Red Hat Enterprise Linux 9

Managing replication in Identity Management

Preparing and verifying replication environments

Last Updated: 2025-05-20

Red Hat Enterprise Linux 9 Managing replication in Identity Management

Preparing and verifying replication environments

Legal Notice

Abstract

In a Red Hat Identity Management (IdM) environment, replication enables failover and load-balancing. You can configure, verify, and stop replication between servers using the command-line, the Web UI, and Ansible Playbooks.

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. MANAGING REPLICATION TOPOLOGY

You can manage replication between servers in an Identity Management (IdM) domain. When you create a replica, Identity Management (IdM) creates a replication agreement between the initial server and the replica. The data that is replicated is then stored in topology suffixes and when two replicas have a replication agreement between their suffixes, the suffixes form a topology segment.

Additional resources

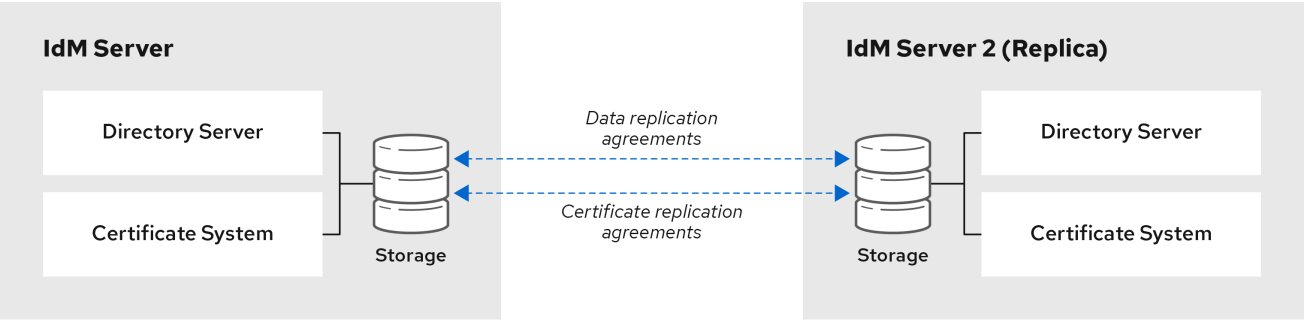
- Planning the replica topology
- Uninstalling an IdM server
- Failover, load-balancing, and high-availability in IdM
- Tuning performance in Identity Management

1.1. REPLICATION AGREEMENTS BETWEEN IDM REPLICAS

When an administrator creates a replica based on an existing server, Identity Management (IdM) creates a *replication agreement* between the initial server and the replica. The replication agreement ensures that the data and configuration is continuously replicated between the two servers.

IdM uses *multiple read/write replica replication* . In this configuration, all replicas joined in a replication agreement receive and provide updates, and are therefore considered suppliers and consumers. Replication agreements are always bilateral.

Figure 1.1. Server and replica agreements



IdM uses two types of replication agreements:

- **Domain replication agreements** replicate the identity information.
- **Certificate replication agreements** replicate the certificate information.

Both replication channels are independent. Two servers can have one or both types of replication agreements configured between them. For example, when server A and server B have only domain replication agreement configured, only identity information is replicated between them, not the certificate information.

1.2. TOPOLOGY SUFFIXES

Topology suffixes store the data that is replicated. IdM supports two types of topology suffixes: **domain** and **ca**. Each suffix represents a separate server, a separate replication topology.

When a replication agreement is configured, it joins two topology suffixes of the same type on two different servers.

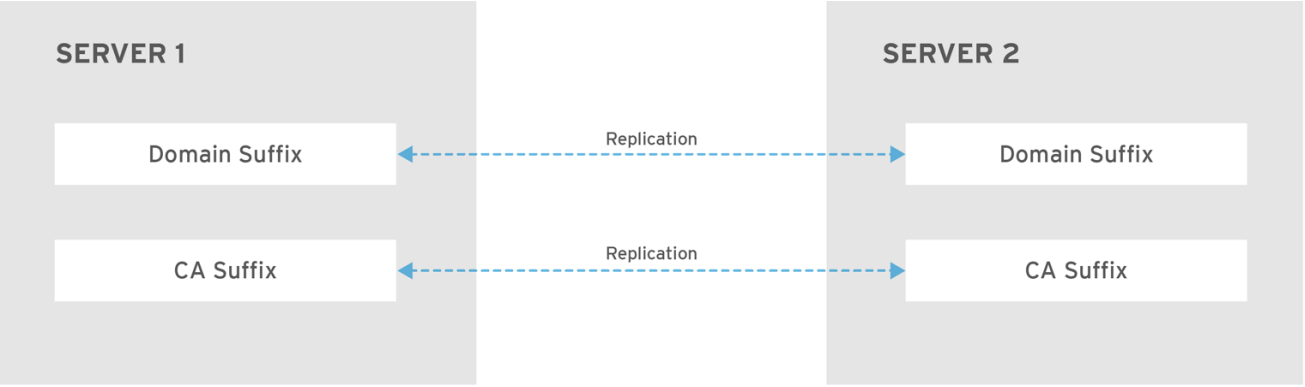
The domain suffix: dc=*example*,dc=*com*

The **domain** suffix contains all domain-related data.
When two replicas have a replication agreement between their **domain** suffixes, they share directory data, such as users, groups, and policies.

The ca suffix: o=ipaca

The **ca** suffix contains data for the Certificate System component. It is only present on servers with a certificate authority (CA) installed.
When two replicas have a replication agreement between their **ca** suffixes, they share certificate data.

Figure 1.2. Topology suffixes



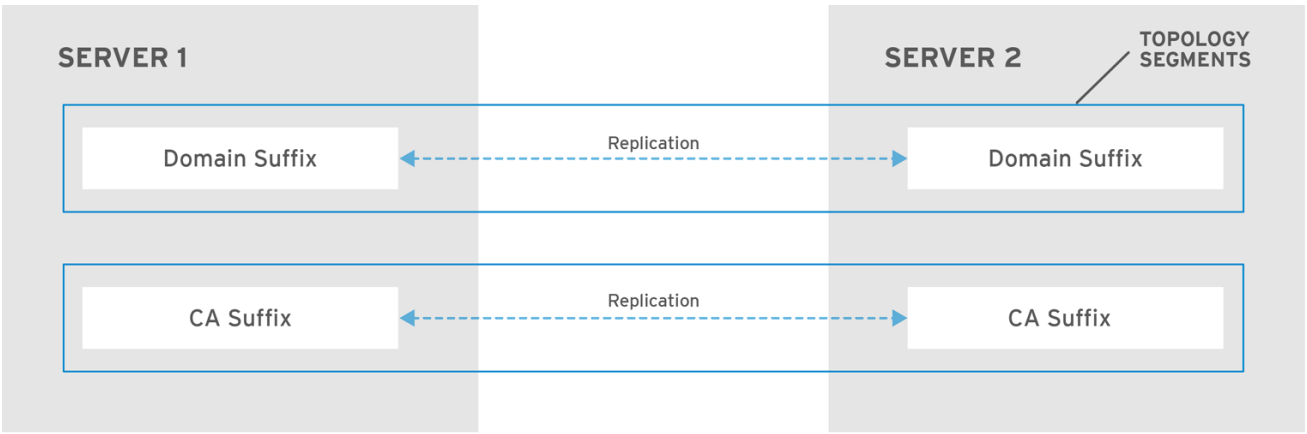
An initial topology replication agreement is set up between two servers by the **ipa-replica-install** script when installing a new replica.

1.3. TOPOLOGY SEGMENTS

When two replicas have a replication agreement between their suffixes, the suffixes form a *topology segment*. Each topology segment consists of a *left node* and a *right node*. The nodes represent the servers joined in the replication agreement.

Topology segments in IdM are always bidirectional. Each segment represents two replication agreements: from server A to server B, and from server B to server A. The data is therefore replicated in both directions.

Figure 1.3. Topology segments



1.4. VIEWING AND MODIFYING THE VISUAL REPRESENTATION OF THE REPLICATION TOPOLOGY USING THE WEBUI

Using the Web UI, you can view, manipulate, and transform the representation of the replication topology. The topology graph in the web UI shows the relationships between the servers in the domain. You can move individual topology nodes by holding and dragging the mouse.

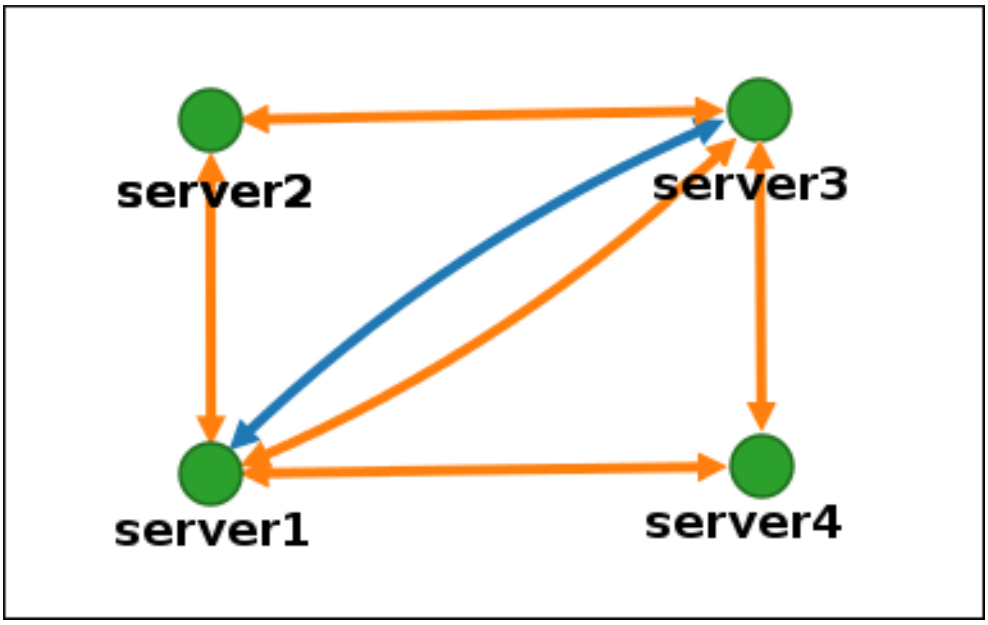
Interpreting the topology graph

Servers joined in a domain replication agreement are connected by an orange arrow. Servers joined in a CA replication agreement are connected by a blue arrow.

Topology graph example: recommended topology

The recommended topology example below shows one of the possible recommended topologies for four servers: each server is connected to at least two other servers, and more than one server is a CA server.

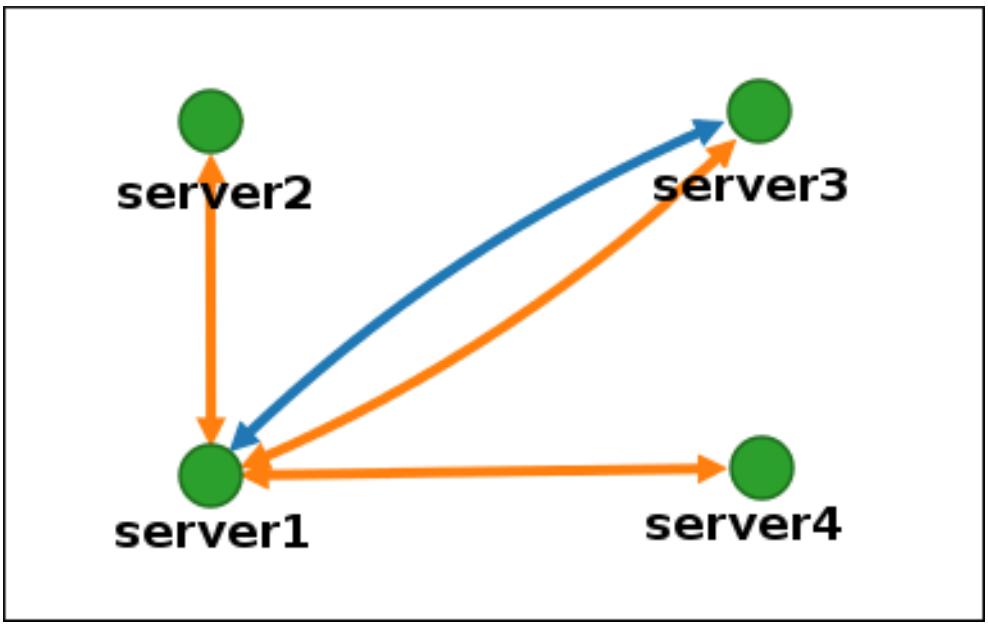
Figure 1.4. Recommended topology example



Topology graph example: discouraged topology

In the discouraged topology example below, **server1** is a single point of failure. All the other servers have replication agreements with this server, but not with any of the other servers. Therefore, if **server1** fails, all the other servers will become isolated.
Avoid creating topologies like this.

Figure 1.5. Discouraged topology example: Single Point of Failure

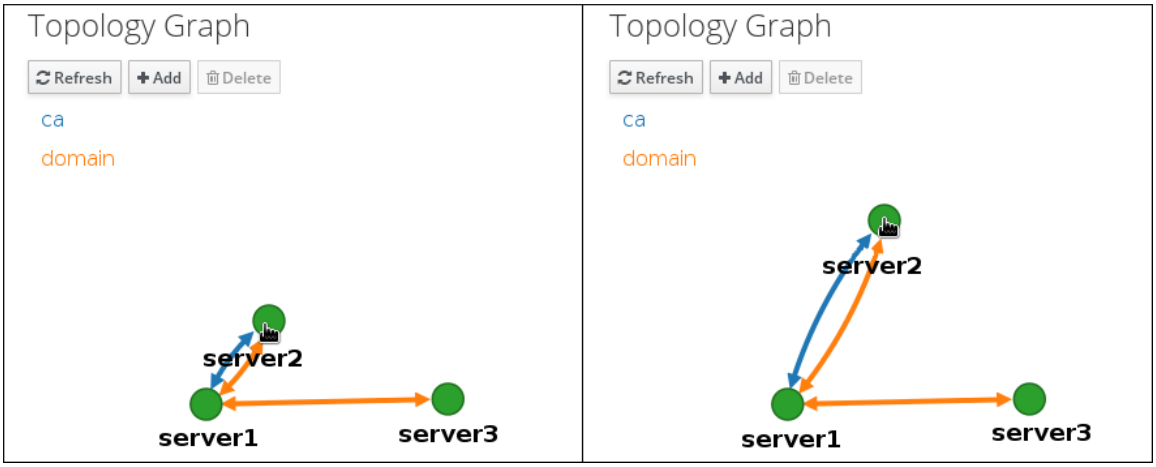


Prerequisites

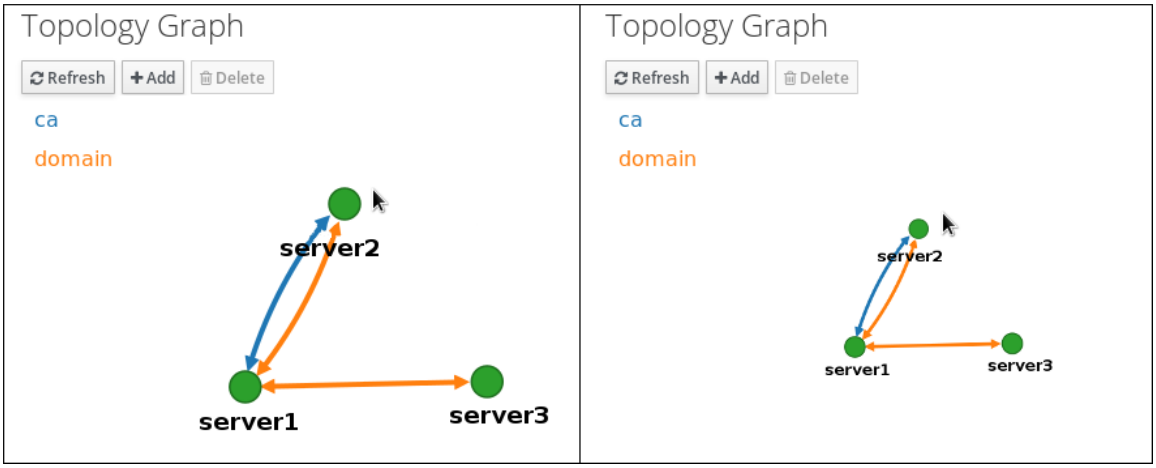
- You are logged in as an IdM administrator.

Procedure

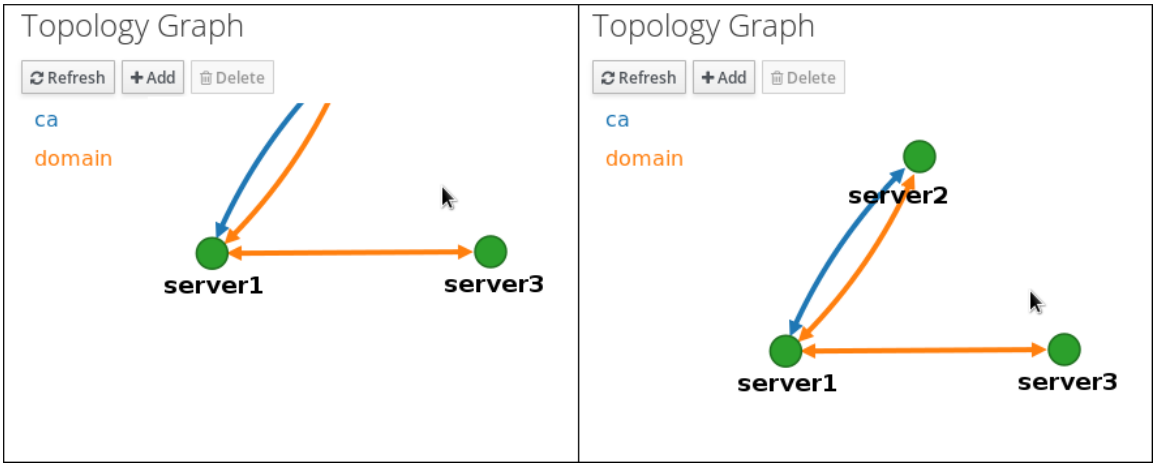
1. Select **IPA Server → Topology → Topology Graph**.
2. Make changes to the topology:
   - You can move the topology graph nodes using the left mouse button:



   - You can zoom in and zoom out the topology graph using the mouse wheel:



- You can move the canvas of the topology graph by holding the left mouse button:



3. If you make any changes to the topology that are not immediately reflected in the graph, click **Refresh**.

1.5. VIEWING TOPOLOGY SUFFIXES USING THE CLI

In a replication agreement, topology suffixes store the data that is replicated. You can view topology suffixes using the CLI.

Procedure

- Enter the **ipa topologysuffix-find** command to display a list of topology suffixes:

```
$ ipa topologysuffix-find
----------------------------
2 topology suffixes matched
----------------------------
  Suffix name: ca
  Managed LDAP suffix DN: o=ipaca

  Suffix name: domain
  Managed LDAP suffix DN: dc=example,dc=com
----------------------------
Number of entries returned 2
----------------------------
```

Additional resources

- Topology suffixes

1.6. VIEWING TOPOLOGY SEGMENTS USING THE CLI

In a replication agreement, when two replicas have a replication agreement between their suffixes, the suffixes form a topology segments. You can view topology segments using the CLI.

Procedure

1. Enter the **ipa topologysegment-find** command to show the current topology segments configured for the domain or CA suffixes. For example, for the domain suffix:

```
$ ipa topologysegment-find
Suffix name: domain
----------------
1 segment matched
----------------
  Segment name: server1.example.com-to-server2.example.com
  Left node: server1.example.com
  Right node: server2.example.com
  Connectivity: both
----------------------------
Number of entries returned 1
----------------------------
```

   In this example, domain-related data is only replicated between two servers: **server1.example.com** and **server2.example.com**.

2. Optional: To display details for a particular segment only, enter the **ipa topologysegment-show** command:

```
$ ipa topologysegment-show
Suffix name: domain
Segment name: server1.example.com-to-server2.example.com
  Segment name: server1.example.com-to-server2.example.com
  Left node: server1.example.com
  Right node: server2.example.com
  Connectivity: both
```

Additional resources

- Topology segments

1.7. SETTING UP REPLICATION BETWEEN TWO SERVERS USING THE WEB UI

Using the Identity Management (IdM) Web UI, you can choose two servers and create a new replication agreement between them.

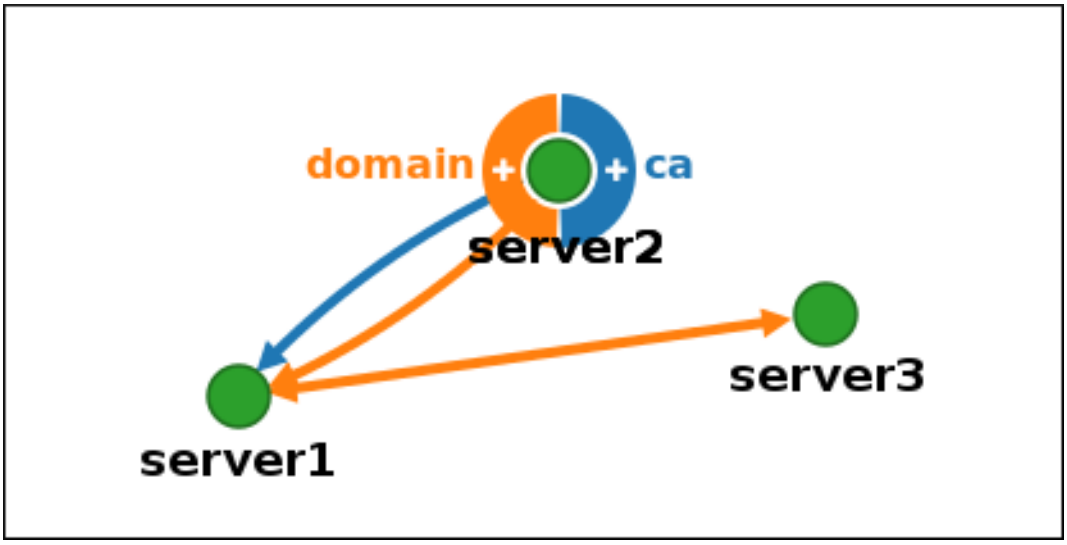
Prerequisites

- You are logged in as an IdM administrator.

Procedure

1. In the topology graph, hover your mouse over one of the server nodes.

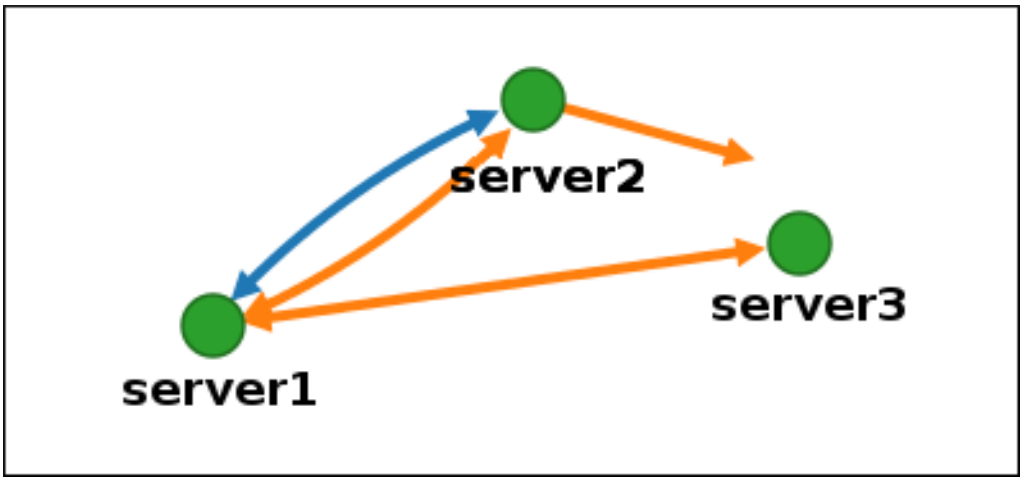
   Figure 1.6. Domain or CA options



2. Click on the **domain** or the **ca** part of the circle depending on what type of topology segment you want to create.

3. A new arrow representing the new replication agreement appears under your mouse pointer. Move your mouse to the other server node, and click on it.

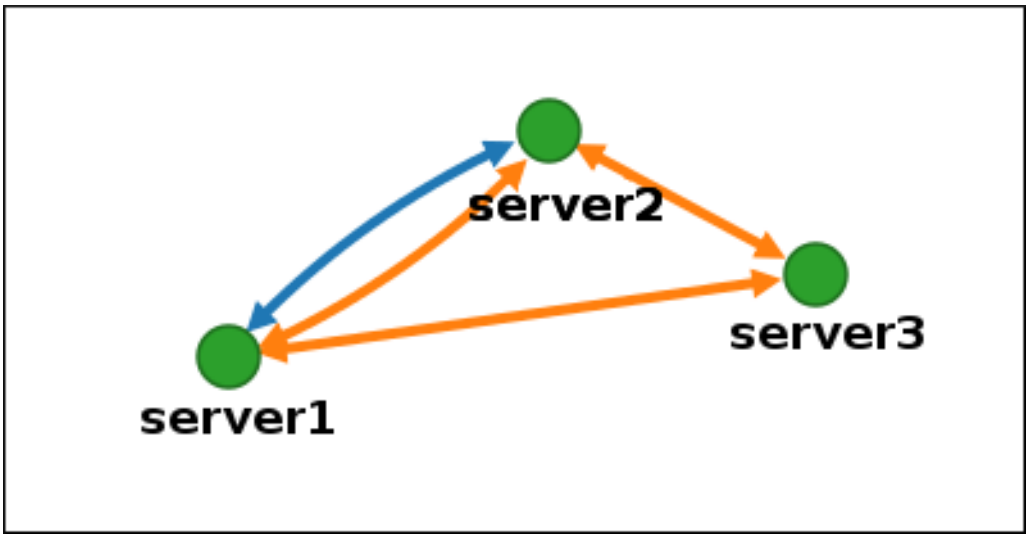
   Figure 1.7. Creating a new segment



4. In the **Add topology segment** window, click **Add** to confirm the properties of the new segment.

The new topology segment between the two servers joins them in a replication agreement. The topology graph now shows the updated replication topology:

Figure 1.8. New segment created



1.8. STOPPING REPLICATION BETWEEN TWO SERVERS USING THE WEB UI

Using the Identity Management (IdM) Web UI, you can remove a replication agreement from servers.

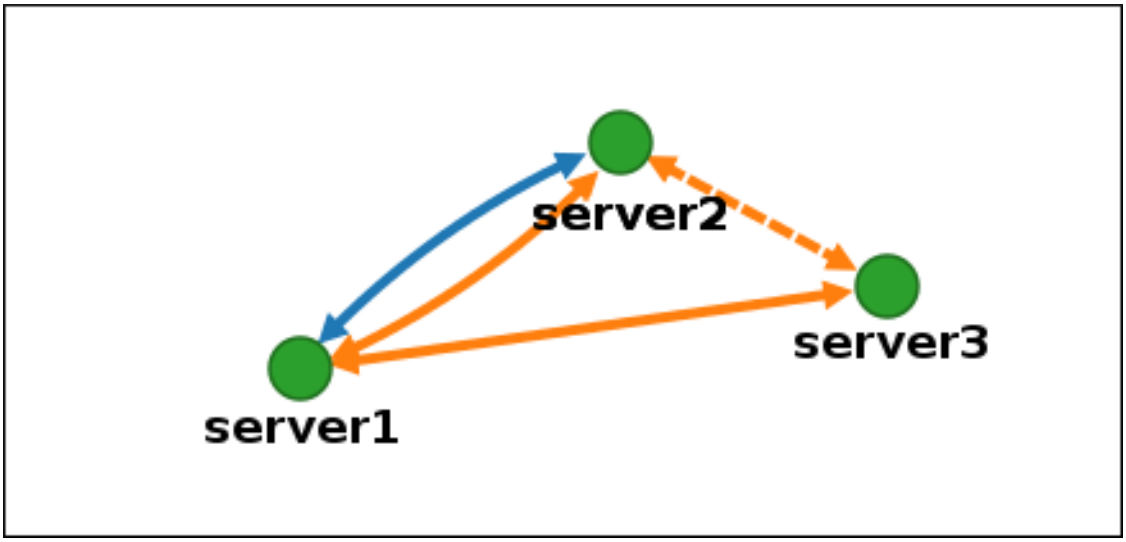
Prerequisites

- You are logged in as an IdM administrator.

Procedure

1. Click on an arrow representing the replication agreement you want to remove. This highlights the arrow.

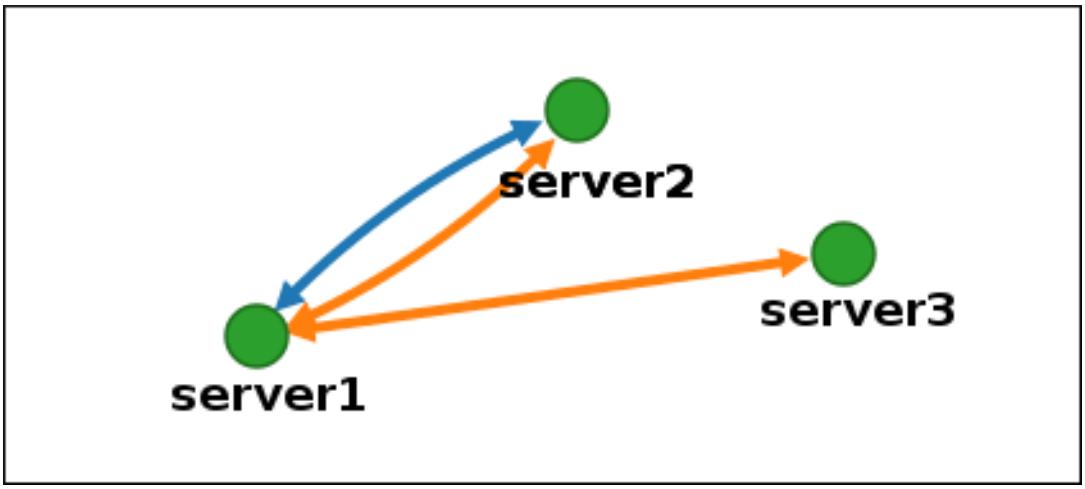
   Figure 1.9. Topology segment highlighted

2. Click **Delete**.

3. In the **Confirmation** window, click **OK**.
   IdM removes the topology segment between the two servers, which deletes their replication agreement. The topology graph now shows the updated replication topology:

Figure 1.10. Topology segment deleted



1.9. SETTING UP REPLICATION BETWEEN TWO SERVERS USING THE CLI

You can configure replication agreements between two servers using the **ipa topologysegment-add** command.

Prerequisites

- You have the IdM administrator credentials.

Procedure

- Create a topology segment for the two servers. When prompted, provide:
  - The required topology suffix: **domain** or **ca**
  - The left node and the right node, representing the two servers
  - Optional: A custom name for the segment
    For example:

```
$ ipa topologysegment-add
Suffix name: domain
Left node: server1.example.com
Right node: server2.example.com
Segment name [server1.example.com-to-server2.example.com]: new_segment
----------------------------
Added segment "new_segment"
----------------------------
  Segment name: new_segment
  Left node: server1.example.com
  Right node: server2.example.com
  Connectivity: both
```

    Adding the new segment joins the servers in a replication agreement.

Verification

- Verify that the new segment is configured:

```
$ ipa topologysegment-show
Suffix name: domain
Segment name: new_segment
  Segment name: new_segment
  Left node: server1.example.com
  Right node: server2.example.com
  Connectivity: both
```

1.10. STOPPING REPLICATION BETWEEN TWO SERVERS USING THE CLI

You can terminate replication agreements from command line using the **ipa topology segment-del** command.

Prerequisites

- You have the IdM administrator credentials.

Procedure

1. Optional: If you do not know the name of the specific replication segment that you want to remove, display all segments available. Use the **ipa topologysegment-find** command. When prompted, provide the required topology suffix: **domain** or **ca**. For example:

```
$ ipa topologysegment-find
Suffix name: domain
------------------
8 segments matched
------------------
  Segment name: new_segment
  Left node: server1.example.com
  Right node: server2.example.com
  Connectivity: both

...

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

   Locate the required segment in the output.

2. Remove the topology segment joining the two servers:

```
$ ipa topologysegment-del
Suffix name: domain
Segment name: new_segment
----------------------------
Deleted segment "new_segment"
----------------------------
```

   Deleting the segment removes the replication agreement.

Verification

- Verify that the segment is no longer listed:

```
$ ipa topologysegment-find
Suffix name: domain
------------------
7 segments matched
------------------
  Segment name: server2.example.com-to-server3.example.com
  Left node: server2.example.com
  Right node: server3.example.com
  Connectivity: both

...

----------------------------
Number of entries returned 7
----------------------------
```

1.11. REMOVING SERVER FROM TOPOLOGY USING THE WEB UI

You can use Identity Management (IdM) web interface to remove a server from the topology. This action does not uninstall the server components from the host.

Prerequisites

- You are logged in as an IdM administrator.
- The server you want to remove is **not** the only server connecting other servers with the rest of the topology; this would cause the other servers to become isolated, which is not allowed.
- The server you want to remove is **not** your last CA or DNS server.

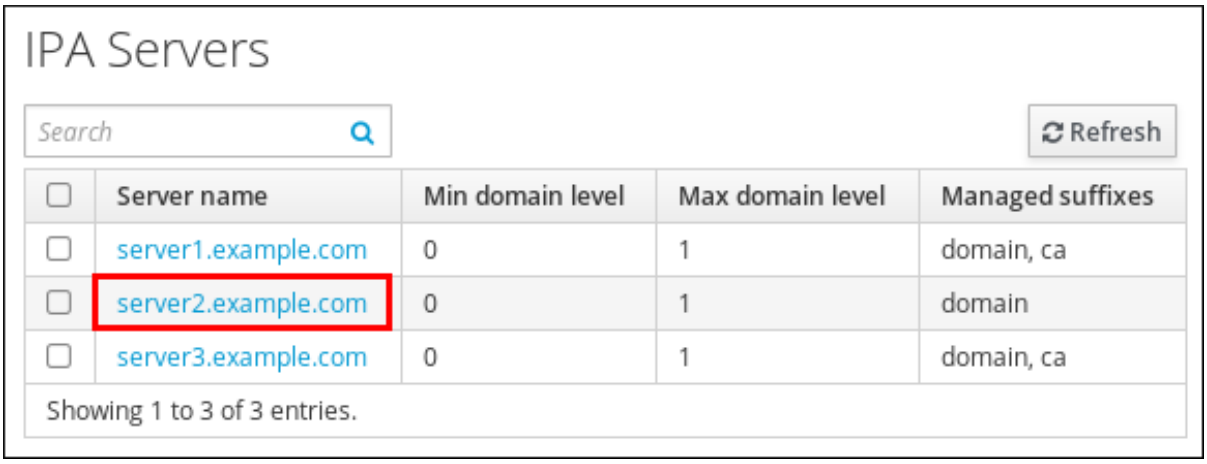
WARNING

Removing a server is an irreversible action. If you remove a server, the only way to introduce it back into the topology is to install a new replica on the machine.

Procedure

1. Select **IPA Server → Topology → IPA Servers**.
2. Click on the name of the server you want to delete.

Figure 1.11. Selecting a server



3. Click **Delete Server**.

Additional resources

- Uninstalling an IdM server

1.12. REMOVING SERVER FROM TOPOLOGY USING THE CLI

You can use the command line to remove an Identity Management (IdM) server from the topology.

Prerequisites

- You have the IdM administrator credentials.
- The server you want to remove is **not** the only server connecting other servers with the rest of the topology; this would cause the other servers to become isolated, which is not allowed.
- The server you want to remove is **not** your last CA or DNS server.

IMPORTANT

Removing a server is an irreversible action. If you remove a server, the only way to introduce it back into the topology is to install a new replica on the machine.

Procedure

To remove **server1.example.com**:

1. On another server, run the **ipa server-del** command to remove **server1.example.com**. The command removes all topology segments pointing to the server:

```
[user@server2 ~]$ ipa server-del
Server name: server1.example.com
Removing server1.example.com from replication topology, please wait...
-----------------------------------------------------------
Deleted IPA server "server1.example.com"
-----------------------------------------------------------
```

2. Optional: On **server1.example.com**, run the **ipa server-install --uninstall** command to uninstall the server components from the machine.

```
[root@server1 ~]# ipa server-install --uninstall
```

1.13. REMOVING OBSOLETE RUV RECORDS

If you remove a server from the IdM topology without properly removing its replication agreements, obsolete replica update vector (RUV) records will remain on one or more remaining servers in the topology. This can happen, for example, due to automation. These servers will then expect to receive updates from the now removed server. In this case, you need to clean the obsolete RUV records from the remaining servers.

Prerequisites

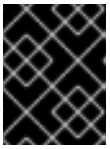
- You have the IdM administrator credentials.
- You know which replicas are corrupted or have been improperly removed.

Procedure

1. List the details about RUVs using the **ipa-replica-manage list-ruv** command. The command displays the replica IDs:

```
$ ipa-replica-manage list-ruv

server1.example.com:389: 6
server2.example.com:389: 5
server3.example.com:389: 4
server4.example.com:389: 12
```



IMPORTANT

The **ipa-replica-manage list-ruv** command lists ALL replicas in the topology, not only the malfunctioning or improperly removed ones.

2. Remove obsolete RUVs associated with a specified replica using the **ipa-replica-manage clean-ruv** command. Repeat the command for every replica ID with obsolete RUVs. For example, if you know **server1.example.com** and **server2.example.com** are the malfunctioning or improperly removed replicas:

```
ipa-replica-manage clean-ruv 6
ipa-replica-manage clean-ruv 5
```

WARNING

Proceed with extreme caution when using **ipa-replica-manage clean-ruv**. Running the command against a valid replica ID will corrupt all the data associated with that replica in the replication database.

If this happens, re-initialize the replica from another replica using **$ ipa-replica-manage re-initialize --from server1.example.com**.

Verification

1. Run **ipa-replica-manage list-ruv** again. If the command no longer displays any corrupt RUVs, the records have been successfully cleaned.
2. If the command still displays corrupt RUVs, clear them manually using this task:

```
dn: cn=clean replica_ID, cn=cleanallruv, cn=tasks, cn=config
objectclass: extensibleObject
replica-base-dn: dc=example,dc=com
replica-id: replica_ID
replica-force-cleaning: no
cn: clean replica_ID
```

1.14. VIEWING AVAILABLE SERVER ROLES IN THE IDM TOPOLOGY USING THE IDM WEB UI

Based on the services installed on an IdM server, it can perform various *server roles*. For example:

- CA server
- DNS server
- Key recovery authority (KRA) server.

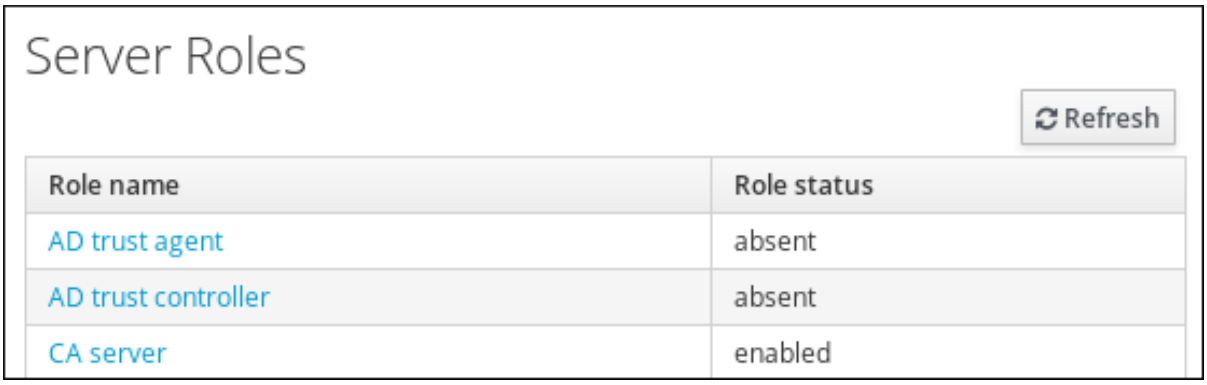
Procedure

- For a complete list of the supported server roles, see **IPA Server → Topology → Server Roles**.

NOTE

- Role status **absent** means that no server in the topology is performing the role.
- Role status **enabled** means that one or more servers in the topology are performing the role.

Figure 1.12. Server roles in the web UI



1.15. VIEWING AVAILABLE SERVER ROLES IN THE IDM TOPOLOGY USING THE IDM CLI

Based on the services installed on an IdM server, it can perform various *server roles*. For example:

- CA server
- DNS server
- Key recovery authority (KRA) server.

Procedure

- To display all CA servers in the topology and the current CA renewal server:

```
$ ipa config-show
  ...
  IPA masters: server1.example.com, server2.example.com, server3.example.com
  IPA CA servers: server1.example.com, server2.example.com
  IPA CA renewal master: server1.example.com
```

- Alternatively, to display a list of roles enabled on a particular server, for example *server.example.com*:

```
$ ipa server-show
Server name: server.example.com
  ...
  Enabled server roles: CA server, DNS server, KRA server
```

- Alternatively, use the **ipa server-find --servrole** command to search for all servers with a particular server role enabled. For example, to search for all CA servers:

```
$ ipa server-find --servrole "CA server"
----------------------
2 IPA servers matched
----------------------
  Server name: server1.example.com
  ...

  Server name: server2.example.com
  ...
```

```
----------------------------
Number of entries returned 2
----------------------------
```

1.16. PROMOTING A REPLICA TO A CA RENEWAL SERVER AND CRL PUBLISHER SERVER

If your IdM deployment uses an embedded certificate authority (CA), one of the IdM CA servers acts as the CA renewal server, a server that manages the renewal of CA subsystem certificates. One of the IdM CA servers also acts as the IdM CRL publisher server, a server that generates certificate revocation lists.

By default, the CA renewal server and CRL publisher server roles are installed on the first server on which the system administrator installed the CA role using the **ipa-server-install** or **ipa-ca-install** command. You can, however, transfer either of the two roles to any other IdM server on which the CA role is enabled.

Prerequisites

- You have the IdM administrator credentials.

1.17. DEMOTING OR PROMOTING HIDDEN REPLICAS

Procedure

After a replica has been installed, you can configure whether the replica is hidden or visible.

For details about hidden replicas, see The hidden replica mode.

Prerequisites

- Ensure that the replica is not the DNSSEC key master. If it is, move the service to another replica before making this replica hidden.
- Ensure that the replica is not a CA renewal server. If it is, move the service to another replica before making this replica hidden. For details, see

Procedure

- To hide a replica:

```
# ipa server-state replica.idm.example.com --state=hidden
```

- To make a replica visible again:

```
# ipa server-state replica.idm.example.com --state=enabled
```

- To view a list of all the hidden replicas in your topology:

```
# ipa config-show
```

  If all of your replicas are enabled, the command output does not mention hidden replicas.

CHAPTER 2. PREPARING YOUR ENVIRONMENT FOR MANAGING IDM USING ANSIBLE PLAYBOOKS

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.

Using 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** and **ipabackup 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:

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

[us]
replica.idm.example.com

[ipaserver:children]
eu
us
```

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
```

These commands require that you enter the IdM **admin** password.

Additional resources

- Installing an Identity Management server using an Ansible playbook
- How to build your inventory

CHAPTER 3. USING ANSIBLE TO MANAGE THE REPLICATION TOPOLOGY IN IDM

You can maintain multiple Identity Management (IdM) servers and let them replicate each other for redundancy purposes to mitigate or prevent server loss. For example, if one server fails, the other servers keep providing services to the domain. You can also recover the lost server by creating a new replica based on one of the remaining servers.

Data stored on an IdM server is replicated based on replication agreements: when two servers have a replication agreement configured, they share their data. The data that is replicated is stored in the topology **suffixes**. When two replicas have a replication agreement between their suffixes, the suffixes form a topology **segment**.

This chapter describes how to use Ansible to manage IdM replication agreements, topology segments, and topology suffixes.

3.1. USING ANSIBLE TO ENSURE A REPLICATION AGREEMENT EXISTS IN IDM

Data stored on an Identity Management (IdM) server is replicated based on replication agreements: when two servers have a replication agreement configured, they share their data. Replication agreements are always bilateral: the data is replicated from the first replica to the other one as well as from the other replica to the first one.

Follow this procedure to use an Ansible playbook to ensure that a replication agreement of the **domain** type exists between **server.idm.example.com** and **replica.idm.example.com**.

Prerequisites

- Ensure that you understand the recommendations for designing your IdM topology listed in Guidelines for connecting IdM replicas in a topology .
- 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.
  - The example assumes that in the ***~/MyPlaybooks/*** directory, you have created an Ansible inventory file with the fully-qualified domain name (FQDN) of the IdM server.
  - The example assumes that the **secret.yml** Ansible vault stores your **ipaadmin_password** and that you have access to a file that stores the password protecting the **secret.yml** file.
- The target node, that is the node on which the **ansible-freeipa** module is executed, is part of the IdM domain as an IdM client, server or replica.

Procedure

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

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

2. Copy the **add-topologysegment.yml** Ansible playbook file provided by the **ansible-freeipa** package:

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/topology/add-topologysegment.yml
add-topologysegment-copy.yml
```

3. Open the **add-topologysegment-copy.yml** file for editing.

4. Adapt the file by setting the following variables in the **ipatopologysegment** task section:

   - Indicate that the value of the **ipaadmin_password** variable is defined in the **secret.yml** Ansible vault file.
   - Set the **suffix** variable to either **domain** or **ca**, depending on what type of segment you want to add.
   - Set the **left** variable to the name of the IdM server that you want to be the left node of the replication agreement.
   - Set the **right** variable to the name of the IdM server that you want to be the right node of the replication agreement.
   - Ensure that the **state** variable is set to **present**.

   This is the modified Ansible playbook file for the current example:

```
---
- name: Playbook to handle topologysegment
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
- name: Add topology segment
    ipatopologysegment:
      ipaadmin_password: "{{ ipaadmin_password }}"
      suffix: domain
      left: server.idm.example.com
      right: replica.idm.example.com
      state: present
```

5. Save the file.

6. Run the Ansible playbook. Specify the playbook file, the file storing the password protecting the **secret.yml** file, and the inventory file:

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory add-
topologysegment-copy.yml
```

Additional resources

- Explaining Replication Agreements, Topology Suffixes, and Topology Segments
- **/usr/share/doc/ansible-freeipa/README-topology.md**

- Sample playbooks in **/usr/share/doc/ansible-freeipa/playbooks/topology**

3.2. USING ANSIBLE TO ENSURE REPLICATION AGREEMENTS EXIST BETWEEN MULTIPLE IDM REPLICAS

Data stored on an Identity Management (IdM) server is replicated based on replication agreements: when two servers have a replication agreement configured, they share their data. Replication agreements are always bilateral: the data is replicated from the first replica to the other one as well as from the other replica to the first one.

Follow this procedure to ensure replication agreements exist between multiple pairs of replicas in IdM.

Prerequisites

- Ensure that you understand the recommendations for designing your IdM topology listed in Connecting the replicas in a topology .
- 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.
  - The example assumes that in the ***~/MyPlaybooks/*** directory, you have created an Ansible inventory file with the fully-qualified domain name (FQDN) of the IdM server.
  - The example assumes that the **secret.yml** Ansible vault stores your **ipaadmin_password** and that you have access to a file that stores the password protecting the **secret.yml** file.
- The target node, that is the node on which the **ansible-freeipa** module is executed, is part of the IdM domain as an IdM client, server or replica.

Procedure

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

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

2. Copy the **add-topologysegments.yml** Ansible playbook file provided by the **ansible-freeipa** package:

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/topology/add-topologysegments.yml
add-topologysegments-copy.yml
```

3. Open the **add-topologysegments-copy.yml** file for editing.
4. Adapt the file by setting the following variables in the **vars** section:
   - Indicate that the value of the **ipaadmin_password** variable is defined in the **secret.yml** Ansible vault file.
   - For every topology segment, add a line in the **ipatopology_segments** section and set the following variables:

- Set the **suffix** variable to either **domain** or **ca**, depending on what type of segment you want to add.
- Set the **left** variable to the name of the IdM server that you want to be the left node of the replication agreement.
- Set the **right** variable to the name of the IdM server that you want to be the right node of the replication agreement.

5. In the **tasks** section of the **add-topologysegments-copy.yml** file, ensure that the **state** variable is set to **present**.
   This is the modified Ansible playbook file for the current example:

```
---
- name: Add topology segments
  hosts: ipaserver
  gather_facts: false

  vars:
    ipaadmin_password: "{{ ipaadmin_password }}"
    ipatopology_segments:
    - {suffix: domain, left: replica1.idm.example.com , right: replica2.idm.example.com }
    - {suffix: domain, left: replica2.idm.example.com , right: replica3.idm.example.com }
    - {suffix: domain, left: replica3.idm.example.com , right: replica4.idm.example.com }
    - {suffix: domain+ca, left: replica4.idm.example.com , right: replica1.idm.example.com }

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Add topology segment
    ipatopologysegment:
      ipaadmin_password: "{{ ipaadmin_password }}"
      suffix: "{{ item.suffix }}"
      name: "{{ item.name | default(omit) }}"
      left: "{{ item.left }}"
      right: "{{ item.right }}"
      state: present
    loop: "{{ ipatopology_segments | default([]) }}"
```

6. Save the file.

7. Run the Ansible playbook. Specify the playbook file, the file storing the password protecting the **secret.yml** file, and the inventory file:

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory add-topologysegments-copy.yml
```

Additional resources

- Explaining Replication Agreements, Topology Suffixes, and Topology Segments
- **/usr/share/doc/ansible-freeipa/README-topology.md**
- Sample playbooks in **/usr/share/doc/ansible-freeipa/playbooks/topology**

3.3. USING ANSIBLE TO CHECK IF A REPLICATION AGREEMENT EXISTS BETWEEN TWO REPLICAS

Data stored on an Identity Management (IdM) server is replicated based on replication agreements: when two servers have a replication agreement configured, they share their data. Replication agreements are always bilateral: the data is replicated from the first replica to the other one as well as from the other replica to the first one.

Follow this procedure to verify that replication agreements exist between multiple pairs of replicas in IdM. In contrast to Using Ansible to ensure a replication agreement exists in IdM , this procedure does not modify the existing configuration.

Prerequisites

- Ensure that you understand the recommendations for designing your Identity Management (IdM) topology listed in Connecting the replicas in a topology .
- 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.
  - The example assumes that in the ***~/MyPlaybooks/*** directory, you have created an Ansible inventory file with the fully-qualified domain name (FQDN) of the IdM server.
  - The example assumes that the **secret.yml** Ansible vault stores your **ipaadmin_password** and that you have access to a file that stores the password protecting the **secret.yml** file.
- The target node, that is the node on which the **ansible-freeipa** module is executed, is part of the IdM domain as an IdM client, server or replica.

Procedure

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

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

2. Copy the **check-topologysegments.yml** Ansible playbook file provided by the **ansible-freeipa** package:

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/topology/check-topologysegments.yml
check-topologysegments-copy.yml
```

3. Open the **check-topologysegments-copy.yml** file for editing.
4. Adapt the file by setting the following variables in the **vars** section:
   - Indicate that the value of the **ipaadmin_password** variable is defined in the **secret.yml** Ansible vault file.
   - For every topology segment, add a line in the **ipatopology_segments** section and set the following variables:
     - Set the **suffix** variable to either **domain** or **ca**, depending on the type of segment you are adding.

- Set the **left** variable to the name of the IdM server that you want to be the left node of the replication agreement.
- Set the **right** variable to the name of the IdM server that you want to be the right node of the replication agreement.

5. In the **tasks** section of the **check-topologysegments-copy.yml** file, ensure that the **state** variable is set to **present**.
   This is the modified Ansible playbook file for the current example:

```
---
- name: Add topology segments
  hosts: ipaserver
  gather_facts: false

  vars:
    ipaadmin_password: "{{ ipaadmin_password }}"
    ipatopology_segments:
    - {suffix: domain, left: replica1.idm.example.com, right: replica2.idm.example.com }
    - {suffix: domain, left: replica2.idm.example.com , right: replica3.idm.example.com }
    - {suffix: domain, left: replica3.idm.example.com , right: replica4.idm.example.com }
    - {suffix: domain+ca, left: replica4.idm.example.com , right:
replica1.idm.example.com }

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Check topology segment
    ipatopologysegment:
      ipaadmin_password: "{{ ipaadmin_password }}"
      suffix: "{{ item.suffix }}"
      name: "{{ item.name | default(omit) }}"
      left: "{{ item.left }}"
      right: "{{ item.right }}"
      state: checked
    loop: "{{ ipatopology_segments | default([]) }}"
```

6. Save the file.

7. Run the Ansible playbook. Specify the playbook file, the file storing the password protecting the **secret.yml** file, and the inventory file:

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory check-
topologysegments-copy.yml
```

Additional resources

- Explaining Replication Agreements, Topology Suffixes, and Topology Segments
- **/usr/share/doc/ansible-freeipa/README-topology.md**
- Sample playbooks in **/usr/share/doc/ansible-freeipa/playbooks/topology**

3.4. USING ANSIBLE TO VERIFY THAT A TOPOLOGY SUFFIX EXISTS IN IDM

In the context of replication agreements in Identity Management (IdM), topology suffixes store the data that is replicated. IdM supports two types of topology suffixes: **domain** and **ca**. Each suffix represents a separate back end, a separate replication topology. When a replication agreement is configured, it joins two topology suffixes of the same type on two different servers.

The **domain** suffix contains all domain-related data, such as data about users, groups, and policies. The **ca** suffix contains data for the Certificate System component. It is only present on servers with a certificate authority (CA) installed.

Follow this procedure to use an Ansible playbook to ensure that a topology suffix exists in IdM. The example describes how to ensure that the **domain** suffix exists in IdM.

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

Procedure

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

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

2. Copy the **verify-topologysuffix.yml** Ansible playbook file provided by the **ansible-freeipa** package:

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/topology/ verify-topologysuffix.yml
verify-topologysuffix-copy.yml
```

3. Open the **verify-topologysuffix-copy.yml** Ansible playbook file for editing.
4. Adapt the file by setting the following variables in the **ipatopologysuffix** section:
   - Indicate that the value of the **ipaadmin_password** variable is defined in the **secret.yml** Ansible vault file.
   - Set the **suffix** variable to **domain**. If you are verifying the presence of the **ca** suffix, set the variable to **ca**.
   - Ensure that the **state** variable is set to **verified**. No other option is possible.

This is the modified Ansible playbook file for the current example:

```
---
- name: Playbook to handle topologysuffix
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Verify topology suffix
    ipatopologysuffix:
      ipaadmin_password: "{{ ipaadmin_password }}"
      suffix: domain
      state: verified
```

5. Save the file.

6. Run the Ansible playbook. Specify the playbook file, the file storing the password protecting the **secret.yml** file, and the inventory file:

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory verify-topologysuffix-copy.yml
```

Additional resources

- Explaining Replication Agreements, Topology Suffixes, and Topology Segments
- **/usr/share/doc/ansible-freeipa/README-topology.md**
- Sample playbooks in **/usr/share/doc/ansible-freeipa/playbooks/topology**

3.5. USING ANSIBLE TO REINITIALIZE AN IDM REPLICA

If a replica has been offline for a long period of time or its database has been corrupted, you can reinitialize it. Reinitialization refreshes the replica with an updated set of data. Reinitialization can, for example, be used if an authoritative restore from backup is required.

NOTE

In contrast to replication updates, during which replicas only send changed entries to each other, reinitialization refreshes the whole database.

The local host on which you run the command is the reinitialized replica. To specify the replica from which the data is obtained, use the **direction** option.

Follow this procedure to use an Ansible playbook to reinitialize the **domain** data on **replica.idm.example.com** from **server.idm.example.com**.

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.
- The example assumes that in the ***~/MyPlaybooks/*** directory, you have created an Ansible inventory file with the fully-qualified domain name (FQDN) of the IdM server.
- The example assumes that the **secret.yml** Ansible vault stores your **ipaadmin_password** and that you have access to a file that stores the password protecting the **secret.yml** file.

- The target node, that is the node on which the **ansible-freeipa** module is executed, is part of the IdM domain as an IdM client, server or replica.

Procedure

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

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

2. Copy the **reinitialize-topologysegment.yml** Ansible playbook file provided by the **ansible-freeipa** package:

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/topology/reinitialize-topologysegment.yml reinitialize-topologysegment-copy.yml
```

3. Open the **reinitialize-topologysegment-copy.yml** file for editing.
4. Adapt the file by setting the following variables in the **ipatopologysegment** section:
   - Indicate that the value of the **ipaadmin_password** variable is defined in the **secret.yml** Ansible vault file.
   - Set the **suffix** variable to **domain**. If you are reinitializing the **ca** data, set the variable to **ca**.
   - Set the **left** variable to the left node of the replication agreement.
   - Set the **right** variable to the right node of the replication agreement.
   - Set the **direction** variable to the direction of the reinitializing data. The **left-to-right** direction means that data flows from the left node to the right node.
   - Ensure that the **state** variable is set to **reinitialized**.
     This is the modified Ansible playbook file for the current example:

```
---
- name: Playbook to handle topologysegment
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Reinitialize topology segment
    ipatopologysegment:
      ipaadmin_password: "{{ ipaadmin_password }}"
      suffix: domain
      left: server.idm.example.com
```

```
right: replica.idm.example.com
direction: left-to-right
state: reinitialized
```

5. Save the file.

6. Run the Ansible playbook. Specify the playbook file, the file storing the password protecting the **secret.yml** file, and the inventory file:

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory reinitialize-topologysegment-copy.yml
```

Additional resources

- Explaining Replication Agreements, Topology Suffixes, and Topology Segments
- **/usr/share/doc/ansible-freeipa/README-topology.md**
- Sample playbooks in **/usr/share/doc/ansible-freeipa/playbooks/topology**

3.6. USING ANSIBLE TO ENSURE A REPLICATION AGREEMENT IS ABSENT IN IDM

Data stored on an Identity Management (IdM) server is replicated based on replication agreements: when two servers have a replication agreement configured, they share their data. Replication agreements are always bilateral: the data is replicated from the first replica to the other one as well as from the other replica to the first one.

Follow this procedure to ensure a replication agreement between two replicas does not exist in IdM. The example describes how to ensure a replication agreement of the **domain** type does not exist between the **replica01.idm.example.com** and **replica02.idm.example.com** IdM servers.

Prerequisites

- You understand the recommendations for designing your IdM topology listed in Connecting the replicas in a topology.
- 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.
  - The example assumes that in the ***~/MyPlaybooks/*** directory, you have created an Ansible inventory file with the fully-qualified domain name (FQDN) of the IdM server.
  - The example assumes that the **secret.yml** Ansible vault stores your **ipaadmin_password** and that you have access to a file that stores the password protecting the **secret.yml** file.
- The target node, that is the node on which the **ansible-freeipa** module is executed, is part of the IdM domain as an IdM client, server or replica.

Procedure

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

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

2. Copy the **delete-topologysegment.yml** Ansible playbook file provided by the **ansible-freeipa** package:

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/topology/delete-topologysegment.yml
delete-topologysegment-copy.yml
```

3. Open the **delete-topologysegment-copy.yml** file for editing.

4. Adapt the file by setting the following variables in the **ipatopologysegment** task section:

   - Indicate that the value of the **ipaadmin_password** variable is defined in the **secret.yml** Ansible vault file.
   - Set the **suffix** variable to **domain**. Alternatively, if you are ensuring that the **ca** data are not replicated between the left and right nodes, set the variable to **ca**.
   - Set the **left** variable to the name of the IdM server that is the left node of the replication agreement.
   - Set the **right** variable to the name of the IdM server that is the right node of the replication agreement.
   - Ensure that the **state** variable is set to **absent**.

   This is the modified Ansible playbook file for the current example:

```
---
- name: Playbook to handle topologysegment
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
- name: Delete topology segment
    ipatopologysegment:
      ipaadmin_password: "{{ ipaadmin_password }}"
      suffix: domain
      left: replica01.idm.example.com
      right: replica02.idm.example.com:
      state: absent
```

5. Save the file.

6. Run the Ansible playbook. Specify the playbook file, the file storing the password protecting the **secret.yml** file, and the inventory file:

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory delete-
topologysegment-copy.yml
```

Additional resources

- Explaining Replication Agreements, Topology Suffixes, and Topology Segments

- **/usr/share/doc/ansible-freeipa/README-topology.md**
- Sample playbooks in **/usr/share/doc/ansible-freeipa/playbooks/topology**

3.7. ADDITIONAL RESOURCES

- Planning the replica topology.
- Installing an IdM replica .

CHAPTER 4. DEMOTING OR PROMOTING HIDDEN REPLICAS

After a replica has been installed, you can configure whether the replica is hidden or visible.

For details about hidden replicas, see The hidden replica mode.

Prerequisites

- Ensure that the replica is not the DNSSEC key master. If it is, move the service to another replica before making this replica hidden.
- Ensure that the replica is not a CA renewal server. If it is, move the service to another replica before making this replica hidden. For details, see

Procedure

- To hide a replica:

```
# ipa server-state replica.idm.example.com --state=hidden
```

- To make a replica visible again:

```
# ipa server-state replica.idm.example.com --state=enabled
```

- To view a list of all the hidden replicas in your topology:

```
# ipa config-show
```

  If all of your replicas are enabled, the command output does not mention hidden replicas.

CHAPTER 5. CHECKING IDM REPLICATION USING HEALTHCHECK

You can test Identity Management (IdM) replication using the Healthcheck tool.

Prerequisites

- You are using RHEL version 8.1 or newer.

5.1. REPLICATION HEALTHCHECK TESTS

The Healthcheck tool tests the Identity Management (IdM) topology configuration and searches for replication conflict issues.

To list all tests, run the **ipa-healthcheck** with the **--list-sources** option:

```
# ipa-healthcheck --list-sources
```

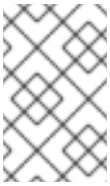
The topology tests are placed under the **ipahealthcheck.ipa.topology** and **ipahealthcheck.ds.replication** sources:

IPATopologyDomainCheck

This test verifies:

- That no single server is disconnected from the topology.
- That servers do not have more than the recommended number of replication agreements.

If the test succeeds, the test returns the configured domains. Otherwise, specific connection errors are reported.



NOTE

The test runs the **ipa topologysuffix-verify** command for the **domain** suffix. It also runs the command for the **ca** suffix if the IdM Certificate Authority server role is configured on this server.

ReplicationConflictCheck

The test searches for entries in LDAP matching **(&(!(objectclass=nstombstone))(nsds5ReplConflict=*))**.

NOTE

Run these tests on all IdM servers when trying to check for issues.

Additional resources

- Solving common replication problems

5.2. SCREENING REPLICATION USING HEALTHCHECK

Follow this procedure to run a standalone manual test of an Identity Management (IdM) replication topology and configuration using the Healthcheck tool.

The Healthcheck tool includes many tests. Therefore, you can shorten the results with:

- Replication conflict test: **--source=ipahealthcheck.ds.replication**
- Correct topology test: **--source=ipahealthcheck.ipa.topology**

Prerequisites

- You are logged in as the **root** user.

Procedure

- To run Healthcheck replication conflict and topology checks, enter:

```
# ipa-healthcheck --source=ipahealthcheck.ds.replication --
source=ipahealthcheck.ipa.topology
```

  Four different results are possible:

  - SUCCESS – the test passed successfully.

```
{
  "source": "ipahealthcheck.ipa.topology",
  "check": "IPATopologyDomainCheck",
  "result": "SUCCESS",
  "kw": {
    "suffix": "domain"
  }
}
```

  - WARNING – the test passed but there might be a problem.
  - ERROR – the test failed.

```
{
  "source": "ipahealthcheck.ipa.topology",
  "check": "IPATopologyDomainCheck",
  "result": "ERROR",
  "uuid": d6ce3332-92da-423d-9818-e79f49ed321f
  "when": 20191007115449Z
  "duration": 0.005943
  "kw": {
    "msg": "topologysuffix-verify domain failed, server2 is not connected
(server2_139664377356472 in MainThread)"
  }
}
```

  - CRITICAL – the test failed and it affects the IdM server functionality.

Additional resources

- **man ipa-healthcheck**

5.3. ADDITIONAL RESOURCES

- Healthcheck in IdM