Red Hat

Red Hat Enterprise Linux 8

Performing disaster recovery with Identity Management

Documentation for recovering from a disaster affecting an Identity Management deployment

Last Updated: 2020-04-28

Red Hat Enterprise Linux 8 Performing disaster recovery with Identity Management

Documentation for recovering from a disaster affecting an Identity Management deployment

Legal Notice

Abstract

This document describes responding to server or data loss among an Identity Management deployment with replication, Virtual Machine snapshots, and backups.

Table of Contents

PROVIDING FEEDBACK ON RED HAT DOCUMENTATION 3

CHAPTER 1. DISASTER SCENARIOS IN IDM 4

CHAPTER 2. RECOVERING FROM SERVER LOSS WITH REPLICATION 5
2.1. RECOVERING FROM LOSING THE CA RENEWAL MASTER 5
2.2. RECOVERING FROM LOSING A REGULAR REPLICA 6
2.3. RECOVERING FROM LOSING MULTIPLE SERVERS 7
2.3.1. Recovering from losing multiple servers in a CA-less deployment 7
2.3.2. Recovering from losing multiple servers when the CA renewal master is unharmed 8
2.3.3. Recovering from losing the CA renewal master and other servers 8
2.3.4. Recovering from losing all CA replicas 8
2.3.5. Recovering from a total infrastructure loss 8

CHAPTER 3. RECOVERING FROM DATA LOSS WITH VM SNAPSHOTS 10
3.1. RECOVERING FROM ONLY A VM SNAPSHOT 10
3.2. RECOVERING FROM A VM SNAPSHOT AMONG A PARTIALLY-WORKING ENVIRONMENT 11
3.3. RECOVERING FROM A VM SNAPSHOT TO ESTABLISH A NEW IDM ENVIRONMENT 13

CHAPTER 4. RECOVERING FROM DATA LOSS WITH IDM BACKUPS 16
4.1. WHEN TO RESTORE FROM AN IDM BACKUP 16
4.2. CONSIDERATIONS WHEN RESTORING FROM AN IDM BACKUP 16
4.3. RESTORING AN IDM SERVER FROM A BACKUP 17
4.4. RESTORING FROM AN ENCRYPTED BACKUP 20

CHAPTER 5. MANAGING DATA LOSS 22
5.1. RESPONDING TO ISOLATED DATA LOSS 22
5.2. RESPONDING TO LIMITED DATA LOSS AMONG ALL SERVERS 23
5.3. RESPONDING TO UNDEFINED DATA LOSS AMONG ALL SERVERS 23

CHAPTER 6. ADJUSTING IDM CLIENTS DURING RECOVERY 25

PROVIDING FEEDBACK ON RED HAT DOCUMENTATION

We appreciate your input on our documentation. Please let us know how we could make it better. To do so:

- For simple comments on specific passages:
  1. Make sure you are viewing the documentation in the *Multi-page HTML* format. In addition, ensure you see the **Feedback** button in the upper right corner of the document.
  2. Use your mouse cursor to highlight the part of text that you want to comment on.
  3. Click the **Add Feedback** pop-up that appears below the highlighted text.
  4. Follow the displayed instructions.
- For submitting more complex feedback, create a Bugzilla ticket:
  1. Go to the Bugzilla website.
  2. As the Component, use **Documentation**.
  3. Fill in the **Description** field with your suggestion for improvement. Include a link to the relevant part(s) of documentation.
  4. Click **Submit Bug**.

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	• Chapter 2, *Recovering from server loss 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	• Chapter 3, *Recovering from data loss with VM snapshots* • Chapter 4, *Recovering from data loss with IdM backups* • Chapter 5, *Managing data loss*

CHAPTER 2. RECOVERING FROM SERVER LOSS WITH REPLICATION

If a 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 master and other replicas.

2.1. RECOVERING FROM LOSING THE CA RENEWAL MASTER

If the Certificate Authority (CA) renewal master is lost, you must first promote another CA replica to fulfill the CA renewal master 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 master 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 on creating and restoring from backups, see Preparing for data loss with IdM backups .

Procedure

1. Remove replication agreements to the lost CA renewal master. See Uninstalling an IdM server.
2. Promote another CA Replica in the environment to act as the new CA renewal master. See Changing and resetting IdM CA Renewal Master.
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 steps

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

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

   [root@master ~]# 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/master.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@master ~]# 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@master ~]# 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

- For more information regarding the IdM CA Renewal Master, see Using IdM CA renewal master

2.2. RECOVERING FROM LOSING A REGULAR REPLICA

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

Prerequisites

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

Procedure

1. Remove replication agreements to the lost server. See Uninstalling an IdM server.
2. Deploy a new replica with the desired 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 steps

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/master.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
```

2.3. RECOVERING FROM LOSING MULTIPLE SERVERS

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.

2.3.1. Recovering from losing multiple servers in a CA-less deployment

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

Procedure

- See Recovering from losing a regular replica .

2.3.2. Recovering from losing multiple servers when the CA renewal master is unharmed

Prerequisites

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

Procedure

- See Recovering from losing a regular replica .

2.3.3. Recovering from losing the CA renewal master and other servers

Prerequisites

- Your deployment uses IdM's internal Certificate Authority (CA).
- At least one CA replica is unharmed.

Procedure

1. Promote another CA replica to fulfill the CA renewal master role. See Recovering from losing the CA renewal master.
2. Replace all other lost replicas. See Recovering from losing a regular replica .

2.3.4. 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 IdM's 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 .

2.3.5. 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

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

CHAPTER 3. 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.

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

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

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

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 steps

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

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

[root@master ~]# 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/master.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@master ~]# 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@master ~]# 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

- For replication topology best-practices, see Planning the replica topology.

3.2. 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 in order 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 steps

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

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

   [root@master ~]# 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/master.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@master ~]# 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@master ~]# 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

- If the database on the restored CA replica does not synchronize or reinitialize, create a new deployment from the restored CA Replica and switch to the new environment. See Recovering from a VM snapshot to establish a new IdM environment.

3.3. 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 steps

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

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

[root@master ~]# 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/master.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@master ~]# 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@master ~]# 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 4. 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.

4.1. 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 in order 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 from server loss with replication.

4.2. 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 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 in a multi-master environment is restored, the restored server becomes the only source of information for IdM. All other master 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.

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 .

4.3. RESTORING AN IDM SERVER FROM A BACKUP

The following procedure describes restoring an IdM server, or its LDAP data, from an IdM backup.

Figure 4.1. Replication Topology used in this example

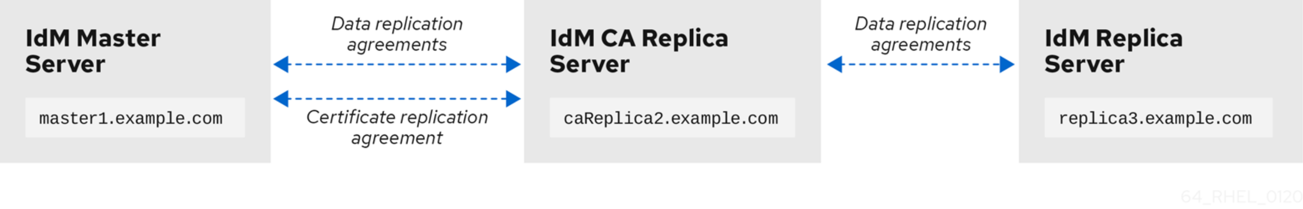

Table 4.1. Server naming conventions used in this example

Server Name	Function
master1.example.com	The server that needs to be restored from backup
caReplica2.example.com	A Certificate Authority (CA) replica connected to master1.example.com.
replica3.example.com	A replica connected to caReplica2.example.com.

Prerequisites

- A full-server or data-only backup of the IdM server was generated with the **ipa-backup** utility. See Creating a backup .

- 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@master1 ~]# 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@master1 ~]# 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 **ipa-restore**:

  ```
  [root@master1 ~]# 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
   master1.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 master1.example.com to caReplica2.example.com
   Disabling CA replication agreement on master1.example.com to caReplica2.example.com
   Disabling replication agreement on caReplica2.example.com to master1.example.com
   Disabling replication agreement on caReplica2.example.com to replica3.example.com
   Disabling CA replication agreement on caReplica2.example.com to master1.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@master1 ~]# ipa topologysegment-find domain
    ------------------
    2 segments matched
    ------------------
      Segment name: master1.example.com-to-caReplica2.example.com
      Left node: master1.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 **master1**.

    ```
    [root@caReplica2 ~]# ipa-replica-manage re-initialize --from=master1.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@master1 ~]# ipa topologysegment-find ca
    -----------------
    1 segment matched
    -----------------
      Segment name: master1.example.com-to-caReplica2.example.com
      Left node: master1.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 **master1**.

    ```
    [root@caReplica2 ~]# ipa-csreplica-manage re-initialize --
    from=master1.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 **master1.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 in order 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.

4.4. RESTORING FROM AN ENCRYPTED BACKUP

The **ipa-restore** utility automatically detects if an IdM backup is encrypted, and restores it using the GPG2 root keyring and **gpg-agent** by default.

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, make sure that the **$GNUPGHOME** environment variable is set to that directory. See Creating a GPG2 key for encrypting IdM backups.

```
[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: │
│ "IPA Backup (IPA Backup) <root@example.com>"              │
│ 2048-bit RSA key, ID BF28FFA302EF4557,                    │
│ created 2020-01-13.                           │
│                                      │
│                                      │
│ Passphrase: SecretPassPhrase42                     │
│                                      │
│     <OK>                  <Cancel>    │
 └──────────────────────────────────────────────────────────────
──────────────┘
```

3. Re-initialize all replicas connected to the restored server. See Restoring an IdM server from backup.

CHAPTER 5. 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.

5.1. 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 from losing multiple servers .

5.2. RESPONDING TO LIMITED DATA LOSS AMONG ALL SERVERS

A data loss event may 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

- For information on restoring from VM snapshots, see Recovering from data loss with VM snapshots.

- For information on backing up and restoring IdM, see Backing Up and Restoring IdM .

5.3. 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

- For information on restoring from VM snapshots, see Recovering from data loss with VM snapshots.

CHAPTER 6. 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 = working-master.example.com:88
      master_kdc = working-master.example.com:88
      admin_server = working-master.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_, master.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 = working-master.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 steps

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)
   ```