_freeipa/playbooks/** directory on the control node.

6. Run the Ansible playbook, specifying the inventory file and the playbook file:

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

7.5. USING ANSIBLE TO REMOVE A BACKUP FROM AN IDM SERVER

You can use an Ansible playbook to automate the removal of old or unnecessary backup files from an IdM server.

Prerequisites

- On the control node:
  - You are using Ansible version 2.15 or later.
  - You have installed the **ansible-freeipa** package.
  - The example assumes that in the ***~/MyPlaybooks/*** directory, you have created an Ansible inventory file with the fully-qualified domain name (FQDN) of the IdM server.
  - The example assumes that the **secret.yml** Ansible vault stores your **ipaadmin_password** and that you have access to a file that stores the password protecting the **secret.yml** file.
- The target node, that is the node on which the **freeipa.ansible_freeipa** module is executed, is part of the IdM domain as an IdM client, server or replica.

Procedure

1. Navigate to the ~/MyPlaybooks/ directory:

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

2. Make a copy of the **remove-backup-from-server.yml** file located in the **/usr/share/ansible/collections/ansible_collections/freeipa/ansible_freeipa/playbooks** directory:

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

3. Open the **remove-backup-from-my-server.yml** file for editing.

4. Adapt the file by setting the following variables:

   a. Set the **hosts** variable to a host group from your inventory file. In this example, set it to the **ipaserver** host group.

   b. Set the **ipabackup_name** variable to the name of the **ipabackup** to remove from your IdM server.

```
---
- 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: freeipa.ansible_freeipa.ipabackup
    state: absent
```

5. Save the file.
   For details about variables and example playbooks in the FreeIPA Ansible collection, see the **/usr/share/ansible/collections/ansible_collections/freeipa/ansible_freeipa/README.md** file and the **/usr/share/ansible/collections/ansible_collections/freeipa/ansible_freeipa/playbooks/** directory on the control node.

6. Run the Ansible playbook, specifying the inventory file and the playbook file:

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

NOTE

To remove **all** IdM backups from the IdM server, set the **ipabackup_name** variable in the Ansible playbook to **all**:

```
vars:
  ipabackup_name: all
```

For an example, see the **remove-all-backups-from-server.yml** Ansible playbook in the **/usr/share/ansible/collections/ansible_collections/freeipa/ansible_freeipa/playbooks** directory.

7.6. USING ANSIBLE TO RESTORE AN IDM SERVER FROM A BACKUP STORED ON THE SERVER

You can use an Ansible playbook to restore an IdM server from a backup stored on that host.

Prerequisites

- On the control node:
  - You are using Ansible version 2.15 or later.
  - You have installed the **ansible-freeipa** package.
  - The example assumes that in the ***~/MyPlaybooks/*** directory, you have created an Ansible inventory file with the fully-qualified domain name (FQDN) of the IdM server.
  - The example assumes that the **secret.yml** Ansible vault stores your **ipaadmin_password** and that you have access to a file that stores the password protecting the **secret.yml** file.
- The target node, that is the node on which the **freeipa.ansible_freeipa** module is executed, is part of the IdM domain as an IdM client, server or replica.
- You know the LDAP Directory Manager password.

Procedure

1. Navigate to the **~/MyPlaybooks/** directory:

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

2. Make a copy of the **restore-server.yml** file located in the **/usr/share/ansible/collections/ansible_collections/freeipa/ansible_freeipa/playbooks** directory:

```
$ cp
/usr/share/ansible/collections/ansible_collections/freeipa/ansible_freeipa/playbooks/restore-server.yml restore-my-server.yml
```

3. Open the **restore-my-server.yml** Ansible playbook file for editing.
4. Adapt the file by setting the following variables:
   a. Set the **hosts** variable to a host group from your inventory file. In this example, set it to the **ipaserver** host group.
   b. Set the **ipabackup_name** variable to the name of the **ipabackup** to restore.
   c. Set the **ipabackup_password** variable to the LDAP Directory Manager password.

```
---
- 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: freeipa.ansible_freeipa.ipabackup
  state: restored
```

5. Save the file.
   For details about variables and example playbooks in the FreeIPA Ansible collection, see the **/usr/share/ansible/collections/ansible_collections/freeipa/ansible_freeipa/README.md** file and the **/usr/share/ansible/collections/ansible_collections/freeipa/ansible_freeipa/playbooks/** directory on the control node.

6. Run the Ansible playbook specifying the inventory file and the playbook file:

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

7.7. USING ANSIBLE TO RESTORE AN IDM SERVER FROM A BACKUP STORED ON YOUR ANSIBLE CONTROLLER

You can use an Ansible playbook to restore an IdM server from a backup stored on your Ansible controller.

Prerequisites

- On the control node:
  - You are using Ansible version 2.15 or later.
  - You have installed the **ansible-freeipa** package.
  - The example assumes that in the ***~/MyPlaybooks/*** directory, you have created an Ansible inventory file with the fully-qualified domain name (FQDN) of the IdM server.
  - The example assumes that the **secret.yml** Ansible vault stores your **ipaadmin_password** and that you have access to a file that stores the password protecting the **secret.yml** file.
- The target node, that is the node on which the **freeipa.ansible_freeipa** module is executed, is part of the IdM domain as an IdM client, server or replica.
- You know the LDAP Directory Manager password.

Procedure

1. Navigate to the **~/MyPlaybooks/** directory:

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

2. Make a copy of the **restore-server-from-controller.yml** file located in the **/usr/share/ansible/collections/ansible_collections/freeipa/ansible_freeipa/playbooks** directory:

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

3. Open the **restore-my-server-from-my-controller.yml** file for editing.

4. Adapt the file by setting the following variables:

   a. Set the **hosts** variable to a host group from your inventory file. In this example, set it to the **ipaserver** host group.

   b. Set the **ipabackup_name** variable to the name of the **ipabackup** to restore.

   c. Set the **ipabackup_password** variable to the LDAP Directory Manager password.

```
---
- 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: true

  roles:
  - role: freeipa.ansible_freeipa.ipabackup
    state: restored
```

5. Save the file.
   For details about variables and example playbooks in the FreeIPA Ansible collection, see the **/usr/share/ansible/collections/ansible_collections/freeipa/ansible_freeipa/README.md** file and the **/usr/share/ansible/collections/ansible_collections/freeipa/ansible_freeipa/playbooks/** directory on the control node.

6. Run the Ansible playbook, specifying the inventory file and the playbook file:

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