



# Red Hat Enterprise Linux 8

## Installing Identity Management

Getting started using Identity Management



# Red Hat Enterprise Linux 8 Installing Identity Management

---

Getting started using Identity Management

## Legal Notice

Copyright © 2022 Red Hat, Inc.

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

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

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

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

## Abstract

This documentation collection provides instructions on how to install Identity Management on Red Hat Enterprise Linux 8 (RHEL) and how to upgrade to it from RHEL 7.

## Table of Contents

<b>MAKING OPEN SOURCE MORE INCLUSIVE</b>	<b>8</b>
<b>PROVIDING FEEDBACK ON RED HAT DOCUMENTATION</b>	<b>9</b>
<b>PART I. INSTALLING IDENTITY MANAGEMENT</b>	<b>10</b>
<b>CHAPTER 1. PREPARING THE SYSTEM FOR IDM SERVER INSTALLATION</b>	<b>11</b>
1.1. AUTHORIZATION REQUIREMENTS WHEN INSTALLING AN IDM SERVER	11
1.2. HARDWARE RECOMMENDATIONS	11
1.3. CUSTOM CONFIGURATION REQUIREMENTS FOR IDM	11
1.4. TIME SERVICE REQUIREMENTS FOR IDM	14
1.4.1. How IdM uses chronyd for synchronization	14
1.4.2. List of NTP configuration options for IdM installation commands	15
1.4.3. Ensuring IdM can reference your NTP time server	15
1.4.4. Additional resources	16
1.5. HOST NAME AND DNS REQUIREMENTS FOR IDM	16
1.6. PORT REQUIREMENTS FOR IDM	19
1.7. OPENING THE PORTS REQUIRED BY IDM	20
1.8. INSTALLING PACKAGES REQUIRED FOR AN IDM SERVER	21
<b>CHAPTER 2. INSTALLING AN IDM SERVER: WITH INTEGRATED DNS, WITH AN INTEGRATED CA AS THE ROOT CA</b>	<b>23</b>
2.1. INTERACTIVE INSTALLATION	23
2.2. NON-INTERACTIVE INSTALLATION	25
<b>CHAPTER 3. INSTALLING AN IDM SERVER: WITH INTEGRATED DNS, WITH AN EXTERNAL CA AS THE ROOT CA</b>	<b>27</b>
3.1. INTERACTIVE INSTALLATION	27
3.2. TROUBLESHOOTING: EXTERNAL CA INSTALLATION FAILS	30
What this means:	30
To fix the problem:	31
<b>CHAPTER 4. INSTALLING AN IDM SERVER: WITH INTEGRATED DNS, WITHOUT A CA</b>	<b>32</b>
4.1. CERTIFICATES REQUIRED TO INSTALL AN IDM SERVER WITHOUT A CA	32
4.2. INTERACTIVE INSTALLATION	33
<b>CHAPTER 5. INSTALLING AN IDM SERVER: WITHOUT INTEGRATED DNS, WITH AN INTEGRATED CA AS THE ROOT CA</b>	<b>37</b>
5.1. INTERACTIVE INSTALLATION	37
5.2. NON-INTERACTIVE INSTALLATION	38
5.3. IDM DNS RECORDS FOR EXTERNAL DNS SYSTEMS	39
<b>CHAPTER 6. INSTALLING AN IDM SERVER: WITHOUT INTEGRATED DNS, WITH AN EXTERNAL CA AS THE ROOT CA</b>	<b>41</b>
6.1. OPTIONS USED WHEN INSTALLING AN IDM CA WITH AN EXTERNAL CA AS THE ROOT CA	41
6.2. INTERACTIVE INSTALLATION	42
6.3. NON-INTERACTIVE INSTALLATION	44
6.4. IDM DNS RECORDS FOR EXTERNAL DNS SYSTEMS	46
<b>CHAPTER 7. INSTALLING AN IDM SERVER OR REPLICA WITH CUSTOM DATABASE SETTINGS FROM AN LDIF FILE</b>	<b>48</b>
<b>CHAPTER 8. OPTIONS FOR THE IPA-SERVER-INSTALL AND IPA-REPLICA-INSTALL COMMANDS</b>	<b>49</b>
<b>CHAPTER 9. TROUBLESHOOTING IDM SERVER INSTALLATION</b>	<b>51</b>

9.1. REVIEWING IDM SERVER INSTALLATION ERROR LOGS	51
9.2. REVIEWING IDM CA INSTALLATION ERRORS	52
9.3. REMOVING A PARTIAL IDM SERVER INSTALLATION	53
9.4. ADDITIONAL RESOURCES	54
<b>CHAPTER 10. UNINSTALLING AN IDM SERVER</b>	<b>55</b>
<b>CHAPTER 11. RENAMING AN IDM SERVER</b>	<b>58</b>
<b>CHAPTER 12. PREPARING THE SYSTEM FOR IDM CLIENT INSTALLATION</b>	<b>59</b>
12.1. DNS REQUIREMENTS FOR IDM CLIENTS	59
12.2. PORT REQUIREMENTS FOR IDM CLIENTS	59
12.3. IPV6 REQUIREMENTS FOR IDM CLIENTS	59
12.4. PACKAGES REQUIRED TO INSTALL AN IDM CLIENT	59
12.4.1. Installing IdM client packages from the idm:client stream	60
12.4.2. Installing IdM client packages from the idm:DL1 stream	60
<b>CHAPTER 13. INSTALLING AN IDM CLIENT: BASIC SCENARIO</b>	<b>61</b>
13.1. PREREQUISITES	61
13.2. INSTALLING A CLIENT BY USING USER CREDENTIALS: INTERACTIVE INSTALLATION	61
13.3. INSTALLING A CLIENT BY USING A ONE-TIME PASSWORD: INTERACTIVE INSTALLATION	63
13.4. INSTALLING A CLIENT: NON-INTERACTIVE INSTALLATION	64
13.5. REMOVING PRE-IDM CONFIGURATION AFTER INSTALLING A CLIENT	66
13.6. TESTING AN IDM CLIENT	66
13.7. CONNECTIONS PERFORMED DURING AN IDM CLIENT INSTALLATION	66
13.8. IDM CLIENT'S COMMUNICATIONS WITH THE SERVER DURING POST-INSTALLATION DEPLOYMENT	67
13.8.1. SSSD communication patterns	68
13.8.2. Certmonger communication patterns	69
<b>CHAPTER 14. INSTALLING AN IDM CLIENT WITH KICKSTART</b>	<b>71</b>
14.1. INSTALLING A CLIENT WITH KICKSTART	71
14.2. KICKSTART FILE FOR CLIENT INSTALLATION	71
14.3. TESTING AN IDM CLIENT	72
<b>CHAPTER 15. TROUBLESHOOTING IDM CLIENT INSTALLATION</b>	<b>73</b>
15.1. REVIEWING IDM CLIENT INSTALLATION ERRORS	73
15.2. RESOLVING ISSUES IF THE CLIENT INSTALLATION FAILS TO UPDATE DNS RECORDS	74
15.3. RESOLVING ISSUES IF THE CLIENT INSTALLATION FAILS TO JOIN THE IDM KERBEROS REALM	74
15.4. ADDITIONAL RESOURCES	75
<b>CHAPTER 16. RE-ENROLLING AN IDM CLIENT</b>	<b>76</b>
16.1. CLIENT RE-ENROLLMENT IN IDM	76
16.1.1. What happens during client re-enrollment	76
16.2. RE-ENROLLING A CLIENT BY USING USER CREDENTIALS: INTERACTIVE RE-ENROLLMENT	76
16.3. RE-ENROLLING A CLIENT BY USING THE CLIENT KEYTAB: NON-INTERACTIVE RE-ENROLLMENT	77
16.4. TESTING AN IDM CLIENT	77
<b>CHAPTER 17. UNINSTALLING AN IDM CLIENT</b>	<b>79</b>
17.1. UNINSTALLING AN IDM CLIENT	79
17.2. UNINSTALLING AN IDM CLIENT: ADDITIONAL STEPS AFTER MULTIPLE PAST INSTALLATIONS	80
<b>CHAPTER 18. RENAMING IDM CLIENT SYSTEMS</b>	<b>82</b>
18.1. PREPARING AN IDM CLIENT FOR ITS RENAMING	82
18.2. UNINSTALLING AN IDM CLIENT	83
18.3. UNINSTALLING AN IDM CLIENT: ADDITIONAL STEPS AFTER MULTIPLE PAST INSTALLATIONS	84

18.4. RENAMING THE HOST SYSTEM	85
18.5. RE-INSTALLING AN IDM CLIENT	85
18.6. RE-ADDING SERVICES, RE-GENERATING CERTIFICATES, AND RE-ADDING HOST GROUPS	85
<b>CHAPTER 19. PREPARING THE SYSTEM FOR IDM REPLICA INSTALLATION</b>	<b>86</b>
19.1. REPLICA VERSION REQUIREMENTS	86
19.2. METHODS FOR DISPLAYING IDM SOFTWARE VERSION	86
19.3. AUTHORIZING THE INSTALLATION OF A REPLICA ON AN IDM CLIENT	87
19.4. AUTHORIZING THE INSTALLATION OF A REPLICA ON A SYSTEM THAT IS NOT ENROLLED INTO IDM	88
<b>CHAPTER 20. INSTALLING AN IDM REPLICA</b>	<b>90</b>
20.1. INSTALLING AN IDM REPLICA WITH INTEGRATED DNS AND A CA	90
20.2. INSTALLING AN IDM REPLICA WITH INTEGRATED DNS AND NO CA	91
20.3. INSTALLING AN IDM REPLICA WITHOUT INTEGRATED DNS AND WITH A CA	92
20.4. INSTALLING AN IDM REPLICA WITHOUT INTEGRATED DNS AND WITHOUT A CA	93
20.5. INSTALLING AN IDM HIDDEN REPLICA	94
20.6. TESTING AN IDM REPLICA	94
20.7. CONNECTIONS PERFORMED DURING AN IDM REPLICA INSTALLATION	95
<b>CHAPTER 21. TROUBLESHOOTING IDM REPLICA INSTALLATION</b>	<b>96</b>
21.1. REVIEWING IDM REPLICA INSTALLATION ERRORS	96
21.2. REVIEWING IDM CA INSTALLATION ERRORS	98
21.3. REMOVING A PARTIAL IDM REPLICA INSTALLATION	99
21.4. RESOLVING INVALID CREDENTIAL ERRORS	100
21.5. ADDITIONAL RESOURCES	101
<b>CHAPTER 22. UNINSTALLING AN IDM REPLICA</b>	<b>102</b>
<b>CHAPTER 23. INSTALLING DNS ON AN EXISTING IDM SERVER</b>	<b>103</b>
<b>CHAPTER 24. MANAGING REPLICATION TOPOLOGY</b>	<b>105</b>
24.1. EXPLAINING REPLICATION AGREEMENTS, TOPOLOGY SUFFIXES AND TOPOLOGY SEGMENTS	105
Replication agreements	105
Topology suffixes	105
Topology segments	106
24.2. USING THE TOPOLOGY GRAPH TO MANAGE REPLICATION TOPOLOGY	108
Accessing the topology graph	108
Interpreting the topology graph	108
Customizing the topology view	109
24.3. SETTING UP REPLICATION BETWEEN TWO SERVERS USING THE WEB UI	110
24.4. STOPPING REPLICATION BETWEEN TWO SERVERS USING THE WEB UI	112
24.5. SETTING UP REPLICATION BETWEEN TWO SERVERS USING THE CLI	113
24.6. STOPPING REPLICATION BETWEEN TWO SERVERS USING THE CLI	114
24.7. REMOVING SERVER FROM TOPOLOGY USING THE WEB UI	115
24.8. REMOVING SERVER FROM TOPOLOGY USING THE CLI	116
24.9. VIEWING SERVER ROLES ON AN IDM SERVER USING THE WEB UI	117
24.10. VIEWING SERVER ROLES ON AN IDM SERVER USING THE CLI	117
24.11. PROMOTING A REPLICA TO A CA RENEWAL SERVER AND CRL PUBLISHER SERVER	118
24.12. DEMOTING OR PROMOTING HIDDEN REPLICAS	118
<b>CHAPTER 25. INSTALLING AND RUNNING THE IDM HEALTHCHECK TOOL</b>	<b>120</b>
25.1. HEALTHCHECK IN IDM	120
25.2. INSTALLING IDM HEALTHCHECK	121
25.3. RUNNING IDM HEALTHCHECK	121

25.4. ADDITIONAL RESOURCES	121
<b>CHAPTER 26. INSTALLING AN IDENTITY MANAGEMENT SERVER USING AN ANSIBLE PLAYBOOK ...</b>	<b>123</b>
26.1. ANSIBLE AND ITS ADVANTAGES FOR INSTALLING IDM	123
Advantages of using Ansible to install IdM	123
26.2. IDM SERVER INSTALLATION USING AN ANSIBLE PLAYBOOK	123
26.3. INSTALLING THE ANSIBLE-FREEIPA PACKAGE	124
26.4. ANSIBLE ROLES LOCATION IN THE FILE SYSTEM	124
26.5. DEPLOYING AN IDM SERVER WITH AN INTEGRATED CA AS THE ROOT CA USING AN ANSIBLE PLAYBOOK	125
26.5.1. Setting the parameters for a deployment with an integrated DNS and an integrated CA as the root CA	125
26.5.2. Setting the parameters for a deployment with external DNS and an integrated CA as the root CA	128
26.5.3. Deploying an IdM server with an integrated CA as the root CA using an Ansible playbook	130
26.6. DEPLOYING AN IDM SERVER WITH AN EXTERNAL CA AS THE ROOT CA USING AN ANSIBLE PLAYBOOK	131
26.6.1. Setting the parameters for a deployment with an integrated DNS and an external CA as the root CA	131
26.6.2. Setting the parameters for a deployment with external DNS and an external CA as the root CA	134
26.6.3. Deploying an IdM server with an external CA as the root CA using an Ansible playbook	136
26.7. ADDITIONAL RESOURCES	138
<b>CHAPTER 27. INSTALLING AN IDENTITY MANAGEMENT REPLICA USING AN ANSIBLE PLAYBOOK ...</b>	<b>139</b>
27.1. IDM REPLICA INSTALLATION USING AN ANSIBLE PLAYBOOK	139
27.2. SETTING THE PARAMETERS OF THE IDM REPLICA DEPLOYMENT	139
27.2.1. Specifying the base, server and client variables for installing the IdM replica	140
27.2.2. Specifying the credentials for installing the IdM replica using an Ansible playbook	143
27.3. DEPLOYING AN IDM REPLICA USING AN ANSIBLE PLAYBOOK	144
<b>CHAPTER 28. INSTALLING AN IDENTITY MANAGEMENT CLIENT USING AN ANSIBLE PLAYBOOK ....</b>	<b>145</b>
28.1. IDM CLIENT INSTALLATION USING AN ANSIBLE PLAYBOOK	145
28.2. SETTING THE PARAMETERS OF THE IDM CLIENT DEPLOYMENT	146
28.2.1. Setting the parameters of the inventory file for the autodiscovery client installation mode	146
28.2.2. Setting the parameters of the inventory file when autodiscovery is not possible during client installation	148
28.2.3. Checking the parameters in the install-client.yml file	150
28.2.4. Authorization options for IdM client enrollment using an Ansible playbook	150
28.3. DEPLOYING AN IDM CLIENT USING AN ANSIBLE PLAYBOOK	152
28.4. TESTING AN IDENTITY MANAGEMENT CLIENT AFTER ANSIBLE INSTALLATION	153
28.5. UNINSTALLING AN IDM CLIENT USING AN ANSIBLE PLAYBOOK	153
<b>PART II. INTEGRATING IDM AND AD .....</b>	<b>155</b>
<b>CHAPTER 29. INSTALLING TRUST BETWEEN IDM AND AD .....</b>	<b>156</b>
29.1. SUPPORTED VERSIONS OF WINDOWS SERVER	156
29.2. HOW THE TRUST WORKS	157
29.3. AD ADMINISTRATION RIGHTS	157
29.4. ENSURING SUPPORT FOR COMMON ENCRYPTION TYPES IN AD AND RHEL	157
29.5. PORTS REQUIRED FOR COMMUNICATION BETWEEN IDM AND AD	158
29.6. CONFIGURING DNS AND REALM SETTINGS FOR A TRUST	161
29.6.1. Unique primary DNS domains	161
29.6.2. Configuring a DNS forward zone in the IdM Web UI	162
29.6.3. Configuring a DNS forward zone in the CLI	166
29.6.4. Configuring DNS forwarding in AD	167
29.6.5. Verifying the DNS configuration	168
29.7. CONFIGURING IDM CLIENTS IN AN ACTIVE DIRECTORY DNS DOMAIN	169



29.7.1. Configuring an IdM client without Kerberos single sign-on	169
29.7.2. Requesting SSL certificates without single sign-on	170
29.7.3. Configuring an IdM client with Kerberos single sign-on	170
29.7.4. Requesting SSL certificates with single sign-on	171
29.8. SETTING UP A TRUST	171
29.8.1. Preparing the IdM server for the trust	172
29.8.2. Setting up a trust agreement using the command line	174
29.8.3. Setting up a trust agreement in the IdM Web UI	175
29.8.4. Verifying the Kerberos configuration	178
29.8.5. Verifying the trust configuration on IdM	178
29.8.6. Verifying the trust configuration on AD	179
29.8.7. Creating a trust agent	181
29.8.8. Enabling automatic private group mapping for a POSIX ID range on the CLI	181
29.8.9. Enabling automatic private group mapping for a POSIX ID range in the IdM WebUI	182
29.9. REMOVING THE TRUST USING THE COMMAND LINE	184
29.10. REMOVING THE TRUST USING THE IDM WEB UI	184
<b>PART III. MIGRATING IDM FROM RHEL 7 TO RHEL 8 AND KEEPING IT UP-TO-DATE</b>	<b>186</b>
<b>CHAPTER 30. MIGRATING YOUR IDM ENVIRONMENT FROM RHEL 7 SERVERS TO RHEL 8 SERVERS</b>	<b>187</b>
30.1. PREREQUISITES FOR MIGRATING IDM FROM RHEL 7 TO 8	188
30.2. INSTALLING THE RHEL 8 REPLICA	189
30.3. ASSIGNING THE CA RENEWAL SERVER ROLE TO THE RHEL 8 IDM SERVER	192
30.4. STOPPING CRL GENERATION ON A RHEL 7 IDM CA SERVER	193
30.5. STARTING CRL GENERATION ON THE NEW RHEL 8 IDM CA SERVER	193
30.6. STOPPING AND DECOMMISSIONING THE RHEL 7 SERVER	194
<b>CHAPTER 31. UPDATING AND DOWNGRADING IDM</b>	<b>196</b>
<b>CHAPTER 32. UPGRADING AN IDM CLIENT FROM RHEL 7 TO RHEL 8</b>	<b>197</b>
32.1. UPDATING THE SSSD CONFIGURATION AFTER UPGRADING TO RHEL 8	197
32.1.1. Switching from the local ID provider to the files ID provider	197
32.1.2. Removing deprecated options	198
32.1.3. Enabling wildcard matching for sudo rules	198
32.2. LIST OF SSSD FUNCTIONALITY REMOVED IN RHEL 8	199
32.3. ADDITIONAL RESOURCES	200
<b>PART IV. MIGRATING TO IDM FROM EXTERNAL SOURCES</b>	<b>201</b>
<b>CHAPTER 33. MIGRATING FROM AN LDAP DIRECTORY TO IDM</b>	<b>202</b>
33.1. CONSIDERATIONS IN MIGRATING FROM LDAP TO IDM	202
33.2. PLANNING THE CLIENT CONFIGURATION WHEN MIGRATING FROM LDAP TO IDM	202
33.2.1. Initial, pre-migration client configuration	203
33.2.2. Recommended configuration for RHEL clients	203
33.2.3. Alternative supported configuration	204
33.3. PLANNING PASSWORD MIGRATION WHEN MIGRATING FROM LDAP TO IDM	205
33.3.1. Methods for migrating passwords when migrating LDAP to IdM	206
33.3.2. Planning the migration of cleartext LDAP passwords	206
33.3.3. Planning the migration of LDAP passwords that do not meet the IdM requirements	207
33.4. FURTHER MIGRATION CONSIDERATIONS AND REQUIREMENTS	207
33.4.1. LDAP servers supported for migration	207
33.4.2. LDAP environment requirements for migration	207
33.4.3. IdM system requirements for migration	208
33.4.4. Considerations about sudo rules	209
33.4.5. LDAP to IdM migration tools	209

33.4.6. Improving LDAP to IdM migration performance	209
33.4.7. LDAP to IdM migration sequence	209
33.5. CUSTOMIZING THE MIGRATION FROM LDAP TO IDM	210
33.5.1. Examples of customizing the Bind DN and Base DN during the migration from LDAP to IdM	210
33.5.2. The migration of specific subtrees	211
33.5.3. The inclusion and exclusion of entries	212
33.5.4. The exclusion of entry attributes	212
33.5.5. The schema to use when migrating from LDAP to IdM and the schema compat feature	213
33.6. MIGRATING AN LDAP SERVER TO IDM	213
33.7. MIGRATING FROM LDAP TO IDM OVER SSL	217



## MAKING OPEN SOURCE MORE INCLUSIVE

Red Hat is committed to replacing problematic language in our code, documentation, and web properties. We are beginning with these four terms: master, slave, blacklist, and whitelist. Because of the enormity of this endeavor, these changes will be implemented gradually over several upcoming releases. For more details, see [our CTO Chris Wright's message](#).

In Identity Management, planned terminology replacements include:

- ***block list*** replaces *blacklist*
- ***allow list*** replaces *whitelist*
- ***secondary*** replaces *slave*
- The word *master* is being replaced with more precise language, depending on the context:
  - ***IdM server*** replaces *IdM master*
  - ***CA renewal server*** replaces *CA renewal master*
  - ***CRL publisher server*** replaces *CRL master*
  - ***multi-supplier*** replaces *multi-master*

# 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**.

## PART I. INSTALLING IDENTITY MANAGEMENT

# CHAPTER 1. PREPARING THE SYSTEM FOR IDM SERVER INSTALLATION

The following sections list the requirements to install an Identity Management (IdM) server. Before the installation, make sure your system meets these requirements.

## 1.1. AUTHORIZATION REQUIREMENTS WHEN INSTALLING AN IDM SERVER

You need **root** privileges to install an Identity Management (IdM) server on your host.

## 1.2. HARDWARE RECOMMENDATIONS

RAM is the most important hardware feature to size properly. Make sure your system has enough RAM available. Typical RAM requirements are:

- For 10,000 users and 100 groups: at least 4 GB of RAM and 4 GB swap space
- For 100,000 users and 50,000 groups: at least 16 GB of RAM and 4 GB of swap space

For larger deployments, it is more effective to increase the RAM than to increase disk space because much of the data is stored in cache. In general, adding more RAM leads to better performance for larger deployments due to caching.



### NOTE

A basic user entry or a simple host entry with a certificate is approximately 5–10 kB in size.

## 1.3. CUSTOM CONFIGURATION REQUIREMENTS FOR IDM

Install an Identity Management (IdM) server on a clean system without any custom configuration for services such as DNS, Kerberos, Apache, or Directory Server.

The IdM server installation overwrites system files to set up the IdM domain. IdM backs up the original system files to `/var/lib/ipa/sysrestore/`. When an IdM server is uninstalled at the end of the lifecycle, these files are restored.

### IPv6 requirements in IdM

The IdM system must have the IPv6 protocol enabled in the kernel. If IPv6 is disabled, then the CLDAP plug-in used by the IdM services fails to initialize.



### NOTE

IPv6 does not have to be enabled on the network.

### Support for encryption types in IdM

Red Hat Enterprise Linux (RHEL) uses Version 5 of the Kerberos protocol, which supports encryption types such as Advanced Encryption Standard (AES), Camellia, and Data Encryption Standard (DES).

### List of supported encryption types

While the Kerberos libraries on IdM servers and clients might support more encryption types, the IdM Kerberos Distribution Center (KDC) only supports the following encryption types:

- **aes256-cts:normal**
- **aes256-cts:special** (default)
- **aes128-cts:normal**
- **aes128-cts:special** (default)
- **aes128-sha2:normal**
- **aes128-sha2:special**
- **aes256-sha2:normal**
- **aes256-sha2:special**
- **camellia128-cts-cmac:normal**
- **camellia128-cts-cmac:special**
- **camellia256-cts-cmac:normal**
- **camellia256-cts-cmac:special**

### RC4 encryption types are disabled by default

The following RC4 encryption types have been deprecated and disabled by default in RHEL 8, as they are considered less secure than the newer AES-128 and AES-256 encryption types:

- **arcfour-hmac:normal**
- **arcfour-hmac:special**

For more information on manually enabling RC4 support for compatibility with legacy Active Directory environments, see [Ensuring support for common encryption types in AD and RHEL](#) .

### Support for DES and 3DES encryption has been removed

Due to security reasons, support for the DES algorithm was deprecated in RHEL 7. The recent rebase of Kerberos packages in RHEL 8.3.0 removes support for single-DES (DES) and triple-DES (3DES) encryption types from RHEL 8.



#### NOTE

Standard RHEL 8 IdM installations do not use DES or 3DES encryption types by default and are unaffected by the Kerberos upgrade.

If you manually configured any services or users to **only** use DES or 3DES encryption (for example, for legacy clients), you might experience service interruptions after updating to the latest Kerberos packages, such as:

- Kerberos authentication errors
- **unknown enctype** encryption errors



- KDCs with DES-encrypted Database Master Keys (**K/M**) fail to start

Red Hat recommends you do not use DES or 3DES encryption in your environment.



#### NOTE

You only need to disable DES and 3DES encryption types if you configured your environment to use them.

### Support for system-wide cryptographic policies in IdM

IdM uses the **DEFAULT** system-wide cryptographic policy. This policy offers secure settings for current threat models. It allows the TLS 1.2 and 1.3 protocols, as well as the IKEv2 and SSH2 protocols. The RSA keys and Diffie-Hellman parameters are accepted if they are at least 2048 bits long. This policy does not allow DES, 3DES, RC4, DSA, TLS v1.0, and other weaker algorithms.



#### NOTE

You cannot install an IdM server while using the **FUTURE** system-wide cryptographic policy. When installing an IdM server, ensure you are using the **DEFAULT** system-wide cryptographic policy.

### Additional Resources

- [System-wide cryptographic policies](#)

### FIPS compliance

With RHEL 8.3.0 or later, you can install a new IdM server or replica on a system with the Federal Information Processing Standard (FIPS) mode enabled.

To install IdM with FIPS, first enable FIPS mode on the host, then install IdM. The IdM installation script detects if FIPS is enabled and configures IdM to only use encryption types that are compliant with FIPS 140-2:

- **aes256-cts:normal**
- **aes256-cts:special**
- **aes128-cts:normal**
- **aes128-cts:special**
- **aes128-sha2:normal**
- **aes128-sha2:special**
- **aes256-sha2:normal**
- **aes256-sha2:special**

For an IdM environment to be FIPS-compliant, **all** IdM replicas must have FIPS mode enabled.

Red Hat recommends that you enable FIPS in IdM clients as well, especially if you might promote those clients to IdM replicas. Ultimately, it is up to administrators to determine how they meet FIPS requirements; Red Hat does not enforce FIPS criteria.

## Support for cross-forest trust with FIPS mode enabled

To establish a cross-forest trust with an Active Directory (AD) domain while FIPS mode is enabled, you must meet the following requirements:

- IdM servers are on RHEL 8.4.0 or later.
- You must authenticate with an AD administrative account when setting up a trust. You cannot establish a trust using a shared secret while FIPS mode is enabled.



### IMPORTANT

RADIUS authentication is not FIPS compliant as the RADIUS protocol uses the MD5 hash function to encrypt passwords between client and server and, in FIPS mode, OpenSSL disables the use of the MD5 digest algorithm. However, if the RADIUS server is running on the same host as the IdM server, you can work around the problem and enable MD5 by performing the steps described in [How to configure FreeRADIUS authentication in FIPS mode](#).

### Additional Resources

- To enable FIPS mode in the RHEL operating system, see [Switching the system to FIPS mode](#) in the *Security Hardening* guide.
- For more details on FIPS 140-2, see the [Security Requirements for Cryptographic Modules](#) on the National Institute of Standards and Technology (NIST) web site.

## 1.4. TIME SERVICE REQUIREMENTS FOR IDM

The following sections discuss using **chronyd** to keep your IdM hosts in sync with a central time source:

- [How IdM uses \*\*chronyd\*\* for synchronization](#)
- [List of NTP configuration options for IdM installation commands](#)
- [Ensuring IdM can reference your NTP time server](#)

### 1.4.1. How IdM uses **chronyd** for synchronization

Kerberos, the underlying authentication mechanism in IdM, uses time stamps as part of its protocol. Kerberos authentication fails if the system time of an IdM client differs by more than five minutes from the system time of the Key Distribution Center (KDC).

To ensure that IdM servers and clients stay in sync with a central time source, IdM installation scripts automatically configure **chronyd** Network Time Protocol (NTP) client software.

If you do not pass any NTP options to the IdM installation command, the installer searches for **\_ntp.\_udp** DNS service (SRV) records that point to the NTP server in your network and configures **chrony** with that IP address. If you do not have any **\_ntp.\_udp** SRV records, **chronyd** uses the configuration shipped with the **chrony** package.



## NOTE

Because **ntpd** has been deprecated in favor of **chronyd** in RHEL 8, IdM servers are no longer configured as Network Time Protocol (NTP) servers and are only configured as NTP clients. The RHEL 7 **NTP Server** IdM server role has also been deprecated in RHEL 8.

### 1.4.2. List of NTP configuration options for IdM installation commands

You can specify the following options with any of the IdM installation commands (**ipa-server-install**, **ipa-replica-install**, **ipa-client-install**) to configure **chronyd** client software during setup.

Table 1.1. List of NTP configuration options for IdM installation commands

Option	Behavior
<b>--ntp-server</b>	Use it to specify one NTP server. You can use it multiple times to specify multiple servers.
<b>--ntp-pool</b>	Use it to specify a pool of multiple NTP servers resolved as one hostname.
<b>-N, --no-ntp</b>	Do not configure, start, or enable <b>chronyd</b> .

### 1.4.3. Ensuring IdM can reference your NTP time server

This procedure verifies you have the necessary configurations in place for IdM to be able to synchronize with your Network Time Protocol (NTP) time server.

#### Prerequisites

- You have configured an NTP time server in your environment. In this example, the hostname of the previously configured time server is **ntpserver.example.com**.

#### Procedure

- Perform a DNS service (SRV) record search for NTP servers in your environment.

```
[user@server ~]$ dig +short -t SRV _ntp._udp.example.com
0 100 123 ntpserver.example.com.
```

- If the previous **dig** search does not return your time server, add a **\_ntp.\_udp** SRV record that points to your time server on port **123**. This process depends on your DNS solution.

#### Verification steps

- Verify that DNS returns an entry for your time server on port **123** when you perform a search for **\_ntp.\_udp** SRV records.

```
[user@server ~]$ dig +short -t SRV _ntp._udp.example.com
0 100 123 ntpserver.example.com.
```

### 1.4.4. Additional resources

- [Implementation of NTP](#)
- [Using the Chrony suite to configure NTP](#)

## 1.5. HOST NAME AND DNS REQUIREMENTS FOR IDM

This section lists the host name and DNS requirements for server and replica systems. It also shows how to verify that the systems meet the requirements.

The requirements in this section apply to all Identity Management (IdM) servers, those with integrated DNS and those without integrated DNS.



### WARNING

DNS records are vital for nearly all IdM domain functions, including running LDAP directory services, Kerberos, and Active Directory integration. Be extremely cautious and ensure that:

- You have a tested and functional DNS service available
- The service is properly configured

This requirement applies to IdM servers with **and** without integrated DNS.

### Verify the server host name

The host name must be a fully qualified domain name, such as ***server.idm.example.com***.



### IMPORTANT

Do not use single-label domain names, for example **.company**: the IdM domain must be composed of one or more subdomains and a top level domain, for example **example.com** or **company.example.com**.

The fully qualified domain name must meet the following conditions:

- It is a valid DNS name, which means only numbers, alphabetic characters, and hyphens (-) are allowed. Other characters, such as underscores (\_), in the host name cause DNS failures.
- It is all lower-case. No capital letters are allowed.
- It does not resolve to the loopback address. It must resolve to the system's public IP address, not to **127.0.0.1**.

To verify the host name, use the **hostname** utility on the system where you want to install:

```
# hostname
server.idm.example.com
```

The output of **hostname** must not be **localhost** or **localhost6**.

## Verify the forward and reverse DNS configuration

1. Obtain the IP address of the server.
  - a. The **ip addr show** command displays both the IPv4 and IPv6 addresses. In the following example, the relevant IPv6 address is **2001:DB8::1111** because its scope is global:

```
[root@server ~]# ip addr show
...
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
state UP group default qlen 1000
link/ether 00:1a:4a:10:4e:33 brd ff:ff:ff:ff:ff:ff
inet 192.0.2.1/24 brd 192.0.2.255 scope global dynamic eth0
    valid_lft 106694sec preferred_lft 106694sec
inet6 2001:DB8::1111/32 scope global dynamic
    valid_lft 2591521sec preferred_lft 604321sec
inet6 fe80::56ee:75ff:fe2b:def6/64 scope link
    valid_lft forever preferred_lft forever
...
```

2. Verify the forward DNS configuration using the **dig** utility.
  - a. Run the command **dig +short server.idm.example.com A**. The returned IPv4 address must match the IP address returned by **ip addr show**:

```
[root@server ~]# dig +short server.idm.example.com A
192.0.2.1
```

- b. Run the command **dig +short server.idm.example.com AAAA**. If it returns an address, it must match the IPv6 address returned by **ip addr show**:

```
[root@server ~]# dig +short server.idm.example.com AAAA
2001:DB8::1111
```



### NOTE

If **dig** does not return any output for the AAAA record, it does not indicate incorrect configuration. No output only means that no IPv6 address is configured in DNS for the system. If you do not intend to use the IPv6 protocol in your network, you can proceed with the installation in this situation.

3. Verify the reverse DNS configuration (PTR records). Use the **dig** utility and add the IP address.  
If the commands below display a different host name or no host name, the reverse DNS configuration is incorrect.

- a. Run the command **dig +short -x IPv4\_address**. The output must display the server host name. For example:

```
[root@server ~]# dig +short -x 192.0.2.1
server.idm.example.com
```

- b. If the command **dig +short -x server.idm.example.com AAAA** in the previous step returned an IPv6 address, use **dig** to query the IPv6 address too. The output must display the server host name. For example:

```
[root@server ~]# dig +short -x 2001:DB8::1111
server.idm.example.com
```



#### NOTE

If **dig +short server.idm.example.com AAAA** in the previous step did not display any IPv6 address, querying the AAAA record does not output anything. In this case, this is normal behavior and does not indicate incorrect configuration.



#### WARNING

If a reverse DNS (PTR record) search returns multiple host names, **httpd** and other software associated with IdM may show unpredictable behavior. Red Hat strongly recommends configuring only one PTR record per IP.

### Verify the standards-compliance of DNS forwarders (required for integrated DNS only)

Ensure that all DNS forwarders you want to use with the IdM DNS server comply with the Extension Mechanisms for DNS (EDNS0) and DNS Security Extensions (DNSSEC) standards. To do this, inspect the output of the following command for each forwarder separately:

```
$ dig +dnssec @IP_address_of_the_DNS_forwarder . SOA
```

The expected output displayed by the command contains the following information:

- status: **NOERROR**
- flags: **ra**
- EDNS flags: **do**
- The **RRSIG** record must be present in the **ANSWER** section

If any of these items is missing from the output, inspect the documentation for your DNS forwarder and verify that EDNS0 and DNSSEC are supported and enabled. In the latest versions of the BIND server, the **dnssec-enable yes**; option must be set in the **/etc/named.conf** file.

Example of the expected output produced by **dig**:

```
:: ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 48655
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
```

```
;; ANSWER SECTION:
```

```
. 31679 IN SOA a.root-servers.net. nstld.verisign-grs.com. 2015100701 1800 900 604800 86400
. 31679 IN RRSIG SOA 8 0 86400 20151017170000 20151007160000 62530 . GNVz7SQs [...]
```

### Verify the `/etc/hosts` file

Verify that the `/etc/hosts` file fulfills one of the following conditions:

- The file does not contain an entry for the host. It only lists the IPv4 and IPv6 localhost entries for the host.
- The file contains an entry for the host and the file fulfills all the following conditions:
  - The first two entries are the IPv4 and IPv6 localhost entries.
  - The next entry specifies the IdM server IPv4 address and host name.
  - The **FQDN** of the IdM server comes before the short name of the IdM server.
  - The IdM server host name is not part of the localhost entry.

The following is an example of a correctly configured `/etc/hosts` file:

```
127.0.0.1 localhost localhost.localdomain \
localhost4 localhost4.localdomain4
::1      localhost localhost.localdomain \
localhost6 localhost6.localdomain6
192.0.2.1 server.idm.example.com server
2001:DB8::1111 server.idm.example.com server
```

## 1.6. PORT REQUIREMENTS FOR IDM

Identity Management (IdM) uses a number of [ports](#) to communicate with its services. These ports must be open and available for incoming connections to the IdM server for IdM to work. They must not be currently used by another service or blocked by a [firewall](#).

**Table 1.2. IdM ports**

Service	Ports	Protocol
HTTP/HTTPS	80, 443	TCP
LDAP/LDAPS	389, 636	TCP
Kerberos	88, 464	TCP and UDP
DNS	53	TCP and UDP (optional)
NTP	123	UDP (optional)

In addition, ports 8080, 8443, and 749 must be free as they are used internally. Do not open these ports and instead leave them blocked by a firewall.

**Table 1.3. firewalld services**

Service name	For details, see:
<b>freeipa-ldap</b>	<code>/usr/lib/firewalld/services/freeipa-ldap.xml</code>
<b>freeipa-ldaps</b>	<code>/usr/lib/firewalld/services/freeipa-ldaps.xml</code>
<b>dns</b>	<code>/usr/lib/firewalld/services/dns.xml</code>

## 1.7. OPENING THE PORTS REQUIRED BY IDM

### Procedure

1. Make sure the **firewalld** service is running.

- To find out if **firewalld** is currently running:

```
# systemctl status firewalld.service
```

- To start **firewalld** and configure it to start automatically when the system boots:

```
# systemctl start firewalld.service
# systemctl enable firewalld.service
```

2. Open the required ports using the **firewall-cmd** utility. Choose one of the following options:

- a. Add the individual ports to the firewall by using the **firewall-cmd --add-port** command. For example, to open the ports in the default zone:

```
# firewall-cmd --permanent --add-port=
{80/tcp,443/tcp,389/tcp,636/tcp,88/tcp,88/udp,464/tcp,464/udp,53/tcp,53/udp,123/udp}
```

- b. Add the **firewalld** services to the firewall by using the **firewall-cmd --add-service** command. For example, to open the ports in the default zone:

```
# firewall-cmd --permanent --add-service={freeipa-ldap,freeipa-ldaps,dns}
```

For details on using **firewall-cmd** to open ports on a system, see the **firewall-cmd(1)** man page.

3. Reload the **firewall-cmd** configuration to ensure that the change takes place immediately:

```
# firewall-cmd --reload
```

Note that reloading **firewalld** on a system in production can cause DNS connection time outs. If required, to avoid the risk of time outs and to make the changes persistent on the running system, use the **--runtime-to-permanent** option of the **firewall-cmd** command, for example:



```
# firewall-cmd --runtime-to-permanent
```

4. **Optional.** To verify that the ports are available now, use the **nc**, **telnet**, or **nmap** utilities to connect to a port or run a port scan.



## NOTE

Note that you also have to open network-based firewalls for both incoming and outgoing traffic.

## 1.8. INSTALLING PACKAGES REQUIRED FOR AN IDM SERVER

In RHEL8, the packages necessary for installing an Identity Management (IdM) server are shipped as a module. The IdM server module stream is called the **DL1** stream, and you need to enable this stream before downloading packages from this stream. The following procedure shows how to download the packages necessary for setting up the IdM environment of your choice.

### Prerequisites

- You have a newly installed RHEL system.
- You have made the required repositories available:
  - If your RHEL system is not running in the cloud, you have registered your system with the Red Hat Subscription Manager (RHSM). For details, see [Registration, attaching, and removing subscriptions in the Subscription Manager command line](#). You have also enabled the **BaseOS** and **AppStream** repositories that IdM uses:

```
# subscription-manager repos --enable=rhel-8-for-x86_64-baseos-rpms
# subscription-manager repos --enable=rhel-8-for-x86_64-appstream-rpms
```

For details on how to enable and disable specific repositories using RHSM, see [Configuring options in Red Hat Subscription Manager](#).

- If your RHEL system is running in the cloud, skip the registration. The required repositories are already available via the Red Hat Update Infrastructure (RHUI).
- You have not previously enabled an IdM module stream.

### Procedure

1. Enable the **idm:DL1** stream:

```
# yum module enable idm:DL1
```

2. Switch to the RPMs delivered through the **idm:DL1** stream:

```
# yum distro-sync
```

3. Choose one of the following options, depending on your IdM requirements:
  - To download the packages necessary for installing an IdM server without an integrated DNS:
    -

```
# yum module install idm:DL1/server
```

- To download the packages necessary for installing an IdM server with an integrated DNS:

```
# yum module install idm:DL1/dns
```

- To download the packages necessary for installing an IdM server that has a trust agreement with Active Directory:

```
# yum module install idm:DL1/adtrust
```

- To download the packages from multiple profiles, for example the **adtrust** and **dns** profiles:

```
# yum module install idm:DL1/{dns,adtrust}
```

- To download the packages necessary for installing an IdM client:

```
# yum module install idm:DL1/client
```



## IMPORTANT

When switching to a new module stream once you have already enabled a different stream and downloaded packages from it, you need to first explicitly remove all the relevant installed content and disable the current module stream before enabling the new module stream. Trying to enable a new stream without disabling the current one results in an error. For details on how to proceed, see [Switching to a later stream](#).



## WARNING

While it is possible to install packages from modules individually, be aware that if you install any package from a module that is not listed as "API" for that module, it is only going to be supported by Red Hat in the context of that module. For example, if you install **bind-dyndb-ldap** directly from the repository to use with your custom 389 Directory Server setup, any problems that you have will be ignored unless they occur for IdM, too.

## CHAPTER 2. INSTALLING AN IDM SERVER: WITH INTEGRATED DNS, WITH AN INTEGRATED CA AS THE ROOT CA

Installing a new Identity Management (IdM) server with integrated DNS has the following advantages:

- You can automate much of the maintenance and DNS record management using native IdM tools. For example, DNS SRV records are automatically created during the setup, and later on are automatically updated.
- You can have a stable connection with the rest of the Internet by setting up global forwarders during the installation of the IdM server. Global forwarders are also useful for trusts with Active Directory.
- You can set up a DNS reverse zone to prevent emails from your domain to be considered spam by email servers outside of the IdM domain.

Installing IdM with integrated DNS has certain limitations:

- IdM DNS is not meant to be used as a general-purpose DNS server. Some of the advanced DNS functions are not supported.

This chapter describes how you can install a new IdM server with an integrated certificate authority (CA) as the root CA.



### NOTE

The default configuration for the **ipa-server-install** command is an integrated CA as the root CA. If no CA option, for example **--external-ca** or **--ca-less** is specified, the IdM server is installed with an integrated CA.

## 2.1. INTERACTIVE INSTALLATION

During the interactive installation using the **ipa-server-install** utility, you are asked to supply basic configuration of the system, for example the realm, the administrator's password and the Directory Manager's password.

The **ipa-server-install** installation script creates a log file at **/var/log/ipaserver-install.log**. If the installation fails, the log can help you identify the problem.

### Procedure

1. Run the **ipa-server-install** utility.

```
# ipa-server-install
```

2. The script prompts to configure an integrated DNS service. Enter **yes**.

```
Do you want to configure integrated DNS (BIND)? [no]: yes
```

3. The script prompts for several required settings and offers recommended default values in brackets.
  - To accept a default value, press **Enter**.

- To provide a custom value, enter the required value.

Server host name [server.example.com]:  
 Please confirm the domain name [example.com]:  
 Please provide a realm name [EXAMPLE.COM]:



### WARNING

Plan these names carefully. You will not be able to change them after the installation is complete.

4. Enter the passwords for the Directory Server superuser (**cn=Directory Manager**) and for the Identity Management (IdM) administration system user account (**admin**).

Directory Manager password:  
 IPA admin password:

5. The script prompts for per-server DNS forwarders.

Do you want to configure DNS forwarders? [yes]:

- To configure per-server DNS forwarders, enter **yes**, and then follow the instructions on the command line. The installation process will add the forwarder IP addresses to the IdM LDAP.
  - For the forwarding policy default settings, see the **--forward-policy** description in the **ipa-dns-install(1)** man page.
- If you do not want to use DNS forwarding, enter **no**.  
 With no DNS forwarders, hosts in your IdM domain will not be able to resolve names from other, internal, DNS domains in your infrastructure. The hosts will only be left with public DNS servers to resolve their DNS queries.

6. The script prompts to check if any DNS reverse (PTR) records for the IP addresses associated with the server need to be configured.

Do you want to search for missing reverse zones? [yes]:

If you run the search and missing reverse zones are discovered, the script asks you whether to create the reverse zones along with the PTR records.

Do you want to create reverse zone for IP 192.0.2.1 [yes]:  
 Please specify the reverse zone name [2.0.192.in-addr.arpa]:  
 Using reverse zone(s) 2.0.192.in-addr.arpa.



### NOTE

Using IdM to manage reverse zones is optional. You can use an external DNS service for this purpose instead.

7. Enter **yes** to confirm the server configuration.

Continue to configure the system with these values? [no]: yes

8. The installation script now configures the server. Wait for the operation to complete.
9. After the installation script completes, update your DNS records in the following way:
  - a. Add DNS delegation from the parent domain to the IdM DNS domain. For example, if the IdM DNS domain is **idm.example.com**, add a name server (NS) record to the **example.com** parent domain.



### IMPORTANT

Repeat this step each time after an IdM DNS server is installed.

- b. Add an **\_ntp.\_udp** service (SRV) record for your time server to your IdM DNS. The presence of the SRV record for the time server of the newly-installed IdM server in IdM DNS ensures that future replica and client installations are automatically configured to synchronize with the time server used by this primary IdM server.

## 2.2. NON-INTERACTIVE INSTALLATION

The **ipa-server-install** installation script creates a log file at **/var/log/ipaserver-install.log**. If the installation fails, the log can help you identify the problem.

### Procedure

1. Run the **ipa-server-install** utility with the options to supply all the required information. The minimum required options for non-interactive installation are:
  - **--realm** to provide the Kerberos realm name
  - **--ds-password** to provide the password for the Directory Manager (DM), the Directory Server super user
  - **--admin-password** to provide the password for **admin**, the Identity Management (IdM) administrator
  - **--unattended** to let the installation process select default options for the host name and domain name

To install a server with integrated DNS, add also these options:

- **--setup-dns** to configure integrated DNS
- **--forwarder** or **--no-forwarders**, depending on whether you want to configure DNS forwarders or not
- **--auto-reverse** or **--no-reverse**, depending on whether you want to configure automatic detection of the reverse DNS zones that must be created in the IdM DNS or no reverse zone auto-detection

For example:

```
# ipa-server-install --realm EXAMPLE.COM --ds-password DM_password --admin-  
password admin_password --unattended --setup-dns --forwarder 192.0.2.1 --no-  
reverse
```

2. After the installation script completes, update your DNS records in the following way:
  - a. Add DNS delegation from the parent domain to the IdM DNS domain. For example, if the IdM DNS domain is ***idm.example.com***, add a name server (NS) record to the ***example.com*** parent domain.



### IMPORTANT

Repeat this step each time after an IdM DNS server is installed.

- b. Add an ***\_ntp.\_udp*** service (SRV) record for your time server to your IdM DNS. The presence of the SRV record for the time server of the newly-installed IdM server in IdM DNS ensures that future replica and client installations are automatically configured to synchronize with the time server used by this primary IdM server.

### Additional resources

- For a complete list of options accepted by ***ipa-server-install***, run the ***ipa-server-install --help*** command.

## CHAPTER 3. INSTALLING AN IDM SERVER: WITH INTEGRATED DNS, WITH AN EXTERNAL CA AS THE ROOT CA

Installing a new Identity Management (IdM) server with integrated DNS has the following advantages:

- You can automate much of the maintenance and DNS record management using native IdM tools. For example, DNS SRV records are automatically created during the setup, and later on are automatically updated.
- You can have a stable connection with the rest of the Internet by setting up global forwarders during the installation of the IdM server. Global forwarders are also useful for trusts with Active Directory.
- You can set up a DNS reverse zone to prevent emails from your domain to be considered spam by email servers outside of the IdM domain.

Installing IdM with integrated DNS has certain limitations:

- IdM DNS is not meant to be used as a general-purpose DNS server. Some of the advanced DNS functions are not supported.

This chapter describes how you can install a new IdM server with an external certificate authority (CA) as the root CA.

### 3.1. INTERACTIVE INSTALLATION

During the interactive installation using the **ipa-server-install** utility, you are asked to supply basic configuration of the system, for example the realm, the administrator's password and the Directory Manager's password.

The **ipa-server-install** installation script creates a log file at **/var/log/ipaserver-install.log**. If the installation fails, the log can help you identify the problem.

This procedure describes how to install a server:

- with integrated DNS
- with an external certificate authority (CA) as the root CA

#### Prerequisites

- Decide on the type of the external CA you use (the **--external-ca-type** option). See the **ipa-server-install(1)** man page for details.
- Alternatively, decide on the **--external-ca-profile** option allowing an alternative Active Directory Certificate Services (AD CS) template to be specified. For example, to specify an AD CS installation-specific object identifier:

```
[root@server ~]# ipa-server-install --external-ca --external-ca-type=ms-cs --external-ca-profile=1.3.6.1.4.1.311.21.8.8950086.10656446.2706058.12775672.480128.147.7130143.4405632:1
```

#### Procedure

1. Run the `ipa-server-install` utility with the `--external-ca` option.

```
# ipa-server-install --external-ca
```

If you are using the Microsoft Certificate Services CA, use also the `--external-ca-type` option. For details, see the `ipa-server-install` (1) man page.

2. The script prompts to configure an integrated DNS service. Enter **yes** or **no**. In this procedure, we are installing a server with integrated DNS.

```
Do you want to configure integrated DNS (BIND)? [no]: yes
```



#### NOTE

If you want to install a server without integrated DNS, the installation script will not prompt you for DNS configuration as described in the steps below. See [Chapter 5, Installing an IdM server: Without integrated DNS, with an integrated CA as the root CA](#) for details on the steps for installing a server without DNS.

3. The script prompts for several required settings and offers recommended default values in brackets.

- To accept a default value, press **Enter**.
- To provide a custom value, enter the required value.

```
Server host name [server.example.com]:
Please confirm the domain name [example.com]:
Please provide a realm name [EXAMPLE.COM]:
```



#### WARNING

Plan these names carefully. You will not be able to change them after the installation is complete.

4. Enter the passwords for the Directory Server superuser (**cn=Directory Manager**) and for the Identity Management (IdM) administration system user account (**admin**).

```
Directory Manager password:
IPA admin password:
```

5. The script prompts for per-server DNS forwarders.

```
Do you want to configure DNS forwarders? [yes]:
```

- To configure per-server DNS forwarders, enter **yes**, and then follow the instructions on the command line. The installation process will add the forwarder IP addresses to the IdM LDAP.



- For the forwarding policy default settings, see the **--forward-policy** description in the **ipa-dns-install(1)** man page.
  - If you do not want to use DNS forwarding, enter **no**.  
With no DNS forwarders, hosts in your IdM domain will not be able to resolve names from other, internal, DNS domains in your infrastructure. The hosts will only be left with public DNS servers to resolve their DNS queries.
6. The script prompts to check if any DNS reverse (PTR) records for the IP addresses associated with the server need to be configured.

Do you want to search for missing reverse zones? [yes]:

If you run the search and missing reverse zones are discovered, the script asks you whether to create the reverse zones along with the PTR records.

Do you want to create reverse zone for IP 192.0.2.1 [yes]:  
Please specify the reverse zone name [2.0.192.in-addr.arpa]:  
Using reverse zone(s) 2.0.192.in-addr.arpa.



#### NOTE

Using IdM to manage reverse zones is optional. You can use an external DNS service for this purpose instead.

7. Enter **yes** to confirm the server configuration.

Continue to configure the system with these values? [no]: yes

8. During the configuration of the Certificate System instance, the utility prints the location of the certificate signing request (CSR): **/root/ipa.csr**:

...

Configuring certificate server (pki-tomcatd): Estimated time 3 minutes 30 seconds  
[1/8]: creating certificate server user  
[2/8]: configuring certificate server instance  
The next step is to get /root/ipa.csr signed by your CA and re-run /sbin/ipa-server-install as:  
/sbin/ipa-server-install --external-cert-file=/path/to/signed\_certificate --external-cert-file=/path/to/external\_ca\_certificate

When this happens:

- a. Submit the CSR located in **/root/ipa.csr** to the external CA. The process differs depending on the service to be used as the external CA.
- b. Retrieve the issued certificate and the CA certificate chain for the issuing CA in a base 64-encoded blob (either a PEM file or a Base\_64 certificate from a Windows CA). Again, the process differs for every certificate service. Usually, a download link on a web page or in the notification email allows the administrator to download all the required certificates.



#### IMPORTANT

Be sure to get the full certificate chain for the CA, not just the CA certificate.

- c. Run **ipa-server-install** again, this time specifying the locations and names of the newly-issued CA certificate and the CA chain files. For example:

```
# ipa-server-install --external-cert-file=/tmp/servercert20170601.pem --external-cert-file=/tmp/cacert.pem
```

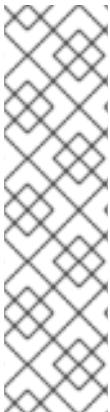
9. The installation script now configures the server. Wait for the operation to complete.
10. After the installation script completes, update your DNS records in the following way:
  - a. Add DNS delegation from the parent domain to the IdM DNS domain. For example, if the IdM DNS domain is **idm.example.com**, add a name server (NS) record to the **example.com** parent domain.



#### IMPORTANT

Repeat this step each time after an IdM DNS server is installed.

- b. Add an **\_ntp.\_udp** service (SRV) record for your time server to your IdM DNS. The presence of the SRV record for the time server of the newly-installed IdM server in IdM DNS ensures that future replica and client installations are automatically configured to synchronize with the time server used by this primary IdM server.



#### NOTE

The **ipa-server-install --external-ca** command can sometimes fail with the following error:

```
ipa      : CRITICAL failed to configure ca instance Command '/usr/sbin/pkispawn -s CA -f /tmp/configuration_file' returned non-zero exit status 1
Configuration of CA failed
```

This failure occurs when the **\*\_proxy** environmental variables are set. For a solution of the problem, see [Troubleshooting: External CA installation fails](#).

## 3.2. TROUBLESHOOTING: EXTERNAL CA INSTALLATION FAILS

The **ipa-server-install --external-ca** command fails with the following error:

```
ipa      : CRITICAL failed to configure ca instance Command '/usr/sbin/pkispawn -s CA -f /tmp/configuration_file' returned non-zero exit status 1
Configuration of CA failed
```

The **env|grep proxy** command displays variables such as the following:

```
# env|grep proxy
http_proxy=http://example.com:8080
ftp_proxy=http://example.com:8080
https_proxy=http://example.com:8080
```

#### What this means:

The **\*\_proxy** environmental variables are preventing the server from being installed.

### To fix the problem:

1. Use the following shell script to unset the **\*\_proxy** environmental variables:

```
# for i in ftp http https; do unset ${i}_proxy; done
```

2. Run the **pkidestroy** utility to remove the unsuccessful certificate authority (CA) subsystem installation:

```
# pkidestroy -s CA -i pki-tomcat; rm -rf /var/log/pki/pki-tomcat /etc/sysconfig/pki-tomcat /etc/sysconfig/pki/tomcat/pki-tomcat /var/lib/pki/pki-tomcat /etc/pki/pki-tomcat /root/ipa.csr
```

3. Remove the failed Identity Management (IdM) server installation:

```
# ipa-server-install --uninstall
```

4. Retry running **ipa-server-install --external-ca**.

## CHAPTER 4. INSTALLING AN IDM SERVER: WITH INTEGRATED DNS, WITHOUT A CA

Installing a new Identity Management (IdM) server with integrated DNS has the following advantages:

- You can automate much of the maintenance and DNS record management using native IdM tools. For example, DNS SRV records are automatically created during the setup, and later on are automatically updated.
- You can have a stable connection with the rest of the Internet by setting up global forwarders during the installation of the IdM server. Global forwarders are also useful for trusts with Active Directory.
- You can set up a DNS reverse zone to prevent emails from your domain to be considered spam by email servers outside of the IdM domain.

Installing IdM with integrated DNS has certain limitations:

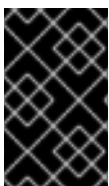
- IdM DNS is not meant to be used as a general-purpose DNS server. Some of the advanced DNS functions are not supported.

This chapter describes how you can install a new IdM server without a certificate authority (CA).

### 4.1. CERTIFICATES REQUIRED TO INSTALL AN IDM SERVER WITHOUT A CA

This section lists:

- the certificates required to install an Identity Management (IdM) server without a certificate authority (CA)
- the command-line options used to provide these certificates to the **ipa-server-install** utility



#### IMPORTANT

You cannot install a server or replica using self-signed third-party server certificates because the imported certificate files must contain the full CA certificate chain of the CA that issued the LDAP and Apache server certificates.

#### The LDAP server certificate and private key

- **--dirsrv-cert-file** for the certificate and private key files for the LDAP server certificate
- **--dirsrv-pin** for the password to access the private key in the files specified in **--dirsrv-cert-file**

#### The Apache server certificate and private key

- **--http-cert-file** for the certificate and private key files for the Apache server certificate
- **--http-pin** for the password to access the private key in the files specified in **--http-cert-file**

#### The full CA certificate chain of the CA that issued the LDAP and Apache server certificates

- **--dirstv-cert-file** and **--http-cert-file** for the certificate files with the full CA certificate chain or a part of it

You can provide the files specified in the **--dirstv-cert-file** and **--http-cert-file** options in the following formats:

- Privacy-Enhanced Mail (PEM) encoded certificate (RFC 7468). Note that the Identity Management installer accepts concatenated PEM-encoded objects.
- Distinguished Encoding Rules (DER)
- PKCS #7 certificate chain objects
- PKCS #8 private key objects
- PKCS #12 archives

You can specify the **--dirstv-cert-file** and **--http-cert-file** options multiple times to specify multiple files.

#### The certificate files to complete the full CA certificate chain (not needed in some environments)

- **--ca-cert-file** for the file or files containing the CA certificate of the CA that issued the LDAP, Apache Server, and Kerberos KDC certificates. Use this option if the CA certificate is not present in the certificate files provided by the other options.

The files provided using **--dirstv-cert-file** and **--http-cert-file** combined with the file provided using **--ca-cert-file** must contain the full CA certificate chain of the CA that issued the LDAP and Apache server certificates.

#### The Kerberos key distribution center (KDC) PKINIT certificate and private key (optional)

- **--pkinit-cert-file** for the Kerberos KDC SSL certificate and private key
- **--pkinit-pin** for the password to access the Kerberos KDC private key in the files specified in **--pkinit-cert-file**
- **--no-pkinit** for disabling pkinit setup steps

If you do not provide the PKINIT certificate, **ipa-server-install** configures the IdM server with a local KDC with a self-signed certificate.

#### Additional resources

- For details on what the certificate file formats these options accept, see the **ipa-server-install(1)** man page.
- For details on PKINIT extensions required to create a RHEL IdM PKINIT certificate, see [this article](#).

## 4.2. INTERACTIVE INSTALLATION

During the interactive installation using the **ipa-server-install** utility, you are asked to supply basic configuration of the system, for example the realm, the administrator's password and the Directory Manager's password.

The **ipa-server-install** installation script creates a log file at **/var/log/ipaserver-install.log**. If the installation fails, the log can help you identify the problem.

## Procedure

1. Run the **ipa-server-install** utility and provide all the required certificates. For example:

```
[root@server ~]# ipa-server-install \
--http-cert-file /tmp/server.crt \
--http-cert-file /tmp/server.key \
--http-pin secret \
--dirsrv-cert-file /tmp/server.crt \
--dirsrv-cert-file /tmp/server.key \
--dirsrv-pin secret \
--ca-cert-file ca.crt
```

See [Certificates required to install an IdM server without a CA](#) for details on the provided certificates.

2. The script prompts to configure an integrated DNS service. Enter **yes** or **no**. In this procedure, we are installing a server with integrated DNS.

```
Do you want to configure integrated DNS (BIND)? [no]: yes
```



### NOTE

If you want to install a server without integrated DNS, the installation script will not prompt you for DNS configuration as described in the steps below. See [Installing an IdM server: Without integrated DNS, with an integrated CA as the root CA](#) for details on the steps for installing a server without DNS.

3. The script prompts for several required settings and offers recommended default values in brackets.
  - To accept a default value, press **Enter**.
  - To provide a custom value, enter the required value.

```
Server host name [server.example.com]:
Please confirm the domain name [example.com]:
Please provide a realm name [EXAMPLE.COM]:
```



### WARNING

Plan these names carefully. You will not be able to change them after the installation is complete.

4. Enter the passwords for the Directory Server superuser (**cn=Directory Manager**) and for the Identity Management (IdM) administration system user account (**admin**).

Directory Manager password:  
IPA admin password:

5. The script prompts for per-server DNS forwarders.

Do you want to configure DNS forwarders? [yes]:

- To configure per-server DNS forwarders, enter **yes**, and then follow the instructions on the command line. The installation process will add the forwarder IP addresses to the IdM LDAP.
  - For the forwarding policy default settings, see the **--forward-policy** description in the **ipa-dns-install(1)** man page.
- If you do not want to use DNS forwarding, enter **no**.  
With no DNS forwarders, hosts in your IdM domain will not be able to resolve names from other, internal, DNS domains in your infrastructure. The hosts will only be left with public DNS servers to resolve their DNS queries.

6. The script prompts to check if any DNS reverse (PTR) records for the IP addresses associated with the server need to be configured.

Do you want to search for missing reverse zones? [yes]:

If you run the search and missing reverse zones are discovered, the script asks you whether to create the reverse zones along with the PTR records.

Do you want to create reverse zone for IP 192.0.2.1 [yes]:  
Please specify the reverse zone name [2.0.192.in-addr.arpa]:  
Using reverse zone(s) 2.0.192.in-addr.arpa.



#### NOTE

Using IdM to manage reverse zones is optional. You can use an external DNS service for this purpose instead.

7. Enter **yes** to confirm the server configuration.

Continue to configure the system with these values? [no]: yes

8. The installation script now configures the server. Wait for the operation to complete.
9. After the installation script completes, update your DNS records in the following way:
  - a. Add DNS delegation from the parent domain to the IdM DNS domain. For example, if the IdM DNS domain is **idm.example.com**, add a name server (NS) record to the **example.com** parent domain.



#### IMPORTANT

Repeat this step each time after an IdM DNS server is installed.

- b. Add an **\_ntp.\_udp** service (SRV) record for your time server to your IdM DNS. The

presence of the SRV record for the time server of the newly-installed IdM server in IdM DNS ensures that future replica and client installations are automatically configured to synchronize with the time server used by this primary IdM server.



## CHAPTER 5. INSTALLING AN IDM SERVER: WITHOUT INTEGRATED DNS, WITH AN INTEGRATED CA AS THE ROOT CA

This chapter describes how you can install a new Identity Management (IdM) server without integrated DNS.



### NOTE

Red Hat strongly recommends installing IdM-integrated DNS for basic usage within the IdM deployment: When the IdM server also manages DNS, there is tight integration between DNS and native IdM tools which enables automating some of the DNS record management.

For more details, see [Planning your DNS services and host names](#).

### 5.1. INTERACTIVE INSTALLATION

During the interactive installation using the **ipa-server-install** utility, you are asked to supply basic configuration of the system, for example the realm, the administrator's password and the Directory Manager's password.

The **ipa-server-install** installation script creates a log file at **/var/log/ipaserver-install.log**. If the installation fails, the log can help you identify the problem.

This procedure installs a server:

- Without integrated DNS
- With integrated Identity Management (IdM) certificate authority (CA) as the root CA, which is the default CA configuration

#### Procedure

1. Run the **ipa-server-install** utility.

```
# ipa-server-install
```

2. The script prompts to configure an integrated DNS service. Press **Enter** to select the default **no** option.

```
Do you want to configure integrated DNS (BIND)? [no]:
```

3. The script prompts for several required settings and offers recommended default values in brackets.
  - To accept a default value, press **Enter**.
  - To provide a custom value, enter the required value.

```
Server host name [server.example.com]:
Please confirm the domain name [example.com]:
Please provide a realm name [EXAMPLE.COM]:
```

**WARNING**

Plan these names carefully. You will not be able to change them after the installation is complete.

4. Enter the passwords for the Directory Server superuser (**cn=Directory Manager**) and for the IdM administration system user account (**admin**).

```
Directory Manager password:
IPA admin password:
```

5. Enter **yes** to confirm the server configuration.

```
Continue to configure the system with these values? [no]: yes
```

6. The installation script now configures the server. Wait for the operation to complete.
7. The installation script produces a file with DNS resource records: **the /tmp/ipa.system.records.UFRPto.db** file in the example output below. Add these records to the existing external DNS servers. The process of updating the DNS records varies depending on the particular DNS solution.

```
...
Restarting the KDC
Please add records in this file to your DNS system:
/tmp/ipa.system.records.UFRBto.db
Restarting the web server
...
```

**IMPORTANT**

The server installation is not complete until you add the DNS records to the existing DNS servers.

**Additional resources**

- For more information about the DNS resource records you must add to your DNS system, see [IdM DNS records for external DNS systems](#).

**5.2. NON-INTERACTIVE INSTALLATION**

This procedure installs a server:

- Without integrated DNS
- With integrated Identity Management (IdM) certificate authority (CA) as the root CA, which is the default CA configuration



## NOTE

The **ipa-server-install** installation script creates a log file at **/var/log/ipaserver-install.log**. If the installation fails, the log can help you identify the problem.

## Procedure

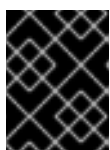
1. Run the **ipa-server-install** utility with the options to supply all the required information. The minimum required options for non-interactive installation are:
  - **--realm** to provide the Kerberos realm name
  - **--ds-password** to provide the password for the Directory Manager (DM), the Directory Server super user
  - **--admin-password** to provide the password for **admin**, the IdM administrator
  - **--unattended** to let the installation process select default options for the host name and domain name

For example:

```
# ipa-server-install --realm EXAMPLE.COM --ds-password DM_password --admin-
password admin_password --unattended
```

2. The installation script produces a file with DNS resource records: **the /tmp/ipa.system.records.UFRPto.db** file in the example output below. Add these records to the existing external DNS servers. The process of updating the DNS records varies depending on the particular DNS solution.

```
...
Restarting the KDC
Please add records in this file to your DNS system:
/tmp/ipa.system.records.UFRBto.db
Restarting the web server
...
```



## IMPORTANT

The server installation is not complete until you add the DNS records to the existing DNS servers.

## Additional resources

- For more information about the DNS resource records you must add to your DNS system, see [IdM DNS records for external DNS systems](#) .
- For a complete list of options accepted by **ipa-server-install**, run the **ipa-server-install --help** command.

## 5.3. IDM DNS RECORDS FOR EXTERNAL DNS SYSTEMS

After installing an IdM server without integrated DNS, you must add LDAP and Kerberos DNS resource records for the IdM server to your external DNS system.

The **ipa-server-install** installation script generates a file containing the list of DNS resource records with a file name in the format **/tmp/ipa.system.records.<random\_characters>.db** and prints instructions to add those records:

Please add records in this file to your DNS system: **/tmp/ipa.system.records.6zджqh3.db**

This is an example of the contents of the file:

```
_kerberos-master._tcp.example.com. 86400 IN SRV 0 100 88 server.example.com.  
_kerberos-master._udp.example.com. 86400 IN SRV 0 100 88 server.example.com.  
_kerberos._tcp.example.com. 86400 IN SRV 0 100 88 server.example.com.  
_kerberos._udp.example.com. 86400 IN SRV 0 100 88 server.example.com.  
_kerberos.example.com. 86400 IN TXT "EXAMPLE.COM"  
_kpasswd._tcp.example.com. 86400 IN SRV 0 100 464 server.example.com.  
_kpasswd._udp.example.com. 86400 IN SRV 0 100 464 server.example.com.  
_ldap._tcp.example.com. 86400 IN SRV 0 100 389 server.example.com.
```



## NOTE

After adding the LDAP and Kerberos DNS resource records for the IdM server to your DNS system, ensure that the DNS management tools have not added PTR records for **ipa-ca**. The presence of PTR records for **ipa-ca** in your DNS could cause subsequent IdM replica installations to fail.

## CHAPTER 6. INSTALLING AN IDM SERVER: WITHOUT INTEGRATED DNS, WITH AN EXTERNAL CA AS THE ROOT CA

This chapter describes how you can install a new Identity Management (IdM) server, without integrated DNS, that uses an external certificate authority (CA) as the root CA.



### NOTE

Red Hat strongly recommends installing IdM-integrated DNS for basic usage within the IdM deployment: When the IdM server also manages DNS, there is tight integration between DNS and native IdM tools which enables automating some of the DNS record management.

For more details, see [Planning your DNS services and host names](#).

### 6.1. OPTIONS USED WHEN INSTALLING AN IDM CA WITH AN EXTERNAL CA AS THE ROOT CA

You may want to install an Identity Management IdM certificate authority (CA) with an external CA as the root CA if one of the following conditions applies:

- You are installing a new IdM server or replica by using the **ipa-server-install** command.
- You are installing the CA component into an existing IdM server by using the **ipa-ca-install** command.

This section describes the options for both commands that you can use for creating a certificate signing request (CSR) during the installation of an IdM CA with an external CA as the root CA.

#### **--external-ca-type=TYPE**

Type of the external CA. Possible values are **generic** and **ms-cs**. The default value is **generic**. Use **ms-cs** to include a template name required by Microsoft Certificate Services (MS CS) in the generated CSR. To use a non-default profile, use the **--external-ca-profile** option in conjunction with **--external-ca-type=ms-cs**.

#### **--external-ca-profile=PROFILE\_SPEC**

Specify the certificate profile or template that you want the MS CS to apply when issuing the certificate for your IdM CA.

Note that the **--external-ca-profile** option can only be used if **--external-ca-type** is **ms-cs**.

You can identify the MS CS template in one of the following ways:

- **<oid>:<majorVersion>[:<minorVersion>]**. You can specify a certificate template by its object identifier (OID) and major version. You can optionally also specify the minor version.
- **<name>**. You can specify a certificate template by its name. The name cannot contain any **:** characters and cannot be an OID, otherwise the OID-based template specifier syntax takes precedence.
- **default**. If you use this specifier, the template name **SubCA** is used.

In certain scenarios, the Active Directory (AD) administrator can use the **Subordinate Certification Authority (SCA)** template, which is a built-in template in AD CS, to create a unique template to better suit the needs of the organization. The new template can, for example, have a customized validity period

and customized extensions. The associated Object Identifier (OID) can be found in the AD **Certificates Template** console.

If the AD administrator has disabled the original, built-in template, you must specify the OID or name of the new template when requesting a certificate for your IdM CA. Ask your AD administrator to provide you with the name or OID of the new template.

If the original SCA AD CS template is still enabled, you can use it by specifying **--external-ca-type=ms-cs** without additionally using the **--external-ca-profile** option. In this case, the **subCA** external CA profile is used, which is the default IdM template corresponding to the SCA AD CS template.

## 6.2. INTERACTIVE INSTALLATION

During the interactive installation using the **ipa-server-install** utility, you are asked to supply basic configuration of the system, for example the realm, the administrator's password and the Directory Manager's password.

The **ipa-server-install** installation script creates a log file at **/var/log/ipaserver-install.log**. If the installation fails, the log can help you identify the problem.

This procedure describes how to install a server:

- Without integrated DNS
- With an external certificate authority (CA) as the root CA

### Procedure

1. Run the **ipa-server-install** utility with the **--external-ca** option.
  - If you are using the Microsoft Certificate Services (MS CS) CA, also use the **--external-ca-type** and **--external-ca-profile** options. For example, to install an IdM server with a CA whose signing certificate is issued using the 1.3.6.1.4.1.311.21.8.8950086.10656446.2706058.12775672.480128.147.7130143.4405632:1 Object Identifier (OID) template:

```
[root@server ~]# ipa-server-install --external-ca --external-ca-type=ms-cs --
external-ca-
profile=1.3.6.1.4.1.311.21.8.8950086.10656446.2706058.12775672.480128.147.7130143
.4405632:1
```

For more information about the **--external-ca-type** and **--external-ca-profile** options, see [Options used when installing an IdM CA with an external CA as the root CA](#).

- If you are not using MS CS to generate the signing certificate for your IdM CA, no other option may be necessary:

```
# ipa-server-install --external-ca
```

2. The script prompts to configure an integrated DNS service. Press **Enter** to select the default **no** option.

```
Do you want to configure integrated DNS (BIND)? [no]:
```

3. The script prompts for several required settings and offers recommended default values in brackets.

- To accept a default value, press **Enter**.
- To provide a custom value, enter the required value.

```
Server host name [server.example.com]:
Please confirm the domain name [example.com]:
Please provide a realm name [EXAMPLE.COM]:
```



### WARNING

Plan these names carefully. You will not be able to change them after the installation is complete.

4. Enter the passwords for the Directory Server superuser (**cn=Directory Manager**) and for the IdM administration system user account (**admin**).

```
Directory Manager password:
IPA admin password:
```

5. Enter **yes** to confirm the server configuration.

```
Continue to configure the system with these values? [no]: yes
```

6. During the configuration of the Certificate System instance, the utility prints the location of the certificate signing request (CSR): **/root/ipa.csr**:

```
...

Configuring certificate server (pki-tomcatd): Estimated time 3 minutes 30 seconds
[1/8]: creating certificate server user
[2/8]: configuring certificate server instance
The next step is to get /root/ipa.csr signed by your CA and re-run /sbin/ipa-server-install as:
/sbin/ipa-server-install --external-cert-file=/path/to/signed_certificate --external-cert-
file=/path/to/external_ca_certificate
```

When this happens:

- a. Submit the CSR located in **/root/ipa.csr** to the external CA. The process differs depending on the service to be used as the external CA.
- b. Retrieve the issued certificate and the CA certificate chain for the issuing CA in a base 64-encoded blob (either a PEM file or a Base\_64 certificate from a Windows CA). Again, the process differs for every certificate service. Usually, a download link on a web page or in the notification email allows the administrator to download all the required certificates.

**IMPORTANT**

Be sure to get the full certificate chain for the CA, not just the CA certificate.

- c. Run **ipa-server-install** again, this time specifying the locations and names of the newly-issued CA certificate and the CA chain files. For example:

```
# ipa-server-install --external-cert-file=/tmp/servercert20170601.pem --external-cert-file=/tmp/cacert.pem
```

7. The installation script now configures the server. Wait for the operation to complete.
8. The installation script produces a file with DNS resource records: **the /tmp/ipa.system.records.UFRPto.db** file in the example output below. Add these records to the existing external DNS servers. The process of updating the DNS records varies depending on the particular DNS solution.

```
...
Restarting the KDC
Please add records in this file to your DNS system:
/tmp/ipa.system.records.UFRBto.db
Restarting the web server
...
```

**IMPORTANT**

The server installation is not complete until you add the DNS records to the existing DNS servers.

**Additional resources**

- For more information about the DNS resource records you must add to your DNS system, see [IdM DNS records for external DNS systems](#) .
- The **ipa-server-install --external-ca** command can sometimes fail with the following error:

```
ipa      : CRITICAL failed to configure ca instance Command '/usr/sbin/pkispawn -s CA -f /tmp/pass:quotes[configuration_file]' returned non-zero exit status 1
Configuration of CA failed
```

This failure occurs when the **\*\_proxy** environmental variables are set. For a solution of the problem, see [Troubleshooting: External CA installation fails](#) .

**6.3. NON-INTERACTIVE INSTALLATION**

This procedure installs a server:

- Without integrated DNS
- with an external certificate authority (CA) as the root CA





## NOTE

The **ipa-server-install** installation script creates a log file at **/var/log/ipaserver-install.log**. If the installation fails, the log can help you identify the problem.

## Prerequisites

- Decide on the type of the external CA you use (the **--external-ca-type** option). See the **ipa-server-install(1)** man page for details.
- Alternatively, decide on the **--external-ca-profile** option allowing an alternative Active Directory Certificate Services (AD CS) template to be specified. For example, to specify an AD CS installation-specific object identifier:

```
[root@server ~]# ipa-server-install --external-ca --external-ca-type=ms-cs --external-ca-profile=1.3.6.1.4.1.311.21.8.8950086.10656446.2706058.12775672.480128.147.7130143.4405632:1
```

## Procedure

1. Run the **ipa-server-install** utility with the options to supply all the required information. The minimum required options for non-interactive installation of an IdM server with an external CA as the root CA are:

- **--external-ca** to specify an external CA is the root CA
  - **--realm** to provide the Kerberos realm name
  - **--ds-password** to provide the password for the Directory Manager (DM), the Directory Server super user
  - **--admin-password** to provide the password for **admin**, the IdM administrator
  - **--unattended** to let the installation process select default options for the host name and domain name
- For example:

```
# ipa-server-install --external-ca --realm EXAMPLE.COM --ds-password DM_password --admin-password admin_password --unattended
```

If you are using the Microsoft Certificate Services CA, use also the **--external-ca-type** option. For details, see the **ipa-server-install(1)** man page.

2. During the configuration of the Certificate System instance, the utility prints the location of the certificate signing request (CSR): **/root/ipa.csr**:

...

Configuring certificate server (pki-tomcatd). Estimated time: 3 minutes

[1/11]: configuring certificate server instance

The next step is to get /root/ipa.csr signed by your CA and re-run /usr/sbin/ipa-server-install as:

```
/usr/sbin/ipa-server-install --external-cert-file=/path/to/signed_certificate --external-cert-file=/path/to/external_ca_certificate
```

The ipa-server-install command was successful

When this happens:

- a. Submit the CSR located in **/root/ipa.csr** to the external CA. The process differs depending on the service to be used as the external CA.
- b. Retrieve the issued certificate and the CA certificate chain for the issuing CA in a base 64-encoded blob (either a PEM file or a Base\_64 certificate from a Windows CA). Again, the process differs for every certificate service. Usually, a download link on a web page or in the notification email allows the administrator to download all the required certificates.



### IMPORTANT

Be sure to get the full certificate chain for the CA, not just the CA certificate.

- c. Run **ipa-server-install** again, this time specifying the locations and names of the newly-issued CA certificate and the CA chain files. For example:

```
# ipa-server-install --external-cert-file=/tmp/servercert20170601.pem --external-cert-
file=/tmp/cacert.pem --realm EXAMPLE.COM --ds-password DM_password --admin-
password admin_password --unattended
```

3. The installation script now configures the server. Wait for the operation to complete.
4. The installation script produces a file with DNS resource records: the **/tmp/ipa.system.records.UFRPto.db** file in the example output below. Add these records to the existing external DNS servers. The process of updating the DNS records varies depending on the particular DNS solution.

```
...
Restarting the KDC
Please add records in this file to your DNS system:
/tmp/ipa.system.records.UFRBto.db
Restarting the web server
...
```



### IMPORTANT

The server installation is not complete until you add the DNS records to the existing DNS servers.

#### Additional resources

- For more information about the DNS resource records you must add to your DNS system, see [IdM DNS records for external DNS systems](#).

## 6.4. IDM DNS RECORDS FOR EXTERNAL DNS SYSTEMS

After installing an IdM server without integrated DNS, you must add LDAP and Kerberos DNS resource records for the IdM server to your external DNS system.

The **ipa-server-install** installation script generates a file containing the list of DNS resource records with a file name in the format **/tmp/ipa.system.records.<random\_characters>.db** and prints instructions to add those records:

-

Please add records in this file to your DNS system: **/tmp/ipa.system.records.6zджqh3.db**

This is an example of the contents of the file:

```
_kerberos-master._tcp.example.com. 86400 IN SRV 0 100 88 server.example.com.
_kerberos-master._udp.example.com. 86400 IN SRV 0 100 88 server.example.com.
_kerberos._tcp.example.com. 86400 IN SRV 0 100 88 server.example.com.
_kerberos._udp.example.com. 86400 IN SRV 0 100 88 server.example.com.
_kerberos.example.com. 86400 IN TXT "EXAMPLE.COM"
_kpasswd._tcp.example.com. 86400 IN SRV 0 100 464 server.example.com.
_kpasswd._udp.example.com. 86400 IN SRV 0 100 464 server.example.com.
_ldap._tcp.example.com. 86400 IN SRV 0 100 389 server.example.com.
```



## NOTE

After adding the LDAP and Kerberos DNS resource records for the IdM server to your DNS system, ensure that the DNS management tools have not added PTR records for **ipa-ca**. The presence of PTR records for **ipa-ca** in your DNS could cause subsequent IdM replica installations to fail.

## CHAPTER 7. INSTALLING AN IDM SERVER OR REPLICA WITH CUSTOM DATABASE SETTINGS FROM AN LDIF FILE

You can install an IdM server and IdM replicas with custom settings for the Directory Server database. The following procedure shows you how to create an LDAP Data Interchange Format (LDIF) file with database settings, and how to pass those settings to the IdM server and replica installation commands.

### Prerequisites

- You have determined custom Directory Server settings that improve the performance of your IdM environment. See [Adjusting IdM Directory Server performance](#).

### Procedure

1. Create a text file in LDIF format with your custom database settings. Separate LDAP attribute modifications with a dash (-). This example sets non-default values for the idle timeout and maximum file descriptors.

```
dn: cn=config
changetype: modify
replace: nsslapd-idletimeout
nsslapd-idletimeout=1800
-
replace: nsslapd-maxdescriptors
nsslapd-maxdescriptors=8192
```

2. Use the **--dirsrv-config-file** parameter to pass the LDIF file to the installation script.

- a. To install an IdM server:

```
# ipa-server-install --dirsrv-config-file filename.ldif
```

- b. To install an IdM replica:

```
# ipa-replica-install --dirsrv-config-file filename.ldif
```

### Additional resources

- Options for the **ipa-server-install** and **ipa-replica-install** commands

## CHAPTER 8. OPTIONS FOR THE IPA-SERVER-INSTALL AND IPA-REPLICA-INSTALL COMMANDS

The **ipa-server-install** and **ipa-replica-install** commands have numerous arguments you can use to supply additional information that is not requested during an interactive installation. You can also use these options to script an unattended installation. The following table displays some of the most common options. For an exhaustive list of options, see the **ipa-server-install(1)** and **ipa-replica-install(1)** man pages.

Table 8.1. Options for the **ipa-server-install** and **ipa-replica-install** commands

Argument	Description
<b>-a &lt;ipa_admin_password&gt;</b>	The password for the <b>admin</b> IdM administrator account to authenticate to the Kerberos realm.
<b>-d, --debug</b>	Enables debug logging for more verbose output.
<b>--dirsrv-config-file &lt;LDIF_file_name&gt;</b>	The path to an LDIF file used to modify the configuration of the directory server instance.
<b>--hostname=server.idm.example.com</b>	The fully-qualified domain name of the IdM server machine. Only numbers, lowercase alphabetic characters, and hyphens (-) are allowed.
<b>--idmax=&lt;number&gt;</b>	Sets the upper bound for IDs which can be assigned by the IdM server. The default value is the ID start value plus 199999.
<b>--idstart=&lt;number&gt;</b>	Sets the lower bound, or starting value, for IDs which can be assigned by the IdM server. The default value is randomly selected.
<b>--ip-address 127.0.0.1</b>	Specifies the IP address of the server. This option only accepts IP addresses associated with the local interface.
<b>-n example.com</b>	The name of the LDAP server domain to use for the IdM domain. This is usually based on the IdM server's hostname.
<b>-p &lt;directory_manager_password&gt;</b>	The password for the superuser, <b>cn=Directory Manager</b> , for the LDAP service.
<b>-P &lt;kerberos_main_password&gt;</b>	The password for the KDC administrator. If you do not specify a value, this is randomly generated.
<b>-r &lt;KERBEROS_REALM_NAME&gt;</b>	The name of the Kerberos realm to create for the IdM domain in uppercase, such as <b>EXAMPLE.COM</b> .

Argument	Description
<b>--setup-ca</b>	Install and configure a CA on this replica. If a CA is not configured, certificate operations are forwarded to another replica with a CA installed.
<b>--forwarder=192.0.2.1</b>	Gives a DNS forwarder to use with the DNS service. To specify more than one forwarder, use this option multiple times.
<b>--no-forwarders</b>	Uses root servers with the DNS service instead of forwarders.
<b>--no-reverse</b>	Does not create a reverse DNS zone when the DNS domain is set up. (If a reverse DNS zone is already configured, then that existing reverse DNS zone is used.) If this option is not used, then the default value is true, which assumes that reverse DNS should be configured by the installation script.
<b>--setup-dns</b>	Tells the installation script to set up a DNS service within the IdM domain. Using an integrated DNS service is optional, so if this option is not passed with the installation script, then no DNS is configured.
<b>-U, --unattended</b>	Enable an unattended installation session that does not prompt for user input.

#### Additional resources

- **ipa-server-install(1)** man page
- **ipa-replica-install(1)** man page

## CHAPTER 9. TROUBLESHOOTING IDM SERVER INSTALLATION

The following sections describe how to gather information about a failing IdM server installation, and how to resolve common installation issues.

- [Reviewing IdM server installation error logs](#)
- [Reviewing IdM CA installation errors](#)
- [Removing a partial IdM server installation](#)

### 9.1. REVIEWING IDM SERVER INSTALLATION ERROR LOGS

When you install an Identity Management (IdM) server, debugging information is appended to the following log files:

- **/var/log/ipaserver-install.log**
- **/var/log/httpd/error\_log**
- **/var/log/dirsrv/slapd-*INSTANCE-NAME*/access**
- **/var/log/dirsrv/slapd-*INSTANCE-NAME*/errors**

The last lines of the log files report success or failure, and the **ERROR** and **DEBUG** entries provide additional context.

To troubleshoot a failing IdM server installation, review the errors at the end of the log files and use this information to resolve any corresponding issues.

#### Prerequisites

- You must have **root** privileges to display the contents of IdM log files.

#### Procedure

1. Use the **tail** command to display the last lines of a log file. The following example displays the last 10 lines of **/var/log/ipaserver-install.log**.

```
[user@server ~]$ sudo tail -n 10 /var/log/ipaserver-install.log
[sudo] password for user:
value = gen.send(prev_value)
File "/usr/lib/python3.6/site-packages/ipapython/install/common.py", line 65, in _install
for unused in self._installer(self.parent):
File "/usr/lib/python3.6/site-packages/ipaserver/install/server/init.py", line 564, in main
master_install(self)
File "/usr/lib/python3.6/site-packages/ipaserver/install/server/install.py", line 291, in decorated
raise ScriptError()

2020-05-27T22:59:41Z DEBUG The ipa-server-install command failed, exception:
ScriptError:
2020-05-27T22:59:41Z ERROR The ipa-server-install command failed. See
/var/log/ipaserver-install.log for more information
```

- To review a log file interactively, open the end of the log file using the **less** utility and use the ↑ and ↓ arrow keys to navigate. The following example opens the **/var/log/ipaserver-install.log** file interactively.

```
[user@server ~]$ sudo less -N +G /var/log/ipaserver-install.log
```

- Gather additional troubleshooting information by repeating this review process with the remaining log files.

```
[user@server ~]$ sudo less -N +G /var/log/httpd/error_log
```

```
[user@server ~]$ sudo less -N +G /var/log/dirsrv/slapd-INSTANCE-NAME/access
```

```
[user@server ~]$ sudo less -N +G /var/log/dirsrv/slapd-INSTANCE-NAME/errors
```

### Additional resources

- If you are unable to resolve a failing IdM server installation, and you have a Red Hat Technical Support subscription, open a Technical Support case at the [Red Hat Customer Portal](#) and provide an **sosreport** of the server.
- The **sosreport** utility collects configuration details, logs and system information from a RHEL system. For more information on the **sosreport** utility, see [What is an sosreport and how to create one in Red Hat Enterprise Linux?](#).

## 9.2. REVIEWING IDM CA INSTALLATION ERRORS

When you install the Certificate Authority (CA) service on an Identity Management (IdM) server, debugging information is appended to the following locations (in order of recommended priority):

Location	Description
<b>/var/log/pki/pki-ca-spawn.\$TIME_OF_INSTALLATION.log</b>	High-level issues and Python traces for the <b>pkispawn</b> installation process
<b>journalctl -u pki-tomcatd@pki-tomcat</b> output	Errors from the <b>pki-tomcatd@pki-tomcat</b> service
<b>/var/log/pki/pki-tomcat/ca/debug.\$DATE.log</b>	Large JAVA stacktraces of activity in the core of the Public Key Infrastructure (PKI) product
<b>/var/log/pki/pki-tomcat/ca/signedAudit/ca_audit</b> log file	Audit log of the PKI product
<ul style="list-style-type: none"> <li><b>/var/log/pki/pki-tomcat/ca/system</b></li> <li><b>/var/log/pki/pki-tomcat/ca/transactions</b></li> <li><b>/var/log/pki/pki-tomcat/catalina.\$DATE.log</b></li> </ul>	Low-level debug data of certificate operations for service principals, hosts, and other entities that use certificates





## NOTE

If a full IdM server installation fails while installing the optional CA component, no details about the CA are logged; a message is logged in the **/var/log/ipaserver-install.log** file indicating that the overall installation process failed. Red Hat recommends reviewing the log files listed above for details specific to the CA installation failure.

The only exception to this behavior is when you are installing the CA service and the root CA is an external CA. If there is an issue with the certificate from the external CA, errors are logged in **/var/log/ipaserver-install.log**.

To troubleshoot a failing IdM CA installation, review the errors at the end of these log files and use this information to resolve any corresponding issues.

## Prerequisites

- You must have **root** privileges to display the contents of IdM log files.

## Procedure

- To review a log file interactively, open the end of the log file using the **less** utility and use the ↑ and ↓ arrow keys to navigate, while searching for **ScriptError** entries. The following example opens **/var/log/pki/pki-ca-spawn.\$TIME\_OF\_INSTALLATION.log**.

```
[user@server ~]$ sudo less -N +G /var/log/pki/pki-ca-spawn.20200527185902.log
```

- Gather additional troubleshooting information by repeating this review process with all the log files listed above.

## Additional resources

- If you are unable to resolve a failing IdM server installation, and you have a Red Hat Technical Support subscription, open a Technical Support case at the [Red Hat Customer Portal](#) and provide an **sosreport** of the server.
- The **sosreport** utility collects configuration details, logs and system information from a RHEL system. For more information on the **sosreport** utility, see [What is an sosreport and how to create one in Red Hat Enterprise Linux?](#).

## 9.3. REMOVING A PARTIAL IDM SERVER INSTALLATION

If an IdM server installation fails, some configuration files can be left behind. Additional attempts to install the IdM server fail and the installation script reports that IPA is already configured.

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

The log file for this installation can be found in **/var/log/ipaserver-install.log**

**IPA server is already configured on this system.**

If you want to reinstall the IPA server, **please uninstall it first using 'ipa-server-install --uninstall'**. The ipa-server-install command failed. See **/var/log/ipaserver-install.log** for more information

To resolve this issue, uninstall the partial IdM server configuration and retry the installation process.

## Prerequisites

- You must have **root** privileges.

## Procedure

1. Uninstall the IdM server software from the host you are trying to configure as an IdM server.

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

2. If you continue to experience difficulty installing an IdM server because of repeated failed installations, reinstall the operating system.

One of the requirements for installing an IdM server is a clean system without any customization. Failed installations may have compromised the integrity of the host by unexpectedly modifying system files.

## Additional resources

- For additional details on uninstalling an IdM server, see [Uninstalling an IdM server](#).
- If installation attempts fail after repeated uninstallation attempts, and you have a Red Hat Technical Support subscription, open a Technical Support case at the [Red Hat Customer Portal](#) and provide an **sosreport** of the server.
- The **sosreport** utility collects configuration details, logs and system information from a RHEL system. For more information on the **sosreport** utility, see [What is an sosreport and how to create one in Red Hat Enterprise Linux?](#).

## 9.4. ADDITIONAL RESOURCES

- To troubleshoot installing an IdM replica, see [Troubleshooting IdM replica installation](#).
- To troubleshoot installing an IdM client, see [Troubleshooting IdM client installation](#).

## CHAPTER 10. UNINSTALLING AN IDM SERVER

This procedure describes how you can uninstall an Identity Management (IdM) server named **server123.idm.example.com**.

### Prerequisites

- You have **root** access to the server.
- On the server, you have obtained the IdM administrator's credentials.

### Procedure

1. If your IdM environment uses integrated DNS, ensure that **server123.idm.example.com** is not the only **enabled** DNS server:

```
[root@server123 ~]# ipa server-role-find --role 'DNS server'
-----
2 server roles matched
-----
Server name: server456.idm.example.com
Role name: DNS server
Role status: enabled
[...]
-----
Number of entries returned 2
-----
```

If **server123** is the only remaining DNS server in the topology, add the DNS server role to another IdM server. For more information, see the **ipa-dns-install(1)** man page.

2. If your IdM environment uses an integrated certificate authority (CA):
  - a. Ensure that **server123.idm.example.com** is not the only **enabled** CA server:

```
[root@server123 ~]# ipa server-role-find --role 'CA server'
-----
2 server roles matched
-----
Server name: server123.idm.example.com
Role name: CA server
Role status: enabled

Server name: r8server.idm.example.com
Role name: CA server
Role status: enabled
-----
Number of entries returned 2
-----
```

If **server123** is the only remaining CA server in the topology, add the CA server role to another IdM server. For more information, see the **ipa-ca-install(1)** man page.

- b. If you have enabled vaults in your IdM environment, ensure that **server123.idm.example.com** is not the only **enabled** Key Recovery Authority (KRA) server:

```
[root@server123 ~]# ipa server-role-find --role 'KRA server'
-----
2 server roles matched
-----
Server name: server123.idm.example.com
Role name: KRA server
Role status: enabled

Server name: r8server.idm.example.com
Role name: KRA server
Role status: enabled
-----
Number of entries returned 2
-----
```

If server123 is the only remaining KRA server in the topology, add the KRA server role to another IdM server. For more information, see **man ipa-kra-install(1)**.

- c. Ensure that server123.idm.example.com is not the CA renewal server:

```
[root@server123 ~]# ipa config-show | grep 'CA renewal'
IPA CA renewal master: r8server.idm.example.com
```

If server123 is the CA renewal server, see [Changing and resetting IdM CA renewal server](#) for more information on how to move the CA renewal server role to another server.

- d. Ensure that server123.idm.example.com is not the current certificate revocation list (CRL) publisher:

```
[root@server123 ~]# ipa-crlgen-manage status
CRL generation: disabled
```

If the output shows that CRL generation is enabled on server123, see [Generating CRL on an IdM CA server](#) for more information on how to move the CRL publisher role to another server.

3. Connect to another IdM server in the topology:

```
$ ssh idm_user@another_server
```

4. On the server, obtain the IdM administrator's credentials:

```
[idm_user@another_server ~]$ kinit admin
```

5. On the server, delete server123.idm.example.com from the topology:

```
[root@another_server ~]$ ipa server-del server123.idm.example.com
```

6. Return to server123.idm.example.com and uninstall the existing IdM installation:

```
[root@server123 ~]# ipa-server-install --uninstall
...
Are you sure you want to continue with the uninstall procedure? [no]: yes
```

7. Make sure all name server (NS) DNS records pointing to server123.idm.example.com are deleted from your DNS zones. This applies regardless of whether you use integrated DNS managed by IdM or external DNS. For more information on how to delete DNS records from IdM, see [Deleting DNS records in the IdM CLI](#) .

## CHAPTER 11. RENAMING AN IDM SERVER

You cannot change the host name of an existing Identity Management (IdM) server. However, you can replace the server with a replica of a different name.

### Procedure

1. Install a new replica that will replace the existing server, ensuring the replica has the required host name and IP address. For details, see [Installing an IdM replica](#).



### IMPORTANT

If the server you are uninstalling is the certificate revocation list (CRL) publisher server, make another server the CRL publisher server before proceeding. For details how this is done in the context of a migration procedure, see the following sections:

- [Stopping CRL generation on a RHEL 7 IdM CA server](#)
- [Starting CRL generation on the new RHEL 8 IdM CA server](#)

2. Stop the existing IdM server instance.

```
[root@old_server ~]# ipactl stop
```

3. Uninstall the existing server as described in [Uninstalling an IdM server](#).

## CHAPTER 12. PREPARING THE SYSTEM FOR IDM CLIENT INSTALLATION

This chapter describes the conditions your system must meet to install an Identity Management (IdM) client.

### 12.1. DNS REQUIREMENTS FOR IDM CLIENTS

Client installer by default tries to search for **\_ldap.\_tcp.DOMAIN** DNS SRV records for all domains that are parent to its hostname. For example, if a client machine has a hostname **client1.idm.example.com**, the installer will try to retrieve an IdM server hostname from **\_ldap.\_tcp.idm.example.com**, **\_ldap.\_tcp.example.com** and **\_ldap.\_tcp.com** DNS SRV records, respectively. The discovered domain is then used to configure client components (for example, SSSD and Kerberos 5 configuration) on the machine.

However, the hostnames of IdM clients are not required to be part of the primary DNS domain. If the client machine hostname is not in a subdomain of an IdM server, pass the IdM domain as the **--domain** option of the **ipa-client-install** command. In that case, after the installation of the client, both SSSD and Kerberos components will have the domain set in their configuration files and will use it to autodiscover IdM servers.

#### Additional resources

- For details on DNS requirements in IdM, see [Host name and DNS requirements for IdM](#).

### 12.2. PORT REQUIREMENTS FOR IDM CLIENTS

Identity Management (IdM) clients connect to a number of ports on IdM servers to communicate with their services.

On IdM client, these ports must be open *in the outgoing direction*. If you are using a firewall that does not filter outgoing packets, such as **firewalld**, the ports are already available in the outgoing direction.

#### Additional resources

- For information about which specific ports are used, see [Port requirements for IdM](#).

### 12.3. IPV6 REQUIREMENTS FOR IDM CLIENTS

Identity Management (IdM) does not require the **IPv6** protocol to be enabled in the kernel of the host that you want to enroll into IdM. For example, if your internal network only uses the **IPv4** protocol, you can configure the System Security Services Daemon (SSSD) to only use **IPv4** to communicate with the IdM server. You can do this by inserting the following line into the **[domain/NAME]** section of the **/etc/sss/sss.conf** file:

```
lookup_family_order = ipv4_only
```

#### Additional resources

- For more information on the **lookup\_family\_order** option, see the **sss.conf(5)** man page.

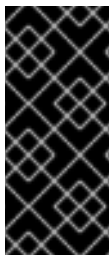
### 12.4. PACKAGES REQUIRED TO INSTALL AN IDM CLIENT

In RHEL8, the packages necessary for installing an Identity Management (IdM) client are shipped as a module. Two IdM streams provide IdM client packages:

- the **idm:client** stream. For details, see [Section 12.4.1, “Installing IdM client packages from the idm:client stream”](#).
- the **idm:DL1** stream. For details, see [Section 12.4.2, “Installing IdM client packages from the idm:DL1 stream”](#).

### 12.4.1. Installing IdM client packages from the idm:client stream

The **idm:client** stream is the default stream of the **idm** module. Use this stream to download the IdM client packages if you do not need to install server components on your machine. Using the **idm:client** stream is especially recommended if you need to consistently use IdM client software that is supported long-term, provided you do not need server components, too.



#### IMPORTANT

When switching to the **idm:client** stream after you previously enabled the **idm:DL1** stream and downloaded packages from it, you need to first explicitly remove all the relevant installed content and disable the **idm:DL1** stream before enabling the **idm:client** stream. Trying to enable a new stream without disabling the current one results in an error. For details on how to proceed, see [Switching to a later stream](#).

#### Procedure

- To download the packages necessary for installing an IdM client:

```
# yum module install idm
```

### 12.4.2. Installing IdM client packages from the idm:DL1 stream

The **idm:DL1** stream needs to be enabled before you can download packages from it. Use this stream to download the IdM client packages if you need to install IdM server components on your machine.



#### IMPORTANT

When switching to the **idm:DL1** stream after you previously enabled the **idm:client** stream and downloaded packages from it, you need to first explicitly remove all the relevant installed content and disable the **idm:client** stream before enabling the **idm:DL1** stream. Trying to enable a new stream without disabling the current one results in an error. For details on how to proceed, see [Switching to a later stream](#).

#### Procedure

1. To switch to the RPMs delivered through the **idm:DL1** stream:

```
# yum module enable idm:DL1  
# yum distro-sync
```

2. To download the packages necessary for installing an IdM client:

```
# yum module install idm:DL1/client
```



## CHAPTER 13. INSTALLING AN IDM CLIENT: BASIC SCENARIO

The following sections describe how to configure a system as an Identity Management (IdM) client by using the **ipa-client-install** utility. Configuring a system as an IdM client enrolls it into an IdM domain and enables the system to use IdM services on IdM servers in the domain.

To install an Identity Management (IdM) client successfully, you must provide credentials that can be used to enroll the client. The following authentication methods are available:

- For installing a client interactively using privileged user's credentials (the default option), see [Installing a client by using user credentials: Interactive installation](#).
- For installing a client interactively using a one-time password, see [Installing a client by using a one-time password: Interactive installation](#).
- For installing a client noninteractively using either a privileged user's credentials, a one-time password or a keytab from a previous enrollment, see [Installing a client: Non-interactive installation](#).

### 13.1. PREREQUISITES

Before you start installing the IdM client, make sure that you have met all the prerequisites. See [Preparing the system for IdM client installation](#).

### 13.2. INSTALLING A CLIENT BY USING USER CREDENTIALS: INTERACTIVE INSTALLATION

This procedure describes installing an Identity Management (IdM) client interactively by using the credentials of an authorized user to enroll the system into the domain.

#### Prerequisites

- Ensure you have the credentials of a user authorized to enroll clients into the IdM domain. This could be, for example, a **hostadmin** user with the Enrollment Administrator role.

#### Procedure

1. Run the **ipa-client-install** utility on the system that you want to configure as an IdM client.

```
# ipa-client-install --mkhomedir
```

Add the **--enable-dns-updates** option to update the DNS records with the IP address of the client system if either of the following conditions applies:

- The IdM server the client will be enrolled with was installed with integrated DNS
- The DNS server on the network accepts DNS entry updates with the GSS-TSIG protocol

```
# ipa-client-install --enable-dns-updates --mkhomedir
```

Enabling DNS updates is useful if your client:

- has a dynamic IP address issued using the Dynamic Host Configuration Protocol

- has a static IP address but it has just been allocated and the IdM server does not know about it
2. The installation script attempts to obtain all the required settings, such as DNS records, automatically.
    - If the SRV records are set properly in the IdM DNS zone, the script automatically discovers all the other required values and displays them. Enter **yes** to confirm.

```
Client hostname: client.example.com
Realm: EXAMPLE.COM
DNS Domain: example.com
IPA Server: server.example.com
BaseDN: dc=example,dc=com
```

```
Continue to configure the system with these values? [no]: yes
```

- To install the system with different values, enter **no**. Then run **ipa-client-install** again, and specify the required values by adding command-line options to **ipa-client-install**, for example:
  - **--hostname**
  - **--realm**
  - **--domain**
  - **--server**
  - **--mkhomedir**



### IMPORTANT

The fully qualified domain name must be a valid DNS name:

- Only numbers, alphabetic characters, and hyphens (-) are allowed. For example, underscores are not allowed and can cause DNS failures.
- The host name must be all lower-case. No capital letters are allowed.

- If the script fails to obtain some settings automatically, it prompts you for the values.
3. The script prompts for a user whose identity will be used to enroll the client. This could be, for example, a **hostadmin** user with the Enrollment Administrator role:

```
User authorized to enroll computers: hostadmin
Password for hostadmin@EXAMPLE.COM:
```

4. The installation script now configures the client. Wait for the operation to complete.

```
Client configuration complete.
```

### Additional resources

- For details on how the client installation script searches for the DNS records, see the **DNS Autodiscovery** section in the **ipa-client-install(1)** man page.

### 13.3. INSTALLING A CLIENT BY USING A ONE-TIME PASSWORD: INTERACTIVE INSTALLATION

This procedure describes installing an Identity Management (IdM) client interactively by using a one-time password to enroll the system into the domain.

#### Prerequisites

- On a server in the domain, add the future client system as an IdM host. Use the **--random** option with the **ipa host-add** command to generate a one-time random password for the enrollment.

```
$ ipa host-add client.example.com --random
```

```
-----  
Added host "client.example.com"  
-----
```

```
Host name: client.example.com
```

```
Random password: W5YpARl=7M.n
```

```
Password: True
```

```
Keytab: False
```

```
Managed by: server.example.com
```



#### NOTE

The generated password will become invalid after you use it to enroll the machine into the IdM domain. It will be replaced with a proper host keytab after the enrollment is finished.

#### Procedure

1. Run the **ipa-client-install** utility on the system that you want to configure as an IdM client. Use the **--password** option to provide the one-time random password. Because the password often contains special characters, enclose it in single quotes (').

```
# ipa-client-install --mkhomedir --password=password
```

Add the **--enable-dns-updates** option to update the DNS records with the IP address of the client system if either of the following conditions applies:

- The IdM server the client will be enrolled with was installed with integrated DNS
- The DNS server on the network accepts DNS entry updates with the GSS-TSIG protocol

```
# ipa-client-install --password 'W5YpARl=7M.n' --enable-dns-updates --mkhomedir
```

Enabling DNS updates is useful if your client:

- has a dynamic IP address issued using the Dynamic Host Configuration Protocol
- has a static IP address but it has just been allocated and the IdM server does not know about it

- The installation script attempts to obtain all the required settings, such as DNS records, automatically.

- If the SRV records are set properly in the IdM DNS zone, the script automatically discovers all the other required values and displays them. Enter **yes** to confirm.

```
Client hostname: client.example.com
Realm: EXAMPLE.COM
DNS Domain: example.com
IPA Server: server.example.com
BaseDN: dc=example,dc=com
```

Continue to configure the system with these values? [no]: yes

- To install the system with different values, enter **no**. Then run **ipa-client-install** again, and specify the required values by adding command-line options to **ipa-client-install**, for example:
  - hostname**
  - realm**
  - domain**
  - server**
  - mkhomedir**



### IMPORTANT

The fully qualified domain name must be a valid DNS name:

- Only numbers, alphabetic characters, and hyphens (-) are allowed. For example, underscores are not allowed and can cause DNS failures.
  - The host name must be all lower-case. No capital letters are allowed.
- If the script fails to obtain some settings automatically, it prompts you for the values.

- The installation script now configures the client. Wait for the operation to complete.

```
Client configuration complete.
```

### Additional resources

- For details on how the client installation script searches for the DNS records, see the **DNS Autodiscovery** section in the **ipa-client-install(1)** man page.

## 13.4. INSTALLING A CLIENT: NON-INTERACTIVE INSTALLATION

For a non-interactive installation, you must provide all required information to the **ipa-client-install** utility using command-line options. The following sections describe the minimum required options for a non-interactive installation.

### Options for the intended authentication method for client enrollment

The available options are:

- **--principal** and **--password** to specify the credentials of a user authorized to enroll clients
- **--random** to specify a one-time random password generated for the client
- **--keytab** to specify the keytab from a previous enrollment

### The option for unattended installation

The **--unattended** option lets the installation run without requiring user confirmation.

If the SRV records are set properly in the IdM DNS zone, the script automatically discovers all the other required values. If the script cannot discover the values automatically, provide them using command-line options, such as:

- **--hostname** to specify a static fully qualified domain name (FQDN) for the client machine.



### IMPORTANT

The FQDN must be a valid DNS name:

- Only numbers, alphabetic characters, and hyphens (-) are allowed. For example, underscores are not allowed and can cause DNS failures.
  - The host name must be all lower-case. No capital letters are allowed.
- **--domain** to specify the primary DNS domain of an existing IdM deployment, e.g. example.com. The name is a lowercase version of the IdM Kerberos realm name.
- **--server** to specify the FQDN of the IdM server to connect to. When this option is used, DNS autodiscovery for Kerberos is disabled and a fixed list of KDC and Admin servers is configured. Under normal circumstances, this option is not needed as the list of servers is retrieved from the primary IdM DNS domain.
- **--realm** to specify the Kerberos realm of an existing IdM deployment. Usually it is an uppercase version of the primary DNS domain used by the IdM installation. Under normal circumstances, this option is not needed as the realm name is retrieved from the IdM server.

An example of a basic **ipa-client-install** command for non-interactive installation:

```
# ipa-client-install --password 'W5YpARl=7M.n' --mkhomedir --unattended
```

An example of an **ipa-client-install** command for non-interactive installation with more options specified:

```
# ipa-client-install --password 'W5YpARl=7M.n' --domain idm.example.com --server  
server.idm.example.com --realm IDM.EXAMPLE.COM --mkhomedir --unattended
```

### Additional resources

- For a complete list of options accepted by **ipa-client-install**, see the **ipa-client-install(1)** man page.

## 13.5. REMOVING PRE-IDM CONFIGURATION AFTER INSTALLING A CLIENT

The **ipa-client-install** script does not remove any previous LDAP and System Security Services Daemon (SSSD) configuration from the **/etc/openldap/ldap.conf** and **/etc/sss/sss.conf** files. If you modified the configuration in these files before installing the client, the script adds the new client values, but comments them out. For example:

```
BASE dc=example,dc=com
URI ldap://ldap.example.com

#URI ldaps://server.example.com # modified by IPA
#BASE dc=ipa,dc=example,dc=com # modified by IPA
```

To apply the new Identity Management (IdM) configuration values:

1. Open **/etc/openldap/ldap.conf** and **/etc/sss/sss.conf**.
2. Delete the previous configuration.
3. Uncomment the new IdM configuration.
4. Server processes that rely on system-wide LDAP configuration might require a restart to apply the changes. Applications that use **openldap** libraries typically import the configuration when started.

## 13.6. TESTING AN IDM CLIENT

The command-line interface informs you that the **ipa-client-install** was successful, but you can also do your own test.

To test that the Identity Management (IdM) client can obtain information about users defined on the server, check that you are able to resolve a user defined on the server. For example, to check the default **admin** user:

```
[user@client ~]$ id admin
uid=1254400000(admin) gid=1254400000(admins) groups=1254400000(admins)
```

To test that authentication works correctly, **su** to a root user from a non-root user:

```
[user@client ~]$ su -
Last login: Thu Oct 18 18:39:11 CEST 2018 from 192.168.122.1 on pts/0
[root@client ~]#
```

## 13.7. CONNECTIONS PERFORMED DURING AN IDM CLIENT INSTALLATION

[Requests performed during an IdM client installation](#) lists the operations performed by **ipa-client-install**, the Identity Management (IdM) client installation tool.

**Table 13.1. Requests performed during an IdM client installation**

Operation	Protocol used	Purpose
DNS resolution against the DNS resolvers configured on the client system	DNS	To discover the IP addresses of IdM servers; (optionally) to add A/AAAA and SSHFP records
Requests to ports 88 (TCP/TCP6 and UDP/UDP6) on an IdM replica	Kerberos	To obtain a Kerberos ticket
JSON-RPC calls to the IdM Apache-based web-service on discovered or configured IdM servers	HTTPS	IdM client enrollment; retrieval of CA certificate chain if LDAP method fails; request for a certificate issuance if required
Requests over TCP/TCP6 to ports 389 on IdM servers, using SASL GSSAPI authentication, plain LDAP, or both	LDAP	IdM client enrollment; identity retrieval by SSSD processes; Kerberos key retrieval for the host principal
Network time protocol (NTP) discovery and resolution (optionally)	NTP	To synchronize time between the client system and an NTP server

## 13.8. IDM CLIENT'S COMMUNICATIONS WITH THE SERVER DURING POST-INSTALLATION DEPLOYMENT

The client side of Identity Management (IdM) framework is implemented with two different applications:

- the **ipa** command-line interface (CLI)
- the browser-based Web UI

The browser-based Web UI is optional.

[CLI post-installation operations](#) shows the operations performed by the CLI during an IdM client post-installation deployment. [Web UI post-installation operations](#) shows the operations performed by the Web UI during an IdM client post-installation deployment.

Two daemons run on the IdM client, the **System Security Services Daemon** (SSSD) and **certmonger**. [SSSD communication patterns](#) and [Certmonger communication patterns](#) describe how these daemons communicate with the services available on the IdM and Active Directory servers.

**Table 13.2. CLI post-installation operations**

Operation	Protocol used	Purpose
DNS resolution against the DNS resolvers configured on the client system	DNS	To discover the IP addresses of IdM servers

Operation	Protocol used	Purpose
Requests to ports 88 (TCP/TCP6 and UDP/UDP6) and 464 (TCP/TCP6 and UDP/UDP6) on an IdM replica	Kerberos	To obtain a Kerberos ticket; change a Kerberos password; authenticate to the IdM Web UI
JSON-RPC calls to the IdM Apache-based web-service on discovered or configured IdM servers	HTTPS	any <b>ipa</b> utility usage

**Table 13.3. Web UI post-installation operations**

Operation	Protocol used	Purpose
JSON-RPC calls to the IdM Apache-based web-service on discovered or configured IdM servers	HTTPS	To retrieve the IdM Web UI pages

### 13.8.1. SSSD communication patterns

The System Security Services Daemon (SSSD) is a system service to access remote directories and authentication mechanisms. If configured on an Identity Management IdM client, it connects to the IdM server, which provides authentication, authorization and other identity and policy information. If the IdM server is in a trust relationships with Active Directory (AD), SSSD also connects to AD to perform authentication for AD users using the Kerberos protocol. By default, SSSD uses Kerberos to authenticate any non-local user. In special situations, SSSD might be configured to use the LDAP protocol instead.

The SSSD can be configured to communicate with multiple servers. [Communication patterns of SSSD on IdM clients when talking to IdM servers](#) and [Communication patterns of SSSD on IdM servers acting as trust agents when talking to Active Directory Domain Controllers](#) show common communication patterns for SSSD in IdM.

**Table 13.4. Communication patterns of SSSD on IdM clients when talking to IdM servers**

Operation	Protocol used	Purpose
DNS resolution against the DNS resolvers configured on the client system	DNS	To discover the IP addresses of IdM servers
Requests to ports 88 (TCP/TCP6 and UDP/UDP6), 464 (TCP/TCP6 and UDP/UDP6), and 749 (TCP/TCP6) on an Identity Management replica and Active Directory domain controllers	Kerberos	To obtain a Kerberos ticket; to change a Kerberos password



Operation	Protocol used	Purpose
Requests over TCP/TCP6 to ports 389 on IdM servers, using SASL GSSAPI authentication, plain LDAP, or both	LDAP	To obtain information about IdM users and hosts, download HBAC and sudo rules, automount maps, the SELinux user context, public SSH keys, and other information stored in IdM LDAP
(optionally) In case of smart-card authentication, requests to the Online Certificate Status Protocol (OCSP) responder, if it is configured. This often is done via port 80, but it depends on the actual value of the OCSP responder URL in a client certificate.	HTTP	To obtain information about the status of the certificate installed in the smart card

**Table 13.5. Communication patterns of SSSD on IdM servers acting as trust agents when talking to Active Directory Domain Controllers**

Operation	Protocol used	Purpose
DNS resolution against the DNS resolvers configured on the client system	DNS	To discover the IP addresses of IdM servers
Requests to ports 88 (TCP/TCP6 and UDP/UDP6), 464 (TCP/TCP6 and UDP/UDP6), and 749 (TCP/TCP6) on an Identity Management replica and Active Directory domain controllers	Kerberos	To obtain a Kerberos ticket; change a Kerberos password; administer Kerberos remotely
Requests to ports 389 (TCP/TCP6 and UDP/UDP6) and 3268 (TCP/TCP6)	LDAP	To query Active Directory user and group information; to discover Active Directory domain controllers
(optionally) In case of smart-card authentication, requests to the Online Certificate Status Protocol (OCSP) responder, if it is configured. This often is done via port 80, but it depends on the actual value of the OCSP responder URL in a client certificate.	HTTP	To obtain information about the status of the certificate installed in the smart card

### 13.8.2. Certmonger communication patterns

**Certmonger** is a daemon running on Identity Management (IdM) servers and IdM clients to allow a timely renewal of SSL certificates associated with the services on the host. The [Table 13.6, “Certmonger communication patterns”](#) shows the operations performed by IdM client’s **certmonger** utility on IdM servers.

**Table 13.6. Certmonger communication patterns**

Operation	Protocol used	Purpose
DNS resolution against the DNS resolvers configured on the client system	DNS	To discover the IP addresses of IdM servers
Requests to ports 88 (TCP/TCP6 and UDP/UDP6) and 464 (TCP/TCP6 and UDP/UDP6) on an IdM replica	Kerberos	To obtain a Kerberos ticket
JSON-RPC calls to the IdM Apache-based web-service on discovered or configured IdM servers	HTTPS	To request new certificates
Access over port 8080 (TCP/TCP6) on the IdM server	HTTP	To obtain an Online Certificate Status Protocol (OCSP) responder and certificate status
(on the first installed server or on the server where certificate tracking has been transferred) Access over port 8443 (TCP/TCP6) on the IdM server	HTTPS	To administer the Certificate Authority on the IdM server (only during IdM server and replica installation)

## CHAPTER 14. INSTALLING AN IDM CLIENT WITH KICKSTART

A Kickstart enrollment automatically adds a new system to the Identity Management (IdM) domain at the time Red Hat Enterprise Linux is installed.

### 14.1. INSTALLING A CLIENT WITH KICKSTART

This procedure describes how to use a Kickstart file to install an Identity Management (IdM) client.

#### Prerequisites

- Do not start the **sshd** service prior to the kickstart enrollment. Starting **sshd** before enrolling the client generates the SSH keys automatically, but the Kickstart file in [Section 14.2, “Kickstart file for client installation”](#) uses a script for the same purpose, which is the preferred solution.

#### Procedure

- Pre-create the host entry on the IdM server, and set a temporary password for the entry:

```
$ ipa host-add client.example.com --password=secret
```

The password is used by Kickstart to authenticate during the client installation and expires after the first authentication attempt. After the client is successfully installed, it authenticates using its keytab.

- Create a Kickstart file with the contents described in [Section 14.2, “Kickstart file for client installation”](#). Make sure that network is configured properly in the Kickstart file using the **network** command.
- Use the Kickstart file to install the IdM client.

### 14.2. KICKSTART FILE FOR CLIENT INSTALLATION

This section describes the contents of a kickstart file that you can use to install an Identity Management (IdM) client.

#### The **ipa-client** package in the list of packages to install

Add the **ipa-client** package to the `%packages` section of the kickstart file. For example:

```
%packages
...
ipa-client
...
```

#### Post-installation instructions for the IdM client

The post-installation instructions must include:

- An instruction for ensuring SSH keys are generated before enrollment
- An instruction to run the **ipa-client-install** utility, while specifying:
  - All the required information to access and configure the IdM domain services

- The password which you set when pre-creating the client host on the IdM server. in [Section 14.1, “Installing a client with Kickstart”](#).

For example, the post-installation instructions for a kickstart installation that uses a one-time password and retrieves the required options from the command line rather than via DNS can look like this:

```
%post --log=/root/ks-post.log

# Generate SSH keys; ipa-client-install uploads them to the IdM server by default
/usr/libexec/openssh/sshd-keygen rsa

# Run the client install script
/usr/sbin/ipa-client-install --hostname=client.example.com --domain=EXAMPLE.COM --enable-
dns-updates --mkhomedir -w secret --realm=EXAMPLE.COM --server=server.example.com
```

Optionally, you can also include other options in the Kickstart file, such as:

- For a non-interactive installation, add the **--unattended** option to **ipa-client-install**.
- To let the client installation script request a certificate for the machine:
  - Add the **--request-cert** option to **ipa-client-install**.
  - Set the system bus address to **/dev/null** for both the **getcert** and **ipa-client-install** utility in the Kickstart **chroot** environment. To do this, add these lines to the post-installation instructions in the Kickstart file before the **ipa-client-install** instruction:

```
# env DBUS_SYSTEM_BUS_ADDRESS=unix:path=/dev/null getcert list
# env DBUS_SYSTEM_BUS_ADDRESS=unix:path=/dev/null ipa-client-install
```

## 14.3. TESTING AN IDM CLIENT

The command-line interface informs you that the **ipa-client-install** was successful, but you can also do your own test.

To test that the Identity Management (IdM) client can obtain information about users defined on the server, check that you are able to resolve a user defined on the server. For example, to check the default **admin** user:

```
[user@client ~]$ id admin
uid=1254400000(admin) gid=1254400000(admins) groups=1254400000(admins)
```

To test that authentication works correctly, **su** to a root user from a non-root user:

```
[user@client ~]$ su -
Last login: Thu Oct 18 18:39:11 CEST 2018 from 192.168.122.1 on pts/0
[root@client ~]#
```

## CHAPTER 15. TROUBLESHOOTING IDM CLIENT INSTALLATION

The following sections describe how to gather information about a failing IdM client installation, and how to resolve common installation issues.

- [Reviewing IdM client installation errors](#)
- [Resolving failing to update IdM DNS records](#)
- [Resolving failing to join the IdM Kerberos realm](#)

### 15.1. REVIEWING IDM CLIENT INSTALLATION ERRORS

When you install an Identity Management (IdM) client, debugging information is appended to **/var/log/ipaclient-install.log**. If a client installation fails, the installer logs the failure and rolls back changes to undo any modifications to the host. The reason for the installation failure may not be at the end of the log file, as the installer also logs the roll back procedure.

To troubleshoot a failing IdM client installation, review lines labeled **ScriptError** in the **/var/log/ipaclient-install.log** file and use this information to resolve any corresponding issues.

#### Prerequisites

- You must have **root** privileges to display the contents of IdM log files.

#### Procedure

1. Use the **grep** utility to retrieve any occurrences of the keyword **ScriptError** from the **/var/log/ipaserver-install.log** file.

```
[user@server ~]$ sudo grep ScriptError /var/log/ipaclient-install.log
[sudo] password for user:
2020-05-28T18:24:50Z DEBUG The ipa-client-install command failed, exception:
ScriptError: One of password / principal / keytab is required.
```

2. To review a log file interactively, open the end of the log file using the **less** utility and use the ↑ and ↓ arrow keys to navigate.

```
[user@server ~]$ sudo less -N +G /var/log/ipaclient-install.log
```

#### Additional resources

- If you are unable to resolve a failing IdM client installation, and you have a Red Hat Technical Support subscription, open a Technical Support case at the [Red Hat Customer Portal](#) and provide an **sosreport** of the client.
- The **sosreport** utility collects configuration details, logs and system information from a RHEL system. For more information on the **sosreport** utility, see [What is an sosreport and how to create one in Red Hat Enterprise Linux?](#).

## 15.2. RESOLVING ISSUES IF THE CLIENT INSTALLATION FAILS TO UPDATE DNS RECORDS

The IdM client installer issues **nsupdate** commands to create PTR, SSHFP, and additional DNS records. However, the installation process fails if the client is unable to update DNS records after installing and configuring the client software.

To fix this problem, verify the configuration and review DNS errors in **/var/log/client-install.log**.

### Prerequisites

- You are using IdM DNS as the DNS solution for your IdM environment

### Procedure

1. Ensure that dynamic updates for the DNS zone the client is in are enabled:

```
[user@server ~]$ ipa dnszone-mod idm.example.com. --dynamic-update=TRUE
```

2. Ensure that the IdM server running the DNS service has port 53 opened for both TCP and UDP protocols.

```
[user@server ~]$ sudo firewall-cmd --permanent --add-port=53/tcp --add-port=53/udp
[sudo] password for user:
success
[user@server ~]$ firewall-cmd --runtime-to-permanent
success
```

3. Use the **grep** utility to retrieve the contents of **nsupdate** commands from **/var/log/client-install.log** to see which DNS record updates are failing.

```
[user@server ~]$ sudo grep nsupdate /var/log/ipaclient-install.log
```

### Additional resources

- If you are unable to resolve a failing installation, and you have a Red Hat Technical Support subscription, open a Technical Support case at the [Red Hat Customer Portal](#) and provide an **sosreport** of the client.
- The **sosreport** utility collects configuration details, logs and system information from a RHEL system. For more information on the **sosreport** utility, see [What is an sosreport and how to create one in Red Hat Enterprise Linux?](#).

## 15.3. RESOLVING ISSUES IF THE CLIENT INSTALLATION FAILS TO JOIN THE IDM KERBEROS REALM

The IdM client installation process fails if the client is unable to join the IdM Kerberos realm.

```
Joining realm failed: Failed to add key to the keytab
child exited with 11
```

```
Installation failed. Rolling back changes.
```

This failure can be caused by an empty Kerberos keytab.

### Prerequisites

- Removing system files requires **root** privileges.

### Procedure

1. Remove **/etc/krb5.keytab**.

```
[user@client ~]$ sudo rm /etc/krb5.keytab
[sudo] password for user:
[user@client ~]$ ls /etc/krb5.keytab
ls: cannot access '/etc/krb5.keytab': No such file or directory
```

2. Retry the IdM client installation.

### Additional resources

- If you are unable to resolve a failing installation, and you have a Red Hat Technical Support subscription, open a Technical Support case at the [Red Hat Customer Portal](#) and provide an **sosreport** of the client.
- The **sosreport** utility collects configuration details, logs and system information from a RHEL system. For more information on the **sosreport** utility, see [What is an sosreport and how to create one in Red Hat Enterprise Linux?](#).

## 15.4. ADDITIONAL RESOURCES

- To troubleshoot installing the first IdM server, see [Troubleshooting IdM server installation](#).
- To troubleshoot installing an IdM replica, see [Troubleshooting IdM replica installation](#).

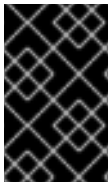
## CHAPTER 16. RE-ENROLLING AN IDM CLIENT

### 16.1. CLIENT RE-ENROLLMENT IN IDM

This section describes how to re-enroll an Identity Management (IdM) client.

If a client machine has been destroyed and lost connection with the IdM servers, for example due to the client's hardware failure, and you still have its keytab, you can re-enroll the client. In this scenario, you want to get the client back in the IdM environment with the same hostname.

During the re-enrollment, the client generates a new Kerberos key and SSH keys, but the identity of the client in the LDAP database remains unchanged. After the re-enrollment, the host has its keys and other information in the same LDAP object with the same **FQDN** as previously, before the machine's loss of connection with the IdM servers.



#### IMPORTANT

You can only re-enroll clients whose domain entry is still active. If you uninstalled a client (using **ipa-client-install --uninstall**) or disabled its host entry (using **ipa host-disable**), you cannot re-enroll it.

You cannot re-enroll a client after you have renamed it. This is because in IdM, the key attribute of the client's entry in LDAP is the client's hostname, its **FQDN**. As opposed to re-enrolling a client, during which the client's LDAP object remains unchanged, the outcome of renaming a client is that the client has its keys and other information in a different LDAP object with a new **FQDN**. Thus the only way to rename a client is to uninstall the host from IdM, change the host's hostname, and install it as an IdM client with a new name. For details on how to rename a client, see [Renaming IdM client systems](#).

#### 16.1.1. What happens during client re-enrollment

During re-enrollment, IdM:

- Revokes the original host certificate
- Creates new SSH keys
- Generates a new keytab

### 16.2. RE-ENROLLING A CLIENT BY USING USER CREDENTIALS: INTERACTIVE RE-ENROLLMENT

This procedure describes re-enrolling an Identity Management (IdM) client interactively by using the credentials of an authorized user.

1. Re-create the client machine with the same host name.
2. Run the **ipa-client-install --force-join** command on the client machine:

```
# ipa-client-install --force-join
```

3. The script prompts for a user whose identity will be used to re-enroll the client. This could be, for example, a **hostadmin** user with the Enrollment Administrator role:



User authorized to enroll computers: **hostadmin**  
 Password for **hostadmin@EXAMPLE.COM**:

### Additional resources

- For a more detailed procedure on enrolling clients by using an authorized user's credentials, see [Installing a client by using user credentials: Interactive installation](#).

## 16.3. RE-ENROLLING A CLIENT BY USING THE CLIENT KEYTAB: NON-INTERACTIVE RE-ENROLLMENT

### Prerequisites

- Back up the original client keytab file, for example in the **/tmp** or **/root** directory.

### Procedure

This procedure describes re-enrolling an Identity Management (IdM) client non-interactively by using the keytab of the client system. For example, re-enrollment using the client keytab is appropriate for an automated installation.

1. Re-create the client machine with the same host name.
2. Copy the keytab file from the backup location to the **/etc/** directory on the re-created client machine.
3. Use the **ipa-client-install** utility to re-enroll the client, and specify the keytab location with the **-keytab** option:

```
# ipa-client-install --keytab /etc/krb5.keytab
```



### NOTE

The keytab specified in the **--keytab** option is only used when authenticating to initiate the enrollment. During the re-enrollment, IdM generates a new keytab for the client.

## 16.4. TESTING AN IDM CLIENT

The command-line interface informs you that the **ipa-client-install** was successful, but you can also do your own test.

To test that the Identity Management (IdM) client can obtain information about users defined on the server, check that you are able to resolve a user defined on the server. For example, to check the default **admin** user:

```
[user@client ~]$ id admin
uid=1254400000(admin) gid=1254400000(admins) groups=1254400000(admins)
```

To test that authentication works correctly, **su** to a root user from a non-root user:

```
[user@client ~]$ su -
```

```
Last login: Thu Oct 18 18:39:11 CEST 2018 from 192.168.122.1 on pts/0
```

```
[root@client ~]#
```

## CHAPTER 17. UNINSTALLING AN IDM CLIENT

As an administrator, you can remove an Identity Management (IdM) client from the environment.

### 17.1. UNINSTALLING AN IDM CLIENT

Uninstalling a client removes the client from the Identity Management (IdM) domain, along with all of the specific IdM configuration of system services, such as System Security Services Daemon (SSSD). This restores the previous configuration of the client system.

#### Procedure

1. Enter the **ipa-client-install --uninstall** command:

```
[root@client ~]# ipa-client-install --uninstall
```

2. Optional: Check that you cannot obtain a Kerberos ticket-granting ticket (TGT) for an IdM user:

```
[root@client ~]# kinit admin
kinit: Client 'admin@EXAMPLE.COM' not found in Kerberos database while getting initial
credentials
[root@client ~]#
```

If a Kerberos TGT ticket has been returned successfully, follow the additional uninstallation steps in [Uninstalling an IdM client: additional steps after multiple past installations](#) .

3. On the client, remove old Kerberos principals from each identified keytab other than **/etc/krb5.keytab**:

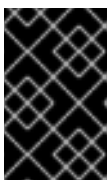
```
[root@client ~]# ipa-rmkeytab -k /path/to/keytab -r EXAMPLE.COM
```

4. On an IdM server, remove all DNS entries for the client host from IdM:

```
[root@server ~]# ipa dnsrecord-del
Record name: old-client-name
Zone name: idm.example.com
No option to delete specific record provided.
Delete all? Yes/No (default No): yes
-----
Deleted record "old-client-name"
```

5. On the IdM server, remove the client host entry from the IdM LDAP server. This removes all services and revokes all certificates issued for that host:

```
[root@server ~]# ipa host-del client.idm.example.com
```



#### IMPORTANT

Removing the client host entry from the IdM LDAP server is crucial if you think you might re-enroll the client in the future, with a different IP address or a different hostname.

## 17.2. UNINSTALLING AN IDM CLIENT: ADDITIONAL STEPS AFTER MULTIPLE PAST INSTALLATIONS

If you install and uninstall a host as an Identity Management (IdM) client multiple times, the uninstallation procedure might not restore the pre-IdM Kerberos configuration.

In this situation, you must manually remove the IdM Kerberos configuration. In extreme cases, you must reinstall the operating system.

### Prerequisites

- You have used the **ipa-client-install --uninstall** command to uninstall the IdM client configuration from the host. However, you can still obtain a Kerberos ticket-granting ticket (TGT) for an IdM user from the IdM server.
- You have checked that the **/var/lib/ipa-client/sysrestore** directory is empty and hence you cannot restore the prior-to-IdM-client configuration of the system using the files in the directory.

### Procedure

1. Check the **/etc/krb5.conf.ipa** file:

- If the contents of the **/etc/krb5.conf.ipa** file are the same as the contents of the **krb5.conf** file prior to the installation of the IdM client, you can:

- i. Remove the **/etc/krb5.conf** file:

```
# rm /etc/krb5.conf
```

- ii. Rename the **/etc/krb5.conf.ipa** file into **/etc/krb5.conf**:

```
# mv /etc/krb5.conf.ipa /etc/krb5.conf
```

- If the contents of the **/etc/krb5.conf.ipa** file are not the same as the contents of the **krb5.conf** file prior to the installation of the IdM client, you can at least restore the Kerberos configuration to the state directly after the installation of the operating system:

- i. Re-install the **krb5-libs** package:

```
# yum reinstall krb5-libs
```

As a dependency, this command will also re-install the **krb5-workstation** package and the original version of the **/etc/krb5.conf** file.

2. Remove the **var/log/ipaclient-install.log** file if present.

### Verification steps

- Try to obtain IdM user credentials. This should fail:

```
[root@r8server ~]# kinit admin
kinit: Client 'admin@EXAMPLE.COM' not found in Kerberos database while getting initial
credentials
```

```
| [root@r8server ~]#
```

The **/etc/krb5.conf** file is now restored to its factory state. As a result, you cannot obtain a Kerberos TGT for an IdM user on the host.

## CHAPTER 18. RENAMING IDM CLIENT SYSTEMS

The following sections describe how to change the host name of an Identity Management (IdM) client system.



### WARNING

Renaming a client is a manual procedure. Do not perform it unless changing the host name is absolutely required.

Renaming an IdM client involves:

1. Preparing the host. For details, see [Preparing an IdM client for its renaming](#).
2. Uninstalling the IdM client from the host. For details, see [Uninstalling a client](#).
3. Renaming the host. For details, see [Renaming a client](#).
4. Installing the IdM client on the host with the new name. For details, see [Reinstalling a client](#).
5. Configuring the host after the IdM client installation. For details, see [Re-adding services, re-generating certificates, and re-adding host groups](#).

### 18.1. PREPARING AN IDM CLIENT FOR ITS RENAMING

Before uninstalling the current client, make note of certain settings for the client. You will apply this configuration after re-enrolling the machine with a new host name.

- Identify which services are running on the machine:
  - Use the **ipa service-find** command, and identify services with certificates in the output:

```
$ ipa service-find old-client-name.example.com
```

- In addition, each host has a default *host* service which does not appear in the **ipa service-find** output. The service principal for the host service, also called a *host principal*, is **host/old-client-name.example.com**.
- For all service principals displayed by **ipa service-find old-client-name.example.com**, determine the location of the corresponding keytabs on the **old-client-name.example.com** system:

```
# find / -name "*.keytab"
```

Each service on the client system has a Kerberos principal in the form *service\_name/host\_name@REALM*, such as **ldap/old-client-name.example.com@EXAMPLE.COM**.

- Identify all host groups to which the machine belongs.

```
# ipa hostgroup-find old-client-name.example.com
```

## 18.2. UNINSTALLING AN IDM CLIENT

Uninstalling a client removes the client from the Identity Management (IdM) domain, along with all of the specific IdM configuration of system services, such as System Security Services Daemon (SSSD). This restores the previous configuration of the client system.

### Procedure

1. Enter the **ipa-client-install --uninstall** command:

```
[root@client ~]# ipa-client-install --uninstall
```

2. Optional: Check that you cannot obtain a Kerberos ticket-granting ticket (TGT) for an IdM user:

```
[root@client ~]# kinit admin
kinit: Client 'admin@EXAMPLE.COM' not found in Kerberos database while getting initial
credentials
[root@client ~]#
```

If a Kerberos TGT ticket has been returned successfully, follow the additional uninstallation steps in [Uninstalling an IdM client: additional steps after multiple past installations](#) .

3. On the client, remove old Kerberos principals from each identified keytab other than **/etc/krb5.keytab**:

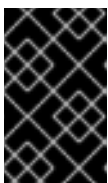
```
[root@client ~]# ipa-rmkeytab -k /path/to/keytab -r EXAMPLE.COM
```

4. On an IdM server, remove all DNS entries for the client host from IdM:

```
[root@server ~]# ipa dnsrecord-del
Record name: old-client-name
Zone name: idm.example.com
No option to delete specific record provided.
Delete all? Yes/No (default No): yes
-----
Deleted record "old-client-name"
```

5. On the IdM server, remove the client host entry from the IdM LDAP server. This removes all services and revokes all certificates issued for that host:

```
[root@server ~]# ipa host-del client.idm.example.com
```



### IMPORTANT

Removing the client host entry from the IdM LDAP server is crucial if you think you might re-enroll the client in the future, with a different IP address or a different hostname.

## 18.3. UNINSTALLING AN IDM CLIENT: ADDITIONAL STEPS AFTER MULTIPLE PAST INSTALLATIONS

If you install and uninstall a host as an Identity Management (IdM) client multiple times, the uninstallation procedure might not restore the pre-IdM Kerberos configuration.

In this situation, you must manually remove the IdM Kerberos configuration. In extreme cases, you must reinstall the operating system.

### Prerequisites

- You have used the **ipa-client-install --uninstall** command to uninstall the IdM client configuration from the host. However, you can still obtain a Kerberos ticket-granting ticket (TGT) for an IdM user from the IdM server.
- You have checked that the **/var/lib/ipa-client/sysrestore** directory is empty and hence you cannot restore the prior-to-IdM-client configuration of the system using the files in the directory.

### Procedure

1. Check the **/etc/krb5.conf.ipa** file:

- If the contents of the **/etc/krb5.conf.ipa** file are the same as the contents of the **krb5.conf** file prior to the installation of the IdM client, you can:

- i. Remove the **/etc/krb5.conf** file:

```
# rm /etc/krb5.conf
```

- ii. Rename the **/etc/krb5.conf.ipa** file into **/etc/krb5.conf**:

```
# mv /etc/krb5.conf.ipa /etc/krb5.conf
```

- If the contents of the **/etc/krb5.conf.ipa** file are not the same as the contents of the **krb5.conf** file prior to the installation of the IdM client, you can at least restore the Kerberos configuration to the state directly after the installation of the operating system:

- i. Re-install the **krb5-libs** package:

```
# yum reinstall krb5-libs
```

As a dependency, this command will also re-install the **krb5-workstation** package and the original version of the **/etc/krb5.conf** file.

2. Remove the **var/log/ipaclient-install.log** file if present.

### Verification steps

- Try to obtain IdM user credentials. This should fail:

```
[root@r8server ~]# kinit admin
kinit: Client 'admin@EXAMPLE.COM' not found in Kerberos database while getting initial
credentials
[root@r8server ~]#
```



■

The `/etc/krb5.conf` file is now restored to its factory state. As a result, you cannot obtain a Kerberos TGT for an IdM user on the host.

## 18.4. RENAMING THE HOST SYSTEM

Rename the machine as required. For example:

```
# hostnamectl set-hostname new-client-name.example.com
```

You can now re-install the Identity Management (IdM) client to the IdM domain with the new host name.

## 18.5. RE-INSTALLING AN IDM CLIENT

Install an client on your renamed host following the procedure described in [Installing a client](#).

## 18.6. RE-ADDING SERVICES, RE-GENERATING CERTIFICATES, AND RE-ADDING HOST GROUPS

1. On the Identity Management (IdM) server, add a new keytab for every service identified in the [Preparing an IdM client for its renaming](#).

```
[root@server ~]# ipa service-add service_name/new-client-name
```

2. Generate certificates for services that had a certificate assigned in the [Preparing an IdM client for its renaming](#). You can do this:
  - Using the IdM administration tools
  - Using the **certmonger** utility
3. Re-add the client to the host groups identified in the [Preparing an IdM client for its renaming](#).

## CHAPTER 19. PREPARING THE SYSTEM FOR IDM REPLICA INSTALLATION

The following links list the requirements to install an Identity Management (IdM) replica. Before the installation, make sure your system meets these requirements.

1. Ensure [the target system meets the general requirements for IdM server installation](#) .
2. Ensure [the target system meets the additional requirements for IdM replica installation](#) .
3. Authorize the target system for enrollment into the IdM domain. For more information, see one of the following sections that best fits your needs:
  - [Authorizing the installation of a replica on an IdM client](#)
  - [Authorizing the installation of a replica on a system that is not enrolled into IdM](#)

### 19.1. REPLICA VERSION REQUIREMENTS

Red Hat Enterprise Linux (RHEL) 8 replicas only work with Identity Management (IdM) servers running on RHEL 7.4 and later. Before introducing IdM replicas running on RHEL 8 into an existing deployment, upgrade all IdM servers to RHEL 7.4 or later, and change the domain level to 1.

In addition, the replica must be running the same or later version of IdM. For example:

- You have an IdM server installed on Red Hat Enterprise Linux 8 and it uses IdM 4.x packages.
- You must install the replica also on Red Hat Enterprise Linux 8 or later and use IdM version 4.x or later.

This ensures that configuration can be properly copied from the server to the replica.

For details on how to display the IdM software version, see [Methods for displaying IdM software version](#) .

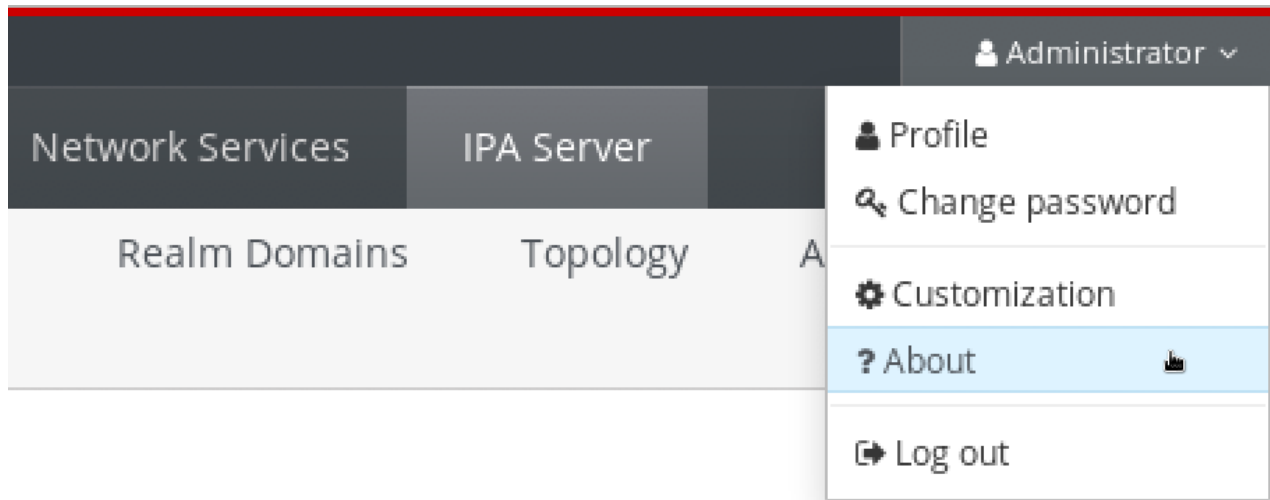
### 19.2. METHODS FOR DISPLAYING IDM SOFTWARE VERSION

You can display the IdM version number with:

- the IdM WebUI
- **ipa** commands
- **rpm** commands

#### Displaying version through the WebUI

In the IdM WebUI, the software version can be displayed by choosing **About** from the username menu at the top-right.



### Displaying version with **ipa** commands

From the command line, use the **ipa --version** command.

```
[root@server ~]# ipa --version
VERSION: 4.8.0, API_VERSION: 2.233
```

### Displaying version with **rpm** commands

If IdM services are not operating properly, you can use the **rpm** utility to determine the version number of the **ipa-server** package that is currently installed.

```
[root@server ~]# rpm -q ipa-server
ipa-server-4.8.0-11.module+el8.1.0+4247+9f3fd721.x86_64
```

## 19.3. AUTHORIZING THE INSTALLATION OF A REPLICA ON AN IDM CLIENT

When [installing a replica](#) on an existing Identity Management (IdM) client by running the **ipa-replica-install** utility, choose **Method 1** or **Method 2** below to authorize the replica installation. Choose **Method 1** if one of the following applies:

- You want a senior system administrator to perform the initial part of the procedure and a junior administrator to perform the rest.
- You want to automate your replica installation.

### Method 1: the **ipaservers** host group

1. Log in to any IdM host as IdM admin:

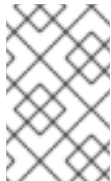
```
$ kinit admin
```

2. Add the client machine to the **ipaservers** host group:

```
$ ipa hostgroup-add-member ipaservers --hosts client.idm.example.com
Host-group: ipaservers
Description: IPA server hosts
```

```
Member hosts: server.idm.example.com, client.idm.example.com
```

```
-----  
Number of members added 1  
-----
```



#### NOTE

Membership in the **ipaservers** group grants the machine elevated privileges similar to the administrator's credentials. Therefore, in the next step, the **ipa-replica-install** utility can be run on the host successfully by a junior system administrator.

### Method 2: a privileged user's credentials

Choose one of the following methods to authorize the replica installation by providing a privileged user's credentials:

- Let Identity Management (IdM) prompt you for the credentials interactively after you start the **ipa-replica-install** utility. This is the default behavior.
- Log in to the client as a privileged user immediately before running the **ipa-replica-install** utility. The default privileged user is **admin**:

```
$ kinit admin
```

### Additional resources

- To start the installation procedure, see [Installing an IdM replica](#).
- You can use an Ansible playbook to install IdM replicas. For more information, see [Installing an Identity Management replica using an Ansible playbook](#).

## 19.4. AUTHORIZING THE INSTALLATION OF A REPLICA ON A SYSTEM THAT IS NOT ENROLLED INTO IDM

When [installing a replica](#) on a system that is not enrolled in the Identity Management (IdM) domain, the **ipa-replica-install** utility first enrolls the system as a client and then installs the replica components. For this scenario, choose **Method 1** or **Method 2** below to authorize the replica installation. Choose **Method 1** if one of the following applies:

- You want a senior system administrator to perform the initial part of the procedure and a junior administrator to perform the rest.
- You want to automate your replica installation.

### Method 1: a random password generated on an IdM server

Enter the following commands on any server in the domain:

1. Log in as the administrator.

```
$ kinit admin
```

2. Add the external system as an IdM host. Use the **--random** option with the **ipa host-add** command to generate a random one-time password to be used for the subsequent replica installation.

```
$ ipa host-add replica.example.com --random
```

```
-----  
Added host "replica.example.com"  
-----
```

```
Host name: replica.example.com  
Random password: W5YpARl=7M.n  
Password: True  
Keytab: False  
Managed by: server.example.com
```

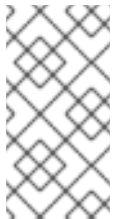
The generated password will become invalid after you use it to enroll the machine into the IdM domain. It will be replaced with a proper host keytab after the enrollment is finished.

3. Add the system to the **ipaservers** host group.

```
$ ipa hostgroup-add-member ipaservers --hosts replica.example.com
```

```
Host-group: ipaservers  
Description: IPA server hosts  
Member hosts: server.example.com, replica.example.com  
-----
```

```
Number of members added 1  
-----
```



## NOTE

Membership in the **ipaservers** group grants the machine elevated privileges similar to the administrator's credentials. Therefore, in the next step, the **ipa-replica-install** utility can be run on the host successfully by a junior system administrator that provides the generated random password.

## Method 2: a privileged user's credentials

Using this method, you authorize the replica installation by providing a privileged user's credentials. The default privileged user is **admin**.

No action is required prior to running the IdM replica installation utility. Add the principal name and password options (**--principal admin --admin-password password**) to the **ipa-replica-install** command directly during the installation.

## Additional resources

- To start the installation procedure, see [Installing an IdM replica](#).
- You can use an Ansible playbook to install IdM replicas. For more information, see [Installing an Identity Management replica using an Ansible playbook](#).

## CHAPTER 20. INSTALLING AN IDM REPLICA

The following sections describe how to install an Identity Management (IdM) replica. The replica installation process copies the configuration of the existing server and installs the replica based on that configuration.

### Prerequisites

- Ensure your system is [prepared for IdM replica installation](#).



#### IMPORTANT

If this preparation is not performed, installing an IdM replica will fail.



#### NOTE

Install one IdM replica at a time. The installation of multiple replicas at the same time is not supported.

### Procedure

For the individual types of replica installation procedures, see:

- [Section 20.1, “Installing an IdM replica with integrated DNS and a CA”](#)
- [Section 20.2, “Installing an IdM replica with integrated DNS and no CA”](#)
- [Section 20.3, “Installing an IdM replica without integrated DNS and with a CA”](#)
- [Section 20.4, “Installing an IdM replica without integrated DNS and without a CA”](#)
- [Section 20.5, “Installing an IdM hidden replica”](#)

To troubleshoot the replica installation procedure, see:

- [Chapter 21, \*Troubleshooting IdM replica installation\*](#)

After the installation, see:

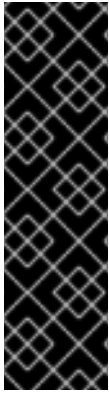
- [Section 20.6, “Testing an IdM replica”](#)

## 20.1. INSTALLING AN IDM REPLICA WITH INTEGRATED DNS AND A CA

This procedure describes installing an Identity Management (IdM) replica:

- With integrated DNS
- With a certificate authority (CA)

You can do this to, for example, replicate the CA service for resiliency after installing an IdM server with an integrated CA.



## IMPORTANT

When configuring a replica with a CA, the CA configuration of the replica must mirror the CA configuration of the other server.

For example, if the server includes an integrated IdM CA as the root CA, the new replica must also be installed with an integrated CA as the root CA. No other CA configuration is available in this case.

Including the **--setup-ca** option in the **ipa-replica-install** command copies the CA configuration of the initial server.

## Prerequisites

- Ensure your system is [prepared for an IdM replica installation](#).

## Procedure

1. Enter **ipa-replica-install** with these options:

- **--setup-dns** to configure the replica as a DNS server
- **--forwarder** to specify a per-server forwarder, or **--no-forwarder** if you do not want to use any per-server forwarders. To specify multiple per-server forwarders for failover reasons, use **--forwarder** multiple times.



## NOTE

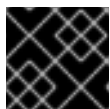
The **ipa-replica-install** utility accepts a number of other options related to DNS settings, such as **--no-reverse** or **--no-host-dns**. For more information about them, see the **ipa-replica-install(1)** man page.

- **--setup-ca** to include a CA on the replica

For example, to set up a replica with an integrated DNS server and a CA that forwards all DNS requests not managed by the IdM servers to the DNS server running on IP 192.0.2.1:

```
# ipa-replica-install --setup-dns --forwarder 192.0.2.1 --setup-ca
```

2. After the installation completes, add a DNS delegation from the parent domain to the IdM DNS domain. For example, if the IdM DNS domain is **idm.example.com**, add a name server (NS) record to the **example.com** parent domain.



## IMPORTANT

Repeat this step each time after you install an IdM DNS server.

## 20.2. INSTALLING AN IDM REPLICA WITH INTEGRATED DNS AND NO CA

This procedure describes installing an Identity Management (IdM) replica:

- With integrated DNS

- Without a certificate authority (CA) in an IdM environment in which a CA is already installed. The replica will forward all certificate operations to the IdM server with a CA installed.

## Prerequisites

- Ensure your system is [prepared for an IdM replica installation](#).

## Procedure

1. Enter **ipa-replica-install** with these options:

- **--setup-dns** to configure the replica as a DNS server
- **--forwarder** to specify a per-server forwarder, or **--no-forwarder** if you do not want to use any per-server forwarders. To specify multiple per-server forwarders for failover reasons, use **--forwarder** multiple times.

For example, to set up a replica with an integrated DNS server that forwards all DNS requests not managed by the IdM servers to the DNS server running on IP 192.0.2.1:

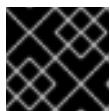
```
# ipa-replica-install --setup-dns --forwarder 192.0.2.1
```



### NOTE

The **ipa-replica-install** utility accepts a number of other options related to DNS settings, such as **--no-reverse** or **--no-host-dns**. For more information about them, see the **ipa-replica-install(1)** man page.

2. After the installation completes, add a DNS delegation from the parent domain to the IdM DNS domain. For example, if the IdM DNS domain is **idm.example.com**, add a name server (NS) record to the **example.com** parent domain.



### IMPORTANT

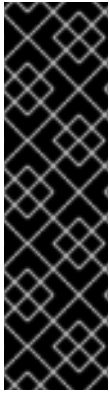
Repeat this step each time after you install an IdM DNS server.

## 20.3. INSTALLING AN IDM REPLICA WITHOUT INTEGRATED DNS AND WITH A CA

This procedure describes installing an Identity Management (IdM) replica:

- Without integrated DNS
- With a certificate authority (CA)





## IMPORTANT

When configuring a replica with a CA, the CA configuration of the replica must mirror the CA configuration of the other server.

For example, if the server includes an integrated IdM CA as the root CA, the new replica must also be installed with an integrated CA as the root CA. No other CA configuration is available in this case.

Including the **--setup-ca** option in the **ipa-replica-install** command copies the CA configuration of the initial server.

### Prerequisites

- Ensure your system is [prepared for an IdM replica installation](#).

### Procedure

1. Enter **ipa-replica-install** with the **--setup-ca** option.

```
# ipa-replica-install --setup-ca
```

2. Add the newly created IdM DNS service records to your DNS server:

- a. Export the IdM DNS service records into a file in the **nsupdate** format:

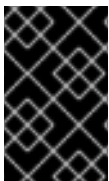
```
$ ipa dns-update-system-records --dry-run --out dns_records_file.nsupdate
```

- b. Submit a DNS update request to your DNS server using the **nsupdate** utility and the **dns\_records\_file.nsupdate** file. For more information, see [Updating External DNS Records Using nsupdate](#) in RHEL 7 documentation. Alternatively, refer to your DNS server documentation for adding DNS records.

## 20.4. INSTALLING AN IDM REPLICA WITHOUT INTEGRATED DNS AND WITHOUT A CA

This procedure describes installing an Identity Management (IdM) replica:

- Without integrated DNS
- Without a certificate authority (CA) by providing the required certificates manually. The assumption here is that the first server was installed without a CA.



## IMPORTANT

You cannot install a server or replica using self-signed third-party server certificates because the imported certificate files must contain the full CA certificate chain of the CA that issued the LDAP and Apache server certificates.

### Prerequisites

- Ensure your system is [prepared for an IdM replica installation](#).

### Procedure

- Enter **ipa-replica-install**, and provide the required certificate files by adding these options:
  - **--dirsrv-cert-file**
  - **--dirsrv-pin**
  - **--http-cert-file**
  - **--http-pin**

For details about the files that are provided using these options, see [Section 4.1, “Certificates required to install an IdM server without a CA”](#).

For example:

```
# ipa-replica-install \
  --dirsrv-cert-file /tmp/server.crt \
  --dirsrv-cert-file /tmp/server.key \
  --dirsrv-pin secret \
  --http-cert-file /tmp/server.crt \
  --http-cert-file /tmp/server.key \
  --http-pin secret
```



#### NOTE

Do not add the **--ca-cert-file** option. The **ipa-replica-install** utility takes this part of the certificate information automatically from the first server you installed.

## 20.5. INSTALLING AN IDM HIDDEN REPLICA

A hidden (unadvertised) replica is an Identity Management (IdM) server that has all services running and available. However, it has no SRV records in DNS, and LDAP server roles are not enabled. Therefore, clients cannot use service discovery to detect these hidden replicas.

For further details about hidden replicas, see [The hidden replica mode](#).

### Prerequisites

- Ensure your system is [prepared for an IdM replica installation](#).

### Procedure

- To install a hidden replica, use the following command:

```
ipa-replica-install --hidden-replica
```

Note that the command installs a replica without DNS SRV records and with disabled LDAP server roles.

You can also change the mode of existing replica to hidden. For details, see [Demotion and promotion of hidden replicas](#).

## 20.6. TESTING AN IDM REPLICA

After creating a replica, check if the replica replicates data as expected. You can use the following procedure.

### Procedure

1. Create a user on the new replica:

```
[admin@new_replica ~]$ ipa user-add test_user
```

2. Make sure the user is visible on another replica:

```
[admin@another_replica ~]$ ipa user-show test_user
```

## 20.7. CONNECTIONS PERFORMED DURING AN IDM REPLICA INSTALLATION

[Requests performed during an IdM replica installation](#) lists the operations performed by **ipa-replica-install**, the Identity Management (IdM) replica installation tool.

**Table 20.1. Requests performed during an IdM replica installation**

Operation	Protocol used	Purpose
DNS resolution against the DNS resolvers configured on the client system	DNS	To discover the IP addresses of IdM servers
Requests to ports 88 (TCP/TCP6 and UDP/UDP6) on the discovered IdM servers	Kerberos	To obtain a Kerberos ticket
JSON-RPC calls to the IdM Apache-based web-service on the discovered or configured IdM servers	HTTPS	IdM client enrollment; replica keys retrieval and certificate issuance if required
Requests over TCP/TCP6 to port 389 on the IdM server, using SASL GSSAPI authentication, plain LDAP, or both	LDAP	IdM client enrollment; CA certificate chain retrieval; LDAP data replication
Requests over TCP/TCP6 to port 22 on IdM server	SSH	To check if the connection is working
(optionally) Access over port 8443 (TCP/TCP6) on the IdM servers	HTTPS	To administer the Certificate Authority on the IdM server (only during IdM server and replica installation)

## CHAPTER 21. TROUBLESHOOTING IDM REPLICA INSTALLATION

The following sections describe the process for gathering information about a failing IdM replica installation, and how to resolve some common installation issues.

- [Reviewing IdM replica installation errors](#)
- [Reviewing IdM CA installation errors](#)
- [Removing a partial IdM replica installation](#)
- [Resolving invalid credentials](#)

### 21.1. REVIEWING IDM REPLICA INSTALLATION ERRORS

When you install an Identity Management (IdM) replica, debugging information is appended to the following log files on the replica:

- **/var/log/ipareplica-install.log**
- **/var/log/ipareplica-conncheck.log**
- **/var/log/ipaclient-install.log**
- **/var/log/httpd/error\_log**
- **/var/log/dirsrv/slapd-*INSTANCE-NAME*/access**
- **/var/log/dirsrv/slapd-*INSTANCE-NAME*/errors**
- **/var/log/ipaserver-install.log**

The replica installation process also appends debugging information to the following log files on the IdM **server** the replica is contacting:

- **/var/log/httpd/error\_log**
- **/var/log/dirsrv/slapd-*INSTANCE-NAME*/access**
- **/var/log/dirsrv/slapd-*INSTANCE-NAME*/errors**

The last line of each log file reports success or failure, and **ERROR** and **DEBUG** entries provide additional context.

To troubleshoot a failing IdM replica installation, review the errors at the end of these log files on both hosts (replica and server) and use this information to resolve any corresponding issues.

#### Prerequisites

- You must have **root** privileges to display the contents of IdM log files.

#### Procedure

1. Use the **tail** command to display the latest errors from the primary log file **/var/log/ipareplica-install.log**. The following example displays the last 10 lines.

```
[user@replica ~]$ sudo tail -n 10 /var/log/ipareplica-install.log
[sudo] password for user:
func(installer)
File "/usr/lib/python3.6/site-packages/ipaserver/install/server/replicainstall.py", line 424, in
decorated
func(installer)
File "/usr/lib/python3.6/site-packages/ipaserver/install/server/replicainstall.py", line 785, in
promote_check
ensure_enrolled(installer)
File "/usr/lib/python3.6/site-packages/ipaserver/install/server/replicainstall.py", line 740, in
ensure_enrolled
raise ScriptError("Configuration of client side components failed!")

2020-05-28T18:24:51Z DEBUG The ipa-replica-install command failed, exception:
ScriptError: Configuration of client side components failed!
2020-05-28T18:24:51Z ERROR Configuration of client side components failed!
2020-05-28T18:24:51Z ERROR The ipa-replica-install command failed. See
/var/log/ipareplica-install.log for more information
```

2. To review the log file interactively, open the end of the log file using the **less** utility and use the **↑** and **↓** arrow keys to navigate.

```
[user@replica ~]$ sudo less -N +G /var/log/ipareplica-install.log
```

3. (Optional) While **/var/log/ipareplica-install.log** is the primary log file for a replica installation, you can gather additional troubleshooting information by repeating this review process with additional files on the replica and the server.

#### On the replica:

```
[user@replica ~]$ sudo less -N +G /var/log/ipareplica-conncheck.log
[user@replica ~]$ sudo less -N +G /var/log/ipaclient-install.log
[user@replica ~]$ sudo less -N +G /var/log/httpd/error_log
[user@replica ~]$ sudo less -N +G /var/log/dirsrv/slaped-INSTANCE-NAME/access
[user@replica ~]$ sudo less -N +G /var/log/dirsrv/slaped-INSTANCE-NAME/errors
[user@replica ~]$ sudo less -N +G /var/log/ipaserver-install.log
```

#### On the server:

```
[user@server ~]$ sudo less -N +G /var/log/httpd/error_log
[user@server ~]$ sudo less -N +G /var/log/dirsrv/slaped-INSTANCE-NAME/access
[user@server ~]$ sudo less -N +G /var/log/dirsrv/slaped-INSTANCE-NAME/errors
```

#### Additional resources

- If you are unable to resolve a failing replica installation, and you have a Red Hat Technical Support subscription, open a Technical Support case at the [Red Hat Customer Portal](#) and provide an **sosreport** of the replica and an **sosreport** of the server.

- The **sosreport** utility collects configuration details, logs and system information from a RHEL system. For more information on the **sosreport** utility, see [What is an sosreport and how to create one in Red Hat Enterprise Linux?](#).

## 21.2. REVIEWING IDM CA INSTALLATION ERRORS

Installing the Certificate Authority (CA) service on an Identity Management (IdM) replica appends debugging information to several locations on the replica and the IdM server the replica communicates with.

Table 21.1. On the replica (in order of recommended priority):

Location	Description
<b>/var/log/pki/pki-ca-spawn.\$TIME_OF_INSTALLATION.log</b>	High-level issues and Python traces for the <b>pkispawn</b> installation process
<b>journalctl -u pki-tomcatd@pki-tomcat</b> output	Errors from the <b>pki-tomcatd@pki-tomcat</b> service
<b>/var/log/pki/pki-tomcat/ca/debug.\$DATE.log</b>	Large JAVA stacktraces of activity in the core of the Public Key Infrastructure (PKI) product
<b>/var/log/pki/pki-tomcat/ca/signedAudit/ca_audit</b>	Audit log of the PKI product
<ul style="list-style-type: none"><li>• <b>/var/log/pki/pki-tomcat/ca/system</b></li><li>• <b>/var/log/pki/pki-tomcat/ca/transactions</b></li><li>• <b>/var/log/pki/pki-tomcat/catalina.\$DATE.log</b></li></ul>	Low-level debug data of certificate operations for service principals, hosts, and other entities that use certificates

On the server contacted by the replica:

- **/var/log/httpd/error\_log** log file

Installing the CA service on an existing IdM replica also writes debugging information to the following log file:

- **/var/log/ipareplica-ca-install.log** log file



### NOTE

If a full IdM replica installation fails while installing the optional CA component, no details about the CA are logged; a message is logged in the **/var/log/ipareplica-install.log** file indicating that the overall installation process failed. Red Hat recommends reviewing the log files listed above for details specific to the CA installation failure.

The only exception to this behavior is when you are installing the CA service and the root CA is an external CA. If there is an issue with the certificate from the external CA, errors are logged in **/var/log/ipareplica-install.log**.

To troubleshoot a failing IdM CA installation, review the errors at the end of these log files and use this information to resolve any corresponding issues.

### Prerequisites

- You must have **root** privileges to display the contents of IdM log files.

### Procedure

1. To review a log file interactively, open the end of the log file using the **less** utility and use the ↑ and ↓ arrow keys to navigate, while searching for **ScriptError** entries. The following example opens `/var/log/pki/pki-ca-spawn.$TIME_OF_INSTALLATION.log`.

```
[user@server ~]$ sudo less -N +G /var/log/pki/pki-ca-spawn.20200527185902.log
```

2. Gather additional troubleshooting information by repeating this review process with all the log files listed above.

### Additional resources

- If you are unable to resolve a failing installation, and you have a Red Hat Technical Support subscription, open a Technical Support case at the [Red Hat Customer Portal](#) and provide an **sosreport** of the replica and an **sosreport** of the server.
- The **sosreport** utility collects configuration details, logs and system information from a RHEL system. For more information on the **sosreport** utility, see [What is an sosreport and how to create one in Red Hat Enterprise Linux?](#).

## 21.3. REMOVING A PARTIAL IDM REPLICA INSTALLATION

If an IdM replica installation fails, some configuration files may be left behind. Additional attempts to install the IdM replica can fail and the installation script reports that IPA is already configured.

```
[root@server ~]# ipa-replica-install
Your system may be partly configured.
Run /usr/sbin/ipa-server-install --uninstall to clean up.
```

```
IPA server is already configured on this system.
If you want to reinstall the IPA server, please uninstall it first using 'ipa-server-install --uninstall'.
The ipa-replica-install command failed. See /var/log/ipareplica-install.log for more information
```

To resolve this issue, uninstall IdM software from the replica, remove the replica from the IdM topology, and retry the installation process.

### Prerequisites

- You must have **root** privileges.

### Procedure

1. Uninstall the IdM server software on the host you are trying to configure as an IdM replica.

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

2. On all other servers in the topology, use the **ipa server-del** command to delete any references to the replica that did not install properly.

```
[root@other-replica ~]# ipa server-del replica.idm.example.com
```

3. Attempt installing the replica.
4. If you continue to experience difficulty installing an IdM replica because of repeated failed installations, reinstall the operating system.  
One of the requirements for installing an IdM replica is a clean system without any customization. Failed installations may have compromised the integrity of the host by unexpectedly modifying system files.

### Additional resources

- For additional details on uninstalling an IdM replica, see [Uninstalling an IdM replica](#).
- If installation attempts fail after repeated uninstallation attempts, and you have a Red Hat Technical Support subscription, open a Technical Support case at the [Red Hat Customer Portal](#) and provide an **sosreport** of the replica and an **sosreport** of the server.
- The **sosreport** utility collects configuration details, logs and system information from a RHEL system. For more information on the **sosreport** utility, see [What is an sosreport and how to create one in Red Hat Enterprise Linux?](#).

## 21.4. RESOLVING INVALID CREDENTIAL ERRORS

If an IdM replica installation fails with an **Invalid credentials** error, the system clocks on the hosts may be out of sync with each other:

```
[27/40]: setting up initial replication
Starting replication, please wait until this has completed.
Update in progress, 15 seconds elapsed
[ldap://server.example.com:389] reports: Update failed! Status: [49 - LDAP error: Invalid credentials]
```

```
[error] RuntimeError: Failed to start replication
Your system may be partly configured.
Run /usr/sbin/ipa-server-install --uninstall to clean up.
```

```
ipa.ipapython.install.cli.install_tool(CompatServerReplicaInstall): ERROR Failed to start replication
ipa.ipapython.install.cli.install_tool(CompatServerReplicaInstall): ERROR The ipa-replica-install
command failed. See /var/log/ipareplica-install.log for more information
```

If you use the **--no-ntp** or **-N** options to attempt the replica installation while clocks are out of sync, the installation fails because services are unable to authenticate with Kerberos.

To resolve this issue, synchronize the clocks on both hosts and retry the installation process.

### Prerequisites

- You must have **root** privileges to change system time.

### Procedure



1. Synchronize the system clocks manually or with **chronyd**.

- **Synchronizing manually:**

Display the system time on the server and set the replica's time to match.

```
[user@server ~]$ date
Thu May 28 21:03:57 EDT 2020

[user@replica ~]$ sudo timedatectl set-time '2020-05-28 21:04:00'
```

- **Synchronizing with chronyd:**

Refer to [Using the Chrony suite to configure NTP](#) to configure and set system time with **chrony** tools.

2. Attempt the IdM replica installation again.

### Additional resources

- If you are unable to resolve a failing replica installation, and you have a Red Hat Technical Support subscription, open a Technical Support case at the [Red Hat Customer Portal](#) and provide an **sosreport** of the replica and an **sosreport** of the server.
- The **sosreport** utility collects configuration details, logs and system information from a RHEL system. For more information on the **sosreport** utility, see [What is an sosreport and how to create one in Red Hat Enterprise Linux?](#).

## 21.5. ADDITIONAL RESOURCES

- To troubleshoot installing the first IdM server, see [Troubleshooting IdM server installation](#).
- To troubleshoot installing an IdM client, see [Troubleshooting IdM client installation](#).

## CHAPTER 22. UNINSTALLING AN IDM REPLICA

As an IdM administrator, you can remove an Identity Management (IdM) replica from the topology. For more information, see [Uninstalling an IdM server](#).

## CHAPTER 23. INSTALLING DNS ON AN EXISTING IDM SERVER

This section describes how to install the DNS service on an Identity Management (IdM) server that was originally installed without it.

### Prerequisites

- You understand the advantages and limitations of using IdM with integrated DNS as described in [Installing an IdM server: With integrated DNS, with an integrated CA as the root CA](#).
- You have **root** access to the IdM server.

### Procedure

1. [Optional] Verify that DNS is not already installed on the IdM server.

```
[root@r8server ~]# ipa server-role-show r8server.idm.example.com
Role name: DNS server
Server name: r8server.idm.example.com
Role name: DNS server
Role status: absent
```

The output confirms that IdM DNS is not available on the server.

2. Enable the **idm:DL1** stream:

```
[root@r8server ~]# yum module enable idm:DL1
```

3. Download the **ipa-dns-server** package and its dependencies:

```
[root@r8server ~]# yum module install idm:DL1/dns
```

4. Start the script to install DNS on the server:

```
[root@r8server ~]# ipa-dns-install
```

- a. The script prompts for per-server DNS forwarders.

```
Do you want to configure DNS forwarders? [yes]:
```

- To configure per-server DNS forwarders, enter **yes**, and then follow the instructions on the command line. The installation process will add the forwarder IP addresses to the IdM LDAP.
  - For the forwarding policy default settings, see the **--forward-policy** description in the **ipa-dns-install(1)** man page.
- If you do not want to use DNS forwarding, enter **no**.  
With no DNS forwarders, hosts in your IdM domain will not be able to resolve names from other, internal, DNS domains in your infrastructure. The hosts will only be left with public DNS servers to resolve their DNS queries.

- b. The script prompts to check if any DNS reverse (PTR) records for the IP addresses associated with the server need to be configured.

Do you want to search for missing reverse zones? [yes]:

If you run the search and missing reverse zones are discovered, the script asks you whether to create the reverse zones along with the PTR records.

Do you want to create reverse zone for IP 192.0.2.1 [yes]:

Please specify the reverse zone name [2.0.192.in-addr.arpa.]:

Using reverse zone(s) 2.0.192.in-addr.arpa.



#### NOTE

Using IdM to manage reverse zones is optional. You can use an external DNS service for this purpose instead.

#### Additional resources

- **man ipa-dns-install(1)**

## CHAPTER 24. MANAGING REPLICATION TOPOLOGY

This chapter describes how to manage replication between servers in an Identity Management (IdM) domain.

### 24.1. EXPLAINING REPLICATION AGREEMENTS, TOPOLOGY SUFFIXES AND TOPOLOGY SEGMENTS

This section explains the following concepts:

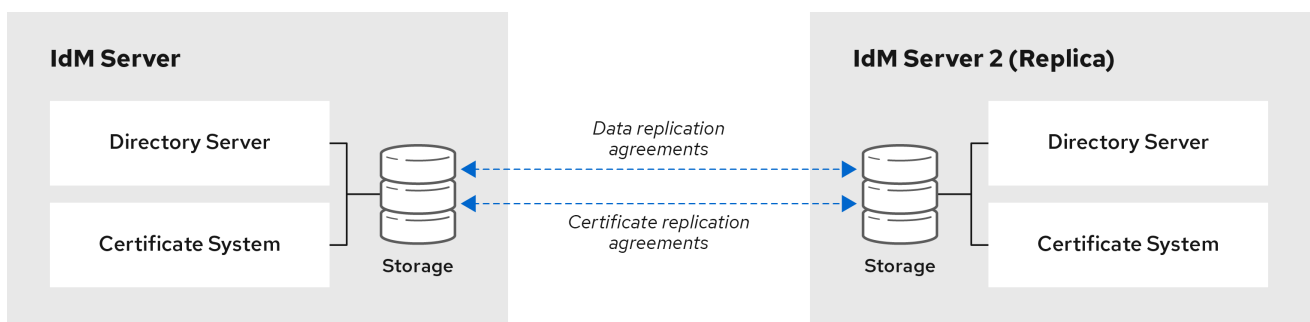
- [Replication agreements](#)
- [Topology suffixes](#)
- [Topology segments](#)

#### Replication agreements

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 24.1. Server and replica agreements



64\_RHEL\_0120

IdM uses two types of replication agreements:

#### Domain replication agreements

These agreements replicate the identity information.

#### Certificate replication agreements

These 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.

#### 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.

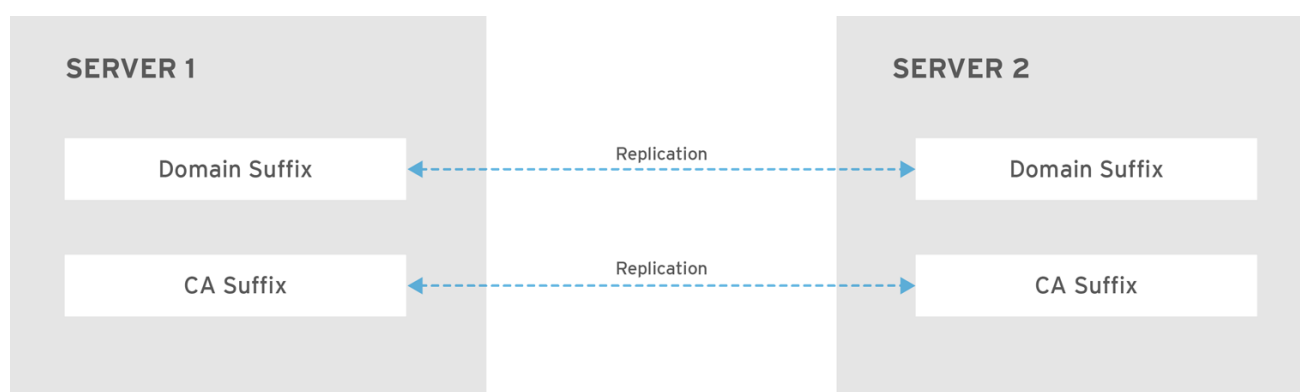
When two replicas have a replication agreement between their **domain** suffixes, they share directory data, such as users, groups, and policies.

### 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 24.2. Topology suffixes



RHEL\_404973\_0916

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

### Example 24.1. Viewing topology suffixes

The **ipa topologysuffix-find** command displays a list of topology suffixes:

```

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

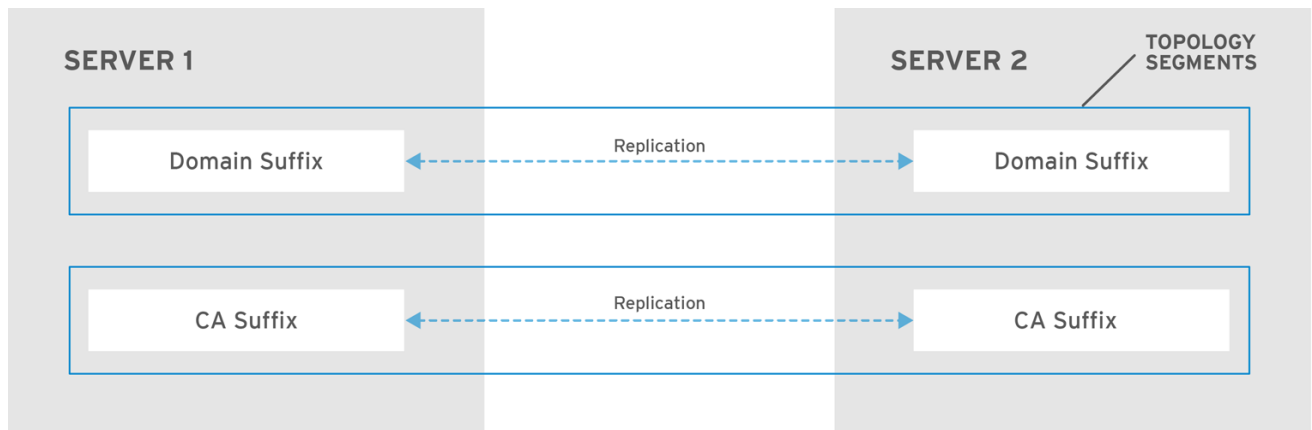
Suffix name: domain
Managed LDAP suffix DN: dc=example,dc=com
-----
Number of entries returned 2
-----
  
```

## 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 24.3. Topology segments**



RHEL\_404973\_0916

### Example 24.2. Viewing topology segments

The **ipa topologysegment-find** command shows 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**.

To display details for a particular segment only, use 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
```

## 24.2. USING THE TOPOLOGY GRAPH TO MANAGE REPLICATION TOPOLOGY

The topology graph in the web UI shows the relationships between the servers in the domain. Using the Web UI, you can manipulate and transform the representation of the topology.

### Accessing the topology graph

To access the topology graph:

1. Select **IPA Server** → **Topology** → **Topology Graph**.
2. If you make any changes to the topology that are not immediately reflected in the graph, click **Refresh**.

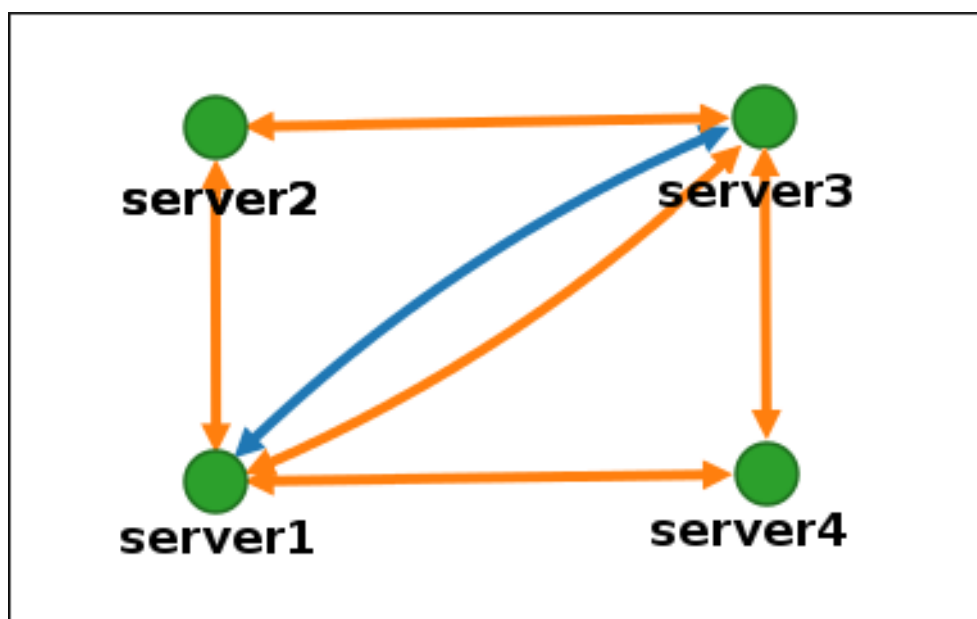
### Interpreting the topology graph

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

Figure 24.4, “Recommended topology example” 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 master.

Figure 24.4. Recommended topology example



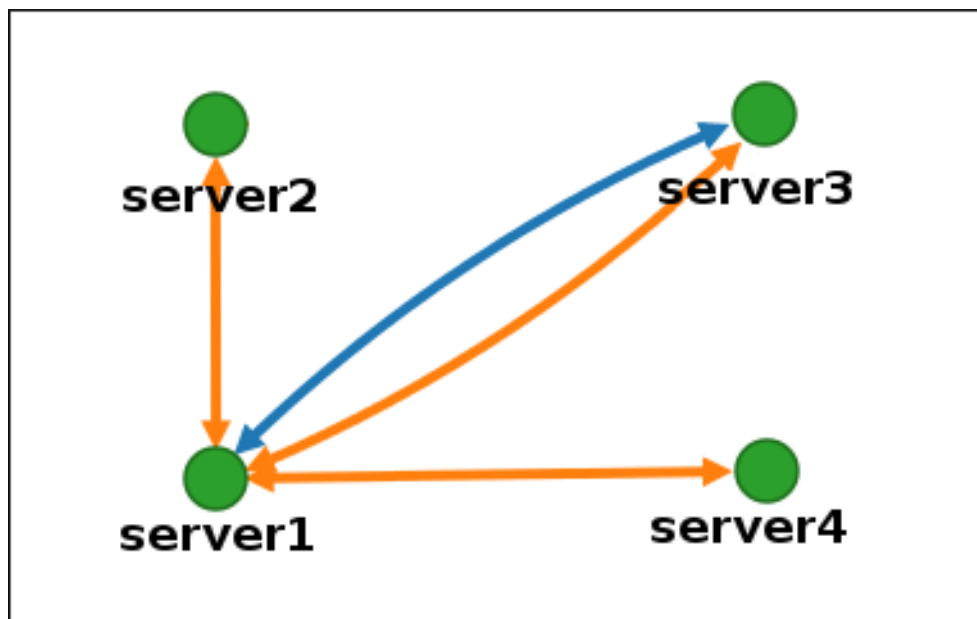
### Topology graph example: discouraged topology

In Figure 24.5, “Discouraged topology example: Single Point of Failure”, **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.



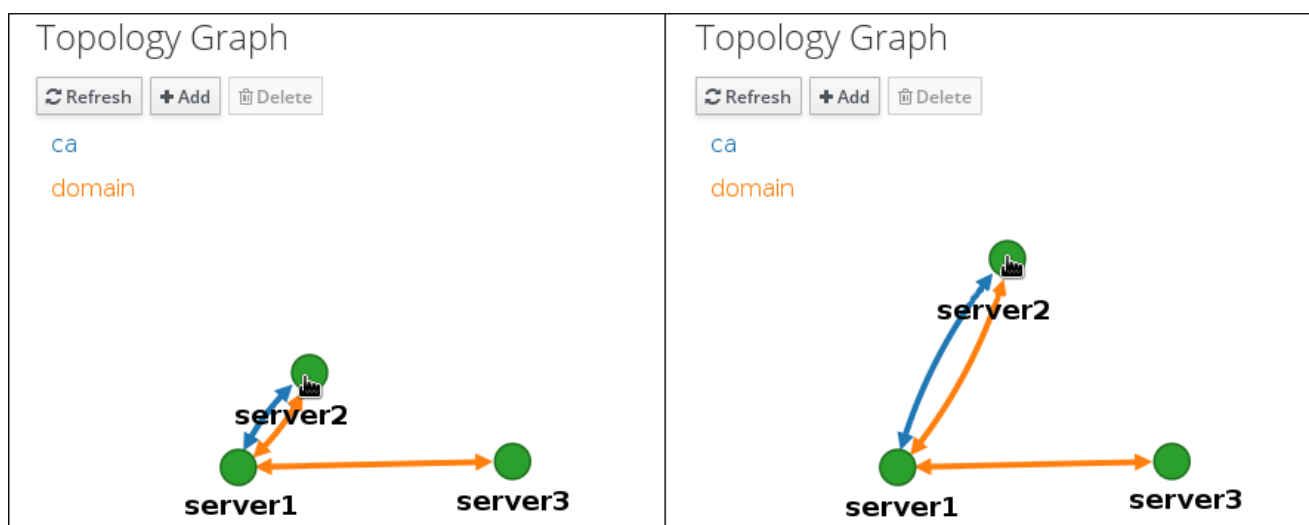
Figure 24.5. Discouraged topology example: Single Point of Failure



### Customizing the topology view

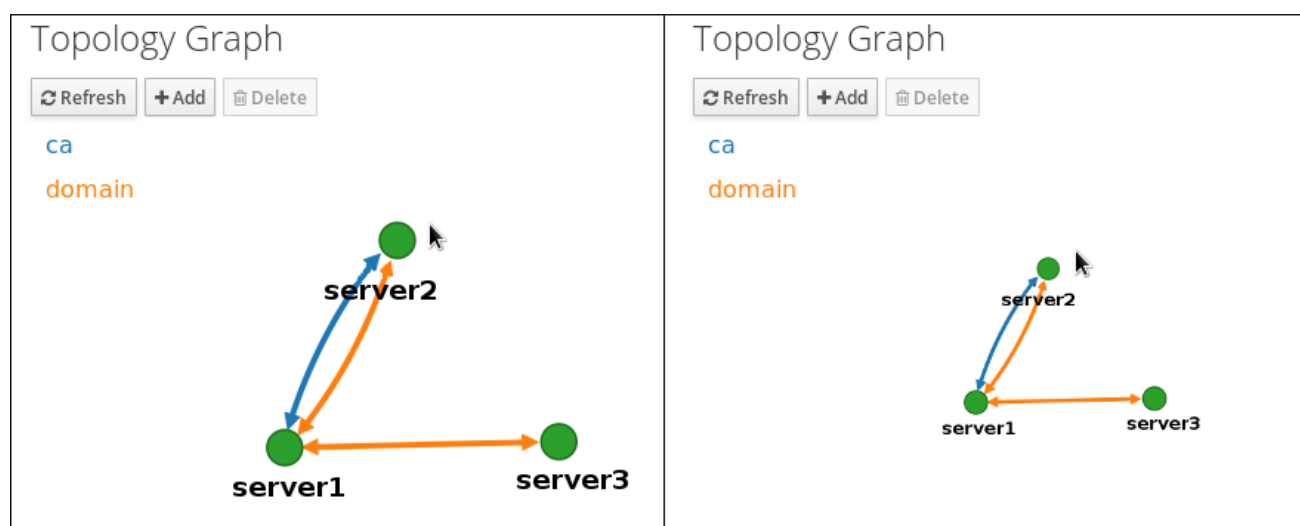
You can move individual topology nodes by dragging the mouse:

Figure 24.6. Moving topology graph nodes



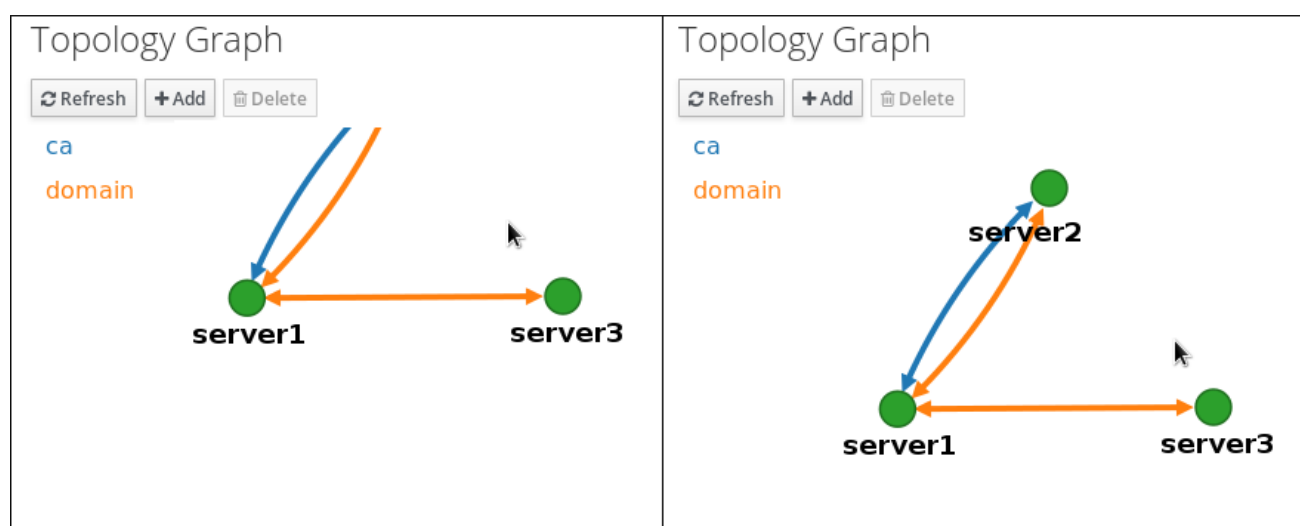
You can zoom in and zoom out the topology graph using the mouse wheel:

Figure 24.7. Zooming the topology graph



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

Figure 24.8. Moving the topology graph canvas



## 24.3. SETTING UP REPLICATION BETWEEN TWO SERVERS USING THE WEB UI

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

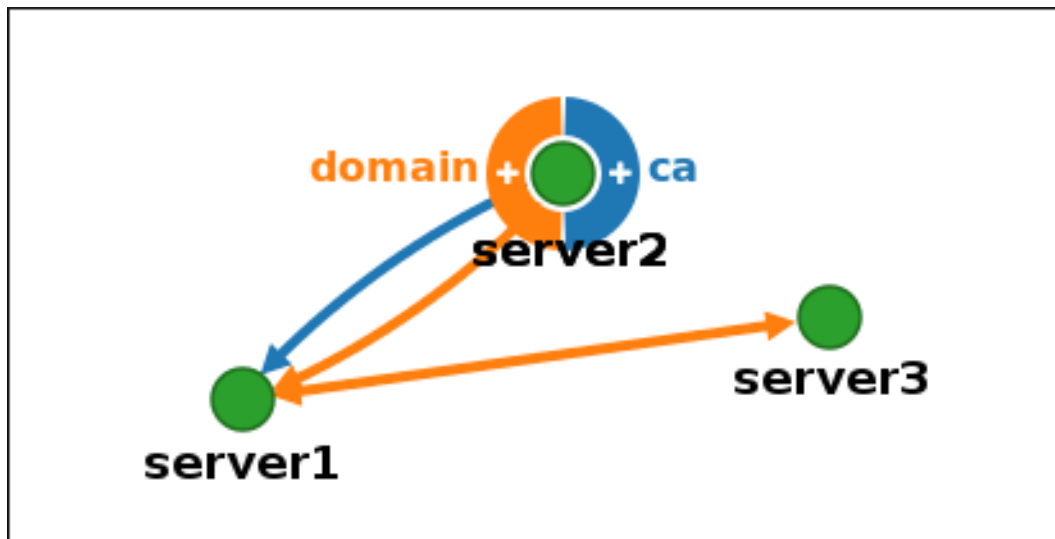
### Prerequisites

- You have the IdM administrator credentials.

### Procedure

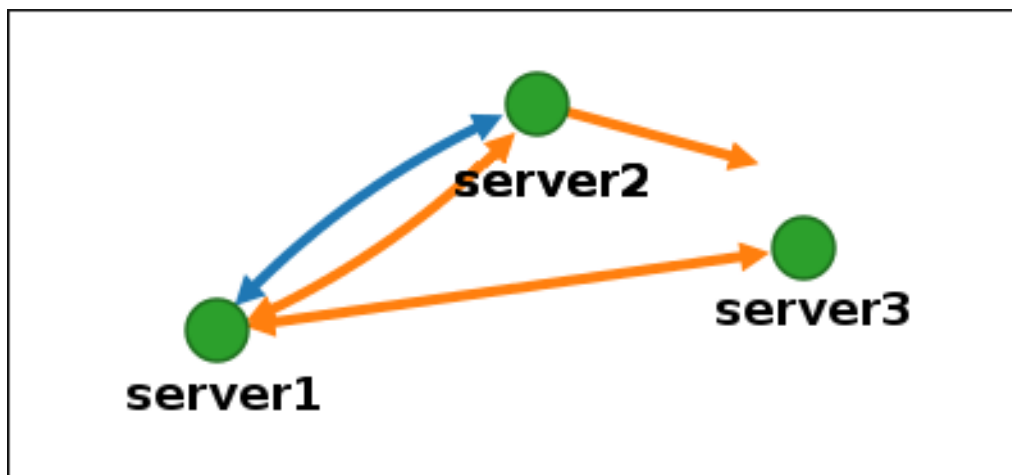
1. In the topology graph, hover your mouse over one of the server nodes.

Figure 24.9. 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.

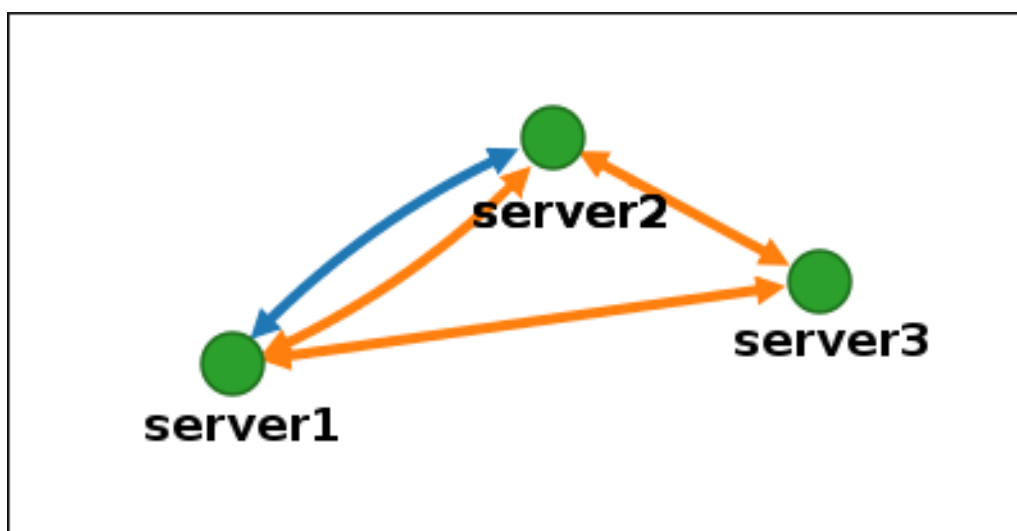
Figure 24.10. Creating a new segment



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 24.11. New segment created



## 24.4. STOPPING REPLICATION BETWEEN TWO SERVERS USING THE WEB UI

Using the web interface of Identity Management (IdM) you can remove a replication agreement from servers.

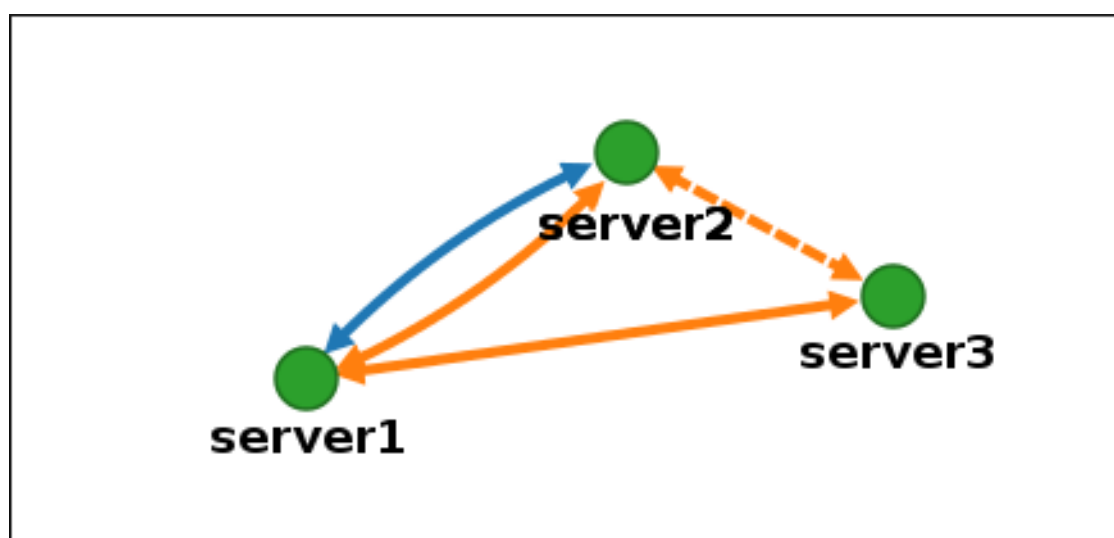
### Prerequisites

- You have the IdM administrator credentials.

### Procedure

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

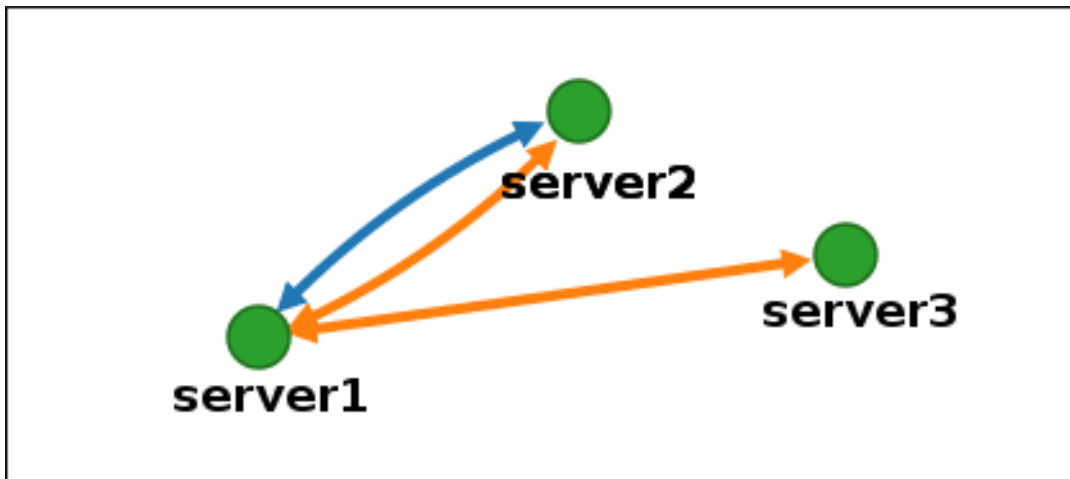
Figure 24.12. Topology segment highlighted



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 24.13. Topology segment deleted



## 24.5. 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

1. Use the **ipa topologysegment-add** command to 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
  - optionally, 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.

2. *Optional.* Use the **ipa topologysegment-show** command to 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
```

## 24.6. 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. To stop replication, you must delete the corresponding replication segment between the servers. To do that, you need to know the segment name.  
If you do not know the name, use the **ipa topologysegment-find** command to display all segments, and locate the required segment in the output. 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
-----
```

2. Use the **ipa topologysegment-del** command to 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.

3. *Optional.* Use the **ipa topologysegment-find** command to 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
-----
```

## 24.7. REMOVING SERVER FROM TOPOLOGY USING THE WEB UI

You can use Identity Management (IdM) web interface to remove a 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.



### 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

To remove a server from the topology without uninstalling the server components from the machine:

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

Figure 24.14. Selecting a server

IPA Servers				
<input type="text" value="Search"/>			<input type="button" value="Refresh"/>	
<input type="checkbox"/>	Server name	Min domain level	Max domain level	Managed suffixes
<input type="checkbox"/>	server1.example.com	0	1	domain, ca
<input type="checkbox"/>	server2.example.com	0	1	domain
<input type="checkbox"/>	server3.example.com	0	1	domain, ca
Showing 1 to 3 of 3 entries.				

- Click **Delete Server**.

## 24.8. REMOVING SERVER FROM TOPOLOGY USING THE CLI

You can use the command line interface to remove a 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**:

- 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"
-----
```

- 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
```

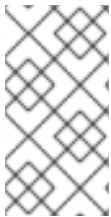


## 24.9. VIEWING SERVER ROLES ON AN IDM SERVER USING THE WEB UI

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.

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 24.15. Server roles in the web UI

Server Roles	
 Refresh	
Role name	Role status
AD trust agent	absent
AD trust controller	absent
CA server	enabled

## 24.10. VIEWING SERVER ROLES ON AN IDM SERVER USING THE 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.

You can view which servers perform which roles in the topology using the following commands.

- The **ipa config-show** command displays all CA servers 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
```

- The **ipa server-show** command displays a list of roles enabled on a particular server. For example, for a list of roles enabled on *server.example.com*:

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

- The **ipa server-find --servrole** searches 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
-----
```

## 24.11. 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.

### Prerequisites

- You have the IdM administrator credentials.

### Procedure

- [Change the current CA renewal master.](#)
- [Configure replica to generate CRLs.](#)

## 24.12. 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](#).

If the replica is a CA renewal server, move the service to another replica before making this replica hidden.

For details, see [Changing and resetting IdM CA renewal server](#).

**NOTE**

The hidden replica feature, introduced in RHEL 8.1 as a Technology Preview, is fully supported starting with RHEL 8.2.

**Procedure**

- To hide the replica, enter:

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

Alternatively, you can make the replica visible with the following command:

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

## CHAPTER 25. INSTALLING AND RUNNING THE IDM HEALTHCHECK TOOL

This chapter describes the IdM Healthcheck tool and how to install and run it.

### Prerequisites

- The Healthcheck tool is only available on RHEL 8.1 or later.

### 25.1. HEALTHCHECK IN IDM

The Healthcheck tool in Identity Management (IdM) helps find issues that may impact the health of your IdM environment.



#### NOTE

The Healthcheck tool is a command line tool that can be used without Kerberos authentication.

### Modules are independent

Healthcheck consists of independent modules which test for:

- Replication issues
- Certificate validity
- Certificate Authority infrastructure issues
- IdM and Active Directory trust issues
- Correct file permissions and ownership settings

### Two output formats

Healthcheck generates the following outputs, which you can set using the **output-type** option:

- **json**: Machine-readable output in JSON format (default)
- **human**: Human-readable output

You can specify a different file destination with the **--output-file** option.

### Results

Each Healthcheck module returns one of the following results:

#### SUCCESS

configured as expected

#### WARNING

not an error, but worth keeping an eye on or evaluating

#### ERROR

not configured as expected

**CRITICAL**

not configured as expected, with a high possibility for impact

## 25.2. INSTALLING IDM HEALTHCHECK

This section describes how to install the IdM Healthcheck tool.

**Procedure**

- Install the **ipa-healthcheck** package:

```
[root@server ~]# yum install ipa-healthcheck
```

**NOTE**

On RHEL 8.1 and 8.2 systems, use the **yum install /usr/bin/ipa-healthcheck** command instead.

**Verification steps**

- Use the **--failures-only** option to have **ipa-healthcheck** only report errors. A fully-functioning IdM installation returns an empty result of [].

```
[root@server ~]# ipa-healthcheck --failures-only  
[]
```

**Additional resources**

- Use **ipa-healthcheck --help** to see all supported arguments.

## 25.3. RUNNING IDM HEALTHCHECK

Healthcheck can be run manually or automatically using [log rotation](#).

**Prerequisites**

- The Healthcheck tool must be installed. See [Installing IdM Healthcheck](#).

**Procedure**

- To run healthcheck manually, enter the **ipa-healthcheck** command.

```
[root@server ~]# ipa-healthcheck
```

**Additional resources**

For all options, see the man page: **man ipa-healthcheck**.

## 25.4. ADDITIONAL RESOURCES

- See the following sections of the [Configuring and managing Identity Management](#) guide for examples of using IdM Healthcheck.
  - [Checking services](#)
  - [Verifying your IdM and AD trust configuration](#)
  - [Verifying certificates](#)
  - [Verifying system certificates](#)
  - [Checking disk space](#)
  - [Verifying permissions of IdM configuration files](#)
  - [Checking replication](#)
- You can also see those chapters organized into a single guide: [Using IdM Healthcheck to monitor your IdM environment](#)

## CHAPTER 26. INSTALLING AN IDENTITY MANAGEMENT SERVER USING AN ANSIBLE PLAYBOOK

### 26.1. ANSIBLE AND ITS ADVANTAGES FOR INSTALLING IDM

Ansible is an automation tool used to configure systems, deploy software, and perform rolling updates. Ansible includes support for Identity Management (IdM), and you can use Ansible modules to automate installation tasks such as the setup of an IdM server, replica, client, or an entire IdM topology.

#### Advantages of using Ansible to install IdM

The following list presents advantages of installing Identity Management using Ansible in contrast to manual installation.

- You do not need to log into the managed node.
- You do not need to configure settings on each host to be deployed individually. Instead, you can have one inventory file to deploy a complete cluster.
- You can reuse an inventory file later for management tasks, for example to add users and hosts. You can reuse an inventory file even for such tasks as are not related to IdM.

### 26.2. IDM SERVER INSTALLATION USING AN ANSIBLE PLAYBOOK

The following sections describe how to configure a system as an IdM server by using [Ansible](#). Configuring a system as an IdM server establishes an IdM domain and enables the system to offer IdM services to IdM clients. The deployment is managed by the **ipaserver** Ansible role.



#### NOTE

Before installing an IdM server using Ansible, ensure that you understand [Ansible](#) and IdM concepts. Ensure that you understand the following terms that are used in this chapter:

- Ansible roles
- Ansible nodes
- Ansible inventory
- Ansible tasks
- Ansible modules
- Ansible plays and playbooks

#### Overview

The installation consists of the following parts:

1. [Installing the ansible-freeipa package](#)
2. [Deploying an IdM server with an integrated CA using an Ansible playbook](#)
3. [Deploying an IdM server with an external CA using an Ansible-playbook](#)

## 26.3. INSTALLING THE ANSIBLE-FREEIPA PACKAGE

### Prerequisites

On the **managed node**:

- Ensure that the managed node is a Red Hat Enterprise Linux 8 system with a static IP address and a working package manager.

On the **controller**:

- Ensure that the controller is a Red Hat Enterprise Linux system with a valid subscription. If this is not the case, see the official Ansible documentation [Installation guide](#) for alternative installation instructions.
- Ensure that you can reach the managed node over the **SSH** protocol from the controller. Check that the managed node is listed in the **/root/.ssh/known\_hosts** file of the controller.

### Procedure

Run the following procedure on the Ansible controller.

1. Enable the required repository:

```
# subscription-manager repos --enable ansible-2.8-for-rhel-8-x86_64-rpms
```

2. Install Ansible:

```
# yum install ansible
```

3. Install the IdM Ansible roles:

```
# yum install ansible-freeipa
```

The roles are installed to the **/usr/share/ansible/roles/** directory.

## 26.4. ANSIBLE ROLES LOCATION IN THE FILE SYSTEM

By default the **ansible-freeipa** roles are installed to the **/usr/share/ansible/roles/** directory. The structure of the **ansible-freeipa** package is as follows:

- The **/usr/share/ansible/roles/** directory stores the **ipaserver**, **ipareplica**, and **ipaclient** roles on the Ansible controller. Each role directory stores examples, a basic overview, the licence and documentation about the role in a README.md Markdown file.

```
[root@server]# ls -l /usr/share/ansible/roles/
ipaclient
ipareplica
ipaserver
```

- The **/usr/share/doc/ansible-freeipa/** directory stores the documentation about individual roles and the topology in README.md Markdown files. It also stores the **playbooks/** subdirectory (see below).

```
[root@server]# ls -l /usr/share/doc/ansible-freeipa/
```



```
playbooks
README-client.md
README.md
README-replica.md
README-server.md
README-topology.md
```

- The `/usr/share/doc/ansible-freeipa/playbooks/` directory stores the example playbooks:

```
[root@server]# ls -l /usr/share/doc/ansible-freeipa/playbooks/
install-client.yml
install-cluster.yml
install-replica.yml
install-server.yml
uninstall-client.yml
uninstall-cluster.yml
uninstall-replica.yml
uninstall-server.yml
```

## 26.5. DEPLOYING AN IDM SERVER WITH AN INTEGRATED CA AS THE ROOT CA USING AN ANSIBLE PLAYBOOK

Before [running the `ansible-playbook` utility](#), set the parameters that correspond to your scenario by choosing one of the following procedures:

- [Procedure with integrated DNS](#)
- [Procedure with external DNS](#)

### NOTE

The inventory files in the following procedures use the **INI** format. You can, alternatively, use the **YAML** or **JSON** formats.

### 26.5.1. Setting the parameters for a deployment with an integrated DNS and an integrated CA as the root CA

Complete this procedure to configure the inventory file for installing an IdM server with an integrated CA as the root CA in an environment that uses the IdM integrated DNS solution.

#### Procedure

1. Open the inventory file for editing. Specify the fully-qualified domain names (**FQDN**) of the host you want to use as an IdM server. Ensure that the **FQDN** meets the following criteria:
  - Only alphanumeric characters and hyphens (-) are allowed. For example, underscores are not allowed and can cause DNS failures.
  - The host name must be all lower-case.
2. Specify the IdM domain and realm information.
3. Specify that you want to use integrated DNS by adding the following option:

```
ipaserver_setup_dns=yes
```

## 4. Specify the DNS forwarding settings. Choose one of the following options:

- Use the **ipaserver\_auto\_forwarders=yes** option if you want the installer to use forwarders from the **/etc/resolv.conf** file. This option is not recommended if the nameserver specified in the **/etc/resolv.conf** file is the localhost 127.0.0.1 address or if you are on a virtual private network and the DNS servers you are using are normally unreachable from the public internet.
- Use the **ipaserver\_forwarders** option to specify your forwarders manually. The installation process adds the forwarder IP addresses to the **/etc/named.conf** file on the installed IdM server.
- Use the **ipaserver\_no\_forwarders=yes** option to configure root DNS servers to be used instead.

**NOTE**

With no DNS forwarders, your environment will be isolated, and names from other DNS domains in your infrastructure will not be resolved.

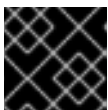
## 5. Specify the DNS reverse record and zone settings. Choose from the following options:

- Use the **ipaserver\_allow\_zone\_overlap=yes** option to allow the creation of a (reverse) zone even if the zone is already resolvable.
- Use the **ipaserver\_reverse\_zones** option to specify your reverse zones manually.
- Use the **ipaserver\_no\_reverse=yes** option if you do not want the installer to create reverse a DNS zone.

**NOTE**

Using IdM to manage reverse zones is optional. You can use an external DNS service for this purpose instead.

- Specify the passwords for **admin** and for the **Directory Manager**. Use the Ansible Vault to store the password, and reference the Vault file from the playbook file. Alternatively and less securely, specify the passwords directly in the inventory file.
- (Optional) Specify a custom **firewalld** zone to be used by the IdM server. If you do not set a custom zone, IdM will add its services to the default **firewalld** zone. The predefined default zone is **public**.

**IMPORTANT**

The specified **firewalld** zone must exist and be permanent.

**Example of an inventory file with the required server information (excluding the passwords)**

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

[ipaserver:vars]
ipaserver_domain=idm.example.com
ipaserver_realm=IDM.EXAMPLE.COM
```

```
ipaserver_setup_dns=yes
ipaserver_auto_forwarders=yes
[...]
```

Example of an inventory file with the required server information (including the passwords)

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

[ipaserver:vars]
ipaserver_domain=idm.example.com
ipaserver_realm=IDM.EXAMPLE.COM
ipaserver_setup_dns=yes
ipaserver_auto_forwarders=yes
ipaadmin_password=MySecretPassword123
ipadm_password=MySecretPassword234

[...]
```

Example of an inventory file with a custom `firewalld` zone

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

[ipaserver:vars]
ipaserver_domain=idm.example.com
ipaserver_realm=IDM.EXAMPLE.COM
ipaserver_setup_dns=yes
ipaserver_auto_forwarders=yes
ipaadmin_password=MySecretPassword123
ipadm_password=MySecretPassword234
ipaserver_firewalld_zone=custom zone
```

Example playbook to set up an IdM server using admin and Directory Manager passwords stored in an Ansible Vault file

```
---
- name: Playbook to configure IPA server
  hosts: ipaserver
  become: true
  vars_files:
    - playbook_sensitive_data.yml

  roles:
    - role: ipaserver
      state: present
```

Example playbook to set up an IdM server using admin and Directory Manager passwords from an inventory file

```
---
- name: Playbook to configure IPA server
```

```
hosts: ipaserver
become: true

roles:
- role: ipaserver
  state: present
```

### Additional resources

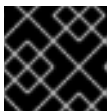
- For the forwarding policy default settings, see the **--forward-policy** description in the **ipa-dns-install(1)** man page.
- For more information about DNS variables used by the **ipaserver** role, see the DNS Variables section in the **README-server.md** file in the **/usr/share/doc/ansible-freeipa** directory.

## 26.5.2. Setting the parameters for a deployment with external DNS and an integrated CA as the root CA

Complete this procedure to configure the inventory file for installing an IdM server with an integrated CA as the root CA in an environment that uses an external DNS solution.

### Procedure

1. Open the inventory file for editing. Specify the fully-qualified domain names (**FQDN**) of the host you want to use as an IdM server. Ensure that the **FQDN** meets the following criteria:
  - Only alphanumeric characters and hyphens (-) are allowed. For example, underscores are not allowed and can cause DNS failures.
  - The host name must be all lower-case.
2. Specify the IdM domain and realm information.
3. Make sure that the **ipaserver\_setup\_dns** option is set to **no** or that it is absent.
4. Specify the passwords for **admin** and for the **Directory Manager**. Use the Ansible Vault to store the password, and reference the Vault file from the playbook file. Alternatively and less securely, specify the passwords directly in the inventory file.
5. (Optional) Specify a custom **firewalld** zone to be used by the IdM server. If you do not set a custom zone, IdM will add its services to the default **firewalld** zone. The predefined default zone is **public**.



### IMPORTANT

The specified **firewalld** zone must exist and be permanent.

### Example of an inventory file with the required server information (excluding the passwords)

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

[ipaserver:vars]
```

```
ipaserver_domain=idm.example.com
ipaserver_realm=IDM.EXAMPLE.COM
ipaserver_setup_dns=no
[...]
```

Example of an inventory file with the required server information (including the passwords)

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

[ipaserver:vars]
ipaserver_domain=idm.example.com
ipaserver_realm=IDM.EXAMPLE.COM
ipaserver_setup_dns=no
ipaadmin_password=MySecretPassword123
ipadm_password=MySecretPassword234

[...]
```

Example of an inventory file with a custom `firewalld` zone

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

[ipaserver:vars]
ipaserver_domain=idm.example.com
ipaserver_realm=IDM.EXAMPLE.COM
ipaserver_setup_dns=no
ipaadmin_password=MySecretPassword123
ipadm_password=MySecretPassword234
ipaserver_firewalld_zone=custom zone
```

Example playbook to set up an IdM server using admin and Directory Manager passwords stored in an Ansible Vault file

```
---
- name: Playbook to configure IPA server
  hosts: ipaserver
  become: true
  vars_files:
    - playbook_sensitive_data.yml

  roles:
    - role: ipaserver
      state: present
```

Example playbook to set up an IdM server using admin and Directory Manager passwords from an inventory file

```
---
- name: Playbook to configure IPA server
  hosts: ipaserver
```

```
become: true

roles:
- role: ipaserver
  state: present
```

### 26.5.3. Deploying an IdM server with an integrated CA as the root CA using an Ansible playbook

Complete this procedure to deploy an IdM server with an integrated certificate authority (CA) as the root CA using an Ansible playbook.

#### Procedure

1. Run the **ansible-playbook** command with the name of the playbook file, for example **install-server.yml**. Specify the inventory file with the **-i** option:

```
$ ansible-playbook -v -i <path_to_inventory_directory>/hosts
<path_to_playbooks_directory>/install-server.yml
```

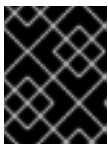
Specify the level of verbosity by using the **-v**, **-vv**, or **-vvv** option.

You can view the output of the Ansible playbook script on the command-line interface (CLI). The following output shows that the script has run successfully as 0 tasks have failed:

```
PLAY RECAP
server.idm.example.com : ok=18  changed=10  unreachable=0  failed=0  skipped=21
rescued=0  ignored=0
```

2. Choose one of the following options:
  - If your IdM deployment uses external DNS: add the DNS resource records contained in the **/tmp/ipa.system.records.UFRPto.db** file to the existing external DNS servers. The process of updating the DNS records varies depending on the particular DNS solution.

```
...
Restarting the KDC
Please add records in this file to your DNS system:
/tmp/ipa.system.records.UFRBto.db
Restarting the web server
...
```



#### IMPORTANT

The server installation is not complete until you add the DNS records to the existing DNS servers.

- If your IdM deployment uses integrated DNS:
  - Add DNS delegation from the parent domain to the IdM DNS domain. For example, if the IdM DNS domain is **idm.example.com**, add a name server (NS) record to the **example.com** parent domain.



## IMPORTANT

Repeat this step each time after an IdM DNS server is installed.

- Add an **\_ntp.\_udp** service (SRV) record for your time server to your IdM DNS. The presence of the SRV record for the time server of the newly-installed IdM server in IdM DNS ensures that future replica and client installations are automatically configured to synchronize with the time server used by this primary IdM server.

## 26.6. DEPLOYING AN IDM SERVER WITH AN EXTERNAL CA AS THE ROOT CA USING AN ANSIBLE PLAYBOOK

Before [running the \*\*ansible-playbook\*\* utility](#), set the parameters that correspond to your scenario by choosing one of the following procedures:

- [Procedure with integrated DNS](#)
- [Procedure with external DNS](#)

### NOTE

The inventory files in the following procedures use the **INI** format. You can, alternatively, use the **YAML** or **JSON** formats.

### 26.6.1. Setting the parameters for a deployment with an integrated DNS and an external CA as the root CA

Complete this procedure to configure the inventory file for installing an IdM server with an external CA as the root CA in an environment that uses the IdM integrated DNS solution.

#### Procedure

1. Open the inventory file for editing. Specify the fully-qualified domain names (**FQDN**) of the host you want to use as an IdM server. Ensure that the **FQDN** meets the following criteria:
  - Only alphanumeric characters and hyphens (-) are allowed. For example, underscores are not allowed and can cause DNS failures.
  - The host name must be all lower-case.
2. Specify the IdM domain and realm information.
3. Specify that you want to use integrated DNS by adding the following option:

```
ipaserver_setup_dns=yes
```

4. Specify the DNS forwarding settings. Choose one of the following options:
  - Use the **ipaserver\_auto\_forwarders=yes** option if you want the installation process to use forwarders from the **/etc/resolv.conf** file. This option is not recommended if the nameserver specified in the **/etc/resolv.conf** file is the localhost 127.0.0.1 address or if you are on a virtual private network and the DNS servers you are using are normally unreachable from the public internet.

- Use the **ipaserver\_forwarders** option to specify your forwarders manually. The installation process adds the forwarder IP addresses to the **/etc/named.conf** file on the installed IdM server.
- Use the **ipaserver\_no\_forwarders=yes** option to configure root DNS servers to be used instead.

**NOTE**

With no DNS forwarders, your environment will be isolated, and names from other DNS domains in your infrastructure will not be resolved.

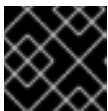
5. Specify the DNS reverse record and zone settings. Choose from the following options:

- Use the **ipaserver\_allow\_zone\_overlap=yes** option to allow the creation of a (reverse) zone even if the zone is already resolvable.
- Use the **ipaserver\_reverse\_zones** option to specify your reverse zones manually.
- Use the **ipaserver\_no\_reverse=yes** option if you do not want the installation process to create reverse a DNS zone.

**NOTE**

Using IdM to manage reverse zones is optional. You can use an external DNS service for this purpose instead.

6. Specify the passwords for **admin** and for the **Directory Manager**. Use the Ansible Vault to store the password, and reference the Vault file from the playbook file. Alternatively and less securely, specify the passwords directly in the inventory file.
7. (Optional) Specify a custom **firewalld** zone to be used by the IdM server. If you do not set a custom zone, IdM will add its services to the default **firewalld** zone. The predefined default zone is **public**.

**IMPORTANT**

The specified **firewalld** zone must exist and be permanent.

### Example of an inventory file with the required server information (excluding the passwords)

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

[ipaserver:vars]
ipaserver_domain=idm.example.com
ipaserver_realm=IDM.EXAMPLE.COM
ipaserver_setup_dns=yes
ipaserver_auto_forwarders=yes
[...]
```

### Example of an inventory file with the required server information (including the passwords)

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



```
[ipaserver:vars]
ipaserver_domain=idm.example.com
ipaserver_realm=IDM.EXAMPLE.COM
ipaserver_setup_dns=yes
ipaserver_auto_forwarders=yes
ipaadmin_password=MySecretPassword123
ipadm_password=MySecretPassword234

[...]
```

### Example of an inventory file with a custom `firewalld` zone

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

[ipaserver:vars]
ipaserver_domain=idm.example.com
ipaserver_realm=IDM.EXAMPLE.COM
ipaserver_setup_dns=yes
ipaserver_auto_forwarders=yes
ipaadmin_password=MySecretPassword123
ipadm_password=MySecretPassword234
ipaserver_firewalld_zone=custom zone

[...]
```

8. Create a playbook for the first step of the installation. Enter instructions for generating the certificate signing request (CSR) and copying it from the controller to the managed node.

```
---
- name: Playbook to configure IPA server Step 1
  hosts: ipaserver
  become: true
  vars_files:
    - playbook_sensitive_data.yml
  vars:
    ipaserver_external_ca: yes

  roles:
    - role: ipaserver
      state: present

  post_tasks:
    - name: Copy CSR /root/ipa.csr from node to "{{ groups.ipaserver[0] + '-ipa.csr' }}"
      fetch:
        src: /root/ipa.csr
        dest: "{{ groups.ipaserver[0] + '-ipa.csr' }}"
        flat: yes
```

9. Create another playbook for the final step of the installation.

```
---
- name: Playbook to configure IPA server Step -1
  hosts: ipaserver
```

```

become: true
vars_files:
- playbook_sensitive_data.yml
vars:
  ipaserver_external_cert_files: "/root/chain.crt"

pre_tasks:
- name: Copy "{{ groups.ipaserver[0] + '-chain.crt' }}" to /root/chain.crt on node
  copy:
    src: "{{ groups.ipaserver[0] + '-chain.crt' }}"
    dest: "/root/chain.crt"
    force: yes

roles:
- role: ipaserver
  state: present

```

### Additional resources

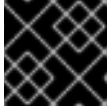
- For the forwarding policy default settings, see the **--forward-policy** description in the **ipa-dns-install(1)** man page.
- For more information about DNS variables used by the **ipaserver** role, see the DNS Variables section in the **README-server.md** file in the **/usr/share/doc/ansible-freeipa** directory.

## 26.6.2. Setting the parameters for a deployment with external DNS and an external CA as the root CA

Complete this procedure to configure the inventory file for installing an IdM server with an external CA as the root CA in an environment that uses an external DNS solution.

### Procedure

1. Open the inventory file for editing. Specify the fully-qualified domain names (**FQDN**) of the host you want to use as an IdM server. Ensure that the **FQDN** meets the following criteria:
  - Only alphanumeric characters and hyphens (-) are allowed. For example, underscores are not allowed and can cause DNS failures.
  - The host name must be all lower-case.
2. Specify the IdM domain and realm information.
3. Make sure that the **ipaserver\_setup\_dns** option is set to **no** or that it is absent.
4. Specify the passwords for **admin** and for the **Directory Manager**. Use the Ansible Vault to store the password, and reference the Vault file from the playbook file. Alternatively and less securely, specify the passwords directly in the inventory file.
5. (Optional) Specify a custom **firewalld** zone to be used by the IdM server. If you do not set a custom zone, IdM will add its services to the default **firewalld** zone. The predefined default zone is **public**.

**IMPORTANT**

The specified **firewalld** zone must exist and be permanent.

**Example of an inventory file with the required server information (excluding the passwords)**

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

[ipaserver:vars]
ipaserver_domain=idm.example.com
ipaserver_realm=IDM.EXAMPLE.COM
ipaserver_setup_dns=no
[...]
```

**Example of an inventory file with the required server information (including the passwords)**

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

[ipaserver:vars]
ipaserver_domain=idm.example.com
ipaserver_realm=IDM.EXAMPLE.COM
ipaserver_setup_dns=no
ipaadmin_password=MySecretPassword123
ipadm_password=MySecretPassword234

[...]
```

**Example of an inventory file with a custom `firewalld` zone**

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

[ipaserver:vars]
ipaserver_domain=idm.example.com
ipaserver_realm=IDM.EXAMPLE.COM
ipaserver_setup_dns=no
ipaadmin_password=MySecretPassword123
ipadm_password=MySecretPassword234
ipaserver_firewalld_zone=custom zone

[...]
```

6. Create a playbook for the first step of the installation. Enter instructions for generating the certificate signing request (CSR) and copying it from the controller to the managed node.

```
---
- name: Playbook to configure IPA server Step 1
  hosts: ipaserver
  become: true
  vars_files:
```

```

- playbook_sensitive_data.yml
vars:
  ipaserver_external_ca: yes

roles:
- role: ipaserver
  state: present

post_tasks:
- name: Copy CSR /root/ipa.csr from node to "{{ groups.ipaserver[0] + '-ipa.csr' }}"
  fetch:
    src: /root/ipa.csr
    dest: "{{ groups.ipaserver[0] + '-ipa.csr' }}"
    flat: yes

```

7. Create another playbook for the final step of the installation.

```

---
- name: Playbook to configure IPA server Step -1
  hosts: ipaserver
  become: true
  vars_files:
  - playbook_sensitive_data.yml
  vars:
    ipaserver_external_cert_files: "/root/chain.crt"

  pre_tasks:
  - name: Copy "{{ groups.ipaserver[0] + '-chain.crt' }}" to /root/chain.crt on node
    copy:
      src: "{{ groups.ipaserver[0] + '-chain.crt' }}"
      dest: "/root/chain.crt"
      force: yes

  roles:
  - role: ipaserver
    state: present

```

### Additional resources

- For details on the options available to you when installing an IdM server with external DNS and an externally signed CA, see [Installing an IdM server: Without integrated DNS, with an external CA as the root CA](#).

### 26.6.3. Deploying an IdM server with an external CA as the root CA using an Ansible playbook

Complete this procedure to deploy an IdM server with an external certificate authority (CA) as the root CA using an Ansible playbook.

#### Procedure

1. Run the **ansible-playbook** command with the name of the playbook file that contains instructions for the first step of the installation, for example **install-server-step1.yml**. Specify the inventory file with the **-i** option:

■

```
$ ansible-playbook -v -i <path_to_inventory_directory>/host.server
<path_to_playbooks_directory>/install-server-step1.yml
```

Specify the level of verbosity by using the **-v**, **-vv** or **-vvv** option.

You can view the output of the Ansible playbook script on the command-line interface (CLI). The following output shows that the script has run successfully as 0 tasks have failed:

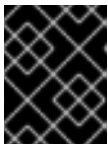
```
PLAY RECAP
server.idm.example.com : ok=18  changed=10  unreachable=0  failed=0  skipped=21
rescued=0  ignored=0
```

2. Locate the **ipa.csr** certificate signing request file on the controller and submit it to the external CA.
3. Place the IdM CA certificate signed by the external CA in the controller file system so that the playbook in the next step can find it.
4. Run the **ansible-playbook** command with the name of the playbook file that contains instructions for the final step of the installation, for example **install-server-step2.yml**. Specify the inventory file with the **-i** option:

```
$ ansible-playbook -v -i <path_to_inventory_directory>/host.server
<path_to_playbooks_directory>/install-server-step2.yml
```

5. Choose one of the following options:
  - If your IdM deployment uses external DNS: add the DNS resource records contained in the **/tmp/ipa.system.records.UFRPto.db** file to the existing external DNS servers. The process of updating the DNS records varies depending on the particular DNS solution.

```
...
Restarting the KDC
Please add records in this file to your DNS system:
/tmp/ipa.system.records.UFRBto.db
Restarting the web server
...
```



### IMPORTANT

The server installation is not complete until you add the DNS records to the existing DNS servers.

- If your IdM deployment uses integrated DNS:
  - Add DNS delegation from the parent domain to the IdM DNS domain. For example, if the IdM DNS domain is **idm.example.com**, add a name server (NS) record to the **example.com** parent domain.



### IMPORTANT

Repeat this step each time after an IdM DNS server is installed.

- Add an **\_ntp.\_udp** service (SRV) record for your time server to your IdM DNS. The presence of the SRV record for the time server of the newly-installed IdM server in IdM DNS ensures that future replica and client installations are automatically configured to synchronize with the time server used by this primary IdM server.

## 26.7. ADDITIONAL RESOURCES

- [Inventory basics: formats, hosts, and groups](#)
- You can see sample Ansible playbooks for installing an IdM server and a list of possible variables in the [ansible-freeipa upstream documentation](#).

## CHAPTER 27. INSTALLING AN IDENTITY MANAGEMENT REPLICA USING AN ANSIBLE PLAYBOOK

### 27.1. IDM REPLICA INSTALLATION USING AN ANSIBLE PLAYBOOK

The following sections describe how to configure a system as an IdM replica by using [Ansible](#). Configuring a system as an IdM replica enrolls it into an IdM domain and enables the system to use IdM services on IdM servers in the domain.

The deployment is managed by the **ipareplica** Ansible role. The role can use the autodiscovery mode for identifying the IdM servers, domain and other settings. However, if you deploy multiple replicas in a tier-like model, with different groups of replicas being deployed at different times, you must define specific servers or replicas for each group.



#### NOTE

Before installing an IdM replica using Ansible, ensure that you understand [Ansible](#) and IdM concepts. Ensure that you understand the following terms that are used in this chapter:

- Ansible roles
- Ansible nodes
- Ansible inventory
- Ansible tasks
- Ansible modules
- Ansible plays and playbooks

#### Overview

The installation consists of the following parts:

1. [Setting the parameters of the IdM replica deployment](#)
  - [Specifying the base, server and client variables for installing the IdM replica](#)
  - [Specifying the credentials for installing the IdM replica using an Ansible playbook](#)
2. [Deploying an IdM replica using an Ansible playbook](#)

#### Prerequisites

- You have installed the [ansible-freeipa](#) package on the Ansible control node.

### 27.2. SETTING THE PARAMETERS OF THE IDM REPLICA DEPLOYMENT

Before you deploy a target host as an IdM replica, configure the following settings:

- [Specify the base, server and client variables for installing the IdM replica.](#)

- [Specify the credentials for installing the IdM replica using an Ansible playbook.](#)

### 27.2.1. Specifying the base, server and client variables for installing the IdM replica

Complete this procedure to configure the inventory file for installing an IdM replica.

#### Procedure

1. Open the inventory file for editing. Specify the fully-qualified domain names (**FQDN**) of the hosts to become IdM replicas. The **FQDNs** must be valid DNS names:
  - Only numbers, alphabetic characters, and hyphens (-) are allowed. For example, underscores are not allowed and can cause DNS failures.
  - The host name must be all lower-case.

#### Example of a simple inventory hosts file with only the replicas' FQDN defined

```
[ipareplicas]
replica1.idm.example.com
replica2.idm.example.com
replica3.idm.example.com
[...]
```

If the IdM server is already deployed and the SRV records are set properly in the IdM DNS zone, the script automatically discovers all the other required values.

2. Optionally, provide additional information in the inventory file based on which of the following scenarios is closest to yours:
  - **Scenario 1**  
If you want to avoid autodiscovery and have all replicas listed in the **[ipareplicas]** section use a specific IdM server, set the server in the **[ipaservers]** section of the inventory file.

#### Example inventory hosts file with the FQDN of the IdM server and replicas defined

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

[ipareplicas]
replica1.idm.example.com
replica2.idm.example.com
replica3.idm.example.com
[...]
```

- **Scenario 2**  
Alternatively, if you want to avoid autodiscovery but want to deploy specific replicas with specific servers, set the servers for specific replicas individually in the **[ipareplicas]** section in the inventory file.

#### Example inventory file with a specific IdM server defined for a specific replica

```
[ipaservers]
```



```
server.idm.example.com
replica1.idm.example.com

[ipareplicas]
replica2.idm.example.com
replica3.idm.example.com ipareplica_servers=replica1.idm.example.com
```

In the example above, **replica3.idm.example.com** uses the already deployed **replica1.idm.example.com** as its replication source.

- **Scenario 3**

If you are deploying several replicas in one batch and time is a concern to you, multitier replica deployment can be useful for you. Define specific groups of replicas in the inventory file, for example **[ipareplicas\_tier1]** and **[ipareplicas\_tier2]**, and design separate plays for each group in the **install-replica.yml** playbook.

#### Example inventory file with replica tiers defined

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

[ipareplicas_tier1]
replica1.idm.example.com

[ipareplicas_tier2]
replica2.idm.example.com \
ipareplica_servers=replica1.idm.example.com,server.idm.example.com
```

The first entry in **ipareplica\_servers** will be used. The second entry will be used as a fallback option. When using multiple tiers for deploying IdM replicas, you must have separate tasks in the playbook to first deploy replicas from tier1 and then replicas from tier2:

#### Example of a playbook file with different plays for different replica groups

```
---
- name: Playbook to configure IPA replicas (tier1)
  hosts: ipareplicas_tier1
  become: true

  roles:
  - role: ipareplica
    state: present

- name: Playbook to configure IPA replicas (tier2)
  hosts: ipareplicas_tier2
  become: true

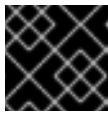
  roles:
  - role: ipareplica
    state: present
```

- **Scenario 4**

If you want the replica to use a specified **firewalld** zone instead of the default one, you can specify it in the inventory file. This can be useful, for example, when you want to use an internal **firewalld** zone for your IdM installation instead of a public zone that is set as

default.

If you do not set a custom zone, IdM will add its services to the default **firewalld** zone. The predefined default zone is **public**.



### IMPORTANT

The specified **firewalld** zone must exist and be permanent.

### Example of a simple inventory hosts file with a custom **firewalld** zone

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

[ipareplicas]
replica1.idm.example.com
replica2.idm.example.com
replica3.idm.example.com
[...]

[ipareplicas:vars]
ipareplica_firewalld_zone=custom zone
```

- Scenario 5

If you want the replica to host the IdM DNS service, add the **ipareplica\_setup\_dns=yes** line to the **[ipareplicas:vars]** section. Additionally, specify if you want to use per-server DNS forwarders:

- To configure per-server forwarders, add the **ipareplica\_forwarders** variable and a list of strings to the **[ipareplicas:vars]** section, for example:  
**ipareplica\_forwarders=192.0.2.1,192.0.2.2**
- To configure no per-server forwarders, add the following line to the **[ipareplicas:vars]** section: **ipareplica\_no\_forwarders=yes**.
- To configure per-server forwarders based on the forwarders listed in the **/etc/resolv.conf** file of the replica, add the **ipareplica\_auto\_forwarders** variable to the **[ipareplicas:vars]** section.

### Example inventory file with instructions to set up DNS and per-server forwarders on the replicas

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

[ipareplicas]
replica1.idm.example.com
replica2.idm.example.com
replica3.idm.example.com
[...]

[ipareplicas:vars]
ipareplica_setup_dns=yes
ipareplica_forwarders=192.0.2.1,192.0.2.2
```

## Additional resources

- For more information on the **ipareplica** variables, see the **/usr/share/ansible/roles/ipareplica/README.md** Markdown file.

### 27.2.2. Specifying the credentials for installing the IdM replica using an Ansible playbook

Complete this procedure to configure the authorization for installing the IdM replica.

#### Procedure

1. Specify the **password of a user authorized to deploy replicas** for example the IdM **admin**.

- Red Hat recommends using the Ansible Vault to store the password, and referencing the Vault file from the playbook file, for example **install-replica.yml**:

**Example playbook file using principal from inventory file and password from an Ansible Vault file**

```
- name: Playbook to configure IPA replicas
  hosts: ipareplicas
  become: true
  vars_files:
    - playbook_sensitive_data.yml

  roles:
    - role: ipareplica
      state: present
```

For details how to use Ansible Vault, see the official [Ansible Vault](#) documentation.

- Less securely, provide the credentials of **admin** directly in the inventory file. Use the **ipaadmin\_password** option in the **[ipareplicas:vars]** section of the inventory file. The inventory file and the **install-replica.yml** playbook file can then look as follows:

**Example inventory hosts.replica file**

```
[...]
[ipareplicas:vars]
ipaadmin_password=Secret123
```

**Example playbook using principal and password from inventory file**

```
- name: Playbook to configure IPA replicas
  hosts: ipareplicas
  become: true

  roles:
    - role: ipareplica
      state: present
```

- Alternatively but also less securely, provide the credentials of another user authorized to deploy a replica directly in the inventory file. To specify a different authorized user, use the

**ipaadmin\_principal** option for the user name, and the **ipaadmin\_password** option for the password. The inventory file and the **install-replica.yml** playbook file can then look as follows:

#### Example inventory hosts.replica file

```
[...]
[ipareplicas:vars]
ipaadmin_principal=my_admin
ipaadmin_password=my_admin_secret123
```

#### Example playbook using principal and password from inventory file

```
- name: Playbook to configure IPA replicas
  hosts: ipareplicas
  become: true

  roles:
    - role: ipareplica
      state: present
```

#### Additional resources

- For details on the options accepted by the **ipareplica** Ansible role, see the **/usr/share/ansible/roles/ipareplica/README.md** Markdown file.

## 27.3. DEPLOYING AN IDM REPLICA USING AN ANSIBLE PLAYBOOK

Complete this procedure to use an Ansible playbook to deploy an IdM replica.

#### Procedure

- To install an IdM replica using an Ansible playbook, use the **ansible-playbook** command with the name of the playbook file, for example **install-replica.yml**. Specify the inventory file with the **-i** option:

```
$ ansible-playbook -v -i <path_to_inventory_directory>/hosts.replica
<path_to_playbooks_directory>/install-replica.yml
```

Specify the level of verbosity by using the **-v**, **-vv** or **-vvv** option.

Ansible informs you about the execution of the Ansible playbook script. The following output shows that the script has run successfully as 0 tasks have failed:

```
PLAY RECAP
 replica.idm.example.com : ok=18  changed=10  unreachable=0  failed=0  skipped=21
 rescued=0  ignored=0
```

You have now installed an IdM replica.

## CHAPTER 28. INSTALLING AN IDENTITY MANAGEMENT CLIENT USING AN ANSIBLE PLAYBOOK

### 28.1. IDM CLIENT INSTALLATION USING AN ANSIBLE PLAYBOOK

The following sections describe how to configure a system as an Identity Management (IdM) client by using [Ansible](#). Configuring a system as an IdM client enrolls it into an IdM domain and enables the system to use IdM services on IdM servers in the domain.

The deployment is managed by the **ipaclient** Ansible role. By default, the role uses the autodiscovery mode for identifying the IdM servers, domain and other settings. The role can be modified to have the Ansible playbook use the settings specified, for example in the inventory file.



#### NOTE

Before installing an IdM client using Ansible, ensure that you understand [Ansible](#) and IdM concepts. Ensure that you understand the following terms that are used in this chapter:

- Ansible roles
- Ansible nodes
- Ansible inventory
- Ansible tasks
- Ansible modules
- Ansible plays and playbooks

#### Overview

The installation consists of the following parts:

1. [Setting the parameters of the IdM client deployment](#) to correspond to your deployment scenario:
  - Setting the parameters of the inventory file [for the autodiscovery client installation mode](#);
  - Setting the parameters of the inventory file [for when autodiscovery is not possible during client installation](#);
2. [Checking the parameters in install-client.yml](#);
3. [Deploying an IdM client using an Ansible playbook](#) ;
4. [Testing an Identity Management client after installation](#) .

The chapter also includes a section describing [how to uninstall an IdM client](#) .

#### Prerequisites

- You have installed the [ansible-freeipa](#) package on the Ansible control node.

## 28.2. SETTING THE PARAMETERS OF THE IDM CLIENT DEPLOYMENT

Before you deploy a target host as an IdM client, configure the [deployment instructions](#) on the control node. Additionally, configure the target host parameters depending on which of the following options you are planning:

- [Using the autodiscovery client installation mode](#)
- [Specifying the \*\*FQDN\*\* of the IdM server and the domain or realm information](#).

### 28.2.1. Setting the parameters of the inventory file for the autodiscovery client installation mode

To install an Identity Management client using an Ansible playbook, provide the following information in an inventory file, for example **inventory/hosts**:

- the information about the host
- the authorization for the task

The inventory file can be in one of many formats, depending on the inventory plugins you have. The **INI-like** format is one of Ansible's defaults and is used in the examples below.



#### NOTE

To use smart cards with the graphical user interface in RHEL, ensure that you include the **ipaclient\_mkhomedir** variable in your Ansible playbook.

#### Procedure

1. Specify the fully-qualified hostname (**FQDN**) of the host to become an IdM client. The fully qualified domain name must be a valid DNS name:
  - Only numbers, alphabetic characters, and hyphens (-) are allowed. For example, underscores are not allowed and can cause DNS failures.
  - The host name must be all lower-case. No capital letters are allowed.  
If the SRV records are set properly in the IdM DNS zone, the script automatically discovers all the other required values.

#### Example of a simple inventory hosts file with only the client FQDN defined

```
[ipaclients]
client.idm.example.com
[...]
```

2. Specify the credentials for enrolling the client. The following authentication methods are available:
  - The **password of a user authorized to enroll clients** This is the default option.
    - Red Hat recommends using the Ansible Vault to store the password, and referencing the Vault file from the playbook file, for example **install-client.yml**, directly:

### Example playbook file using principal from inventory file and password from an Ansible Vault file

```
- name: Playbook to configure IPA clients with username/password
  hosts: ipaclients
  become: true
  vars_files:
    - playbook_sensitive_data.yml

  roles:
    - role: ipaclient
      state: present
```

- Less securely, provide the credentials of **admin** using the **ipaadmin\_password** option in the **[ipaclients:vars]** section of the **inventory/hosts** file. Alternatively, to specify a different authorized user, use the **ipaadmin\_principal** option for the user name, and the **ipaadmin\_password** option for the password. The **inventory/hosts** inventory file and the **install-client.yml** playbook file can then look as follows:

### Example inventory hosts file

```
[...]
[ipaclients:vars]
ipaadmin_principal=my_admin
ipaadmin_password=Secret123
```

### Example Playbook using principal and password from inventory file

```
- name: Playbook to unconfigure IPA clients
  hosts: ipaclients
  become: true

  roles:
    - role: ipaclient
      state: true
```

- The **client keytab** from the previous enrollment if it is still available:
  - This option is available if the system was previously enrolled as an Identity Management client. To use this authentication method, uncomment the **#ipaclient\_keytab** option, specifying the path to the file storing the keytab, for example in the **[ipaclient:vars]** section of **inventory/hosts**.
- A **random, one-time password**(OTP) to be generated during the enrollment. To use this authentication method, use the **ipaclient\_use\_otp=yes** option in your inventory file. For example, you can uncomment the **ipaclient\_use\_otp=yes** option in the **[ipaclients:vars]** section of the **inventory/hosts** file. Note that with OTP you must also specify one of the following options:
  - The **password of a user authorized to enroll clients** for example by providing a value for **ipaadmin\_password** in the **[ipaclients:vars]** section of the **inventory/hosts** file.
  - The **admin keytab**, for example by providing a value for **ipaadmin\_keytab** in the **[ipaclients:vars]** section of **inventory/hosts**.

## Additional resources

- For details on the options accepted by the **ipacient** Ansible role, see the **/usr/share/ansible/roles/ipacient/README.md** README file.

### 28.2.2. Setting the parameters of the inventory file when autodiscovery is not possible during client installation

To install an Identity Management client using an Ansible playbook, provide the following information in an inventory file, for example **inventory/hosts**:

- the information about the host, the IdM server and the IdM domain or the IdM realm
- the authorization for the task

The inventory file can be in one of many formats, depending on the inventory plugins you have. The **INI-like** format is one of Ansible's defaults and is used in the examples below.



#### NOTE

To use smart cards with the graphical user interface in RHEL, ensure that you include the **ipacient\_mkhomedir** variable in your Ansible playbook.

## Procedure

1. Specify the fully-qualified hostname (**FQDN**) of the host to become an IdM client. The fully qualified domain name must be a valid DNS name:
  - Only numbers, alphabetic characters, and hyphens (-) are allowed. For example, underscores are not allowed and can cause DNS failures.
  - The host name must be all lower-case. No capital letters are allowed.
2. Specify other options in the relevant sections of the **inventory/hosts** file:
  - the **FQDN** of the servers in the **[ipaservers]** section to indicate which IdM server the client will be enrolled with
  - one of the two following options:
    - the **ipacient\_domain** option in the **[ipaclients:vars]** section to indicate the DNS domain name of the IdM server the client will be enrolled with
    - the **ipacient\_realm** option in the **[ipaclients:vars]** section to indicate the name of the Kerberos realm controlled by the IdM server

**Example of an inventory hosts file with the client FQDN, the server FQDN and the domain defined**

```
[ipaclients]
client.idm.example.com

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



```
[ipaclients:vars]
ipaclient_domain=idm.example.com
[...]
```

3. Specify the credentials for enrolling the client. The following authentication methods are available:

- The **password of a user authorized to enroll clients** This is the default option.
  - Red Hat recommends using the Ansible Vault to store the password, and referencing the Vault file from the playbook file, for example **install-client.yml**, directly:

#### Example playbook file using principal from inventory file and password from an Ansible Vault file

```
- name: Playbook to configure IPA clients with username/password
  hosts: ipaclients
  become: true
  vars_files:
  - playbook_sensitive_data.yml

  roles:
  - role: ipaclient
    state: present
```

- Less securely, provide the credentials of **admin** using the **ipaadmin\_password** option in the **[ipaclients:vars]** section of the **inventory/hosts** file. Alternatively, to specify a different authorized user, use the **ipaadmin\_principal** option for the user name, and the **ipaadmin\_password** option for the password. The **install-client.yml** playbook file can then look as follows:

#### Example inventory hosts file

```
[...]
[ipaclients:vars]
ipaadmin_principal=my_admin
ipaadmin_password=Secret123
```

#### Example Playbook using principal and password from inventory file

```
- name: Playbook to unconfigure IPA clients
  hosts: ipaclients
  become: true

  roles:
  - role: ipaclient
    state: true
```

- The **client keytab** from the previous enrollment if it is still available:
  - This option is available if the system was previously enrolled as an Identity Management client. To use this authentication method, uncomment the **ipaclient\_keytab** option, specifying the path to the file storing the keytab, for example in the **[ipaclient:vars]** section of **inventory/hosts**.

- A **random, one-time password**(OTP) to be generated during the enrollment. To use this authentication method, use the **ipaclient\_use\_otp=yes** option in your inventory file. For example, you can uncomment the **#ipaclient\_use\_otp=yes** option in the **[ipaclients:vars]** section of the **inventory/hosts** file. Note that with OTP you must also specify one of the following options:
  - The **password of a user authorized to enroll clients** for example by providing a value for **ipaadmin\_password** in the **[ipaclients:vars]** section of the **inventory/hosts** file.
  - The **admin keytab**, for example by providing a value for **ipaadmin\_keytab** in the **[ipaclients:vars]** section of **inventory/hosts**.

### Additional resources

- For details on the options accepted by the **ipaclient** Ansible role, see the **/usr/share/ansible/roles/ipaclient/README.md** README file.

### 28.2.3. Checking the parameters in the install-client.yml file

The **install-client.yml** playbook file contains instructions for the IdM client deployment.

- Open the file and check if the instructions in the playbook correspond to what you are planning for your deployment. The contents typically look like this:

```
---
- name: Playbook to configure IPA clients with username/password
  hosts: ipaclients
  become: true

  roles:
  - role: ipaclient
    state: present
```

This is what the individual entries mean:

- The **hosts** entry specifies the section of the **inventory/hosts** file where the ansible script searches the **FQDNs** of the hosts on which the **ipa-client-install** script shall be run.
- The **become: true** entry specifies that root's credentials will be invoked during the execution of the **ipa-client-install** script.
- The **role: ipaclient** entry specifies the role that will be installed on the host: in this case, it is the ipa client role.
- The **state: present** entry specifies that the client should be installed rather than uninstalled (**absent**).

### 28.2.4. Authorization options for IdM client enrollment using an Ansible playbook

This referential section presents individual authorization options for IdM client enrollment with examples of inventory and playbook files.

**Table 28.1. Authorization options for IdM client enrollment using Ansible**

Authorization option	Note	Example inventory file	Example install-client.yml playbook file
Password of a user authorized to enroll a client: Option 1	Password stored in Ansible vault	<pre>[ipaclients:vars] [...]</pre>	<pre>- name: Playbook to configure IPA clients with   username/password   hosts: ipaclients   become: true   vars_files:   - playbook_sensitive_data.yml    roles:   - role: ipaclient     state: present</pre>
Password of a user authorized to enroll a client: Option 2	Password stored in inventory file	<pre>[ipaclients:vars] ipaadmin_password=Secret123</pre>	<pre>- name: Playbook to configure IPA clients   hosts: ipaclients   become: true    roles:   - role: ipaclient     state: true</pre>
A random, one-time password (OTP): Option 1	OTP + administrat or password	<pre>[ipaclients:vars] ipaadmin_password=Secret123 ipaclient_use_otp=yes</pre>	<pre>- name: Playbook to configure IPA clients   hosts: ipaclients   become: true    roles:   - role: ipaclient     state: true</pre>
A random, one-time password (OTP): Option 2	OTP + an admin keytab	<pre>[ipaclients:vars] ipaadmin_keytab=/tmp/admin.keytab ipaclient_use_otp=yes</pre>	<pre>- name: Playbook to configure IPA clients   hosts: ipaclients   become: true    roles:   - role: ipaclient     state: true</pre>

Authorization option	Note	Example inventory file	Example <code>install-client.yml</code> playbook file
The client keytab from the previous enrollment		<pre>[ipaclients:vars] ipaclient_keytab=/tmp/krb5.keytab</pre>	<pre>- name: Playbook to configure IPA clients   hosts: ipaclients   become: true    roles:   - role: ipaclient     state: true</pre>

### 28.3. DEPLOYING AN IDM CLIENT USING AN ANSIBLE PLAYBOOK

Complete this procedure to use an Ansible playbook to deploy an IdM client in your IdM environment.

#### Procedure

- To install an IdM client using an Ansible playbook, use the **ansible-playbook** command with the name of the playbook file, for example **install-client.yml**. Specify the inventory file with the **-i** option:

```
$ ansible-playbook -v -i inventory/hosts install-client.yml
```

Specify the level of verbosity by using the **-v**, **-vv** or **-vvv** option.

Ansible informs you about the execution of the Ansible playbook script. The following output shows that the script has run successfully as no tasks have failed:

```
PLAY RECAP
client1.idm.example.com : ok=18 changed=10 unreachable=0 failed=0 skipped=21
rescued=0 ignored=0
```



#### NOTE

Ansible uses different colors to provide different types of information about the running process. You can modify the default colors in the **[colors]** section of the **/etc/ansible/ansible.cfg** file:

```
[colors]
[...]
#error = red
#debug = dark gray
#deprecate = purple
#skip = cyan
#unreachable = red
#ok = green
#changed = yellow
[...]
```

You have now installed an IdM client on your host using an Ansible playbook.

## 28.4. TESTING AN IDENTITY MANAGEMENT CLIENT AFTER ANSIBLE INSTALLATION

The command-line interface (CLI) informs you that the **ansible-playbook** command was successful, but you can also do your own test.

To test that the Identity Management client can obtain information about users defined on the server, check that you are able to resolve a user defined on the server. For example, to check the default **admin** user:

```
[user@client1 ~]$ id admin
uid=1254400000(admin) gid=1254400000(admins) groups=1254400000(admins)
```

To test that authentication works correctly, **su -** as another already existing IdM user:

```
[user@client1 ~]$ su - idm_user
Last login: Thu Oct 18 18:39:11 CEST 2018 from 192.168.122.1 on pts/0
[idm_user@client1 ~]$
```

## 28.5. UNINSTALLING AN IDM CLIENT USING AN ANSIBLE PLAYBOOK

Complete this procedure to use an Ansible playbook to uninstall your host as an IdM client.

### Prerequisites

- IdM administrator credentials.

### Procedure

- To uninstall the IdM client, use the **ansible-playbook** command with the name of the playbook file, for example **uninstall-client.yml**. Specify the inventory file with the **-i** option and, optionally, specify the level of verbosity by using the **-v**, **-vv** or **-vvv** options:

```
$ ansible-playbook -v -i inventory/hosts uninstall-client.yml
```



## IMPORTANT

The uninstallation of the client only removes the basic IdM configuration from the host but leaves the configuration files on the host in case you decide to re-install the client. In addition, the uninstallation has the following limitations:

- It does not remove the client host entry from the IdM LDAP server. The uninstallation only unenrolls the host.
- It does not remove any services residing on the client from IdM.
- It does not remove the DNS entries for the client from the IdM server.
- It does not remove the old principals for keytabs other than **/etc/krb5.keytab**.

Note that the uninstallation does remove all certificates that were issued for the host by the IdM CA.

### Additional resources

- For more information on how to remove the IdM client configuration from both the host and the IdM environment completely, see the manual procedure for [Uninstalling an IdM client](#).

## PART II. INTEGRATING IDM AND AD

## CHAPTER 29. INSTALLING TRUST BETWEEN IDM AND AD

This chapter aims to help you create a trust between the Identity Management IdM server and Active Directory (AD), where both servers are located in the same forest.

### Prerequisites

- First, read the [Planning a cross-forest trust between Identity Management and Active Directory](#) document.
- AD is installed with a domain controller on it.
- The IdM server is installed and running.
  - For details, see [Installing Identity Management](#).
- Both the AD server and the IdM server must have their clocks in sync because Kerberos requires max 5 mins delay in communication.
- Unique NetBIOS names for each of the servers placed in the trust because the NetBIOS names are critical for identifying the Active Directory domain.
  - The NetBIOS name of an Active Directory or IdM domain is usually the first part of the corresponding DNS domain. If the DNS domain is **ad.example.com**, the NetBIOS name is typically **AD**. However, it is not required. Important is that the NetBIOS name is just one word without periods. The maximum length of a NetBIOS name is 15 characters.
- The IdM system must have the IPv6 protocol enabled in the kernel.
  - If IPv6 is disabled, then the CLDAP plug-in used by the IdM services fails to initialize.

### 29.1. SUPPORTED VERSIONS OF WINDOWS SERVER

In RHEL 8.4, Identity Management (IdM) does not support establishing trust to Active Directory with Active Directory domain controllers running Windows Server 2008 R2 or earlier versions. RHEL IdM now requires SMB encryption when establishing the trust relationship, which is only supported in Windows Server 2012 or later.

You can establish a trust relationship with Active Directory (AD) forests that use the following forest and domain functional levels:

- Forest functional level range: Windows Server 2012 – Windows Server 2016
- Domain functional level range: Windows Server 2012 – Windows Server 2016

Identity Management (IdM) supports establishing a trust with Active Directory domain controllers running the following operating systems:

- Windows Server 2012
- Windows Server 2012 R2
- Windows Server 2016
- Windows Server 2019



## 29.2. HOW THE TRUST WORKS

The trust between Identity Management IdM and Active Directory (AD) is established on the Cross-realm Kerberos trust. This solution uses the Kerberos capability to establish trusts between different identity sources. Therefore, all AD users can:

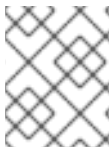
- Log in to access Linux systems and resources.
- Use single sign-on (SSO).

All IdM objects are managed in IdM in the trust.

All AD objects are managed in AD in the trust.

In complex environments, a single IdM forest can be connected to multiple AD forests. This setup enables better separation of duties for different functions in the organization. AD administrators can focus on users and policies related to users while Linux administrators have full control over the Linux infrastructure. In such a case, the Linux realm controlled by IdM is analogous to an AD resource domain or realm but with Linux systems in it.

From the perspective of AD, Identity Management represents a separate AD forest with a single AD domain. When cross-forest trust between an AD forest root domain and an IdM domain is established, users from the AD forest domains can interact with Linux machines and services from the IdM domain.



### NOTE

In trust environments, IdM enables you to use ID views to configure POSIX attributes for AD users on the IdM server.

## 29.3. AD ADMINISTRATION RIGHTS

When you want to build a trust between AD (Active Directory) and IdM (Identity Management), you will need to use an AD administrator account with appropriate AD privileges.

Such an AD administrator must be a member of one of the following groups:

- Enterprise Admin group in the AD forest
- Domain Admins group in the forest root domain for your AD forest

### Additional resources

- For details about Enterprise Admins, see [Enterprise Admins](#).
- For details about Domain Admins, see [Domain Admins](#).
- For details about AD trust, see [How Domain and Forest Trusts Work](#).

## 29.4. ENSURING SUPPORT FOR COMMON ENCRYPTION TYPES IN AD AND RHEL

By default, Identity Management establishes a cross-realm trust with support for RC4, AES-128, and AES-256 Kerberos encryption types.

RC4 encryption has been deprecated and disabled by default, as it is considered less secure than the newer AES-128 and AES-256 encryption types. In contrast, Active Directory (AD) user credentials and trusts between AD domains support RC4 encryption and they might not support AES encryption types.

Without any common encryption types, communication between IdM and AD child domains might not work, or some AD accounts might not be able to authenticate. To remedy this situation, modify one of the following configurations:

- **Enable AES encryption support in Active Directory (recommended option)** To ensure trusts between AD domains in an AD forest support strong AES encryption types, see the following Microsoft article: [AD DS: Security: Kerberos "Unsupported etype" error when accessing a resource in a trusted domain](#)
- **Enable RC4 support in RHEL:** On every IdM trust controller, trust agent, and client where authentication against AD Domain Controllers takes place:
  1. Use the **update-crypto-policies** command to enable the **AD-SUPPORT** cryptographic subpolicy in addition to the **DEFAULT** cryptographic policy.

```
[root@host ~]# update-crypto-policies --set DEFAULT:AD-SUPPORT
Setting system policy to DEFAULT:AD-SUPPORT
Note: System-wide crypto policies are applied on application start-up.
It is recommended to restart the system for the change of policies
to fully take place.
```

2. Restart the host.

## IMPORTANT

The **AD-SUPPORT** cryptographic subpolicy is only available on RHEL 8.3 and newer.

- To enable support for RC4 in RHEL 8.2, create and enable a custom cryptographic module policy with **cipher = RC4-128+**. For more details, see [Customizing system-wide cryptographic policies with policy modifiers](#).
- To enable support for RC4 in RHEL 8.0 and RHEL 8.1, add **+rc4** to the **permitted\_etypes** option in the **/etc/crypto-policies/back-ends/krb5.config** file:

```
[libdefaults]
permitted_etypes = aes256-cts-hmac-sha1-96 aes256-cts-hmac-sha384-
192 camellia256-cts-cmac aes128-cts-hmac-sha1-96 aes128-cts-hmac-
sha256-128 camellia128-cts-cmac +rc4
```

## Additional resources

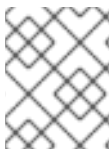
- For more information on working with RHEL cryptographic policies, see [Using system-wide cryptographic policies](#) in the Security Hardening guide.
- For more information on IdM trust agents and trust controllers, see [Trust controllers and trust agents](#) in the Planning Identity Management guide.

## 29.5. PORTS REQUIRED FOR COMMUNICATION BETWEEN IDM AND AD

To enable communication between your Active Directory (AD) and Identity Management (IdM) environments, open the following ports on the firewalls of your AD Domain Controllers and IdM servers.

**Table 29.1. Ports required for an AD trust**

Service	Port	Protocol
Endpoint resolution portmapper	135	TCP
NetBIOS-DGM	138	TCP and UDP
NetBIOS-SSN	139	TCP and UDP
Microsoft-DS	445	TCP and UDP
Dynamic RPC	49152-65535	TCP
AD Global Catalog	3268	TCP
LDAP	389	TCP and UDP



#### NOTE

The TCP port 389 is not required to be open on IdM servers for trust, but it is necessary for clients communicating with the IdM server.

To open ports, you can use the following methods:

- Firewall service – you can enable the particular ports or enable the following services which includes the ports:
  - FreeIPA trust setup
  - FreeIPA with LDAP
  - Kerberos
  - DNS

For details, see [Controlling ports using CLI](#).

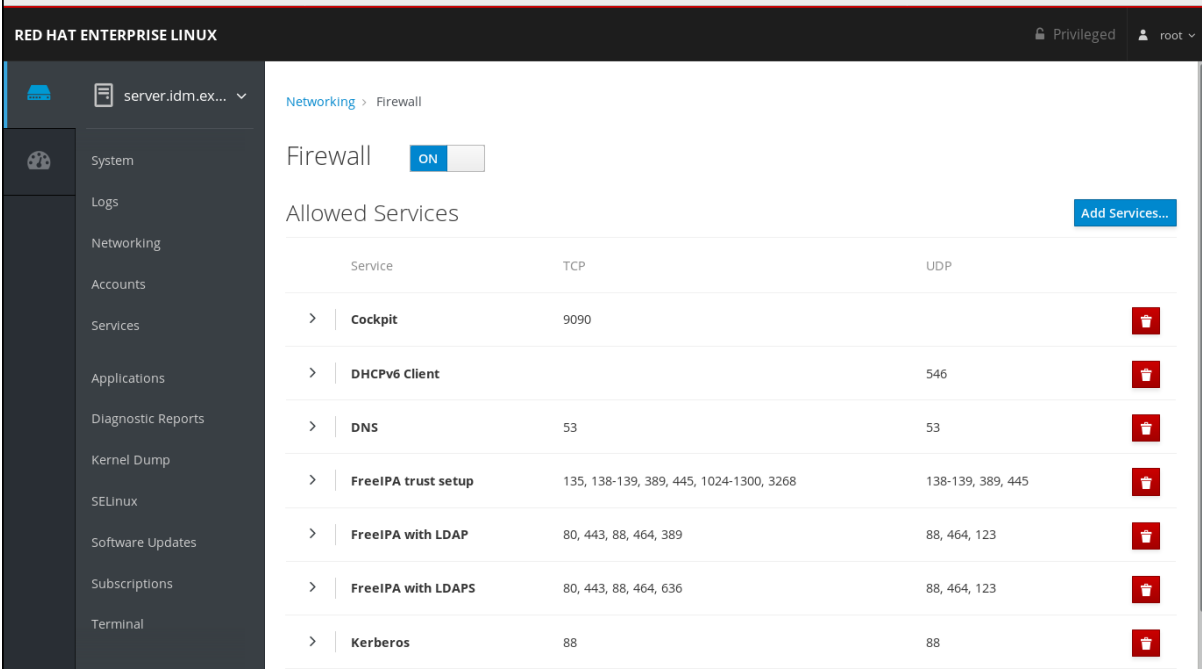


#### NOTE

The **freeipa-trust** Firewall service currently includes an RPC port range of **1024-1300**, but this range has been updated to **49152-65535** in Windows Server 2008 and later. The **freeipa-trust** Firewall service will be updated to reflect this new range, and this issue is tracked in [Bug 1850418 - update freeipa-trust.xml definition to include correct dynamic RPC range](#).

Until that bug has been resolved, manually open the TCP port range **49152-65535** in addition to enabling the **freeipa-trust** Firewall service.

- The RHEL web console, which is a UI with firewall settings based on the **firewalld** service.



For details about firewall configuration through the web console, see [Enabling services on the firewall using the web console](#).



**NOTE**

The **FreeIPA Trust Setup** service currently includes an RPC port range of **1024-1300**, but this range has been updated to **49152-65535** in Windows Server 2008 and later. The **FreeIPA Trust Setup** firewall service definition will be updated, and this issue is tracked in [Bug 1850418 - update freeipa-trust.xml definition to include correct dynamic RPC range](#).

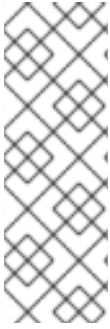
Until that bug has been resolved, manually open the TCP port range **49152-65535** in addition to enabling the **FreeIPA Trust Setup** service in the RHEL web console.

Table 29.2. Ports required by IdM servers in a trust

Service	Port	Protocol
Kerberos	88, 464	TCP and UDP
LDAP	389	TCP
DNS	53	TCP and UDP

Table 29.3. Ports required by IdM clients in an AD trust

Service	Port	Protocol
Kerberos	88	UDP and TCP



## NOTE

The **libkrb5** library uses UDP and falls back to the TCP protocol if the data sent from the Key Distribution Centre (KDC) is too large. Active Directory attaches a Privilege Attribute Certificate (PAC) to the Kerberos ticket, which increases the size and requires to use the TCP protocol. To avoid the fall-back and resending the request, by default, SSSD in Red Hat Enterprise Linux 7.4 and later uses TCP for user authentication. If you want to configure the size before libkrb5 uses TCP, set the **udp\_preference\_limit** in the **/etc/krb5.conf** file. For details, see the **krb5.conf(5)** man page.

## Additional resources

- For more information on the Dynamic RPC port range in Windows Server 2008 and later, see [The default dynamic port range for TCP/IP has changed since Windows Vista and in Windows Server 2008](#).

## 29.6. CONFIGURING DNS AND REALM SETTINGS FOR A TRUST

Before you connect Identity Management (IdM) and Active Directory (AD) in a trust, you need to ensure that servers see each other and resolve domain names correctly. This scenario describes configuring DNS to allow using domain names between:

- One primary IdM server using integrated DNS server and Certification Authority.
- One AD Domain Controller.

DNS settings require:

- Configuring DNS zones in the IdM server
- Configuring conditional DNS forwarding in AD
- Verifying correctness of the DNS configuration

### 29.6.1. Unique primary DNS domains

In Windows, every domain is a Kerberos realm and a DNS domain at the same time. Every domain managed by the domain controller needs to have its own dedicated DNS zone. The same applies when Identity Management (IdM) is trusted by Active Directory (AD) as a forest. AD expects IdM to have its own DNS domain. For the trust setup to work, the DNS domain needs to be dedicated to the Linux environment.

Each system must have its own unique primary DNS domain configured. For example:

- **ad.example.com** for AD and **idm.example.com** for IdM
- **example.com** for AD and **idm.example.com** for IdM
- **ad.example.com** for AD and **example.com** for IdM

The most convenient management solution is an environment where each DNS domain is managed by integrated DNS servers, but it is possible to use any other standard-compliant DNS server as well.

### Kerberos realm names as upper-case versions of primary DNS domain names

Kerberos realm names must be the same as the primary DNS domain names, with all letters uppercase. For example, if the domain names are **ad.example.com** for AD and **idm.example.com**

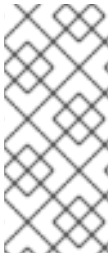
for IdM, the Kerberos realm names are required to be **AD.EXAMPLE.COM** and **IDM.EXAMPLE.COM**.

### DNS records resolvable from all DNS domains in the trust

All machines must be able to resolve DNS records from all DNS domains involved in the trust relationship.

### IdM and AD DNS Domains

Systems joined to IdM can be distributed over multiple DNS domains. Red Hat recommends that you deploy IdM clients in a DNS zone different to the ones owned by Active Directory. The primary IdM DNS domain must have proper SRV records to support AD trusts.



#### NOTE

In some environments with trusts between IdM and Active Directory, you can install an IdM client on a host that is part of the Active Directory DNS domain. The host can then benefit from the Linux-focused features of IdM. This is not a recommended configuration and has some limitations. See [Configuring IdM clients in an Active Directory DNS domain](#) for more details.

You can acquire a list of the required SRV records specific to your system setup by running the following command:

```
$ ipa dns-update-system-records --dry-run
```

The generated list can look for example like this:

#### IPA DNS records:

```
_kerberos-master._tcp.idm.example.com. 86400 IN SRV 0 100 88 server.idm.example.com.
_kerberos-master._udp.idm.example.com. 86400 IN SRV 0 100 88 server.idm.example.com.
_kerberos._tcp.idm.example.com. 86400 IN SRV 0 100 88 server.idm.example.com.
_kerberos._tcp.idm.example.com. 86400 IN SRV 0 100 88 server.idm.example.com.
_kerberos.idm.example.com. 86400 IN TXT "IDM.EXAMPLE.COM"
_kpasswd._tcp.idm.example.com. 86400 IN SRV 0 100 464 server.idm.example.com.
_kpasswd._udp.idm.example.com. 86400 IN SRV 0 100 464 server.idm.example.com.
_ldap._tcp.idm.example.com. 86400 IN SRV 0 100 389 server.idm.example.com.
_ipa-ca.idm.example.com. 86400 IN A 192.168.122.2
```

For other DNS domains that are part of the same IdM realm, it is not required for the SRV records to be configured when the trust to AD is configured. The reason is that AD domain controllers do not use SRV records to discover KDCs but rather base the KDC discovery on name suffix routing information for the trust.

## 29.6.2. Configuring a DNS forward zone in the IdM Web UI

This section describes how to add a new DNS forward zone to the Identity Management (IdM) server using the IdM Web UI.

With DNS forward zones, you can forward DNS queries for a specific zone to a different DNS server. For example, you can forward DNS queries for the Active Directory (AD) domain to an AD DNS server.

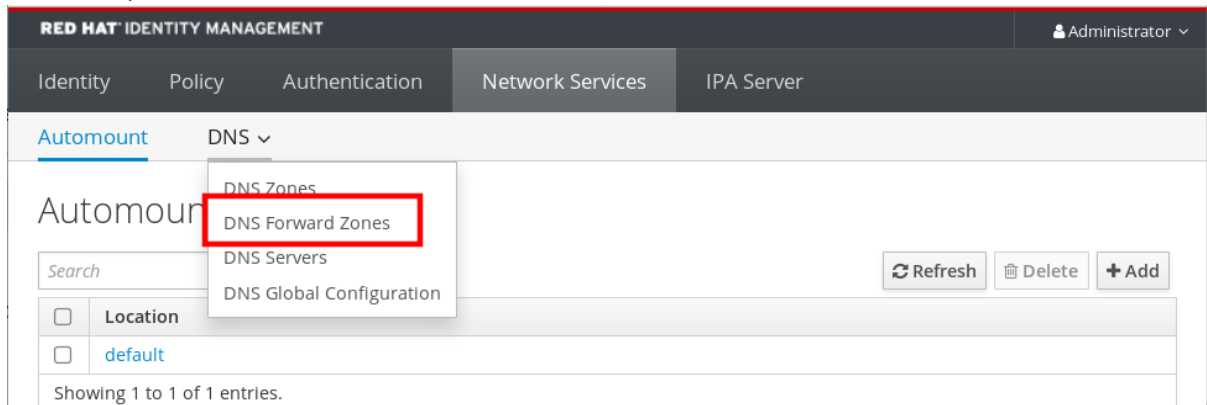
### Prerequisites

- Access to the IdM Web UI with a user account that has administrator rights.

- Correctly configured DNS server.

## Procedure

1. Log in to the IdM Web UI with administrator privileges. For details, see [Accessing the IdM Web UI in a web browser](#).
2. Click on the **Network Services** tab.
3. Click on the **DNS** tab.
4. In the drop down menu, click on the **DNS Forward Zones** item.



5. Click on the **Add** button.
6. In the **Add DNS forward zone** dialog box, add a zone name.
7. In the **Zone forwarders** item, click on the **Add** button.
8. In the **Zone forwarders** field, add the IP address of the server for which you want to create the new forward zone.
9. Click on the **Add** button.

Add DNS forward zone

Zone name \*

ad.example.com

Reverse zone

IP network

Zone forwarders \*

192.168.122.3

Undo

Undo

Add

Forward policy

Forward first

Forward only

Forwarding disabled

Skip overlap check

i

\* Required field

Add

Add and Add Another

Add and Edit

Cancel

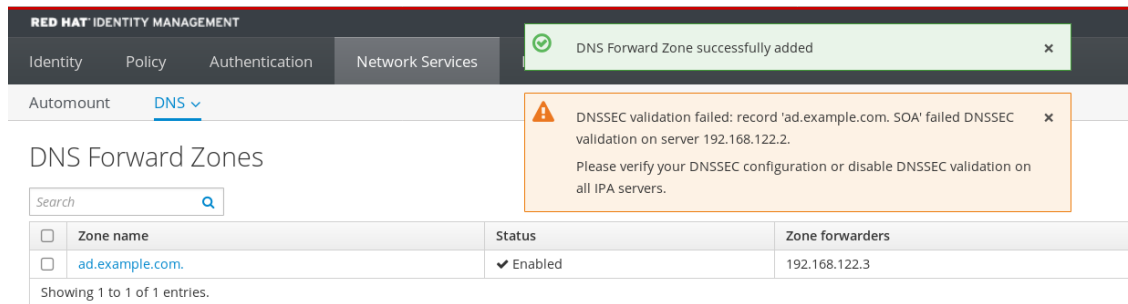
The forwarded zone has been added to the DNS settings and you can verify it in the DNS Forward Zones settings. The Web UI informs you about success with the following pop-up message: **DNS Forward Zone successfully added.**

164



## NOTE

The Web UI might display a warning about a DNSSEC validation failure after adding a new forward zone to the configuration.



RED HAT IDENTITY MANAGEMENT

Identity Policy Authentication Network Services

Automount DNS

DNS Forward Zones

Search

Zone name	Status	Zone forwarders
ad.example.com	✓ Enabled	192.168.122.3

Showing 1 to 1 of 1 entries.

DNSSEC validation failed: record 'ad.example.com. SOA' failed DNSSEC validation on server 192.168.122.2.  
Please verify your DNSSEC configuration or disable DNSSEC validation on all IPA servers.

DNSSEC (Domain Name System Security Extensions) secures DNS data with a digital signature to protect DNS from attacks. This service is enabled by default in the IdM server. The warning appears because the remote DNS server does not use DNSSEC. Red Hat recommends that you enable DNSSEC on the remote DNS server.

If you cannot enable DNSSEC validation on the remote server, you can disable DNSSEC in the IdM server:

1. Choose the appropriate configuration file to edit:

- If your IdM server is using RHEL 8.0 or RHEL 8.1, open the **/etc/named.conf** file.
- If your IdM server is using RHEL 8.2 or later, open the **/etc/named/ipa-options-ext.conf** file.

2. Add the following DNSSEC parameters:

```
dnssec-enable no;
dnssec-validation no;
```

3. Save and close the configuration file.

4. Restart the DNS service:

```
# systemctl restart named-pkcs11
```

## Verification steps

- Use the **nslookup** command with the name of the remote DNS server:

```
$ nslookup ad.example.com
Server:      192.168.122.2
Address:     192.168.122.2#53
```

```
No-authoritative answer:
Name:       ad.example.com
Address:    192.168.122.3
```

If the domain forwarding is configured correctly, the **nslookup** request displays an IP address of the remote DNS server.

### 29.6.3. Configuring a DNS forward zone in the CLI

This section describes how to add a new DNS forward zone to the Identity Management (IdM) server using the command line interface (CLI).

With DNS forward zones, you can forward DNS queries for a specific zone to a different DNS server. For example, you can forward DNS queries for the Active Directory (AD) domain to an AD DNS server.

#### Prerequisites

- Access to the CLI with a user account that has administrator rights.
- Correctly configured DNS server.

#### Procedure

- Create a DNS forward zone for the AD domain, and specify the IP address of the remote DNS server with the **--forwarder** option:

```
# ipa dnsforwardzone-add ad.example.com --forwarder=192.168.122.3 --forward-policy=first
```

**NOTE**

You might see a warning about a DNSSEC validation failure in the `/var/log/messages` system logs after adding a new forward zone to the configuration:

```
named-pkcs11[2572]: no valid DS resolving 'host.ad.example.com/A/IN':
192.168.100.25#53
```

DNSSEC (Domain Name System Security Extensions) secures DNS data with a digital signature to protect DNS from attacks. This service is enabled by default in the IdM server. The warning appears because the remote DNS server does not use DNSSEC. Red Hat recommends that you enable DNSSEC on the remote DNS server.

If you cannot enable DNSSEC validation on the remote server, you can disable DNSSEC in the IdM server:

1. Choose the appropriate configuration file to edit:
  - If your IdM server is using RHEL 8.0 or RHEL 8.1, open the `/etc/named.conf` file.
  - If your IdM server is using RHEL 8.2 or later, open the `/etc/named/ipa-options-ext.conf` file.
2. Add the following DNSSEC parameters:

```
dnssec-enable no;
dnssec-validation no;
```

3. Save and close the configuration file.
4. Restart the DNS service:

```
# systemctl restart named-pkcs11
```

**Verification steps**

- Use the **nslookup** command with the name of the remote DNS server:

```
$ nslookup ad.example.com
Server:      192.168.122.2
Address:     192.168.122.2#53

No-authoritative answer:
Name:       ad.example.com
Address:    192.168.122.3
```

If the domain forwarding is configured correctly, the **nslookup** request displays an IP address of the remote DNS server.

**29.6.4. Configuring DNS forwarding in AD**

This section describes how to set up a DNS forwarding in Active Directory (AD) for the Identity Management (IdM) server.

### Prerequisites

- Windows Server with AD installed.
- DNS port open on both servers.

### Procedure

1. Log in to the Windows Server.
2. Open **Server Manager**.
3. Open **DNS Manager**.
4. In **Conditional Forwarders**, add a new conditional forwarder with:
  - The IdM server IP address
  - A fully qualified domain name, for example, ***server.idm.example.com***
5. Save the settings.

## 29.6.5. Verifying the DNS configuration

Before configuring trust, verify that the Identity Management (IdM) and Active Directory (AD) servers can resolve themselves and each other.

### Prerequisites

- You need to be logged in with sudo permissions.

### Procedure

1. Run a DNS query for the Kerberos over UDP and LDAP over TCP service records.

```
[admin@server ~]# dig +short -t SRV _kerberos._udp.idm.example.com.  
0 100 88 server.idm.example.com.
```

```
[admin@server ~]# dig +short -t SRV _ldap._tcp.idm.example.com.  
0 100 389 server.idm.example.com.
```

The commands are expected to list all IdM servers.

2. Run a DNS query for the TXT record with the IdM Kerberos realm name. The obtained value is expected to match the Kerberos realm you specified when installing IdM.

```
[admin@server ~]# dig +short -t TXT _kerberos.idm.example.com.  
"IDM.EXAMPLE.COM"
```

If the previous steps did not return all the expected records, update the DNS configuration with the missing records:

- If your IdM environment uses an integrated DNS server, enter the **ipa dns-update-system-records** command without any options to update your system records:

```
[admin@server ~]$ ipa dns-update-system-records
```

- If your IdM environment does not use an integrated DNS server:

1. On the IdM server, export the IdM DNS records into a file:

```
[admin@server ~]$ ipa dns-update-system-records --dry-run --out  
dns_records_file.nsupdate
```

The command creates a file named **dns\_records\_file.nsupdate** with the relevant IdM DNS records.

2. Submit a DNS update request to your DNS server using the **nsupdate** utility and the **dns\_records\_file.nsupdate** file. For more information, see [Updating External DNS Records Using nsupdate](#) in RHEL 7 documentation. Alternatively, refer to your DNS server documentation for adding DNS records.
3. Verify that IdM is able to resolve service records for AD with a command that runs a DNS query for Kerberos and LDAP over TCP service records:

```
[admin@server ~]# dig +short -t SRV _kerberos._tcp.dc._msdcs.ad.example.com.  
0 100 88 addc1.ad.example.com.
```

```
[admin@server ~]# dig +short -t SRV _ldap._tcp.dc._msdcs.ad.example.com.  
0 100 389 addc1.ad.example.com.
```

## 29.7. CONFIGURING IDM CLIENTS IN AN ACTIVE DIRECTORY DNS DOMAIN

If you have client systems in a DNS domain controlled by Active Directory and you require those clients to be able to join the IdM Server to benefit from its RHEL features, you can configure users to access a client using a host name from the Active Directory DNS domain.



### IMPORTANT

This is not a recommended configuration and has some limitations. Red Hat recommends to always deploy IdM clients in a DNS zone different from the ones owned by Active Directory and access IdM clients through their IdM host names.

Your IdM client configuration depends on whether you require single sign-on with Kerberos.

### 29.7.1. Configuring an IdM client without Kerberos single sign-on

Password authentication is the only authentication method that is available for users to access resources on IdM clients if the IdM clients are in an Active Directory DNS domain. This procedure describes how to configure your client without Kerberos single sign-on.

#### Procedure

1. Install the IdM client with the **--domain=IPA\_DNS\_Domain** option to ensure the System Security Services Daemon (SSSD) can communicate with the IdM servers:

```
[root@idm-client.ad.example.com ~]# ipa-client-install --domain=idm.example.com
```

-

This option disables the SRV record auto-detection for the Active Directory DNS domain.

2. Open the **/etc/krb5.conf** configuration file and locate the existing mapping for the Active Directory domain in the **[domain\_realm]** section.

```
.ad.example.com = IDM.EXAMPLE.COM
ad.example.com = IDM.EXAMPLE.COM
```

3. Replace both lines with an entry mapping the fully qualified domain name (FQDN) of the Linux clients in the Active Directory DNS zone to the IdM realm:

```
idm-client.ad.example.com = IDM.EXAMPLE.COM
```

By replacing the default mapping, you prevent Kerberos from sending its requests for the Active Directory domain to the IdM Kerberos Distribution Center (KDC). Instead Kerberos uses auto-discovery through SRV DNS records to locate the KDC.

### 29.7.2. Requesting SSL certificates without single sign-on

SSL-based services require a certificate with **dnsName** extension records that cover all system host names, because both original (A/AAAA) and CNAME records must be in the certificate. Currently, IdM only issues certificates to host objects in the IdM database.

In the described setup without single sign-on available, IdM already has a host object for the FQDN in the database, and **certmonger** can request a certificate using this name.

#### Prerequisites

- Installed and configured the IdM client by following the procedure in [Configuring an IdM client without Kerberos single sign-on](#).

#### Procedure

- Use **certmonger** to request a certificate using the FQDN:

```
[root@idm-client.ad.example.com ~]# ipa-getcert request -r \
-f /etc/httpd/alias/server.crt \
-k /etc/httpd/alias/server.key \
-N CN=ipa-client.ad.example.com \
-D ipa-client.ad.example.com \
-K host/idm-client.ad.example.com@IDM.EXAMPLE.COM \
-U id-kp-serverAuth
```

The **certmonger** service uses the default host key stored in the **/etc/krb5.keytab** file to authenticate to the IdM Certificate Authority (CA).

### 29.7.3. Configuring an IdM client with Kerberos single sign-on

If you require Kerberos single sign-on to access resources on the IdM client, the client must be within the IdM DNS domain, for example **idm-client.idm.example.com**. You must create a CNAME record **idm-client.ad.example.com** in the Active Directory DNS domain pointing to the A/AAAA record of the IdM client.

For Kerberos-based application servers, MIT Kerberos supports a method to allow the acceptance of any host-based principal available in the application's keytab.

### Procedure

- On the IdM client, disable the strict checks on what Kerberos principal is used to target the Kerberos server by setting the following option in the **[libdefaults]** section of the **/etc/krb5.conf** configuration file:

```
ignore_acceptor_hostname = true
```

### 29.7.4. Requesting SSL certificates with single sign-on

SSL-based services require a certificate with **dnsName** extension records that cover all system host names, because both original (A/AAAA) and CNAME records must be in the certificate. Currently, IdM only issues certificates to host objects in the IdM database.

This procedure describes how to create a host object for **ipa-client.example.com** in IdM and make sure the real IdM machine's host object is able to manage this host.

### Prerequisites

- You have disabled the strict checks on what Kerberos principal is used to target the Kerberos server as outlined in [Configuring an IdM client with Kerberos single sign-on](#).

### Procedure

1. Create a new host object on the IdM server:

```
[root@idm-server.idm.example.com ~]# ipa host-add idm-client.ad.example.com --force
```

Use the **--force** option, because the host name is a CNAME and not an A/AAAA record.

2. On the IdM server, allow the IdM DNS host name to manage the Active Directory host entry in the IdM database:

```
[root@idm-server.idm.example.com ~]# ipa host-add-managedby idm-client.ad.example.com \
--hosts=idm-client.idm.example.com
```

3. You can now request an SSL certificate for your IdM client with the **dnsName** extension record for its host name within the Active Directory DNS domain:

```
[root@idm-client.idm.example.com ~]# ipa-getcert request -r \
-f /etc/httpd/alias/server.crt \
-k /etc/httpd/alias/server.key \
-N CN=`hostname --fqdn` \
-D `hostname --fqdn` \
-D idm-client.ad.example.com \
-K host/idm-client.idm.example.com@IDM.EXAMPLE.COM \
-U id-kp-serverAuth
```

## 29.8. SETTING UP A TRUST

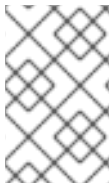
This section describes how to configure the Identity Management (IdM)/Active Directory (AD) trust on the IdM side using the command line.

### Prerequisites

- DNS is correctly configured. Both IdM and AD servers must be able to resolve each other names. For details, see [Configuring DNS and realm settings for a trust](#).
- Supported versions of AD and IdM are deployed. For details, see [Supported versions of Windows Server](#).
- You have obtained a Kerberos ticket. For details, see [Using kinit to log in to IdM manually](#).

### 29.8.1. Preparing the IdM server for the trust

Before you can establish a trust with AD, you must prepare the IdM domain using the **ipa-adtrust-install** utility on an IdM server.



#### NOTE

Any system where you run the **ipa-adtrust-install** command automatically becomes an AD trust controller. However, you must run **ipa-adtrust-install** only once on an IdM server.

### Prerequisites

- IdM server is installed.
- You need root privileges to install packages and restart IdM services.

### Procedure

1. Install the required packages:

```
[root@ipaserver ~]# yum install ipa-server-trust-ad samba-client
```

2. Authenticate as the IdM administrative user:

```
[root@ipaserver ~]# kinit admin
```

3. Run the **ipa-adtrust-install** utility:

```
[root@ipaserver ~]# ipa-adtrust-install
```

The DNS service records are created automatically if IdM was installed with an integrated DNS server.

If you installed IdM without an integrated DNS server, **ipa-adtrust-install** prints a list of service records that you must manually add to DNS before you can continue.

4. The script prompts you that the **/etc/samba/smb.conf** already exists and will be rewritten:

```
WARNING: The smb.conf already exists. Running ipa-adtrust-install will break your existing Samba configuration.
```



Do you wish to continue? [no]: **yes**

- The script prompts you to configure the **slapi-nis** plug-in, a compatibility plug-in that allows older Linux clients to work with trusted users:

Do you want to enable support for trusted domains in Schema Compatibility plugin?  
This will allow clients older than SSSD 1.9 and non-Linux clients to work with trusted users.

Enable trusted domains support in slapi-nis? [no]: **yes**

- When prompted, enter the NetBIOS name for the IdM domain or press **Enter** to accept the name suggested:

Trust is configured but no NetBIOS domain name found, setting it now.  
Enter the NetBIOS name for the IPA domain.  
Only up to 15 uppercase ASCII letters, digits and dashes are allowed.  
Example: EXAMPLE.

NetBIOS domain name [IDM]:

- You are prompted to run the SID generation task to create a SID for any existing users:

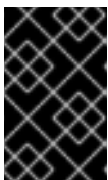
Do you want to run the ipa-sidgen task? [no]: **yes**

This is a resource-intensive task, so if you have a high number of users, you can run this at another time.

- (Optional) By default, the Dynamic RPC port range is defined as **49152-65535** for Windows Server 2008 and later. If you need to define a different Dynamic RPC port range for your environment, configure Samba to use different ports and open those ports in your firewall settings. The following example sets the port range to **55000-65000**.

```
[root@ipaserver ~]# net conf setparm global 'rpc server dynamic port range' 55000-65000
[root@ipaserver ~]# firewall-cmd --add-port=55000-65000/tcp
[root@ipaserver ~]# firewall-cmd --runtime-to-permanent
```

- Make sure that DNS is properly configured, as described in [Verifying the DNS configuration for a trust](#).



### IMPORTANT

Red Hat strongly recommends you verify the DNS configuration as described in [Verifying the DNS configuration for a trust](#) every time after running **ipa-adtrust-install**, especially if IdM or AD do not use integrated DNS servers.

- Restart the **ipa** service:

```
[root@ipaserver ~]# ipactl restart
```

- Use the **smbclient** utility to verify that Samba responds to Kerberos authentication from the IdM side:

```
[root@ipaserver ~]# smbclient -L server.idm.example.com -k
lp_load_ex: changing to config backend registry
Sharename      Type      Comment
-----
IPC$           IPC       IPC Service (Samba 4.12.3)
...
```

### 29.8.2. Setting up a trust agreement using the command line

This section describes how to set up the trust agreement using the command line. The Identity Management (IdM) server allows you to configure three types of trust agreements:

- **One-way trust** – default option. One-way trust enables Active Directory (AD) users and groups to access resources in IdM, but not the other way around. The IdM domain trusts the AD forest, but the AD forest does not trust the IdM domain.
- **Two-way trust** – Two-way trust enables AD users and groups to access resources in IdM. You must configure a two-way trust for solutions such as Microsoft SQL Server that expect the **S4U2Self** and **S4U2Proxy** Microsoft extensions to the Kerberos protocol to work over a trust boundary. An application on a RHEL IdM host might request **S4U2Self** or **S4U2Proxy** information from an Active Directory domain controller about an AD user, and a two-way trust provides this feature.

Note that this two-way trust functionality does not allow IdM users to login to Windows systems, and the two-way trust in IdM does not give the users any additional rights compared to the one-way trust solution in AD.

- To create the two-way trust, add the following option to the command: **--two-way=true**
- **External trust** – a trust relationship between IdM and an AD domain in different forests. While a forest trust always requires establishing a trust between IdM and the root domain of an Active Directory forest, an external trust can be established from IdM to a domain within a forest. This is only recommended if it is not possible to establish a forest trust between forest root domains due to administrative or organizational reasons.
  - To create the external trust, add the following option to the command: **--external=true**

In this section, the steps below shows you how to create a one-way trust agreement.

#### Prerequisites

- User name and password of a Windows administrator.
- You have [prepared the IdM server for the trust](#).

#### Procedure

- Create a trust agreement for the AD domain and the IdM domain by using the **ipa trust-add** command:
  - To have SSSD automatically generate UIDs and GIDs for AD users based on their SID, create a trust agreement with the the **Active Directory domain** ID range type. This is the most common configuration.

```
[root@server ~]# ipa trust-add --type=ad ad.example.com --admin
<ad_admin_username> --password --range-type=ipa-ad-trust
```

- If you have configured POSIX attributes for your users in Active Directory (such as **uidNumber** and **gidNumber**) and you want SSSD to process this information, create a trust agreement with the **Active Directory domain with POSIX attributes** ID range type:

```
[root@server ~]# ipa trust-add --type=ad ad.example.com --admin
<ad_admin_username> --password --range-type=ipa-ad-trust-posix
```



### WARNING

If you do not specify an ID Range type when creating a trust, IdM attempts to automatically select the appropriate range type by requesting details from AD domain controllers in the forest root domain. If IdM does not detect any POSIX attributes, the trust installation script selects the **Active Directory domain** ID range.

If IdM detects any POSIX attributes in the forest root domain, the trust installation script selects the **Active Directory domain with POSIX attributes** ID range and assumes that UIDs and GIDs are correctly defined in AD. If POSIX attributes are not correctly set in AD, you will not be able to resolve AD users.

For example, if the users and groups that need access to IdM systems are not part of the forest root domain, but instead are located in a child domain of the forest domain, the installation script may not detect the POSIX attributes defined in the child AD domain. In this case, Red Hat recommends that you explicitly choose the POSIX ID range type when establishing the trust.

### 29.8.3. Setting up a trust agreement in the IdM Web UI

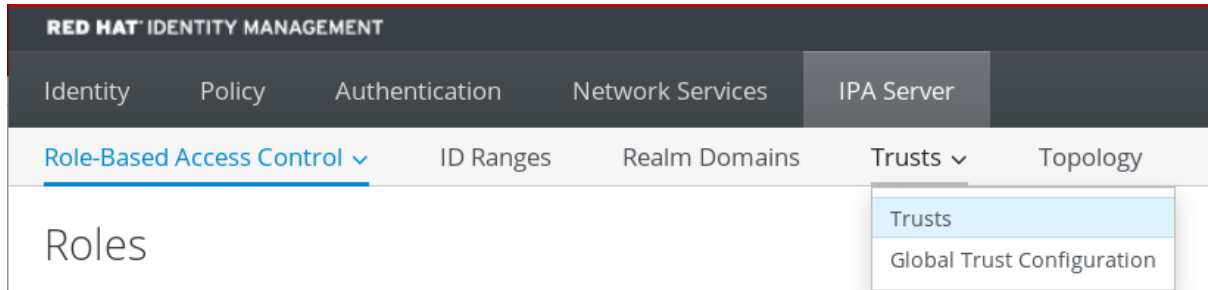
This section describes how to configure the Identity Management (IdM)/Active Directory (AD) trust agreement on the IdM side using the IdM Web UI.

#### Prerequisites

- DNS is correctly configured. Both IdM and AD servers must be able to resolve each other names.
- Supported versions of AD and IdM are deployed.
- You have obtained a Kerberos ticket.
- Before creating a trust in the Web UI, prepare the IdM server for the trust as described in: [Preparing the IdM server for the trust](#).
- You need to be logged in as an IdM administrator.

#### Procedure

1. Log in to the IdM Web UI with administrator privileges. For details, see [Accessing the IdM Web UI in a web browser](#).
2. In the IdM Web UI, click the **IPA Server** tab.
3. In the **IPA Server** tab, click the **Trusts** tab.
4. In the drop down menu, select the **Trusts** option.



5. Click the **Add** button.
6. In the **Add Trust** dialog box, enter the name of the Active Directory domain.
7. In the **Account** and **Password** fields, add the administrator credentials of the Active Directory administrator.

8. (Optional) Select **Two-way trust**, if you want to enable AD users and groups to access resources in IdM. However, the two-way trust in IdM does not give the users any additional rights compared to the one-way trust solution in AD. Both solutions are considered equally secure because of default cross-forest trust SID filtering settings.
9. (Optional) Select **External trust** if you are configuring a trust with an AD domain that is not the root domain of an AD forest. While a forest trust always requires establishing a trust between IdM and the root domain of an Active Directory forest, you can establish an external trust from IdM to any domain within an AD forest.
10. (Optional) By default, the trust installation script tries to detect the appropriate ID range type. You can also explicitly set the ID range type by choosing one of the following options:

- To have SSSD automatically generate UUIDs and GIDs for AD users based on their SID

- a. To have SSSD automatically generate UIDs and GIDs for AD users based on their SID, select the **Active Directory domain** ID range type. This is the most common configuration.
- b. If you have configured POSIX attributes for your users in Active Directory (such as **uidNumber** and **gidNumber**) and you want SSSD to process this information, select the **Active Directory domain with POSIX attributes** ID range type.

Range type	<input checked="" type="radio"/> Detect <input type="radio"/> Active Directory domain <input type="radio"/> Active Directory domain with POSIX attributes
------------	---



### WARNING

If you leave the **Range type** setting on the default **Detect** option, IdM attempts to automatically select the appropriate range type by requesting details from AD domain controllers in the forest root domain. If IdM does not detect any POSIX attributes, the trust installation script selects the **Active Directory domain** ID range.

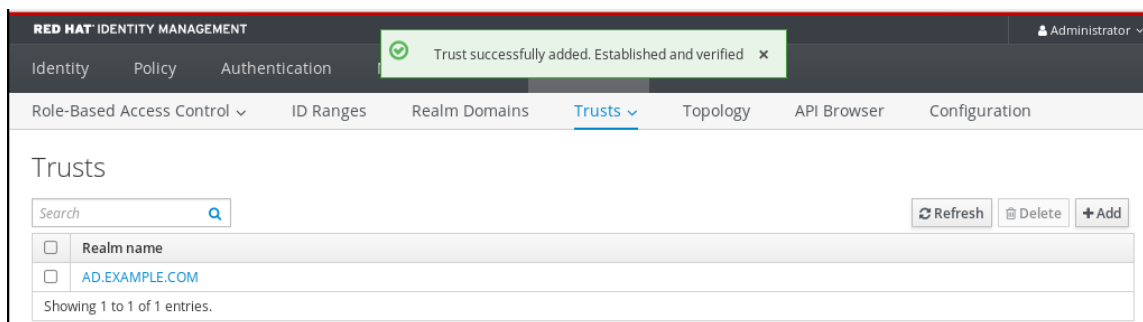
If IdM detects any POSIX attributes in the forest root domain, the trust installation script selects the **Active Directory domain with POSIX attributes** ID range and assumes that UIDs and GIDs are correctly defined in AD. If POSIX attributes are not correctly set in AD, you will not be able to resolve AD users.

For example, if the users and groups that need access to IdM systems are not part of the forest root domain, but instead are located in a child domain of the forest domain, the installation script may not detect the POSIX attributes defined in the child AD domain. In this case, Red Hat recommends that you explicitly choose the POSIX ID range type when establishing the trust.

11. Click **Add**.

### Verification steps

- If the trust has been successfully added to the IdM server, you can see the green pop-up window in the IdM Web UI. It means that the:
  - Domain name exists
  - User name and password of the Windows Server has been added correctly.



Now you can continue to test the trust connection and Kerberos authentication.

### 29.8.4. Verifying the Kerberos configuration

To verify the Kerberos configuration, test if it is possible to obtain a ticket for an Identity Management (IdM) user and if the IdM user can request service tickets.

#### Procedure

1. Request a ticket for an Active Directory (AD) user:

```
[root@ipaserver ~]# kinit user@AD.EXAMPLE.COM
```

2. Request service tickets for a service within the IdM domain:

```
[root@server ~]# kvno -S host server.idm.example.com
```

If the AD service ticket is successfully granted, there is a cross-realm ticket-granting ticket (TGT) listed with all of the other requested tickets. The TGT is named `krbtgt/IPA.DOMAIN@AD.DOMAIN`.

```
[root@server ~]# klist
Ticket cache: KEYRING:persistent:0:krb_ccache_hRtox00
Default principal: user@AD.EXAMPLE.COM
```

```
Valid starting    Expires          Service principal
03.05.2016 18:31:06 04.05.2016 04:31:01 host/server.idm.example.com@IDM.EXAMPLE.COM
renew until 04.05.2016 18:31:00
03.05.2016 18:31:06 04.05.2016 04:31:01 krbtgt/IDM.EXAMPLE.COM@AD.EXAMPLE.COM
renew until 04.05.2016 18:31:00
03.05.2016 18:31:01 04.05.2016 04:31:01 krbtgt/AD.EXAMPLE.COM@AD.EXAMPLE.COM
renew until 04.05.2016 18:31:00
```

The **localauth** plug-in maps Kerberos principals to local System Security Services Daemon (SSSD) user names. This allows AD users to use Kerberos authentication and access Linux services, which support GSSAPI authentication directly.

### 29.8.5. Verifying the trust configuration on IdM

Before configuring trust, verify that the Identity Management (IdM) and Active Directory (AD) servers can resolve themselves and each other.

#### Prerequisites

- You need to be logged in with administrator privileges.

### Procedure

1. Run a DNS query for the MS DC Kerberos over UDP and LDAP over TCP service records.

```
[root@server ~]# dig +short -t SRV _kerberos._udp.dc._msdcs.idm.example.com.
0 100 88 server.idm.example.com.
```

```
[root@server ~]# dig +short -t SRV _ldap._tcp.dc._msdcs.idm.example.com.
0 100 389 server.idm.example.com.
```

These commands list all IdM servers on which **ipa-adtrust-install** has been executed. The output is empty if **ipa-adtrust-install** has not been executed on any IdM server, which is typically before establishing the first trust relationship.

2. Run a DNS query for the Kerberos and LDAP over TCP service records to verify that IdM is able to resolve service records for AD:

```
[root@server ~]# dig +short -t SRV _kerberos._tcp.dc._msdcs.ad.example.com.
0 100 88 addc1.ad.example.com.
```

```
[root@ipaserver ~]# dig +short -t SRV _ldap._tcp.dc._msdcs.ad.example.com.
0 100 389 addc1.ad.example.com.
```

### 29.8.6. Verifying the trust configuration on AD

After configuring the trust, verify that:

- The Identity Management (IdM)-hosted services are resolvable from the Active Directory (AD) server.
- AD services are resolvable from the AD server.

### Prerequisites

- You need to be logged in with administrator privileges.

### Procedure

1. On the AD server, set the **nslookup.exe** utility to look up service records.

```
C:\>nslookup.exe
> set type=SRV
```

2. Enter the domain name for the Kerberos over UDP and LDAP over TCP service records.

```
> _kerberos._udp.idm.example.com.
_kerberos._udp.idm.example.com.    SRV service location:
priority          = 0
weight            = 100
port              = 88
```

```

    svr hostname = server.idm.example.com
> _ldap._tcp.idm.example.com
_ldap._tcp.idm.example.com    SRV service location:
    priority      = 0
    weight        = 100
    port          = 389
    svr hostname  = server.idm.example.com

```

3. Change the service type to TXT and run a DNS query for the TXT record with the IdM Kerberos realm name.

```

C:\>nslookup.exe
> set type=TXT
> _kerberos.idm.example.com.
_kerberos.idm.example.com.    text =

    "IDM.EXAMPLE.COM"

```

4. Run a DNS query for the MS DC Kerberos over UDP and LDAP over TCP service records.

```

C:\>nslookup.exe
> set type=SRV
> _kerberos._udp.dc._msdcs.idm.example.com.
_kerberos._udp.dc._msdcs.idm.example.com.    SRV service location:
    priority = 0
    weight = 100
    port = 88
    svr hostname = server.idm.example.com
> _ldap._tcp.dc._msdcs.idm.example.com.
_ldap._tcp.dc._msdcs.idm.example.com.    SRV service location:
    priority = 0
    weight = 100
    port = 389
    svr hostname = server.idm.example.com

```

Active Directory only expects to discover domain controllers that can respond to AD-specific protocol requests, such as other AD domain controllers and IdM trust controllers. Use the **ipa-adtrust-install** tool to promote an IdM server to a trust controller, and you can verify which servers are trust controllers with the **ipa server-role-find --role 'AD trust controller'** command.

5. Verify that AD services are resolvable from the AD server.

```

C:\>nslookup.exe
> set type=SRV

```

6. Enter the domain name for the Kerberos over UDP and LDAP over TCP service records.

```

> _kerberos._udp.dc._msdcs.ad.example.com.
_kerberos._udp.dc._msdcs.ad.example.com.    SRV service location:
    priority = 0
    weight = 100
    port = 88
    svr hostname = addc1.ad.example.com
> _ldap._tcp.dc._msdcs.ad.example.com.
_ldap._tcp.dc._msdcs.ad.example.com.    SRV service location:

```



```
priority = 0
weight = 100
port = 389
svr hostname = addc1.ad.example.com
```

### 29.8.7. Creating a trust agent

A trust agent is an IdM server that can perform identity lookups against AD domain controllers.

For example, if you are creating a replica of an IdM server that has a trust with Active Directory, you can set up the replica as a trust agent. A replica does not automatically have the AD trust agent role installed.

#### Prerequisites

- IdM is installed with an Active Directory trust.
- The **sssd-tools** package is installed.

#### Procedure

1. On an existing trust controller, run the **ipa-adtrust-install --add-agents** command:

```
[root@existing_trust_controller]# ipa-adtrust-install --add-agents
```

The command starts an interactive configuration session and prompts you for the information required to set up the agent.

2. Restart the IdM service on the trust agent.

```
[root@new_trust_agent]# ipactl restart
```

3. Remove all entries from the SSSD cache on the trust agent:

```
[root@new_trust_agent]# sssctl cache-remove
```

4. Verify that the replica has the AD trust agent role installed:

```
[root@existing_trust_controller]# ipa server-show new_replica.idm.example.com
...
Enabled server roles: CA server, NTP server, AD trust agent
```

#### Additional resources

- For further information about the **--add-agents** option, see the **ipa-adtrust-install(1)** man page.
- For more information on trust agents, see [Trust controllers and trust agents](#) in the Planning Identity Management guide.

### 29.8.8. Enabling automatic private group mapping for a POSIX ID range on the CLI

By default, SSSD does not map private groups for Active Directory (AD) users if you have established a POSIX trust that relies on POSIX data stored in AD. If any AD users do not have primary groups configured, IdM is not be able to resolve them.

This procedure explains how to enable automatic private group mapping for an ID range by setting the **hybrid** option for the **auto\_private\_groups** SSSD parameter on the command line. As a result, IdM is able to resolve AD users that do not have primary groups configured in AD.

## Prerequisites

- You have successfully established a POSIX cross-forest trust between your IdM and AD environments.

## Procedure

1. Display all ID ranges and make note of the AD ID range you want to modify.

```
[root@server ~]# ipa idrange-find
-----
2 ranges matched
-----
Range name: IDM.EXAMPLE.COM_id_range
First Posix ID of the range: 882200000
Number of IDs in the range: 200000
Range type: local domain range

Range name: AD.EXAMPLE.COM_id_range
First Posix ID of the range: 1337000000
Number of IDs in the range: 200000
Domain SID of the trusted domain: S-1-5-21-4123312420-990666102-3578675309
Range type: Active Directory trust range with POSIX attributes
-----
Number of entries returned 2
-----
```

2. Adjust the automatic private group behavior for the AD ID range with the **ipa idrange-mod** command.

```
[root@server ~]# ipa idrange-mod --auto-private-groups=hybrid
AD.EXAMPLE.COM_id_range
```

3. Reset the SSSD cache to enable the new setting.

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

## Additional resources

- [Options for automatically mapping private groups for AD users](#)

## 29.8.9. Enabling automatic private group mapping for a POSIX ID range in the IdM WebUI

By default, SSSD does not map private groups for Active Directory (AD) users if you have established a POSIX trust that relies on POSIX data stored in AD. If any AD users do not have primary groups configured, IdM is not be able to resolve them.

This procedure explains how to enable automatic private group mapping for an ID range by setting the **hybrid** option for the **auto\_private\_groups** SSSD parameter in the Identity Management (IdM) WebUI. As a result, IdM is able to resolve AD users that do not have primary groups configured in AD.

## Prerequisites

- You have successfully established a POSIX cross-forest trust between your IdM and AD environments.

## Procedure

1. Log into the IdM Web UI with your user name and password.
2. Open the **IPA Server** → **ID Ranges** tab.
3. Select the ID range you want to modify, such as **AD.EXAMPLE.COM\_id\_range**.
4. From the **Auto private groups** drop down menu, select the **hybrid** option.

The screenshot shows the IdM WebUI interface. At the top, there are tabs for Identity, Policy, Authentication, Network Services, and IPA Server. The IPA Server tab is selected, and the ID Ranges sub-tab is active. The breadcrumb trail shows 'ID Ranges > AD.EXAMPLE.COM\_id\_range'. The main heading is 'ID Range: AD.EXAMPLE.COM\_id\_range'. Below this, there are buttons for Settings, Refresh, Revert, and Save. The 'Range Settings' section displays the following information:

- Range name: AD.EXAMPLE.COM\_id\_range
- Range type: Active Directory trust range with POSIX attributes
- Base ID \*: 1045000000
- Range size \*: 200000
- Domain SID: S-1-5-21-4029230055-4155305145-370140224
- Auto private groups: A dropdown menu is open, showing three options: 'true', 'false', and 'hybrid'. The 'hybrid' option is selected and highlighted in blue.

5. Click the **Save** button to save your changes.

### Additional resources

- [Options for automatically mapping private groups for AD users](#)

## 29.9. REMOVING THE TRUST USING THE COMMAND LINE

This section describes how to remove the Identity Management (IdM)/Active Directory (AD) trust on the IdM side using the command line interface.

### Prerequisites

- You have obtained a Kerberos ticket as an IdM administrator. For details, see [Logging in to IdM in the Web UI: Using a Kerberos ticket](#).

### Procedure

1. Use the **ipa trust-del** command to remove the trust configuration from IdM.

```
[root@server ~]# ipa trust-del ad_domain_name
```

```
-----  
Deleted trust "ad_domain_name"  
-----
```

2. Remove the trust object from your Active Directory configuration.

### Verification steps

- Use the **ipa trust-show** command to confirm that the trust has been removed.

```
[root@server ~]# ipa trust-show ad.example.com  
ipa: ERROR: ad.example.com: trust not found
```

## 29.10. REMOVING THE TRUST USING THE IDM WEB UI

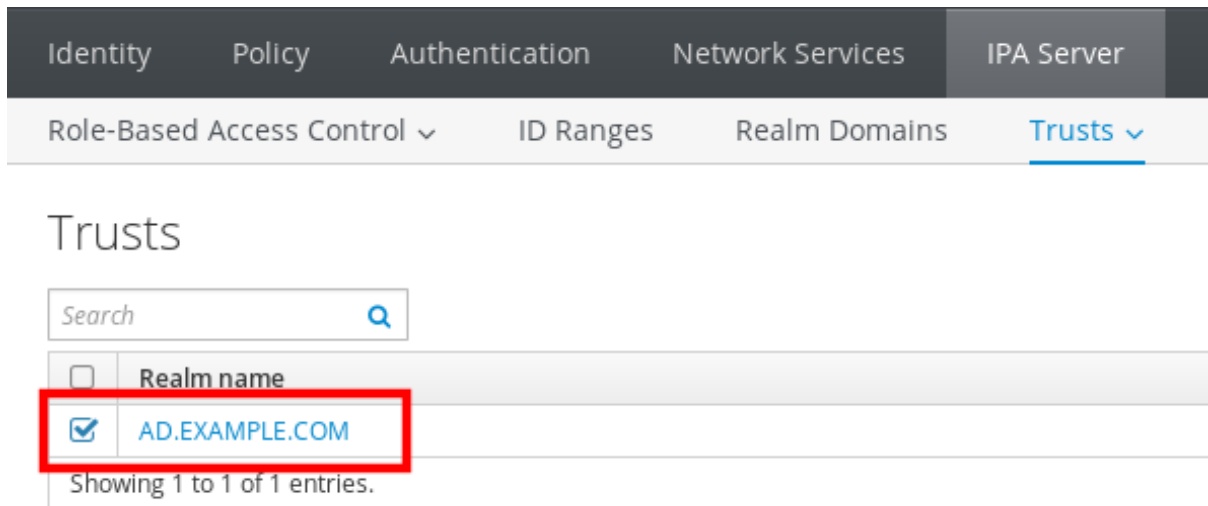
This section describes how to remove the Identity Management (IdM)/Active Directory (AD) trust using the IdM Web UI.

### Prerequisites

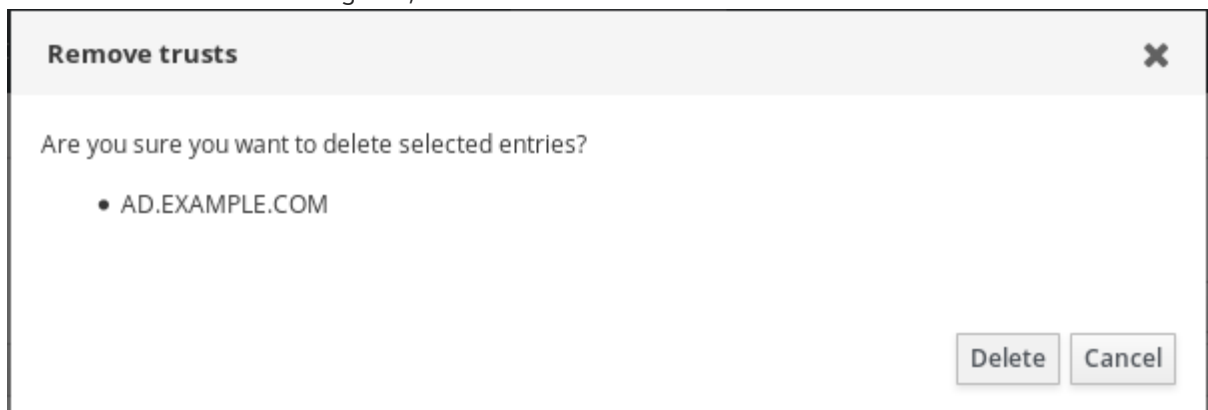
- You have obtained a Kerberos ticket. For details, see [Logging in to IdM in the Web UI: Using a Kerberos ticket](#).

### Procedure

1. Log in to the IdM Web UI with administrator privileges. For details, see [Accessing the IdM Web UI in a web browser](#).
2. In the IdM Web UI, click the **IPA Server** tab.
3. In the **IPA Server** tab, click the **Trusts** tab.
4. Select the trust you want to remove.



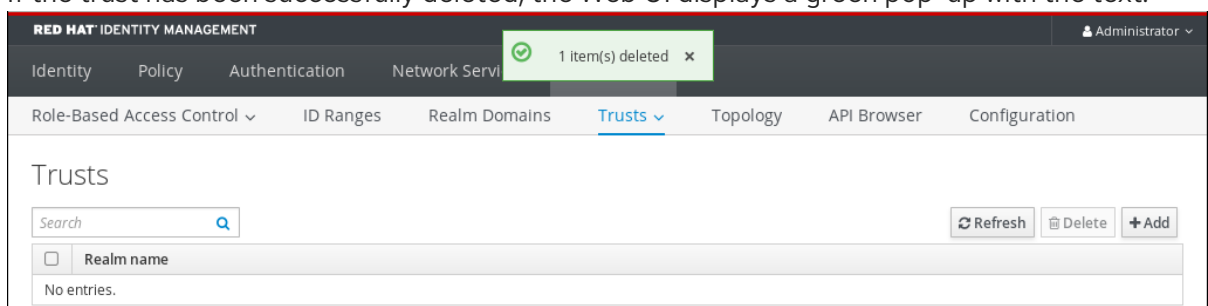
5. Click the **Delete** button.
6. In the **Remove trusts** dialog box, click **Delete**.



7. Remove the trust object from your Active Directory configuration.

## Verification steps

- If the trust has been successfully deleted, the Web UI displays a green pop-up with the text:



## **PART III. MIGRATING IDM FROM RHEL 7 TO RHEL 8 AND KEEPING IT UP-TO-DATE**

## CHAPTER 30. MIGRATING YOUR IDM ENVIRONMENT FROM RHEL 7 SERVERS TO RHEL 8 SERVERS

To upgrade a RHEL 7 IdM environment to RHEL 8, you must first add new RHEL 8 IdM replicas to your RHEL 7 IdM environment, and then retire the RHEL 7 servers.



### WARNING

Performing an in-place upgrade of RHEL 7 IdM servers to RHEL 8 is not supported.

This section describes how to **migrate** all Identity Management (IdM) data and configuration from a Red Hat Enterprise Linux (RHEL) 7 server to a RHEL 8 server.

The migration procedure includes:

1. Configuring a RHEL 8 IdM server and adding it as a replica to your current RHEL 7 IdM environment. For details, see [Installing the RHEL 8 Replica](#).
2. Making the RHEL 8 server the certificate authority (CA) renewal server. For details, see [Assigning the CA renewal server role to the RHEL 8 IdM server](#).
3. Stopping the generation of the certificate revocation list (CRL) on the RHEL 7 server and redirecting CRL requests to RHEL 8. For details, see [Stopping CRL generation on a RHEL 7 IdM CA server](#).
4. Starting the generation of the CRL on the RHEL 8 server. For details, see [Starting CRL generation on the new RHEL 8 IdM CA server](#).
5. Stopping and decommissioning the original RHEL 7 CA renewal server. For details, see [Stopping and decommissioning the RHEL 7 server](#).

In the following procedures:

- **rhel8.example.com** is the RHEL 8 system that will become the new CA renewal server.
- **rhel7.example.com** is the original RHEL 7 CA renewal server. To identify which Red Hat Enterprise Linux 7 server is the CA renewal server, run the following command on any IdM server:

```
[root@rhel7 ~]# ipa config-show | grep "CA renewal"  
IPA CA renewal master: rhel7.example.com
```

If your IdM deployment is CA-less, any IdM server running on RHEL 7 can be **rhel7.example.com**.

**NOTE**

Complete the steps in the following sections **only** if your IdM deployment uses an embedded certificate authority (CA):

- [Assigning the CA renewal server role to the RHEL 8 IdM server](#)
- [Stopping CRL generation on a RHEL 7 IdM CA server](#)
- [Starting CRL generation on the new RHEL 8 IdM CA server](#)

## 30.1. PREREQUISITES FOR MIGRATING IDM FROM RHEL 7 TO 8

On **rhel7.example.com**:

1. Upgrade the system to the latest RHEL 7 version.
2. Ensure that the domain level for your domain is set to 1. For more information, see [Displaying and Raising the Domain Level](#) in the *Linux Domain Identity, Authentication, and Policy Guide* for RHEL 7.
3. Update the **ipa-\*** packages to their latest version:

```
[root@rhel7 ~]# yum update ipa-*
```

**WARNING**

When upgrading multiple Identity Management (IdM) servers, wait at least 10 minutes between each upgrade.

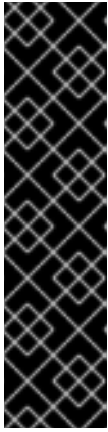
When two or more servers are upgraded simultaneously or with only short intervals between the upgrades, there is not enough time to replicate the post-upgrade data changes throughout the topology, which can result in conflicting replication events.

On **rhel8.example.com**:

1. The latest version of Red Hat Enterprise Linux is installed on the system. For more information, see [Performing a standard RHEL installation](#).
2. Ensure the system is an IdM client enrolled into the domain for which **rhel7.example.com** IdM server is authoritative. For more information, see [Installing an IdM client: Basic scenario](#).
3. Ensure the system meets the requirements for [IdM server installation](#).
4. Ensure you know the time server **rhel7.example.com** is synchronized with:

```
[root@rhel7 ~]# ntpstat
synchronised to NTP server (ntp.example.com) at stratum 3
time correct to within 42 ms
polling server every 1024 s
```





## IMPORTANT

In RHEL 8, IdM does not provide its own time server: the installation of IdM on **rhel8.example.com** does not result in the installation of an NTP server on the host. Therefore, you need to use a separate NTP server, for example **ntp.example.com**. For more information, see [Migrating to chrony](#) and [Time service requirements for IdM](#).

While **rhel7.example.com** can be used in an NTP server role, you will decommission the server as part of the migration process. Therefore, **rhel8.example.com** needs to be synchronized directly with **ntp.example.com** instead. You can specify this during the installation process.

5. Ensure the system is [authorized for the installation of an IdM replica](#) .
6. Update the **ipa-\*** packages to their latest version:

```
[root@rhel7 ~]# yum update ipa-*
```

### Additional resources

- To decide which server roles you want to install on the new IdM primary server, **rhel8.example.com**, see the following links:
  - For details on the CA server role in IdM, see [Planning your CA services](#) .
  - For details on the DNS server role in IdM, see [Planning your DNS services and host names](#) .
  - For details on integration based on cross-forest trust between an IdM and Active Directory (AD), see [Indirect integration](#) .
- To be able to install specific server roles for IdM in RHEL 8, you need to download packages from specific IdM module streams: [Installing packages required for an IdM server](#) .
- To upgrade a system from RHEL 7 to RHEL 8, see [Upgrading from RHEL 7 to RHEL 8](#) .

## 30.2. INSTALLING THE RHEL 8 REPLICA

1. List which server roles are present in your RHEL 7 environment:

```
[root@rhel7 ~]# ipa server-role-find --status enabled --server rhel7.example.com
-----
3 server roles matched
-----
Server name: rhel7.example.com
Role name: CA server
Role status: enabled

Server name: rhel7.example.com
Role name: DNS server
Role status: enabled

Server name: rhel7.example.com
```

```

Role name: NTP server
Role status: enabled
[... output truncated ...]

```

2. [Optional] If you want to use the same per-server forwarders for **rhel8.example.com** that **rhel7.example.com** is using, view the per-server forwarders for **rhel7.example.com**:

```

[root@rhel7 ~]# ipa dnsserver-show rhel7.example.com
-----
1 DNS server matched
-----
Server name: rhel7.example.com
SOA mname: rhel7.example.com.
Forwarders: 192.0.2.20
Forward policy: only
-----
Number of entries returned 1
-----

```

3. Install the IdM server on **rhel8.example.com** as a replica of the IdM RHEL 7 server, including all the server roles present on your **rhel7.example.com** except the NTP server role. To install the roles from the example above, use these options with the **ipa-replica-install** command:

- **--setup-ca** to set up the Certificate System component
- **--setup-dns** and **--forwarder** to configure an integrated DNS server and set a per-server forwarder to take care of DNS queries that go outside the IdM domain



#### NOTE

Additionally, if your IdM deployment is in a trust relationship with Active Directory (AD), add the **--setup-adtrust** option to the **ipa-replica-install** command to configure AD trust capability on **rhel8.example.com**.

- **--ntp-server** to specify an NTP server or **--ntp-pool** to specify a pool of NTP servers



#### IMPORTANT

If you do not specify an NTP server manually, it will be automatically set from DNS records. This can lead to **rhel8.example.com** synchronizing with **rhel7.example.com**. This will cause issues when the RHEL 7 server is decommissioned.

To set up an IdM server with the IP address of 192.0.2.1 that uses a per-server forwarder with the IP address of 192.0.2.20 and synchronizes with the **ntp.example.com** NTP server:

```

[root@rhel8 ~]# ipa-replica-install --setup-ca --ip-address 192.0.2.1 --setup-dns --
forwarder 192.0.2.20 --ntp-server ntp.example.com

```

You do not need to specify the RHEL 7 IdM server itself because if DNS is working correctly, **rhel8.example.com** will find it using DNS autodiscovery.

4. [Optional] Add an **\_ntp.\_udp** service (SRV) record for your external **NTP** time server to the DNS of the newly-installed IdM server, **rhel8.example.com**. Doing this is recommended

because IdM in RHEL 8 does not provide its own time service. The presence of the SRV record for the time server in IdM DNS ensures that future RHEL 8 replica and client installations are automatically configured to synchronize with the time server used by **rhel8.example.com**. This is because **ipa-client-install** looks for the **\_ntp.\_udp** DNS entry unless **--ntp-server** or **--ntp-pool** options are provided on the install command-line interface (CLI).

## Verification

1. Verify that the IdM services are running on **rhel8.example.com**:

```
[root@rhel8 ~]# ipactl status
Directory Service: RUNNING
[... output truncated ...]
ipa: INFO: The ipactl command was successful
```

2. Verify that server roles for **rhel8.example.com** are the same as for **rhel7.example.com** except the NTP server role:

```
[root@rhel8 ~]$ kinit admin
[root@rhel8 ~]$ ipa server-role-find --status enabled --server rhel8.example.com
-----
2 server roles matched
-----
Server name: rhel8.example.com
Role name: CA server
Role status: enabled

Server name: rhel8.example.com
Role name: DNS server
Role status: enabled
```

3. [Optional] Display details about the replication agreement between **rhel7.example.com** and **rhel8.example.com**:

```
[root@rhel8 ~]# ipa-csreplica-manage list --verbose rhel8.example.com
Directory Manager password:

rhel7.example.com
last init status: None
last init ended: 1970-01-01 00:00:00+00:00
last update status: Error (0) Replica acquired successfully: Incremental update succeeded
last update ended: 2019-02-13 13:55:13+00:00
```

4. [Optional] If your IdM deployment is in a trust relationship with AD, verify that it is working:
  - a. [Verify the Kerberos configuration](#)
  - b. Attempt to resolve an AD user on **rhel8.example.com**:

```
[root@rhel8 ~]# id aduser@ad.domain
```

5. Verify that **rhel8.example.com** is synchronized with the **NTP** server:

```
[root@rhel8 ~]# chronyc tracking
```

```
Reference ID   : CB00710F (ntp.example.com)
Stratum       : 3
Ref time (UTC) : Tue Nov 16 09:49:17 2021
[... output truncated ...]
```

#### Additional resources

- [DNS configuration priorities](#)
- [Time service requirements for IdM](#)
- [Migrating to chrony](#)

## 30.3. ASSIGNING THE CA RENEWAL SERVER ROLE TO THE RHEL 8 IDM SERVER



### NOTE

Complete the steps in this section only if your IdM deployment uses an embedded certificate authority (CA).

On **rhel8.example.com**, configure **rhel8.example.com** as the new CA renewal server:

1. Configure **rhel8.example.com** to handle CA subsystem certificate renewal:

```
[root@rhel8 ~]# ipa config-mod --ca-renewal-master-server rhel8.example.com
...
IPA masters: rhel7.example.com, rhel8.example.com
IPA CA servers: rhel7.example.com, rhel8.example.com
IPA NTP servers: rhel7.example.com, rhel8.example.com
IPA CA renewal master: rhel8.example.com
```

The output confirms that the update was successful.

2. On **rhel8.example.com**, enable the certificate updater task:
  - a. Open the **/etc/pki/pki-tomcat/ca/CS.cfg** configuration file for editing.
  - b. Remove the **ca.certStatusUpdateInterval** entry, or set it to the desired interval in seconds. The default value is **600**.
  - c. Save and close the **/etc/pki/pki-tomcat/ca/CS.cfg** configuration file.
  - d. Restart IdM services:

```
[user@rhel8 ~]$ ipactl restart
```

3. On **rhel7.example.com**, disable the certificate updater task:
  - a. Open the **/etc/pki/pki-tomcat/ca/CS.cfg** configuration file for editing.
  - b. Change **ca.certStatusUpdateInterval** to **0**, or add the following entry if it does not exist:

```
ca.certStatusUpdateInterval=0
```

- c. Save and close the `/etc/pki/pki-tomcat/ca/CS.cfg` configuration file.
- d. Restart IdM services:

```
[user@rhel7 ~]$ ipactl restart
```

## 30.4. STOPPING CRL GENERATION ON A RHEL 7 IDM CA SERVER



### NOTE

Complete the steps in this section only if your IdM deployment uses an embedded certificate authority (CA).

This section describes how to stop generating the Certificate Revocation List (CRL) on the `rhel7.example.com` CA server using the `ipa-crlgen-manage` command.

### Prerequisites

- You must be logged in as root.

### Procedure

1. Optionally, check if `rhel7.example.com` is generating the CRL:

```
[root@rhel7 ~]# ipa-crlgen-manage status
CRL generation: enabled
Last CRL update: 2019-10-31 12:00:00
Last CRL Number: 6
The ipa-crlgen-manage command was successful
```

2. Stop generating the CRL on the `rhel7.example.com` server:

```
[root@rhel7 ~]# ipa-crlgen-manage disable
Stopping pki-tomcatd
Editing /var/lib/pki/pki-tomcat/conf/ca/CS.cfg
Starting pki-tomcatd
Editing /etc/httpd/conf.d/ipa-pki-proxy.conf
Restarting httpd
CRL generation disabled on the local host. Please make sure to configure CRL generation on
another master with ipa-crlgen-manage enable.
The ipa-crlgen-manage command was successful
```

3. Optionally, check if the `rhel7.example.com` server stopped generating the CRL:

```
[root@rhel7 ~]# ipa-crlgen-manage status
```

The `rhel7.example.com` server stopped generating the CRL. The next step is to enable generating the CRL on `rhel8.example.com`.

## 30.5. STARTING CRL GENERATION ON THE NEW RHEL 8 IDM CA SERVER

**NOTE**

Complete the steps in this section only if your IdM deployment uses an embedded certificate authority (CA).

**Prerequisites**

- You must be logged in as root on the **rhel8.example.com** machine.

**Procedure**

1. To start generating CRL on **rhel8.example.com**, use the **ipa-crlgen-manage enable** command:

```
[root@rhel8 ~]# ipa-crlgen-manage enable
Stopping pki-tomcatd
Editing /var/lib/pki/pki-tomcat/conf/ca/CS.cfg
Starting pki-tomcatd
Editing /etc/httpd/conf.d/ipa-pki-proxy.conf
Restarting httpd
Forcing CRL update
CRL generation enabled on the local host. Please make sure to have only a single CRL
generation master.
The ipa-crlgen-manage command was successful
```

2. To check if CRL generation is enabled, use the **ipa-crlgen-manage status** command:

```
[root@rhel8 ~]# ipa-crlgen-manage status
CRL generation: enabled
Last CRL update: 2019-10-31 12:10:00
Last CRL Number: 7
The ipa-crlgen-manage command was successful
```

**30.6. STOPPING AND DECOMMISSIONING THE RHEL 7 SERVER**

1. Make sure that all data, including the latest changes, have been correctly migrated from **rhel7.example.com** to **rhel8.example.com**. For example:

- a. Add a new user on **rhel7.example.com**:

```
[root@rhel7 ~]# ipa user-add random_user
First name: random
Last name: user
```

- b. Check that the user has been replicated to **rhel8.example.com**:

```
[root@rhel8 ~]# ipa user-find random_user
-----
1 user matched
-----
User login: random_user
First name: random
Last name: user
```

2. Stop all IdM services on **rhel7.example.com** to force domain discovery to the new **rhel8.example.com** server.

```
[root@rhel7 ~]# ipactl stop
Stopping CA Service
Stopping pki-ca: [ OK ]
Stopping HTTP Service
Stopping httpd: [ OK ]
Stopping MEMCACHE Service
Stopping ipa_memcached: [ OK ]
Stopping DNS Service
Stopping named: . [ OK ]
Stopping KPASSWD Service
Stopping Kerberos 5 Admin Server: [ OK ]
Stopping KDC Service
Stopping Kerberos 5 KDC: [ OK ]
Stopping Directory Service
Shutting down dirsrv:
  EXAMPLE-COM... [ OK ]
  PKI-IPA... [ OK ]
```

After this, the **ipa** utility will contact the new server through a remote procedure call (RPC).

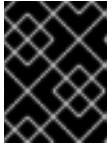
3. Remove the RHEL 7 server from the topology by executing the removal commands on the RHEL 8 server. For details, see [Uninstalling an IdM server](#).

## CHAPTER 31. UPDATING AND DOWNGRADING IDM

You can use the **yum** utility to update the Identity Management (IdM) packages on the system.

- To update all IdM packages that are relevant for your profile and that have updates available:

```
# yum upgrade ipa-*
```



### IMPORTANT

Before installing an update, make sure you have applied all previously released errata relevant to the RHEL system.

- Alternatively, to install or update packages to match the latest version available for your profile from any enabled repository:

```
# yum distro-sync ipa-*
```

After you update the IdM packages on at least one server, all other servers in the topology receive the updated schema, even if you do not update their packages. This ensures that any new entries which use the new schema can be replicated among the other servers.

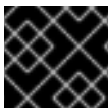


### WARNING

When updating multiple IdM servers, wait at least 10 minutes after updating one server before updating another server. However, the actual time required for a server's successful update depends on the topology deployed, the latency of the connections, and the number of changes generated by the update.

When two or more servers are updated simultaneously or with only short intervals between the upgrades, there is not enough time to replicate the post-upgrade data changes throughout the topology, which can result in conflicting replication events.

Downgrading IdM packages manually is not supported. Use **yum distro-sync** to update and downgrade packages in modules.



### IMPORTANT

Do not run the **yum downgrade** command on any of the **ipa-\*** packages.

### Additional resources

- For details on using the **yum** utility, see the **yum(8)** manual pages.



## CHAPTER 32. UPGRADING AN IDM CLIENT FROM RHEL 7 TO RHEL 8

Unlike IdM servers, performing an in-place upgrade of an IdM client from RHEL 7 to RHEL 8 is supported.

In RHEL 8, some uncommon options and unused functionality have been removed from the System Security Services Daemon (SSSD), the service responsible for authentication in an IdM environment. See the following sections for steps to remove those options.

- [Updating the SSSD configuration after upgrading to RHEL 8](#)
- [List of SSSD functionality removed in RHEL 8](#)

### 32.1. UPDATING THE SSSD CONFIGURATION AFTER UPGRADING TO RHEL 8

After upgrading an Identity Management (IdM) client from Red Hat Enterprise Linux (RHEL) 7 to RHEL 8, the **leapp** upgrade application might display a warning that some SSSD configuration options are no longer supported.

The following procedures describe how to update your SSSD configuration to address these issues.

#### Prerequisites

- You have upgraded an IdM client from RHEL 7 to RHEL 8.
- You have **root** permissions to edit `/etc/sss/sssd.conf`.

#### 32.1.1. Switching from the local ID provider to the files ID provider

If you see the following error, replace the **local** ID provider with the **files** ID provider:

SSSD Domain "example.com": local provider is no longer supported and the domain will be ignored.  
**Local provider is no longer supported.**

#### Procedure

1. Ensure any users and groups you retrieved with the **local** ID provider are also in the `/etc/passwd` and `/etc/group` files. This ensures that the **files** provider can access those users and groups.
  - a. If you need to create users, use the **useradd** command. If you need to specify the UID, add the **-u** option:

```
[root@client ~]# useradd -u 3001 username
```

- b. If you need to create groups, use the **groupadd** command. If you need to specify the GID, add the **-g** option:

```
[root@client ~]# groupadd -g 5001 groupname
```

2. Open the **/etc/sss/sss.conf** configuration file in a text editor.
3. Replace **id\_provider=local** with **id\_provider=files**.

```
[domain/example.com]
id_provider = files
...
```

4. Save the **/etc/sss/sss.conf** configuration file.
5. Restart SSSD to load the configuration changes.

```
[root@client ~]# systemctl restart sssd
```

### 32.1.2. Removing deprecated options

If you see either of the following errors regarding deprecated options, Red Hat recommends removing those options from the **/etc/sss/sss.conf** configuration file:

```
SSSD Domain "example.com": option ldap_groups_use_matching_rule_in_chain has no longer
any effect
Option ldap_groups_use_matching_rule_in_chain was removed and it will be ignored.
```

```
SSSD Domain "example.com": option ldap_initgroups_use_matching_rule_in_chain has no
longer any effect
Option ldap_initgroups_use_matching_rule_in_chain was removed and it will be ignored.
```

#### Procedure

1. Open the **/etc/sss/sss.conf** configuration file in a text editor.
2. Remove any occurrences of **ldap\_groups\_use\_matching\_rule\_in\_chain** or **ldap\_initgroups\_use\_matching\_rule\_in\_chain** options.
3. Save the **/etc/sss/sss.conf** configuration file.
4. Restart SSSD to load the configuration changes.

```
[root@client ~]# systemctl restart sssd
```

### 32.1.3. Enabling wildcard matching for sudo rules

The following warning indicates that **sudo** rules with wildcards in them will not work by default in RHEL 8, as the **ldap\_sudo\_include\_regexp** option is now set to **false** by default.

```
SSSD Domain "example.com": sudo rules containing wildcards will stop working.
Default value of ldap_sudo_include_regexp changed from true to false for performance reason.
```

If you use **sudo** rules with wildcards and want to enable wildcard matching, manually set the **ldap\_sudo\_include\_regexp** option to **true**.



## NOTE

Red Hat recommends against using wildcards to match **sudo** rules.

If the **ldap\_sudo\_include\_regexp** option is set to **true**, SSSD downloads every **sudo** rule that contains a wildcard in the **sudoHost** attribute, which negatively impacts LDAP search performance.

## Procedure

1. Open the **/etc/sss/sss.conf** configuration file in a text editor.
2. In the **example.com** domain, set **ldap\_sudo\_include\_regexp=true**.

```
[domain/example.com]
...
ldap_sudo_include_regexp = true
...
```

3. Save the **/etc/sss/sss.conf** configuration file.
4. Restart SSSD to load the configuration changes.

```
[root@client ~]# systemctl restart sssd
```

## 32.2. LIST OF SSSD FUNCTIONALITY REMOVED IN RHEL 8

The following SSSD functionality has been removed in RHEL 8.

### The local ID provider has been removed

The **local** ID provider, used to serve user information from the local SSSD cache, was deprecated in RHEL 7 and is no longer supported in RHEL 8. If you have a domain with **id\_provider=local** in your **/etc/sss/sss.conf** configuration, SSSD ignores this domain and starts normally.

### Command line tools to manage users and groups in local domains have been removed

The following commands, which only affected **local** domains, have been removed:

- **sss\_useradd**
- **sss\_userdel**
- **sss\_groupadd**
- **sss\_groupdel**

### Support for the **ldap\_groups\_use\_matching\_rule\_in\_chain** option has been removed

This Active Directory-specific option does not provide a significant performance benefit and is ignored in any RHEL 8 **sss.conf** configuration.

### Support for the **ldap\_initgroups\_use\_matching\_rule\_in\_chain** option has been removed

This Active Directory-specific option does not provide a significant performance benefit and is ignored in any RHEL 8 **sss.conf** configuration.

### The **ldap\_sudo\_include\_regexp** option now defaults to **false**

In RHEL 7, this option was set to **true** by default. If this option is set to **true**, SSSD downloads every **sudo** rule that contains a wildcard in the **sudoHost** attribute, which negatively impacts LDAP search performance.

#### The **sssd-secrets** responder has been removed

As the Kerberos Cache Manager (KCM) no longer relies on the **sssd-secrets** responder, and no other IdM process uses it, it has been removed.

## 32.3. ADDITIONAL RESOURCES

- For more details on upgrading to RHEL 8, see [Upgrading from RHEL 7 to RHEL 8](#) .

## PART IV. MIGRATING TO IDM FROM EXTERNAL SOURCES

## CHAPTER 33. MIGRATING FROM AN LDAP DIRECTORY TO IDM

If you previously deployed an LDAP server for identity and authentication lookups, you can migrate the lookup service to Identity Management (IdM). IdM offers a migration tool to help you with the following tasks:

- Transferring user accounts, including passwords and group membership, without losing data.
- Avoiding expensive configuration updates on the clients.

The migration process described here assumes a simple deployment scenario with one namespace in LDAP and one in IdM. For more complex environments, such as those with multiple namespaces or custom schemas, contact the Red Hat support services.

### 33.1. CONSIDERATIONS IN MIGRATING FROM LDAP TO IDM

The process of moving from an LDAP server to Identity Management (IdM) has the following stages:

- Migrating the *clients*. Plan this stage carefully. Determine which services each client in your current infrastructure uses. These may include for example Kerberos or Systems Security Services Daemon (SSSD). Then determine which of these services you can use in the final IdM deployment. See [Planning the client configuration when migrating from LDAP to IdM](#) for more information.
- Migrating the *data*.
- Migrating the *passwords*. Plan this stage carefully. IdM requires Kerberos hashes for every user account in addition to passwords. Some of the considerations and migration paths for passwords are covered in [Planning password migration when migrating from LDAP to IdM](#).

You can first migrate the server part and then the clients or first the clients and then the server. For more information about the two types of migration, see [LDAP to IdM migration sequence](#).



#### IMPORTANT

It is strongly recommended that you set up a test LDAP environment and test the migration process before attempting to migrate the real LDAP environment. When testing the environment, do the following:

1. Create a test user in IdM and compare the output of migrated users to that of the test user. Ensure that the migrated users contain the minimal set of attributes and object classes present on the test user.
2. Compare the output of migrated users, as seen on IdM, to the source users, as seen on the original LDAP server. Ensure that imported attributes are not copied twice and that they have the correct values.

### 33.2. PLANNING THE CLIENT CONFIGURATION WHEN MIGRATING FROM LDAP TO IDM

Identity Management (IdM) can support a number of different client configurations, with varying degrees of functionality, flexibility, and security. Decide which configuration is best for each individual client based on its operating system and your IT maintenance priorities. Consider also the client's

functional area: a development machine typically requires a different configuration than production servers or user laptops do.



## IMPORTANT

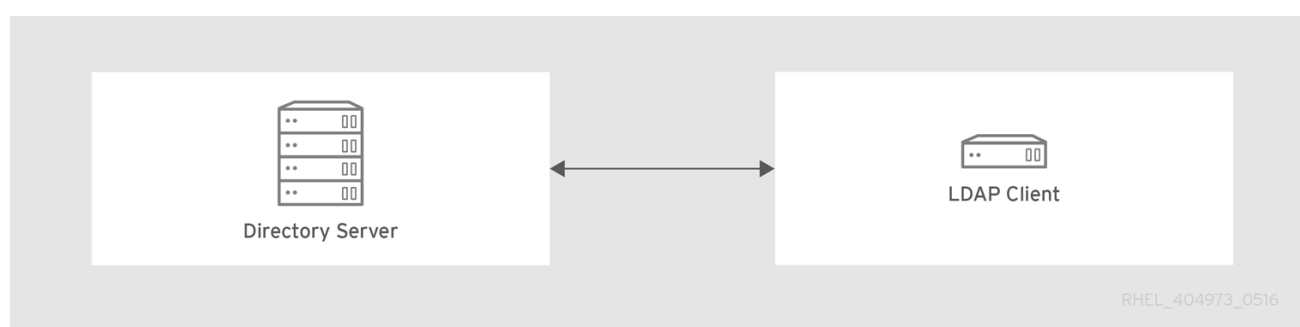
Most environments have a mixture of different ways in which clients connect to the IdM domain. Administrators must decide which scenario is best for each individual client.

### 33.2.1. Initial, pre-migration client configuration

Before deciding on the specifics of the client configuration in Identity Management (IdM), first establish the specifics of the current, pre-migration configuration.

The initial state for almost all LDAP deployments that are to be migrated is that there is an LDAP service providing identity and authentication services.

**Figure 33.1. Basic LDAP directory and client configuration**



Linux and Unix clients use the PAM\_LDAP and NSS\_LDAP libraries to connect directly to the LDAP services. These libraries allow clients to retrieve user information from the LDAP directory as if the data were stored in **/etc/passwd** or **/etc/shadow**. In real life, the infrastructure may be more complex if a client uses LDAP for identity lookups and Kerberos for authentication or other configurations.

There are structural differences between an LDAP directory and an Identity Management (IdM) server, particularly in schema support and the structure of the directory tree. For more background on those differences, see [Contrasting IdM with a Standard LDAP Directory](#). Those differences may impact data, especially with the directory tree, which affects entry names. However, the differences have little impact on the client configuration and on migrating clients to IdM.

### 33.2.2. Recommended configuration for RHEL clients



## NOTE

The client configuration described in this section is only supported for RHEL 6.1 and later and RHEL 5.7 later, which support the latest versions of SSSD and the **ipa-client** package. Older versions of RHEL can be configured as described in [Alternative supported configuration](#).

The System Security Services Daemon (SSSD) in Red Hat Enterprise Linux (RHEL) uses special PAM and NSS libraries, **pam\_sss** and **nss\_sss**. Using these libraries, SSSD can integrate very closely with Identity Management (IdM) and benefit from its full authentication and identity features. SSSD has a number of useful features, such as caching identity information so that users can log in even if the connection to the central server is lost.

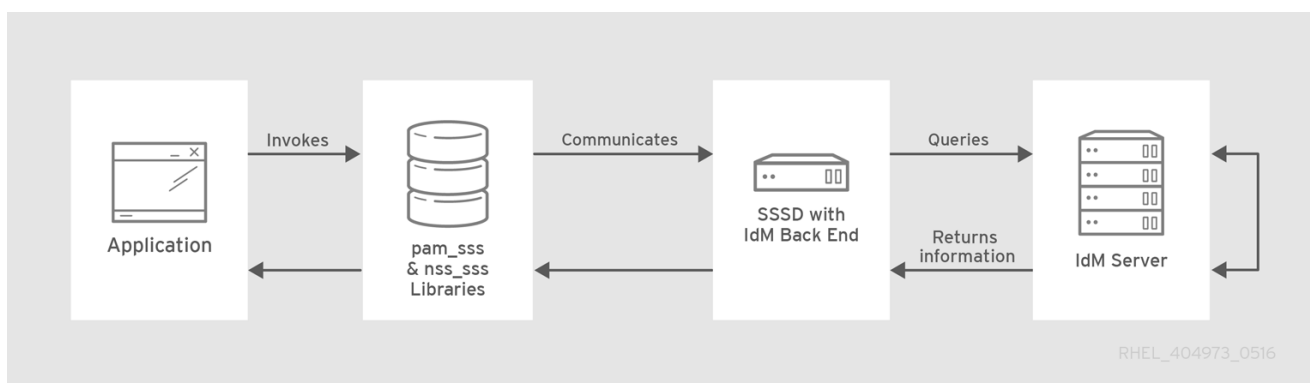
Unlike generic LDAP directory services that use the **pam\_ldap** and **nss\_ldap** libraries, SSSD establishes relationships between identity and authentication information by defining *domains*. A domain in SSSD defines the following back end functions:

- Authentication
- Identity lookups
- Access
- Password changes

The SSSD domain is then configured to use a *provider* to supply the information for any one, or all, of these functions. The domain configuration always requires an *identity* provider. The other three providers are optional; if an authentication, access, or password provider is not defined, then the identity provider is used for that function.

SSSD can use IdM for all of its back end functions. This is the ideal configuration because it provides the full range of IdM functionality, unlike generic LDAP identity providers or Kerberos authentication. For example, during daily operation, SSSD enforces host-based access control rules and security features in IdM.

**Figure 33.2. Clients and SSSD with an IdM back end**



The **ipa-client-install** script automatically configures SSSD to use IdM for all its back end services, so that RHEL clients are set up with the recommended configuration by default.

#### Additional information

- [Understanding SSSD and its benefits](#)

### 33.2.3. Alternative supported configuration

Unix and Linux systems such as Mac, Solaris, HP-UX, AIX, and Scientific Linux support all of the services that Identity Management (IdM) manages but do not use SSSD. Similarly, older Red Hat Enterprise Linux (RHEL) versions, specifically 6.1 and 5.6, support SSSD but have an older version, which does not support IdM as an identity provider.

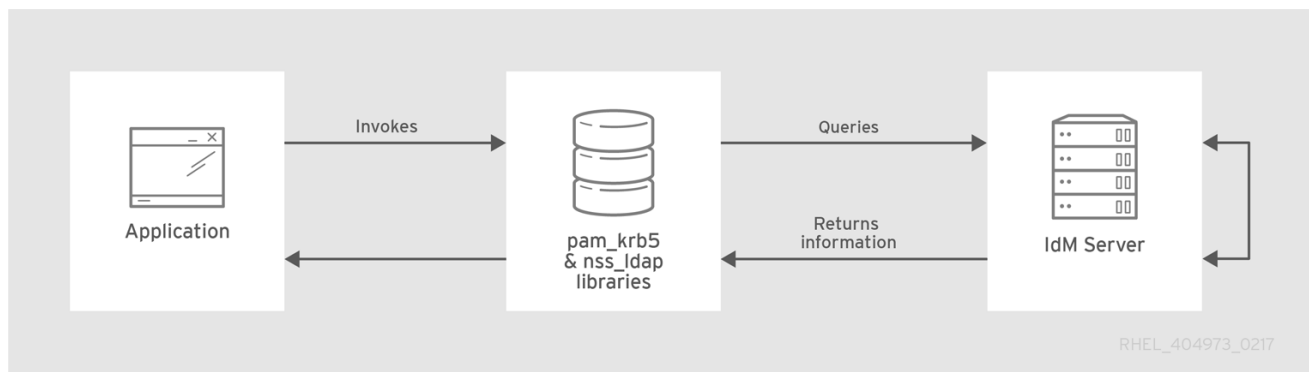
If it is not possible to use a modern version of SSSD on a system, then clients can be configured in the following way:

- The client connects to the IdM server as if it were an LDAP directory server for identity lookups, by using **nss\_ldap**.
- The client connects to the IdM server as if it were a regular Kerberos KDC, by using **pam\_krb5**.



For more information on configuring a *RHEL client with an older version of SSSD* to use the IdM server as its identity provider and its Kerberos authentication domain, see the [Configuring identity and authentication providers for SSSD](#) section of the RHEL 7 *System-Level Authentication Guide*.

**Figure 33.3. Clients and IdM with LDAP and Kerberos**



It is generally best practice to use the most secure configuration possible for a client. This means SSSD or LDAP for identities and Kerberos for authentication. However, for some maintenance situations and IT structures, you may need to resort to the simplest possible scenario: configuring LDAP to provide both identity and authentication by using the **nss\_ldap** and **pam\_ldap** libraries on the clients.

### 33.3. PLANNING PASSWORD MIGRATION WHEN MIGRATING FROM LDAP TO IDM

A crucial question to answer before migrating users from LDAP to Identity Management (IdM) is whether to migrate user passwords or not. The following options are available:

#### Migrating users without passwords

Can be performed more quickly but requires more manual work by administrators and users. In certain situations, this is the only available option: for example, if the [original LDAP environment stored cleartext user passwords](#) or if the [passwords do not meet the password policy requirements defined in IdM](#).

When migrating user accounts without passwords, you reset all user passwords. The migrated users are assigned a temporary password that they change at the first login. For more information on how to reset passwords, see [Changing and resetting user passwords](#).

#### Migrating users with their passwords

Provides a smoother transition but also requires parallel management of LDAP directory and IdM during the migration and transition process. The reason for this is that by default, IdM uses Kerberos for authentication and requires that each user has a Kerberos hash stored in the IdM Directory Server in addition to the standard user password. To generate the hash, the user password needs to be available to the IdM server in clear text. When you create a new user password, the password is available in clear text before it is hashed and stored in IdM. However, when the user is migrated from an LDAP directory, the associated user password is already hashed, so the corresponding Kerberos key cannot be generated.



#### IMPORTANT

By default, users cannot authenticate to the IdM domain or access IdM resources until they have Kerberos hashes – even if the user accounts already exist. One workaround is available: using LDAP authentication in IdM instead of Kerberos authentication. With this workaround, Kerberos hashes are not required for users. However, this workaround limits the capabilities of IdM and is not recommended.

You can migrate users with their passwords:

- [Using a web page](#)
- [Using SSSD](#)

#### Additional resources

- [Planning the migration of cleartext LDAP passwords](#)
- [Planning the migration of LDAP passwords that do not meet the IdM requirements](#)
- [Methods for migrating passwords when migrating LDAP to IdM](#)

### 33.3.1. Methods for migrating passwords when migrating LDAP to IdM

To migrate user accounts from LDAP to Identity Management (IdM) without forcing the users to change their passwords, you can use the following methods:

#### Method 1: Using the migration web page

Tell users to enter their LDAP credentials once into a special page in the IdM Web UI, <https://ipaserver.example.com/ipa/migration>. A script running in the background then captures the clear text password and properly updates the user account with the password and an appropriate Kerberos hash.

#### Method 2 (recommended): Using SSSD

Mitigate the user impact of the migration by using the System Security Services Daemon (SSSD) to generate the required user keys. For deployments with a lot of users or where users should not be burdened with password changes, this is the best scenario.

#### Workflow

1. A user tries to log into a machine with SSSD.
2. SSSD attempts to perform Kerberos authentication against the IdM server.
3. Even though the user exists in the system, the authentication fails with the error *key type is not supported* because the Kerberos hashes do not exist yet.
4. SSSD performs a plain text LDAP bind over a secure connection.
5. IdM intercepts this bind request. If the user has a Kerberos principal but no Kerberos hashes, then the IdM identity provider generates the hashes and stores them in the user entry.
6. If authentication is successful, SSSD disconnects from IdM and tries Kerberos authentication again. This time, the request succeeds because the hash exists in the entry.

With method 2, the entire process is invisible to the users. They log in to a client service without noticing that their password has been moved from LDAP to IdM.

### 33.3.2. Planning the migration of cleartext LDAP passwords

Although in most deployments LDAP passwords are stored encrypted, there may be some users or some environments that use cleartext passwords for user entries.

When users are migrated from the LDAP server to the IdM server, their cleartext passwords are not migrated over because IdM does not allow cleartext passwords. Instead, a Kerberos principal is created for each user, the keytab is set to true, and the password is set as expired. This means that IdM requires the user to reset the password at the next login. For more information, see [Planning the migration of LDAP passwords that do not meet the IdM requirements](#).

### 33.3.3. Planning the migration of LDAP passwords that do not meet the IdM requirements

If user passwords in the original directory do not meet the password policies defined in Identity Management (IdM), the passwords become invalid after the migration.

Password reset is done automatically the first time a user attempts to obtain a Kerberos ticket-granting ticket (TGT) in the IdM domain by entering **kinit**. The user is forced to change his or her password:

```
[migrated_idm_user@idmclient ~]$ kinit
Password for migrated_idm_user@IDM.EXAMPLE.COM:
Password expired. You must change it now.
Enter new password:
Enter it again:
```

## 33.4. FURTHER MIGRATION CONSIDERATIONS AND REQUIREMENTS

As you are planning a migration from an LDAP server to Identity Management (IdM), ensure that your LDAP environment is able to work with the IdM migration script.

### 33.4.1. LDAP servers supported for migration

The migration process from an LDAP server to IdM uses a special script, **ipa migrate-ds**, to perform the migration. This script has specific requirements regarding the structure of the LDAP directory and LDAP entries. Migration is supported only for LDAPv3-compliant directory services, which include several common directories:

- Sun ONE Directory Server
- Apache Directory Server
- OpenLDAP

Migration from an LDAP server to IdM has been tested with Red Hat Directory Server and OpenLDAP.



#### NOTE

Migration using the migration script is *not* supported for Microsoft Active Directory because it is not an LDAPv3-compliant directory. For assistance with migrating from Active Directory, contact Red Hat Professional Services.

### 33.4.2. LDAP environment requirements for migration

Many different possible configuration scenarios exist for LDAP servers and for Identity Management (IdM), which affects the smoothness of the migration process. For the example migration procedures in this chapter, these are the assumptions about the environment:

- A single LDAP directory domain is being migrated to one IdM realm. No consolidation is involved.
- A user password is stored as a hash in the LDAP directory. For a list of supported hashes, see the Password Storage Schemes section in the *Configuration, Command, and File Reference* title available in the Red Hat Directory Server 10 section of [Red Hat Directory Server Documentation](#).
- The LDAP directory instance is both the identity store and the authentication method. Client machines are configured to use the **pam\_ldap** or **nss\_ldap** library to connect to the LDAP server.
- Entries use only the standard LDAP schema. Entries that contain custom object classes or attributes are not migrated to IdM.
- The **migrate-ds** command only migrates the following accounts:
  - Those containing a **gidNumber** attribute. The attribute is required by the **posixAccount** object class.
  - Those containing an **sn** attribute. The attribute is required by the **person** object class.

### 33.4.3. IdM system requirements for migration

With a moderately-sized directory of around 10,000 users and 10 groups, it is necessary to have a powerful enough target IdM system to allow the migration to proceed. The minimum requirements for a migration are:

- 4 cores
- 4GB of RAM
- 30GB of disk space
- A SASL buffer size of 2MB, which is the default for an IdM server  
In case of migration errors, increase the buffer size:

```
[root@ipaserver ~]# ldapmodify -x -D 'cn=directory manager' -w password -h ipaserver.example.com -p 389
```

```
dn: cn=config
changetype: modify
replace: nsslapd-sasl-max-buffer-size
nsslapd-sasl-max-buffer-size: 4194304
```

```
modifying entry "cn=config"
```

Set the **nsslapd-sasl-max-buffer-size** value in bytes.

#### Additional resources

- [IdM server hardware recommendations](#)

### 33.4.4. Considerations about sudo rules

If you are using **sudo** with LDAP, you must migrate the **sudo** rules stored in LDAP to Identity Management (IdM) manually. Red Hat recommends that you recreate netgroups in IdM as hostgroups. IdM presents hostgroups automatically as traditional netgroups for **sudo** configurations that do not use the SSSD **sudo** provider.

### 33.4.5. LDAP to IdM migration tools

Identity Management (IdM) uses a specific command, **ipa migrate-ds**, to execute the migration process so that LDAP directory data are properly formatted and imported cleanly into the IdM server. When using **ipa migrate-ds**, the remote system user, specified by the **--bind-dn** option, must have read access to the **userPassword** attribute, otherwise passwords will not be migrated.

The IdM server must be configured to run in migration mode, and then the migration script can be used. For details, see [Migrating an LDAP server to IdM](#).

### 33.4.6. Improving LDAP to IdM migration performance

An LDAP migration is essentially a specialized import operation for the 389 Directory Server (DS) instance within the IdM server. Tuning the 389 DS instance for better import operation performance can help improve the overall migration performance.

There are two parameters that directly affect import performance:

- The **nsslapd-cachememsize** attribute, which defines the size allowed for the entry cache. This is a buffer that is automatically set to 80% of the total cache memory size. For large import operations, you can increase this parameter and possibly the memory cache itself. This increase will improve the efficiency of the directory service in handling a large number of entries or entries with large attributes.  
For details on how to modify the attribute using the **dsconf** command, see [Adjusting the entry cache size](#).
- The system **ulimit** configuration option sets the maximum number of allowed processes for a system user. Processing a large database can exceed the limit. If this happens, increase the value:

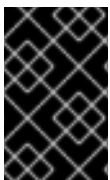
```
[root@server ~]# ulimit -u 4096
```

#### Additional resources

- [Adjusting IdM Directory Server performance](#)

### 33.4.7. LDAP to IdM migration sequence

There are four major steps when migrating to IdM, but their order varies depending on whether you want to first migrate the *server* or the *clients*.



#### IMPORTANT

Both the client-first and server-first migrations provide a general migration procedure, but they may not work in every environment. Set up a test LDAP environment and test the migration process before attempting to migrate the real LDAP environment.

### Client-first migration

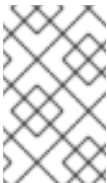
SSSD is used to change the client configuration while an Identity Management (IdM) server is configured:

1. Deploy SSSD.
2. Reconfigure clients to connect to the current LDAP server and then fail over to IdM.
3. Install the IdM server.
4. Migrate the user data using the IdM **ipa migrate-ds** script. This exports the data from the LDAP directory, formats for the IdM schema, and then imports it into IdM.
5. Take the LDAP server offline and allow clients to fail over to IdM transparently.

### Server-first migration

The LDAP to IdM migration comes first:

1. Install the IdM server.
2. Migrate the user data using the IdM **ipa migrate-ds** script. This exports the data from the LDAP directory, formats it for the IdM schema, and then imports it into IdM.
3. *Optional.* Deploy SSSD.
4. Reconfigure clients to connect to IdM. It is not possible to simply replace the LDAP server. The IdM directory tree – and therefore user entry DNs – is different from the previous directory tree.  
While it is required that clients must be reconfigured, clients do not need to be reconfigured immediately. Updated clients can point to the IdM server while other clients point to the old LDAP directory, allowing a reasonable testing and transition phase after the data are migrated.



#### NOTE

Do not run both an LDAP directory service and the IdM server for very long in parallel. This introduces the risk of user data becoming inconsistent between the two services.

## 33.5. CUSTOMIZING THE MIGRATION FROM LDAP TO IDM

You can migrate your authentication and authorization services from an LDAP server to Identity Management (IdM) using the **ipa migrate-ds** command. Without additional options, the command takes the LDAP URL of the directory to migrate and exports the data based on common default settings.

You can customize the migration process and how data is identified and exported by using different **ipa migrate-ds** command options. Customize the migration if your LDAP directory tree has a unique structure or if you know you must exclude certain entries or attributes within entries.

### 33.5.1. Examples of customizing the Bind DN and Base DN during the migration from LDAP to IdM

Use the **ipa migrate-ds** command to migrate from LDAP to Identity Management (IdM). Without additional options, the command takes the LDAP URL of the directory to migrate and exports the data based on common default settings. This section describes examples of modifying the default settings.

```
# ipa migrate-ds ldap://ldap.example.com:389
```

### Customizing the Bind DN

By default, the DN "**cn=Directory Manager**" is used to bind to the remote LDAP directory. Use the **--bind-dn** option to specify a custom bind DN:

```
# ipa migrate-ds ldap://ldap.example.com:389 --bind-dn=cn=Manager,dc=example,dc=com
```

### Customizing the naming context

If the LDAP server naming context differs from the one used in IdM, the base DN's for objects are transformed. For example: **uid=user,ou=people,dc=ldap,dc=example,dc=com** is migrated to **uid=user,ou=people,dc=idm,dc=example,dc=com**. Using the **--base-dn** option, you can change the target for container subtrees and thus set the base DN used on the remote LDAP server for the migration:

```
# ipa migrate-ds --base-dn="ou=people,dc=example,dc=com" ldap://ldap.example.com:389
```

### Additional resources

- **ipa migrate-ds --help**

## 33.5.2. The migration of specific subtrees

The default directory structure places person entries in the **ou=People** subtree and group entries in the **ou=Groups** subtree. These subtrees are container entries for those different types of directory data. If you do not use any options with the **migrate-ds** command, then the utility assumes that the given LDAP directory uses the **ou=People** and **ou=Groups** structure.

Many deployments may have an entirely different directory structure or you may only want to export certain parts of the original directory tree. As an administrator, you can use the following options to specify the RDN of a different user or group subtree on the source LDAP server:

- **--user-container**
- **--group-container**



### NOTE

In both cases, the subtree must be a relative distinguished name (RDN) and must be relative to the base DN. For example, you can migrate the **>ou=Employees,dc=example,dc=com** directory tree by using **--user-container=ou=Employees**.

For example:

```
[ipaserver ~]# ipa migrate-ds --user-container=ou=employees \
--group-container="ou=employee groups" ldap://ldap.example.com:389
```

Optionally, add the **--scope** option to the **ipa migrate-ds** command to set the scope:

- **onelevel**: Default. Only entries in the specified container are migrated.
- **subtree**: Entries in the specified container and all subcontainers are migrated.
- **base**: Only the specified object itself is migrated.

### 33.5.3. The inclusion and exclusion of entries

By default, the **ipa migrate-ds** script imports every user entry with the **person** object class and every group entry with the **groupOfUniqueNames** or **groupOfNames** object class.

In some migration paths, only specific types of users and groups may need to be exported, or, alternatively, specific users and groups may need to be excluded. You can select which *types* of users and groups to include by setting which object classes to search for when looking for user or group entries.

This option is particularly useful when you use custom object classes for different *user* types. For example, the following command migrates only users with the custom **fullTimeEmployee** object class:

```
[root@ipaserver ~]# ipa migrate-ds --user-objectclass=fullTimeEmployee
ldap://ldap.example.com:389
```

Because of the different types of groups, this is also very useful for migrating only certain types of *groups*, such as user groups, while excluding other types of groups, like certificate groups. For example:

```
[root@ipaserver ~]# ipa migrate-ds --group-objectclass=groupOfNames --group-
objectclass=groupOfUniqueNames ldap://ldap.example.com:389
```

Specifying user and group entries to migrate based on object class implicitly excludes all other users and groups from migration.

Alternatively, it can be useful to migrate all user and group entries except for just a small handful of entries. You can exclude specific user or group accounts while migrating all others of that type. For example, this excludes only a hobbies group and two users:

```
[root@ipaserver ~]# ipa migrate-ds --exclude-groups="Golfers Group" --exclude-
users=idmuser101 --exclude-users=idmuser102 ldap://ldap.example.com:389
```

Exclude statements are applied to users matching the pattern in the **uid** and to groups matching it in the **cn** attribute.

You can migrate a general object class but exclude specific entries of that class. For example, this specifically includes users with the **fullTimeEmployee** object class, yet excludes three managers:

```
[root@ipaserver ~]# ipa migrate-ds --user-objectclass=fullTimeEmployee --exclude-
users=jsmith --exclude-users=bjensen --exclude-users=mreynolds
ldap://ldap.example.com:389
```

### 33.5.4. The exclusion of entry attributes

By default, every attribute and object class for a user or group entry is migrated. In certain scenarios, that may not be realistic, either because of bandwidth and network constraints or because the attribute



data are no longer relevant. For example, if users are going to be assigned new user certificates as they join the Identity Management (IdM) domain, then migrating the **userCertificate** attribute would be useless.

You can ignore specific object classes and attributes by using the following options with the **migrate-ds** command:

- **--user-ignore-objectclass**
- **--user-ignore-attribute**
- **--group-ignore-objectclass**
- **--group-ignore-attribute**

For example, to exclude the **userCertificate** attribute and **strongAuthenticationUser** object class for users and the **groupOfCertificates** object class for groups:

```
[root@ipaserver ~]# ipa migrate-ds --user-ignore-attribute=userCertificate --user-ignore-objectclass=strongAuthenticationUser --group-ignore-objectclass=groupOfCertificates ldap://ldap.example.com:389
```



#### NOTE

Make sure not to ignore any required attributes. Also, when excluding object classes, make sure to exclude any attributes that only that object class supports.

#### Additional resources

- [LDAP environment requirements for migration](#)

### 33.5.5. The schema to use when migrating from LDAP to IdM and the schema compat feature

Identity Management (IdM) uses the RFC2307bis schema to define user, host, host group, and other network identities. However, if the LDAP server used as the source for the migration uses the RFC2307 schema instead, specify the **--schema** option with the **ipa migrate-ds** command:

```
[root@ipaserver ~]# ipa migrate-ds --schema=RFC2307 ldap://ldap.example.com:389
```

Alternatively, IdM has a built-in **schema compat feature** that allows IdM to reformat data for systems that do not support RFC2307bis. The compat plugin is enabled by default, which means that the directory server computes an alternate view of the users and groups and provides this view in the **cn=users,cn=compat,dc=example,dc=com** container entry. It does this by precomputing the contents of its entries at startup-time and refreshing its entries as needed.

It is recommended that this feature is disabled during the migration to reduce system overhead.

## 33.6. MIGRATING AN LDAP SERVER TO IDM

You can migrate your authentication and authorization services from an LDAP server to Identity Management (IdM) using the **ipa migrate-ds** command.

**WARNING**

This is a general migration procedure that may not work in every environment.

It is strongly recommended that you set up a test LDAP environment and test the migration process before attempting to migrate the real LDAP environment. When testing the environment, do the following:

1. Create a test user in IdM and compare the output of migrated users to that of the test user.
2. Compare the output of migrated users, as seen on IdM, to the source users, as seen on the original LDAP server.

For more guidance, see the **Verification** section below.

**Prerequisites**

- You have administrator privileges to the LDAP directory.
- If IdM is already installed, you have administrator privileges to IdM.
- You are logged in as **root** on the RHEL system on which you are executing the procedure below.
- You have read and understood the following chapters:
  - [Considerations in migrating from LDAP to IdM](#) .
  - [Planning the client configuration when migrating from LDAP to IdM](#) .
  - [Planning password migration when migrating from LDAP to IdM](#) .
  - [Further migration considerations and requirements](#) .
  - [Customizing the migration from LDAP to IdM](#) .

**Procedure**

1. If IdM is not yet installed: install the IdM server, including any custom LDAP directory schema, on a different machine from the one on which the existing LDAP directory is installed. For details, see [Installing Identity Management](#) .

**NOTE**

Custom user or group schemas have limited support in IdM. They can cause problems during the migration because of incompatible object definitions.

2. For performance reasons, disable the compat plug-in:

```
# ipa-compat-manage disable
```

For more information on the schema compat feature and the benefits of disabling it for the migration, see [The schema to use when migrating from LDAP to IdM and the schema compat feature](#).

- Restart the IdM Directory Server instance:

```
# systemctl restart dirsrv.target
```

- Configure the IdM server to allow migration:

```
# ipa config-mod --enable-migration=TRUE
```

By setting **--enable-migration** to TRUE, you do the following:

- Allow pre-hashed passwords during an LDAP add operation.
- Configure SSSD to try the password migration sequence if the initial Kerberos authentication fails. For more information, see the Workflow section in [Using SSSD when migrating passwords from LDAP to IdM](#).

- Run the IdM migration script, **ipa migrate-ds**, with the options that are relevant for your use case. For more information, see [Customizing the migration from LDAP to IdM](#).

```
# ipa migrate-ds --your-options ldap://ldap.example.com:389
```



#### NOTE

If you did not disable the compat plug-in in one of the previous steps, add the **--with-compat** option to **ipa migrate-ds**:

```
# ipa migrate-ds --your-options --with-compat  
ldap://ldap.example.com:389
```

- Re-enable the compat plug-in:

```
# ipa-compat-manage enable
```

- Restart the IdM Directory Server:

```
# systemctl restart dirsrv.target
```

- When all users have had their passwords migrated, disable the migration mode:

```
# ipa config-mod --enable-migration=FALSE
```

- [Optional] When all of the users have been migrated, reconfigure non-SSSD clients to use Kerberos authentication, that is **pam\_krb5**, instead of LDAP authentication, that is **pam\_ldap**. For more information, see [Configuring a Kerberos Client](#) in the System-level Authentication Guide.

- Have users generate their hashed Kerberos passwords. Choose one of the methods described in [Planning password migration when migrating from LDAP to IdM](#).

For more information, see [Using SSSD with IdM](#).

- If you decide on the [SSSD method](#):
  - Move clients that have SSSD installed from the LDAP directory to the IdM directory, and enroll them as clients with IdM. This downloads the required keys and certificates. On Red Hat Enterprise Linux clients, this can be done using the **ipa-client-install** command. For example:

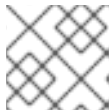
```
# ipa-client-install --enable-dns-update
```

- If you decide on the [IdM migration web page](#) method:
  - Instruct users to log into IdM using the migration web page:

```
https://ipaserver.example.com/ipa/migration
```

11. To monitor the user migration process, query the existing LDAP directory to see which user accounts have a password but do not yet have a Kerberos principal key.

```
$ ldapsearch -LL -x -D 'cn=Directory Manager' -w secret -b  
'cn=users,cn=accounts,dc=example,dc=com' '(&(!(krbprincipalkey=))(userpassword=))'  
uid
```



#### NOTE

Include the single quotes around the filter so that it is not interpreted by the shell.

12. When the migration of all clients and users is complete, decommission the LDAP directory.

## Verification

1. Create a test user in IdM by using the **ipa user-add** command. Compare the output of migrated users to that of the test user. Ensure that the migrated users contain the minimal set of attributes and object classes present on the test user. For example:

```
$ ipa user-show --all testing_user  
dn: uid=testing_user,cn=users,cn=accounts,dc=idm,dc=example,dc=com  
User login: testing_user  
First name: testing  
Last name: user  
Full name: testing user  
Display name: testing user  
Initials: tu  
Home directory: /home/testing_user  
GECOS: testing user  
Login shell: /bin/sh  
Principal name: testing_user@IDM.EXAMPLE.COM  
Principal alias: testing_user@IDM.EXAMPLE.COM  
Email address: testing_user@idm.example.com  
UID: 1689700012  
GID: 1689700012  
Account disabled: False  
Preserved user: False  
Password: False  
Member of groups: ipausers
```

```

Kerberos keys available: False
ipauniqueid: 843b1ac8-6e38-11ec-8dfe-5254005aad3e
mepmanagedentry: cn=testing_user,cn=groups,cn=accounts,dc=idm,dc=example,dc=com
objectclass: top, person, organizationalperson, inetorgperson, inetuser, posixaccount,
krbprincipalaux, krbticketpolicyaux, ipaobject,
ipasshuser, ipaSshGroupOfPubKeys, mepOriginEntry

```

2. Compare the output of migrated users, as seen on IdM, to the source users, as seen on the original LDAP server. Ensure that imported attributes are not copied twice and that they have the correct values.

### Additional resources

- [Migrating from LDAP to IdM over SSL](#)

## 33.7. MIGRATING FROM LDAP TO IDM OVER SSL

You can migrate your authentication and authorization services from an LDAP server to Identity Management (IdM) using the **ipa migrate-ds** command. This section describes how to encrypt the data transmitted during the migration.



### WARNING

This is a general migration procedure that may not work in every environment.

It is strongly recommended that you set up a test LDAP environment and test the migration process before attempting to migrate the real LDAP environment. When testing the environment, do the following:

1. Create a test user in IdM and compare the output of migrated users to that of the test user.
2. Compare the output of migrated users, as seen on IdM, to the source users, as seen on the original LDAP server.

For more guidance, see the **Verification** section below.

### Prerequisites

- You have administrator privileges to the LDAP directory.
- If IdM is already installed, you have administrator privileges to IdM.
- You are logged in as **root** on the RHEL system on which you are executing the procedure below.
- You have read and understood the following chapters:
  - [Considerations in migrating from LDAP to IdM](#) .
  - [Planning the client configuration when migrating from LDAP to IdM](#) .

- [Planning password migration when migrating from LDAP to IdM](#) .
- [Further migration considerations and requirements](#) .
- [Customizing the migration from LDAP to IdM](#) .

## Procedure

1. Store the certificate of the CA that issued the remote LDAP server certificate in a file on the future IdM server. For example: **/tmp/remote.crt**.
2. Follow the steps described in [Migrating an LDAP server to IdM](#) . However, for an encrypted LDAP connection during the migration, use the **ldaps** protocol in the URL and pass the **--ca-cert-file** option to the **ipa migrate-ds** command. For example:

```
# ipa migrate-ds --ca-cert-file=/tmp/remote.crt --your-other-options
ldaps://ldap.example.com:636
```

## Verification

1. Create a test user in IdM by using the **ipa user-add** command. Compare the output of migrated users to that of the test user. Ensure that the migrated users contain the minimal set of attributes and object classes present on the test user. For example:

```
$ ipa user-show --all testing_user
dn: uid=testing_user,cn=users,cn=accounts,dc=idm,dc=example,dc=com
User login: testing_user
First name: testing
Last name: user
Full name: testing user
Display name: testing user
Initials: tu
Home directory: /home/testing_user
GECOS: testing user
Login shell: /bin/sh
Principal name: testing_user@IDM.EXAMPLE.COM
Principal alias: testing_user@IDM.EXAMPLE.COM
Email address: testing_user@idm.example.com
UID: 1689700012
GID: 1689700012
Account disabled: False
Preserved user: False
Password: False
Member of groups: ipausers
Kerberos keys available: False
ipauniqueid: 843b1ac8-6e38-11ec-8dfe-5254005aad3e
mepmanagedentry: cn=testing_user,cn=groups,cn=accounts,dc=idm,dc=example,dc=com
objectclass: top, person, organizationalperson, inetorgperson, inetuser, posixaccount,
krbprincipalaux, krbticketpolicyaux, ipaobject,
ipaasshuser, ipaSshGroupOfPubKeys, mepOriginEntry
```

2. Compare the output of migrated users, as seen on IdM, to the source users, as seen on the original LDAP server. Ensure that imported attributes are not copied twice and that they have the correct values.

