Red Hat Enterprise Linux 9.0

Performing disaster recovery with Identity Management

Recovering IdM after a server or data loss

Red Hat Enterprise Linux 9.0 Performing disaster recovery with Identity Management

Recovering IdM after a server or data loss

Legal Notice

Abstract

Server and data loss scenarios, for example due to a hardware failure, are the highest risks in IT environments. In case of such an event in a Red Hat Identity Management (IdM) environment, the recovery process depends on the type of problem, the IdM topology, and the actions that have been taken to mitigate such situations. For example, you can recover single and multiple servers in an IdM replication topology, and you can recover data by using IdM backups and snapshots. During or after the recovery, it can be necessary to adjust client settings, such as DNS servers and the Kerberos configuration.

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 SCENARIOS IN IDM

There are two main classes of disaster scenarios: *server loss* and *data loss*.

Table 1.1. Server loss vs. data loss

Disaster type	Example causes	How to respond
Server loss: The IdM deployment loses one or several servers.	• Hardware malfunction	• 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	• Recovering from data loss with VM snapshots • Recovering from data loss with IdM backups • Managing data loss

CHAPTER 2. RECOVERING A SINGLE SERVER WITH REPLICATION

If a single server is severely disrupted or lost, having multiple replicas ensures you can create a replacement replica and quickly restore the former level of redundancy.

If your IdM topology contains an integrated Certificate Authority (CA), the steps for removing and replacing a damaged replica differ for the CA renewal server and other replicas.

2.1. RECOVERING FROM LOSING THE CA RENEWAL SERVER

If the Certificate Authority (CA) renewal server is lost, you must first promote another CA replica to fulfill the CA renewal server role, and then deploy a replacement CA replica.

Prerequisites

- Your deployment uses IdM's internal Certificate Authority (CA).

- Another Replica in the environment has CA services installed.

> **WARNING**
>
> An IdM deployment is unrecoverable if:
>
> 1. The CA renewal server has been lost.
>
> 2. No other server has a CA installed.
>
> 3. No backup of a replica with the CA role exists.
>   It is critical to make backups from a replica with the CA role so certificate data is protected. For more information about creating and restoring from backups, see Preparing for data loss with IdM backups .

Procedure

1. From another replica in your environment, promote another CA replica in the environment to act as the new CA renewal server. See Changing and resetting IdM CA renewal server .

2. From another replica in your environment, remove replication agreements to the lost CA renewal server. See Removing server from topology using the CLI .

3. Install a new CA Replica to replace the lost CA replica. See Installing an IdM replica with a CA .

4. Update DNS to reflect changes in the replica topology. If IdM DNS is used, DNS service records are updated automatically.

5. Verify IdM clients can reach IdM servers. See Adjusting IdM clients during recovery .

Verification

1. Test the Kerberos server on the new replica by successfully retrieving a Kerberos Ticket-Granting-Ticket as an IdM user.

   ```
   [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. Test the Directory Server and SSSD configuration by retrieving user information.

   ```
   [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. Test the CA configuration with the **ipa cert-show** command.

   ```
   [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
   ```

Additional resources

- Using IdM CA renewal server

2.2. RECOVERING FROM LOSING A REGULAR REPLICA

To replace a replica that is not the Certificate Authority (CA) renewal server, remove the lost replica from the topology and install a new replica in its place.

Prerequisites

- The CA renewal server is operating properly. If the CA renewal server has been lost, see Recovering from losing the CA renewal server .

Procedure

1. Remove replication agreements to the lost server. See Uninstalling an IdM server .

2. Deploy a new replica with the corresponding services (CA, KRA, DNS). See Installing an IdM replica.

3. Update DNS to reflect changes in the replica topology. If IdM DNS is used, DNS service records are updated automatically.

4. Verify IdM clients can reach IdM servers. See Adjusting IdM clients during recovery .

Verification

1. Test the Kerberos server on the new replica by successfully retrieving a Kerberos Ticket-Granting-Ticket as an IdM user.

   ```
   [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. Test the Directory Server and SSSD configuration on the new replica by retrieving user information.

   ```
   [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
   ```

CHAPTER 3. RECOVERING MULTIPLE SERVERS WITH REPLICATION

If multiple servers are lost at the same time, determine if the environment can be rebuilt by seeing which one of the following five scenarios applies to your situation.

3.1. RECOVERING FROM LOSING MULTIPLE SERVERS IN A CA-LESS DEPLOYMENT

Servers in a CA-less deployment are all considered equal, so you can rebuild the environment by removing and replacing lost replicas in any order.

Prerequisites

- Your deployment uses an external Certificate Authority (CA).

Procedure

- See Recovering from losing a regular replica .

3.2. RECOVERING FROM LOSING MULTIPLE SERVERS WHEN THE CA RENEWAL SERVER IS UNHARMED

If the CA renewal server is intact, you can replace other servers in any order.

Prerequisites

- Your deployment uses the IdM internal Certificate Authority (CA).

Procedure

- See Recovering from losing a regular replica .

3.3. RECOVERING FROM LOSING THE CA RENEWAL SERVER AND OTHER SERVERS

If you lose the CA renewal server and other servers, promote another CA server to the CA renewal server role before replacing other replicas.

Prerequisites

- Your deployment uses the IdM internal Certificate Authority (CA).

- At least one CA replica is unharmed.

Procedure

1. Promote another CA replica to fulfill the CA renewal server role. See Recovering from losing the CA renewal server.

2. Replace all other lost replicas. See Recovering from losing a regular replica .

3.4. RECOVERING FROM LOSING ALL CA REPLICAS

Without any Certificate Authority (CA) replicas, the IdM environment has lost the ability to deploy additional replicas and rebuild itself.

Prerequisites

- Your deployment uses the IdM internal Certificate Authority (CA).

Procedure

- This situation is a total loss.

Additional resources

- To prepare for total infrastructure loss, see Preparing for data loss with VM snapshots .

3.5. RECOVERING FROM A TOTAL INFRASTRUCTURE LOSS

If all servers are lost at once, and there are no virtual machine (VM) snapshots or data backups to restore from, this situation is unrecoverable.

Procedure

- This situation is a total loss.

Additional resources

- Preparing for data loss with VM snapshots

CHAPTER 4. RECOVERING FROM DATA LOSS WITH VM SNAPSHOTS

If a data loss event occurs, you can restore a Virtual Machine (VM) snapshot of a Certificate Authority (CA) replica to repair the lost data, or deploy a new environment from it.

4.1. RECOVERING FROM ONLY A VM SNAPSHOT

If a disaster affects all IdM servers, and only a snapshot of an IdM CA replica virtual machine (VM) is left, you can recreate your deployment by removing all references to the lost servers and installing new replicas.

Prerequisites

- You have prepared a VM snapshot of a CA replica VM. See Preparing for data loss with VM snapshots.

Procedure

1. Boot the desired snapshot of the CA replica VM.

2. Remove replication agreements to any lost replicas.

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

3. Install a second CA replica. See Installing an IdM replica .

4. The VM CA replica is now the CA renewal server. Red Hat recommends promoting another CA replica in the environment to act as the CA renewal server. See Changing and resetting IdM CA renewal server.

5. Recreate the desired replica topology by deploying additional replicas with the desired services (CA, DNS). See Installing an IdM replica

6. Update DNS to reflect the new replica topology. If IdM DNS is used, DNS service records are updated automatically.

7. Verify that IdM clients can reach the IdM servers. See Adjusting IdM Clients during recovery .

Verification

1. Test the Kerberos server on every replica by successfully retrieving a Kerberos ticket-granting ticket as an IdM user.

   ```
   [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. Test the Directory Server and SSSD configuration on every replica by retrieving user information.

   > [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. Test the CA server on every CA replica with the **ipa cert-show** command.

   > [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

Additional resources

- Planning the replica topology

4.2. RECOVERING FROM A VM SNAPSHOT AMONG A PARTIALLY-WORKING ENVIRONMENT

If a disaster affects some IdM servers while others are still operating properly, you may want to restore the deployment to the state captured in a Virtual Machine (VM) snapshot. For example, if all Certificate Authority (CA) Replicas are lost while other replicas are still in production, you will need to bring a CA Replica back into the environment.

In this scenario, remove references to the lost replicas, restore the CA replica from the snapshot, verify replication, and deploy new replicas.

Prerequisites

- You have prepared a VM snapshot of a CA replica VM. See Preparing for data loss with VM snapshots.

Procedure

1. Remove all replication agreements to the lost servers. See Uninstalling an IdM server.

2. Boot the desired snapshot of the CA replica VM.

3. Remove any replication agreements between the restored server and any lost servers.

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

4. If the restored server does not have replication agreements to any of the servers still in production, connect the restored server with one of the other servers to update the restored server.

   ```
   [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. Review Directory Server error logs at **/var/log/dirsrv/slapd-YOUR-INSTANCE/errors** to see if the CA replica from the snapshot correctly synchronizes with the remaining IdM servers.

6. If replication on the restored server fails because its database is too outdated, reinitialize the restored server.

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

7. If the database on the restored server is correctly synchronized, continue by deploying additional replicas with the desired services (CA, DNS) according to Installing an IdM replica.

Verification

1. Test the Kerberos server on every replica by successfully retrieving a Kerberos ticket–granting ticket as an IdM user.

   ```
   [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. Test the Directory Server and SSSD configuration on every replica by retrieving user information.

```
[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. Test the CA server on every CA replica with the **ipa cert-show** command.

```
[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
```

Additional resources

- Recovering from a VM snapshot to establish a new IdM environment

4.3. RECOVERING FROM A VM SNAPSHOT TO ESTABLISH A NEW IDM ENVIRONMENT

If the Certificate Authority (CA) replica from a restored Virtual Machine (VM) snapshot is unable to replicate with other servers, create a new IdM environment from the VM snapshot.

To establish a new IdM environment, isolate the VM server, create additional replicas from it, and switch IdM clients to the new environment.

Prerequisites

- You have prepared a VM snapshot of a CA replica VM. See Preparing for data loss with VM snapshots.

Procedure

1. Boot the desired snapshot of the CA replica VM.

2. Isolate the restored server from the rest of the current deployment by removing all of its replication topology segments.

   a. First, display all **domain** replication topology segments.

   ```
   [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. Next, delete every **domain** topology segment involving the restored server.

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

   c. Finally, perform the same actions with any **ca** topology segments.

   ```
   [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. Install a sufficient number of IdM replicas from the restored server to handle the deployment load. There are now two disconnected IdM deployments running in parallel.

4. Switch the IdM clients to use the new deployment by hard-coding references to the new IdM replicas. See Adjusting IdM clients during recovery .

5. Stop and uninstall IdM servers from the previous deployment. See Uninstalling an IdM server .

Verification

1. Test the Kerberos server on every new replica by successfully retrieving a Kerberos ticket-granting ticket as an IdM user.

   ```
   [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. Test the Directory Server and SSSD configuration on every new replica by retrieving user information.

   ```
   [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. Test the CA server on every new CA replica with the **ipa cert-show** command.

   ```
   [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
   ```

CHAPTER 5. RECOVERING FROM DATA LOSS WITH IDM BACKUPS

You can use the **ipa-restore** utility to restore an IdM server to a previous state captured in an IdM backup.

5.1. WHEN TO RESTORE FROM AN IDM BACKUP

You can respond to several 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.

5.2. CONSIDERATIONS WHEN RESTORING FROM AN IDM BACKUP

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

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 .

5.3. RESTORING AN IDM SERVER FROM A BACKUP

Restore an IdM server, or its LDAP data, from an IdM backup.

Figure 5.1. Replication topology used in this example

Table 5.1. Server naming conventions used in this example

Server host name	Function
server1.example.com	The server that needs to be restored 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.

   ```
   [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.

Additional resources

- The **ipa-restore (1)** man page also covers in detail how to handle complex replication scenarios during restoration.

5.4. RESTORING FROM AN ENCRYPTED BACKUP

This procedure restores 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 6. RESTORING IDM SERVERS USING ANSIBLE PLAYBOOKS

Using the **ipabackup** Ansible role, you can automate restoring an IdM server from a backup and transferring backup files between servers and your Ansible controller.

6.1. PREPARING YOUR ANSIBLE CONTROL NODE FOR MANAGING IDM

As a system administrator managing Identity Management (IdM), when working with Red Hat Ansible Engine, it is good practice to do the following:

- Create a subdirectory dedicated to Ansible playbooks in your home directory, for example **~/MyPlaybooks**.

- Copy and adapt sample Ansible playbooks from the **/usr/share/doc/ansible-freeipa/*** and **/usr/share/doc/rhel-system-roles/*** directories and subdirectories into your **~/MyPlaybooks** directory.

- Include your inventory file in your **~/MyPlaybooks** directory.

By following this practice, you can find all your playbooks in one place and you can run your playbooks without invoking root privileges.

> **NOTE**
>
> You only need **root** privileges on the managed nodes to execute the **ipaserver**, **ipareplica**, **ipaclient**, **ipabackup**, **ipasmartcard_server** and **ipasmartcard_client** **ansible-freeipa** roles. These roles require privileged access to directories and the **dnf** software package manager.

Follow this procedure to create the **~/MyPlaybooks** directory and configure it so that you can use it to store and run Ansible playbooks.

Prerequisites

- You have installed an IdM server on your managed nodes, *server.idm.example.com* and *replica.idm.example.com*.

- You have configured DNS and networking so you can log in to the managed nodes, *server.idm.example.com* and *replica.idm.example.com*, directly from the control node.

- You know the IdM **admin** password.

Procedure

1. Create a directory for your Ansible configuration and playbooks in your home directory:

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

2. Change into the **~/MyPlaybooks/** directory:

   ```
   $ cd ~/MyPlaybooks
   ```

3. Create the **~/MyPlaybooks/ansible.cfg** file with the following content:

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

   [privilege_escalation]
   become=True
   ```

4. Create the **~/MyPlaybooks/inventory** file with the following content:

   ```
   [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
   ```

   This configuration defines two host groups, **eu** and **us**, for hosts in these locations. Additionally, this configuration defines the **ipaserver** host group, which contains all hosts from the   **eu** and **us** groups.

5. Optional: Create an SSH public and private key. To simplify access in your test environment, do not set a password on the private key:

   ```
   $ ssh-keygen
   ```

6. Copy the SSH public key to the IdM **admin** account on each managed node:

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

   You must enter the IdM **admin** password when you enter these commands.

Additional resources

- Installing an Identity Management server using an Ansible playbook

- How to build your inventory

6.2. 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

- You have configured your Ansible control node to meet the following requirements:

  - You are using Ansible version 2.14 or later.

  - You have installed the **ansible-freeipa** package on the Ansible controller.

  - 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**.

- The target node, that is the node on which the **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**/**doc**/**ansible-freeipa**/**playbooks** directory:

   ```
   $ cp /usr/share/doc/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: ipabackup
        state: restored
```

5. Save the file.

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
   ```

Additional resources

- The **README.md** file in the **/usr/share/doc/ansible-freeipa/roles/ipabackup** directory.

- The **/usr/share/doc/ansible-freeipa/playbooks/** directory.

6.3. 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

- You have configured your Ansible control node to meet the following requirements:

  - You are using Ansible version 2.14 or later.

  - You have installed the **ansible-freeipa** package on the Ansible controller.

  - 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**.

- The target node, that is the node on which the **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/doc/ansible-freeipa/playbooks** directory:

   ```
   $ cp /usr/share/doc/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: ipabackup
       state: restored
   ```

5. Save the file.

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
   ```

Additional resources

- The **README.md** file in the **/usr/share/doc/ansible-freeipa/roles/ipabackup** directory.

- The **/usr/share/doc/ansible-freeipa/playbooks/** directory.

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

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

Prerequisites

- You have configured your Ansible control node to meet the following requirements:

  - You are using Ansible version 2.14 or later.

  - You have installed the **ansible-freeipa** package on the Ansible controller.

  - 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**.

- The target node, that is the node on which the **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/doc/ansible-freeipa/playbooks** directory:

   ```
   $ cp /usr/share/doc/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: ipabackup
       state: present
   ```

6. Save the file.

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/doc/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
  ```

Additional resources

- The **README.md** file in the **/usr/share/doc/ansible-freeipa/roles/ipabackup** directory.

- The **/usr/share/doc/ansible-freeipa/playbooks/** directory.

6.5. 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 a backup 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 2.14 or later.

  - You have installed the **ansible-freeipa** package on the Ansible controller.

  - 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**.

- The target node, that is the node on which the **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/doc/ansible-freeipa/playbooks** directory:

   ```
   $ cp /usr/share/doc/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: ipabackup
       state: copied
   ```

5. Save the file.

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
   ```

Additional resources

- The **README.md** file in the **/usr/share/doc/ansible-freeipa/roles/ipabackup** directory.

- The **/usr/share/doc/ansible-freeipa/playbooks/** directory.

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

You can use an Ansible playbook to remove a backup from an IdM server.

Prerequisites

- You have configured your Ansible control node to meet the following requirements:

  - You are using Ansible version 2.14 or later.

  - You have installed the **ansible-freeipa** package on the Ansible controller.

- 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**.

- The target node, that is the node on which the **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/doc/ansible-freeipa/playbooks** directory:

   ```
   $ cp /usr/share/doc/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: ipabackup
       state: absent
   ```

5. Save the file.

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/doc/ansible-freeipa/playbooks** directory.

Additional resources

- The **README.md** file in the **/usr/share/doc/ansible-freeipa/roles/ipabackup** directory.

- The **/usr/share/doc/ansible-freeipa/playbooks/** directory.

CHAPTER 7. MANAGING DATA LOSS

The proper response to a data loss event will depend on the number of replicas that have been affected and the type of lost data.

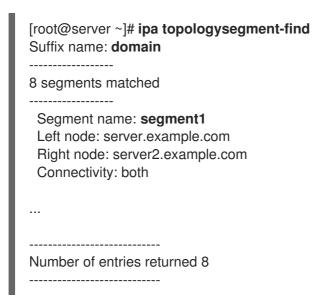## 7.1. RESPONDING TO ISOLATED DATA LOSS

When a data loss event occurs, minimize replicating the data loss by immediately isolating the affected servers. Then create replacement replicas from the unaffected remainder of the environment.

Prerequisites

- A robust IdM replication topology with multiple replicas. See Preparing for server loss with replication.

Procedure

1. To limit replicating the data loss, disconnect all affected replicas from the rest of the topology by removing their replication topology segments.

   a. Display all **domain** replication topology segments in the deployment.

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

   b. Delete all **domain** topology segments involving the affected servers.

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

   c. Perform the same actions with any **ca** topology segments involving any affected servers.

      ```
      [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. The servers affected by the data loss must be abandoned. To create replacement replicas, see Recovering multiple servers with replication .

7.2. RESPONDING TO LIMITED DATA LOSS AMONG ALL SERVERS

A data loss event can affect all replicas in the environment, such as replication carrying out an accidental deletion among all servers. If data loss is known and limited, manually re-add lost data.

Prerequisites

- A Virtual Machine (VM) snapshot or IdM backup of an IdM server that contains the lost data.

Procedure

1. If you need to review any lost data, restore the VM snapshot or backup to an isolated server on a separate network.

2. Add the missing information to the database using **ipa** or **ldapadd** commands.

Additional resources

- Recovering from data loss with VM snapshots

- Backing Up and Restoring IdM

7.3. RESPONDING TO UNDEFINED DATA LOSS AMONG ALL SERVERS

If data loss is severe or undefined, deploy a new environment from a Virtual Machine (VM) snapshot of a server.

Prerequisites

- A Virtual Machine (VM) snapshot contains the lost data.

Procedure

1. Restore an IdM Certificate Authority (CA) Replica from a VM snapshot to a known good state, and deploy a new IdM environment from it. See Recovering from only a VM snapshot .

2. Add any data created after the snapshot was taken using **ipa** or **ldapadd** commands.

Additional resources

- [Recovering from data loss with VM snapshots](#)

CHAPTER 8. ADJUSTING IDM CLIENTS DURING RECOVERY

While IdM servers are being restored, you may need to adjust IdM clients to reflect changes in the replica topology.

Procedure

1. **Adjusting DNS configuration**:

   a. If **/etc/hosts** contains any references to IdM servers, ensure that hard-coded IP-to-hostname mappings are valid.

   b. If IdM clients are using IdM DNS for name resolution, ensure that the **nameserver** entries in **/etc/resolv.conf** point to working IdM replicas providing DNS services.

2. **Adjusting Kerberos configuration**:

   a. By default, IdM clients look to DNS Service records for Kerberos servers, and will adjust to changes in the replica topology:

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

   b. If IdM clients have been hard-coded to use specific IdM servers in **/etc/krb5.conf**:

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

      make sure **kdc**, **master_kdc** and **admin_server** entries in **/etc/krb5.conf** are pointing to IdM servers that work properly:

      ```
      [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. **Adjusting SSSD configuration**:

   a. By default, IdM clients look to DNS Service records for LDAP servers and adjust to changes in the replica topology:

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

   b. If IdM clients have been hard-coded to use specific IdM servers in **/etc/sssd/sssd.conf**, make sure the **ipa_server** entry points to IdM servers that are working properly:

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

4. **Clearing SSSD's cached information**

- The SSSD cache may contain outdated information pertaining to lost servers. If users experience inconsistent authentication problems, purge the SSSD cache :

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

Verification

1. Verify the Kerberos configuration by retrieving a Kerberos Ticket-Granting-Ticket as an IdM user.

   ```
   [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. Verify the SSSD configuration by retrieving IdM user information.

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