



# **Red Hat Enterprise Linux 8.0 Beta**

## **Installing Identity Management and Access Control**

Getting started using your Identity Management and Access Control



# Red Hat Enterprise Linux 8.0 Beta Installing Identity Management and Access Control

---

Getting started using your Identity Management and Access Control

## Legal Notice

Copyright © 2019 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, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

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

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

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

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

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

The OpenStack ® 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 plan, install, configure and maintain Identity Management on Red Hat Enterprise Linux 8.

## Table of Contents

<b>THIS IS A BETA VERSION!</b> .....	<b>7</b>
<b>PROVIDING FEEDBACK ON RED HAT DOCUMENTATION</b> .....	<b>8</b>
<b>PART I. PLANNING IDENTITY MANAGEMENT AND ACCESS CONTROL</b> .....	<b>9</b>
<b>CHAPTER 1. OVERVIEW OF PLANNING FOR IDENTITY MANAGEMENT AND ACCESS CONTROL IN RED HAT ENTERPRISE LINUX</b> .....	<b>10</b>
1.1. INTRODUCTION TO IDENTITY MANAGEMENT .....	10
1.2. INTRODUCTION TO IDENTITY MANAGEMENT SERVERS AND CLIENTS .....	12
1.3. IDENTITY MANAGEMENT AND ACCESS CONTROL IN RED HAT ENTERPRISE LINUX: CENTRAL VS. LOCAL .....	13
1.4. ADDITIONAL RESOURCES .....	14
<b>CHAPTER 2. PLANNING THE REPLICA TOPOLOGY</b> .....	<b>15</b>
2.1. MULTIPLE REPLICA SERVERS AS A SOLUTION FOR HIGH PERFORMANCE AND DISASTER RECOVERY .....	15
2.2. IDENTITY MANAGEMENT SERVERS AND CLIENTS .....	15
2.3. REPLICATION AGREEMENTS .....	16
2.4. DETERMINING THE APPROPRIATE NUMBER OF REPLICAS .....	16
2.5. CONNECTING THE REPLICAS IN A TOPOLOGY .....	17
2.6. REPLICA TOPOLOGY EXAMPLES .....	17
<b>CHAPTER 3. PLANNING INTEGRATION WITH ACTIVE DIRECTORY</b> .....	<b>20</b>
3.1. DIRECT INTEGRATION .....	20
Recommendations .....	20
3.2. INDIRECT INTEGRATION .....	21
3.3. DECIDING BETWEEN INDIRECT AND DIRECT INTEGRATION .....	22
Number of systems to be connected to Active Directory .....	22
Frequency of deploying new systems and their type .....	22
Active Directory is the required authentication provider .....	22
<b>CHAPTER 4. PLANNING A CROSS-FOREST TRUST BETWEEN IDENTITY MANAGEMENT AND ACTIVE DIRECTORY</b> .....	<b>23</b>
4.1. CROSS-FOREST TRUSTS BETWEEN IDENTITY MANAGEMENT AND ACTIVE DIRECTORY .....	23
An external trust to an Active Directory domain .....	23
4.2. TRUST CONTROLLERS AND TRUST AGENTS .....	23
4.3. ONE-WAY TRUSTS AND TWO-WAY TRUSTS .....	24
4.4. NON-POSIX EXTERNAL GROUPS AND SECURITY ID MAPPING .....	24
4.5. SETTING UP DNS .....	25
4.6. NETBIOS NAMES .....	25
4.7. CONFIGURING ACTIVE DIRECTORY SERVER DISCOVERY AND AFFINITY .....	26
Options for configuring LDAP and Kerberos on the Identity Management client for communication with local Identity Management servers .....	26
Options for configuring Kerberos on the Identity Management client for communication with local Active Directory servers .....	27
Options for configuring embedded clients on Identity Management servers for communication with local Active Directory servers over Kerberos and LDAP .....	27
4.8. OPERATIONS PERFORMED DURING INDIRECT INTEGRATION OF IDENTITY MANAGEMENT TO ACTIVE DIRECTORY .....	27
<b>PART II. INSTALLING IDENTITY MANAGEMENT</b> .....	<b>30</b>
<b>CHAPTER 5. PREPARING THE SYSTEM FOR IDENTITY MANAGEMENT SERVER INSTALLATION</b> .....	<b>31</b>
5.1. HARDWARE RECOMMENDATIONS .....	31

5.2. CUSTOM CONFIGURATION REQUIREMENTS FOR IDENTITY MANAGEMENT	31
5.2.1. IPv6 requirements in Identity Management	31
5.3. HOST NAME AND DNS REQUIREMENTS FOR IDENTITY MANAGEMENT	31
5.4. PORT REQUIREMENTS FOR IDENTITY MANAGEMENT	35
Opening the required ports	35
5.5. PACKAGES REQUIRED FOR AN IDENTITY MANAGEMENT SERVER	36
<b>CHAPTER 6. INSTALLING AN IDENTITY MANAGEMENT SERVER: WITH INTEGRATED DNS, WITH AN INTEGRATED CA</b>	<b>38</b>
6.1. INTERACTIVE INSTALLATION	38
Procedure	38
6.2. NON-INTERACTIVE INSTALLATION	40
Procedure	40
Additional resources	41
<b>CHAPTER 7. INSTALLING AN IDENTITY MANAGEMENT SERVER: WITH INTEGRATED DNS, WITH AN EXTERNAL CA</b>	<b>42</b>
7.1. INTERACTIVE INSTALLATION	42
Prerequisites	42
Procedure	42
7.2. TROUBLESHOOTING: EXTERNAL CA INSTALLATION FAILS	45
What this means:	45
To fix the problem:	45
<b>CHAPTER 8. INSTALLING AN IDENTITY MANAGEMENT SERVER: WITH INTEGRATED DNS, WITHOUT A CA</b>	<b>47</b>
8.1. CERTIFICATES REQUIRED TO INSTALL AN IDENTITY MANAGEMENT SERVER WITHOUT A CA	47
Additional resources	48
8.2. INTERACTIVE INSTALLATION	48
Procedure	48
<b>CHAPTER 9. INSTALLING AN IDENTITY MANAGEMENT SERVER: WITHOUT INTEGRATED DNS, WITH AN INTEGRATED CA</b>	<b>51</b>
9.1. INTERACTIVE INSTALLATION	51
Procedure	51
9.2. NON-INTERACTIVE INSTALLATION	52
Procedure	52
Additional resources	52
<b>CHAPTER 10. UNINSTALLING AN IDENTITY MANAGEMENT SERVER</b>	<b>53</b>
Prerequisites	53
Procedure	53
<b>CHAPTER 11. RENAMING AN IDENTITY MANAGEMENT SERVER</b>	<b>54</b>
Procedure	54
<b>CHAPTER 12. PREPARING THE SYSTEM FOR IDENTITY MANAGEMENT CLIENT INSTALLATION</b>	<b>55</b>
12.1. DNS REQUIREMENTS FOR IDENTITY MANAGEMENT CLIENTS	55
Additional resources	55
12.2. PORT REQUIREMENTS FOR IDENTITY MANAGEMENT CLIENTS	55
Additional resources	55
12.3. PACKAGES REQUIRED TO INSTALL AN IDENTITY MANAGEMENT CLIENT	55
<b>CHAPTER 13. INSTALLING AN IDENTITY MANAGEMENT CLIENT: BASIC SCENARIO</b>	<b>57</b>
13.1. PREREQUISITES	57
13.2. AN OVERVIEW OF THE IDENTITY MANAGEMENT CLIENT INSTALLATION OPTIONS	57

Additional resources	57
13.3. INSTALLING A CLIENT BY USING USER CREDENTIALS: INTERACTIVE INSTALLATION	58
Prerequisites	58
Procedure	58
Additional resources	59
13.4. INSTALLING A CLIENT BY USING A ONE-TIME PASSWORD: INTERACTIVE INSTALLATION	59
Prerequisites	59
Procedure	60
Additional resources	61
13.5. INSTALLING A CLIENT: NON-INTERACTIVE INSTALLATION	61
Additional resources	62
13.6. REMOVING PRE-IDENTITY MANAGEMENT CONFIGURATION AFTER INSTALLING A CLIENT	62
13.7. TESTING AN IDENTITY MANAGEMENT CLIENT	63
13.8. CONNECTIONS PERFORMED DURING AN IDENTITY MANAGEMENT CLIENT INSTALLATION	63
13.9. IDENTITY MANAGEMENT CLIENT'S COMMUNICATIONS WITH THE SERVER DURING POST- INSTALLATION DEPLOYMENT	64
13.9.1. SSSD communication patterns	64
13.9.2. Certmonger communication patterns	66
<b>CHAPTER 14. INSTALLING AN IDENTITY MANAGEMENT CLIENT WITH KICKSTART</b>	<b>67</b>
14.1. INSTALLING A CLIENT WITH KICKSTART	67
Prerequisites	67
Procedure	67
14.2. KICKSTART FILE FOR CLIENT INSTALLATION	67
14.3. TESTING AN IDENTITY MANAGEMENT CLIENT	68
<b>CHAPTER 15. RE-ENROLLING AN IDENTITY MANAGEMENT CLIENT</b>	<b>69</b>
15.1. WHAT HAPPENS DURING CLIENT RE-ENROLLMENT	69
15.2. RE-ENROLLING A CLIENT BY USING USER CREDENTIALS: INTERACTIVE RE-ENROLLMENT	69
Additional resources	70
15.3. RE-ENROLLING A CLIENT BY USING THE CLIENT KEYTAB: NON-INTERACTIVE RE-ENROLLMENT	70
Prerequisites	70
Procedure	70
15.4. TESTING AN IDENTITY MANAGEMENT CLIENT	70
<b>CHAPTER 16. UNINSTALLING AN IDENTITY MANAGEMENT CLIENT</b>	<b>72</b>
16.1. UNINSTALLING AN IDENTITY MANAGEMENT CLIENT	72
Procedure	72
<b>CHAPTER 17. RENAMING IDENTITY MANAGEMENT CLIENT SYSTEMS</b>	<b>73</b>
17.1. PREREQUISITES	73
17.2. UNINSTALLING AN IDENTITY MANAGEMENT CLIENT	74
Procedure	74
17.3. RENAMING THE HOST SYSTEM	74
17.4. RE-INSTALLING AN IDENTITY MANAGEMENT CLIENT	74
17.5. RE-ADDING SERVICES, RE-GENERATING CERTIFICATES, AND RE-ADDING HOST GROUPS	74
<b>CHAPTER 18. PREPARING THE SYSTEM FOR IDENTITY MANAGEMENT REPLICA INSTALLATION</b>	<b>76</b>
18.1. REPLICA VERSION REQUIREMENTS	76
<b>CHAPTER 19. INSTALLING AN IDENTITY MANAGEMENT REPLICA</b>	<b>77</b>
19.1. PREREQUISITES FOR INSTALLING A REPLICA ON AN IDENTITY MANAGEMENT CLIENT	77
19.2. PREREQUISITES FOR INSTALLING A REPLICA ON A SYSTEM OUTSIDE THE IDENTITY MANAGEMENT DOMAIN	78

19.3. INSTALLING AN IDENTITY MANAGEMENT REPLICA WITH INTEGRATED DNS	79
Procedure	79
19.4. INSTALLING AN IDENTITY MANAGEMENT REPLICA WITH A CA	79
Procedure	80
19.5. INSTALLING AN IDENTITY MANAGEMENT REPLICA WITHOUT A CA	80
Procedure	80
19.6. TESTING AN IDENTITY MANAGEMENT REPLICA	81
Procedure	81
19.7. CONNECTIONS PERFORMED DURING AN IDENTITY MANAGEMENT REPLICA INSTALLATION	81
<b>CHAPTER 20. UNINSTALLING AN IDENTITY MANAGEMENT REPLICA</b>	<b>83</b>
Prerequisites	83
Procedure	83
<b>PART III. MANAGING IDENTITY MANAGEMENT</b>	<b>84</b>
<b>CHAPTER 21. CONFIGURING IDENTITY MANAGEMENT FOR AUTHENTICATING WITH A CERTIFICATE</b>	<b>85</b>
21.1. CONFIGURING THE IDENTITY MANAGEMENT SERVER FOR CERTIFICATE AUTHENTICATION IN THE WEB UI	85
Procedure	85
21.2. REQUESTING A NEW USER CERTIFICATE AND EXPORTING IT TO THE CLIENT	86
Procedure	86
21.3. MAKING SURE THE CERTIFICATE AND USER ARE LINKED TOGETHER	88
21.4. CONFIGURING A BROWSER TO ENABLE CERTIFICATE AUTHENTICATION	88
Procedure	88
21.5. AUTHENTICATING TO THE IDENTITY MANAGEMENT WEB UI WITH A CERTIFICATE AS AN IDENTITY MANAGEMENT USER	91
Procedure	91
Additional Resources	92
21.6. CONFIGURING AN IDM CLIENT TO ENABLE AUTHENTICATING TO THE CLI USING A CERTIFICATE	92
Procedure	92
<b>CHAPTER 22. ENABLING AD USERS TO ADMINISTER IDM</b>	<b>93</b>
22.1. ID OVERRIDES FOR AD USERS	93
22.2. USING ID OVERRIDES TO ENABLE AD USERS TO ADMINISTER IDM	93
PREREQUISITES	93
PROCEDURE	93
22.3. MANAGING IDM COMMAND-LINE INTERFACE (CLI) AS AN AD USER	94
<b>PART IV. CONFIGURING AUTHENTICATION ON A RED HAT ENTERPRISE LINUX HOST</b>	<b>95</b>
<b>CHAPTER 23. USING AUTHSELECT</b>	<b>96</b>
23.1. EXPLAINING AUTHSELECT	96
23.2. CHOOSING AN AUTHSELECT PROFILE	97
Procedure	97
23.3. MODIFYING A READY-MADE AUTHSELECT PROFILE	98
Procedure	99
23.4. CREATING AND DEPLOYING YOUR OWN CUSTOM AUTHSELECT PROFILE	99
Procedure	99
Example	100
23.5. CONVERTING YOUR SCRIPTS FROM AUTHCONFIG TO AUTHSELECT	100
<b>PART V. STARTING TO USE THE SESSION RECORDING SOLUTION</b>	<b>103</b>
<b>CHAPTER 24. GETTING STARTED WITH SESSION RECORDING ON RED HAT ENTERPRISE LINUX</b>	<b>104</b>



24.1. SESSION RECORDING IN RED HAT ENTERPRISE LINUX	104
24.2. COMPONENTS OF SESSION RECORDING	104
<b>CHAPTER 25. DEPLOYING SESSION RECORDING ON RED HAT ENTERPRISE LINUX .....</b>	<b>105</b>
25.1. INSTALLING TLOG	105
25.2. INSTALLING COCKPIT-SESSION-RECORDING	105
25.3. CONFIGURING THE RECORDED USERS OR USER GROUPS WITH SSSD FROM THE CLI	105
25.4. CONFIGURING THE RECORDED USERS OR USER GROUPS WITH SSSD FROM WEB UI	106
25.5. CONFIGURING RECORDED USERS OR USER GROUPS WITHOUT SSSD	107
<b>CHAPTER 26. PLAYING BACK RECORDED SESSIONS .....</b>	<b>108</b>
26.1. PLAYBACK WITH COCKPIT	108
26.2. PLAYBACK WITH TLOG-PLAY	108
26.3. PLAYING BACK RECORDED SESSIONS WITH TLOG-PLAY	108



## THIS IS A BETA VERSION!

Thank you for your interest in Red Hat Enterprise Linux 8.0 Beta. Be aware that:

- Beta code should not be used with production data or on production systems.
- Beta does not include a guarantee of support.
- Feedback and bug reports are welcome. Discussions with your account representative, partner contact, and Technical Account Manager (TAM) are also welcome.
- Upgrades to or from a Beta are not supported or recommended.

## 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, make sure you are viewing the documentation in the Multi-page HTML format. Highlight the part of text that you want to comment on. Then, click the **Add Feedback** pop-up that appears below the highlighted text, and 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. PLANNING IDENTITY MANAGEMENT AND ACCESS CONTROL**

# CHAPTER 1. OVERVIEW OF PLANNING FOR IDENTITY MANAGEMENT AND ACCESS CONTROL IN RED HAT ENTERPRISE LINUX

The following sections provide an overview of the options for identity management and access control in Red Hat Enterprise Linux. After reading these sections, you will be able to approach the planning stage for your environment.

## 1.1. INTRODUCTION TO IDENTITY MANAGEMENT

This module explains the purpose of Identity Management in Red Hat Enterprise Linux. It also provides basic information about the Identity Management domain, including the client and server machines that are part of the domain.

### The goal of Identity Management in Red Hat Enterprise Linux

Identity Management in Red Hat Enterprise Linux (IdM) provides a centralized and unified way to manage identity stores, authentication, policies, and authorization policies in a Linux-based domain. IdM significantly reduces the administrative overhead of managing different services individually and using different tools on different machines.

IdM is one of the few centralized identity, policy, and authorization software solutions that support:

- Advanced features of Linux operating system environments
- Unifying large groups of Linux machines
- Native integration with Active Directory

IdM creates a Linux-based and Linux-controlled domain:

- IdM builds on existing, native Linux tools and protocols. It has its own processes and configuration, but its underlying technologies are well-established on Linux systems and trusted by Linux administrators.
- IdM servers and clients are Red Hat Enterprise Linux machines. IdM clients can also be other Linux and UNIX distributions if they support standard protocols. Windows client cannot be a member of the IdM domain but user logged into Windows systems managed by Active Directory (AD) can connect to Linux clients or access services managed by IdM. This is accomplished by establishing cross forest trust between AD and IdM domains.

### Managing identities and policies on multiple Linux servers

*Without IdM:* Each server is administered separately. All passwords are saved on the local machines. The IT administrator manages users on every machine, sets authentication and authorization policies separately, and maintains local passwords. However, more often the users rely on other centralized solution, for example direct integration with Active Directory (AD). Systems can be directly integrated with AD using several different solutions: \* Legacy Linux tools (not recommended to use) \* Solution based on Samba winbind (recommended for specific use cases) \* Solution based on a third-party software (usually require a license from another vendor) \* Solution based on SSSD (native Linux and recommended for the majority of use cases)

*With IdM:* The IT administrator can:

- Maintain the identities in one central place: the IdM server

- Apply policies uniformly to multiples of machines at the same time
- Set different access levels for users by using host-based access control, delegation, and other rules
- Centrally manage privilege escalation rules
- Define how home directories are mounted

## Enterprise single sign-on

In case of Identity Management Enterprise, SSO (single sign-on) is implemented leveraging the Kerberos protocol. This protocol is popular in the infrastructure level and enables SSO with services such as SSH, LDAP, NFS, CUPS, or DNS. Web services using different web stacks (Apache, EAP, Django, and others) can also be enabled to use Kerberos for SSO. However, practice shows that using OpenID Connect or SAML based on SSO is more convenient for web applications. To bridge the two layers, it is recommended to deploy an Identity Provider solution (IdP) that would be able to convert Kerberos authentication into a OpenID Connect ticket or SAML assertion. Red Hat SSO technology based on the Keycloak open source project is an example of such IdP

*Without IdM:* Users log in to the system and are prompted for a password every single time they access a service or application. These passwords might be different, and the users have to remember which credential to use for which application.

*With IdM:* After users log in to the system, they can access multiple services and applications without being repeatedly asked for their credentials. This helps to:

- Improve usability
- Reduce the security risk of passwords being written down or stored insecurely
- Boost user productivity

## Managing a mixed Linux and Windows environment

*Without IdM:* Windows systems are managed in an Active Directory forest, but development, production, and other teams have many Linux systems. The Linux systems are excluded from the Active Directory environment.

*With IdM:* The IT administrator can:

- Manage the Linux systems using native Linux tools
- Integrate the Linux systems into the environments centrally managed by Active Directory, thus preserving a centralized user store.
- Easily deploy new Linux systems at scale or as needed.
- Quickly react to business needs and make decisions related to management of the Linux infrastructure without dependency on other teams avoiding delays.

## Contrasting Identity Management with a Standard LDAP Directory

A standard LDAP directory, such as Red Hat Directory Server, is a general-purpose directory: it can be customized to fit a broad range of use cases.

- Schema: a flexible schema that can be customized for a vast array of entries, such as users, machines, network entities, physical equipment, or buildings.

- Typically used as: a back-end directory to store data for other applications, such as business applications that provide services on the Internet.

Identity Management (IdM) has a specific purpose: managing internal, inside-the-enterprise identities as well as authentication and authorization policies that relate to these identities.

- Schema: a specific schema that defines a particular set of entries relevant to its purpose, such as entries for user or machine identities.
- Typically used as: the identity and authentication server to manage identities within the boundaries of an enterprise or a project.

The underlying directory server technology is the same for both Red Hat Directory Server and IdM. However, IdM is optimized to manage identities inside the enterprise. This limits its general extensibility, but also brings certain benefits: simpler configuration, better automation of resource management, and increased efficiency in managing enterprise identities.

### Additional Resources

- [Identity Management or Red Hat Directory Server – Which One Should I Use?](#) on the Red Hat Enterprise Linux Blog.
- Knowledge Base article about [Standard protocols](#).
- Red Hat Enterprise Linux 8 Beta Release Notes

## 1.2. INTRODUCTION TO IDENTITY MANAGEMENT SERVERS AND CLIENTS

The Identity Management domain includes the following types of systems:

### Identity Management servers

Identity Management servers are Red Hat Enterprise Linux systems that work as domain controllers (DCs). In most deployments, an integrated certificate authority (CA) is also installed with the IdM server.

Servers are the central repositories for identity and policy information. They also host the services used by domain members.

### Identity Management clients

Identity Management clients are Red Hat Enterprise Linux systems enrolled with the servers and configured to use the Identity Management services on these servers.

Clients interact with the Identity Management servers to access services provided by them. For example, clients use the Kerberos protocol to perform authentication and acquire tickets for enterprise SSO, use LDAP to get identity and policy information, use DNS to detect where the servers and services are located and how to connect to them.

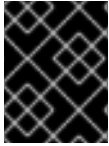
Identity Management servers are also embedded Identity Management clients. As clients enrolled with themselves, the servers provide the same functionality as other clients.

To provide services for large numbers of clients, as well as for redundancy and availability, Identity Management allows deployment on multiple IdM servers in a single domain. It is possible to deploy up to 60 servers. This is the maximum number of IdM servers, also called replicas, that is currently supported in the IdM domain. Identity Management servers provide different services for the client. Not all the servers need to provide all the possible services. Some server components like



Kerberos and LDAP are always available on every server. Other services like Certificate authority (CA), DNS, Trust Controller or Vault are optional. This means that different servers in general play different roles in the deployment.

The first server installed to create the domain is the *master server*. If your Identity Management topology contains an integrated Certificate Authority (CA), this server is the *CRL generation master* and the *CA renewal master*: the only system in the domain responsible for tracking CA subsystem certificates and keys and for generating the certificate revocation list (CRL).



### IMPORTANT

The CRL generation master role is critical because it is performed by only one server in the topology.

For redundancy and load balancing, administrators create additional servers by creating a *replica* of any existing server, either the master server or another replica. When creating a replica, Identity Management clones the configuration of the existing server. A replica shares with the initial server its core configuration, including internal information about users, systems, certificates, and configured policies.



### NOTE

A replica and the server it was created from are functionally identical except for the role of the CRL generation master. Therefore, the term *server* and *replica* are used interchangeably here depending on the context.

## 1.3. IDENTITY MANAGEMENT AND ACCESS CONTROL IN RED HAT ENTERPRISE LINUX: CENTRAL VS. LOCAL

In Red Hat Enterprise Linux, you can manage identities and access control policies using centralized tools for a whole domain of systems, or using local tools for a single system.

### Managing identities and policies on multiple Red Hat Enterprise Linux servers: With and without Identity Management

With Identity Management, the IT administrator can:

- Maintain the identities and grouping mechanisms in one central place: the Identity Management server
- Centrally manage different types of credentials such as passwords, PKI certificates, OTP tokens, or SSH keys
- Apply policies uniformly to multiples of machines at the same time
- Manage POSIX and other attributes for external Active Directory users
- Set different access levels for users by using host-based access control, delegation, and other rules
- Centrally manage privilege escalation rules (sudo) and mandatory access control (SELinux user mapping)
- Maintain central PKI infrastructure and secrets store

- Define how home directories are mounted

Without Identity Management:

- Each server is administered separately.
- All passwords are saved on the local machines.
- The IT administrator manages users on every machine, sets authentication and authorization policies separately, and maintains local passwords.

## 1.4. ADDITIONAL RESOURCES

- For general information on Red Hat Identity Management, see the [Red Hat Identity Management product page](#) on the Red Hat Customer Portal.

## CHAPTER 2. PLANNING THE REPLICA TOPOLOGY

The following sections provide advice on determining the appropriate replica topology for your use case.

### 2.1. MULTIPLE REPLICA SERVERS AS A SOLUTION FOR HIGH PERFORMANCE AND DISASTER RECOVERY

Continuous functionality and high availability of Identity Management services is vital for users who access resources. One of the built-in solutions for accomplishing continuous functionality and high availability of the Identity Management infrastructure through load balancing is the replication of the central directory by creating replica servers of the master server.

Identity Management allows placing additional servers in geographically dispersed data centers to reflect your enterprise organizational structure. In this way, the path between Identity Management clients and the nearest accessible server is shortened. In addition, having multiple servers allows spreading the load and scaling for more clients.

Maintaining multiple redundant Identity Management servers and letting them replicate with each other is also a common backup mechanism to mitigate or prevent server loss. For example, if one server fails, the other servers keep providing services to the domain. You can also recover the lost server by creating a new replica based on one of the remaining servers.

### 2.2. IDENTITY MANAGEMENT SERVERS AND CLIENTS

The Identity Management domain includes the following types of systems:

#### Identity Management servers

Identity Management servers are Red Hat Enterprise Linux systems that work as domain controllers (DCs). In most deployments, an integrated certificate authority (CA) is also installed with the IdM server.

Servers are the central repositories for identity and policy information. They also host the services used by domain members.

Identity Management servers are also embedded Identity Management clients. As clients enrolled with themselves, the servers provide the same functionality as other clients.

#### Identity Management clients

Identity Management clients are Red Hat Enterprise Linux systems enrolled with the servers and configured to use the Identity Management services on these servers.

Clients interact with the Identity Management servers to access domain resources. For example, clients belong to the Kerberos domain configured on the servers, receive certificates and tickets issued by the servers, and use other centralized services for authentication and authorization.

The first server installed to create the domain is the *master server*. If your Identity Management topology contains an integrated Certificate Authority (CA), this server is the *CRL generation master* and the *CA\_renewal\_master*: the only system in the domain responsible for tracking CA subsystem certificates and keys and for generating the certificate revocation list (CRL).



#### IMPORTANT

The CRL generation master role is critical because it is performed by only one server in the topology.

For redundancy and load balancing, administrators create additional servers by creating a *replica* of any existing server, either the master server or another replica. When creating a replica, Identity Management clones the configuration of the existing server. A replica shares with the initial server its core configuration, including internal information about users, systems, certificates, and configured policies.



## NOTE

A replica and the server it was created from are functionally identical except for the role of the CRL generation master. Therefore, the term *server* and *replica* are used interchangeably here depending on the context.

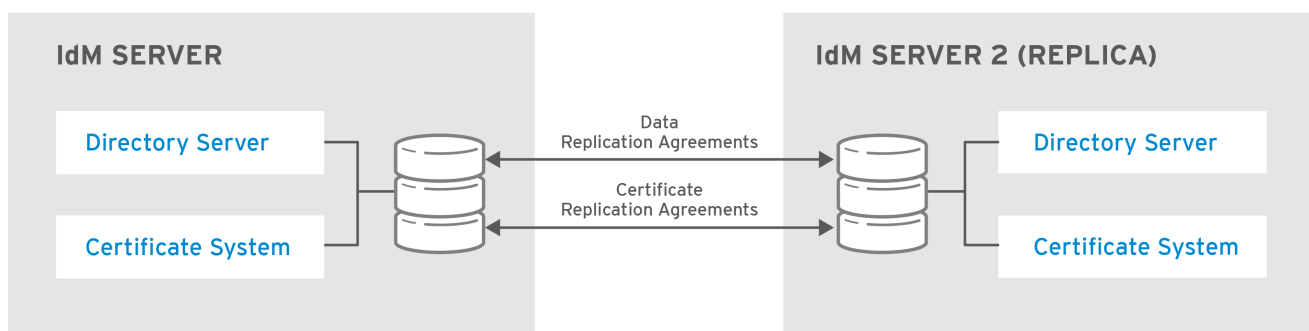
## 2.3. REPLICATION AGREEMENTS

When an administrator creates a replica based on an existing server, Identity Management 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.

Replication agreements are always bilateral: the data is replicated from one server to the other as well as from the other server to the first server.

Identity Management uses *multi-master replication*. In multi-master replication, all replicas joined in a replication agreement receive updates, and are therefore considered data masters.

**Figure 2.1. Server and replica agreements**



RHEL\_404973\_0516

Identity Management 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.

## 2.4. DETERMINING THE APPROPRIATE NUMBER OF REPLICAS

**Set up at least two replicas in each data center (not a hard requirement)**

A data center can be, for example, a main office or a geographical location.

### **Set up a sufficient number of servers to serve your clients**

One Identity Management server can provide services to 2000 - 3000 clients. This assumes the clients query the servers multiple times a day, but not, for example, every minute. If you expect more frequent queries, plan for more servers.

### **Set up a maximum of 60 replicas in a single Identity Management domain**

Red Hat guarantees to support environments with 60 replicas or fewer.

## **2.5. CONNECTING THE REPLICAS IN A TOPOLOGY**

### **Connect each replica to at least two other replicas**

Configuring additional replication agreements ensures that information is replicated not just between the initial replica and the master server, but between other replicas as well.

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

A large number of replication agreements per server does not bring significant additional benefits. One consumer replica can only be updated by one other replica at a time. Meanwhile, the other replication agreements are idle.

Configuring too many replication agreements can also have a negative impact on overall performance.

### **Connect the replicas in a data center with each other**

This ensures domain replication within the data center.

### **Connect each data center to at least two other data centers**

This ensures domain replication between data centers.

### **Connect data centers using at least a pair of replication agreements**

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

## **2.6. REPLICA TOPOLOGY EXAMPLES**

The figures below show examples of Identity Management topologies based on the guidelines for creating a reliable topology.

[Figure 2.2, “Replica Topology Example 1”](#) shows four data centers, each with four servers. The servers are connected with replication agreements.

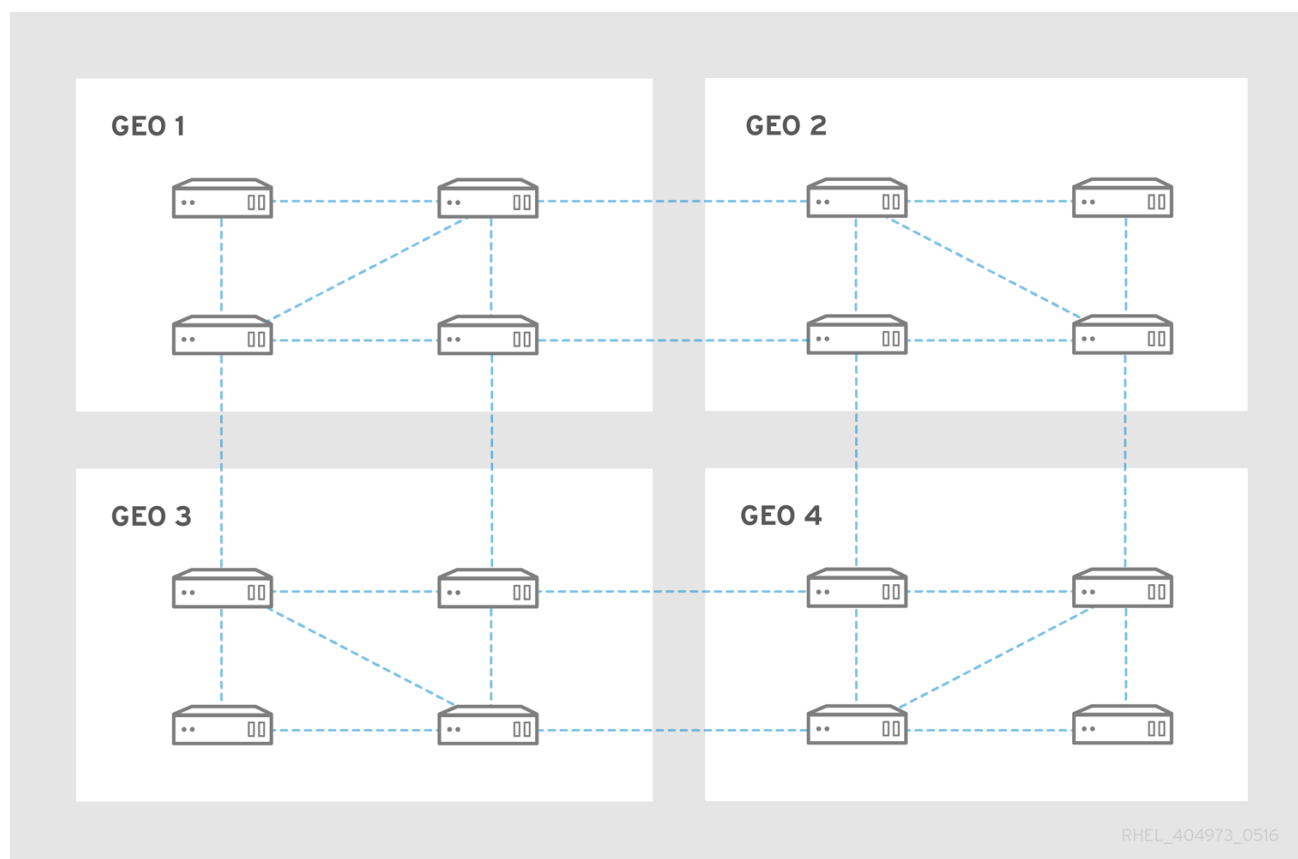
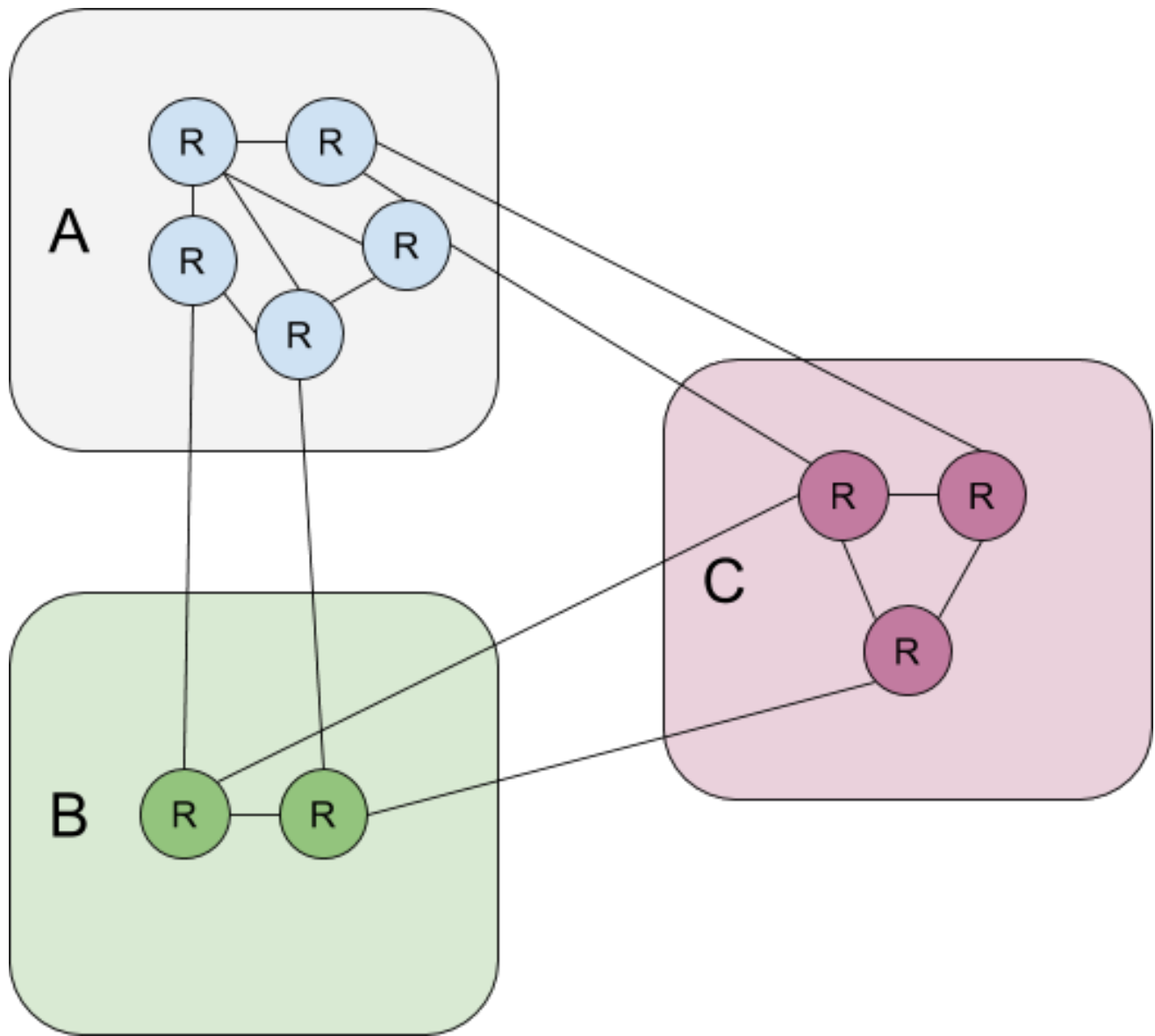
**Figure 2.2. Replica Topology Example 1**

Figure 2.3, “[Replica Topology Example 2](#)” shows three data centers, each with a different number of servers. The servers are connected with replication agreements.

Figure 2.3. Replica Topology Example 2



## CHAPTER 3. PLANNING INTEGRATION WITH ACTIVE DIRECTORY

The following sections introduce the options for integrating Red Hat Enterprise Linux with Active Directory.

- For an overview of direct integration, see [Section 3.1, “Direct integration”](#).
- For an overview of indirect integration, see [Section 3.2, “Indirect integration”](#).
- For advice on how to decide between them, see [Section 3.3, “Deciding between indirect and direct integration”](#).

### 3.1. DIRECT INTEGRATION

In direct integration, Linux systems are connected directly to Active Directory. The following types of integration are possible:

#### Integration with the System Security Services Daemon (SSSD)

SSSD can connect a Linux system with various identity and authentication stores: Active Directory, Identity Management, or a generic LDAP or Kerberos server.

Notable requirements for integration with SSSD:

- When integrating with Active Directory, SSSD works only within a single AD forest by default. For multi-forest setup, configure manual domain enumeration.
- Remote Active Directory forests must trust the local forest to ensure that the `idmap_ad` plug-in handles remote forest users correctly.

SSSD supports both direct and indirect integration. It also enables switching from one integration approach to the other without significant migration costs.

#### Integration with Samba Winbind

The Winbind component of the Samba suite emulates a Windows client on a Linux system and communicates with Active Directory servers.

Notable requirements for integration with Samba Winbind:

- Direct integration with Winbind in a multi-forest Active Directory setup requires bidirectional trusts.
- A bidirectional path from the local domain of a Linux system must exist to the domain of a user in a remote Active Directory forest to allow full information about the user from the remote Active Directory domain to be available to the `idmap_ad` plug-in.

#### Recommendations

- SSSD satisfies most of the use cases for AD integration and provides a robust solution as a generic gateway between a client system and different types of identity and authentication providers - AD, IdM, Kerberos, and LDAP.
- Winbind is recommended for deployment on those AD domain member servers on which you plan to deploy Samba FS.



## 3.2. INDIRECT INTEGRATION

In indirect integration, Linux systems are first connected to a central server which is then connected to Active Directory. Indirect integration enables the administrator to manage Linux systems and policies centrally, while users from Active Directory can transparently access Linux systems and services.

### Integration based on cross-forest trust with Active Directory

The Identity Management server acts as the central server to control Linux systems. A cross-realm Kerberos trust with Active Directory is established, enabling users from Active Directory to log on to access Linux systems and resources. Identity Management presents itself to Active Directory as a separate forest and takes advantage of the forest-level trusts supported by Active Directory.

When using a trust:

- Active Directory users can access Identity Management resources.
- Identity Management servers and clients can resolve the identities of Active Directory users and groups.
- Active Directory users and groups access Identity Management under the conditions defined by Identity Management, such as host-based access control.
- Active Directory users and groups continue being managed on the Active Directory side.

### Integration based on synchronization

This approach is based on the WinSync tool. A WinSync replication agreement synchronizes user accounts from Active Directory to Identity Management.



#### **WARNING**

WinSync is no longer actively developed in Red Hat Enterprise Linux 8. The preferred solution for indirect integration is cross-forest trust.

The limitations of integration based on synchronization include:

- Groups are not synchronized from Identity Management to Active Directory.
- Users are duplicated in Active Directory and Identity Management.
- WinSync supports only a single Active Directory domain.
- Only one domain controller in Active Directory can be used to synchronize data to one instance of Identity Management
- User passwords must be synchronized, which requires the PassSync component to be installed on all domain controllers in the Active Directory domain.
- After configuring the synchronization, all Active Directory users must manually change passwords before PassSync can synchronize them.

### 3.3. DECIDING BETWEEN INDIRECT AND DIRECT INTEGRATION

The guidelines in this section can help decide which type of integration fits your use case.

#### **Number of systems to be connected to Active Directory**

##### **Connecting less than 30-50 systems (not a hard limit)**

If you connect less than 30-50 systems, consider direct integration. Indirect integration might introduce unnecessary overhead.

##### **Connecting more than 30-50 systems (not a hard limit)**

If you connect more than 30-50 systems, consider indirect integration with Identity Management. With this approach, you can benefit from the centralized management for Linux systems.

##### **Managing a small number of Linux systems, but expecting the number to grow rapidly**

In this scenario, consider indirect integration to avoid having to migrate the environment later.

#### **Frequency of deploying new systems and their type**

##### **Deploying bare metal systems on an irregular basis**

If you deploy new systems rarely and they are usually bare metal systems, consider direct integration. In such cases, direct integration is usually simplest and easiest.

##### **Deploying virtual systems frequently**

If you deploy new systems often and they are usually virtual systems provisioned on demand, consider indirect integration. With indirect integration, you can use a central server to manage the new systems dynamically and integrate with orchestration tools, such as Red Hat Satellite.

#### **Active Directory is the required authentication provider**

##### **Do your internal policies state that all users must authenticate against Active Directory?**

You can choose either direct or indirect integration. If you use indirect integration with a trust between Identity Management and Active Directory, the users that access Linux systems authenticate against Active Directory. Policies that exist in Active Directory are executed and enforced during authentication.

## CHAPTER 4. PLANNING A CROSS-FOREST TRUST BETWEEN IDENTITY MANAGEMENT AND ACTIVE DIRECTORY

Active Directory and Identity Management are two alternative environments managing a variety of core services, such as Kerberos, LDAP, DNS, and certificate services. A *cross-forest trust* relationship transparently integrates these two diverse environments by enabling all core services to interact seamlessly. The following sections provide advice on how to plan and design a cross-forest trust deployment.

### 4.1. CROSS-FOREST TRUSTS BETWEEN IDENTITY MANAGEMENT AND ACTIVE DIRECTORY

In a pure Active Directory environment, a cross-forest trust connects two separate Active Directory forest root domains. When you create a cross-forest trust between Active Directory and Identity Management, the Identity Management domain presents itself to Active Directory as a separate forest with a single domain. A trust relationship is then established between the Active Directory forest root domain and the Identity Management domain. As a result, users from the Active Directory forest can access the resources in the Identity Management domain.

Identity Management can establish a trust with one Active Directory forest or multiple unrelated forests.



#### NOTE

Two separate Kerberos realms can be connected in a *cross-realm trust*. However, a Kerberos realm only concerns authentication, not other services and protocols involved in identity and authorization operations. Therefore, establishing a Kerberos cross-realm trust is not enough to enable users from one realm to access resources in another realm.

#### An external trust to an Active Directory domain

An external trust is a trust relationship between Identity Management and an Active Directory domain. While a forest trust always requires establishing a trust between Identity Management and the root domain of an Active Directory forest, an external trust can be established from Identity Management to any domain within a forest.

### 4.2. TRUST CONTROLLERS AND TRUST AGENTS

Identity Management provides the following types of Identity Management servers that support trust to Active Directory:

#### Trust agents

Identity Management servers that can perform identity lookups against Active Directory domain controllers.

#### Trust controllers

Trust agents that also run the Samba suite. Active Directory domain controllers contact trust controllers when establishing and verifying the trust to Active Directory.

The first trust controller is created when you configure the trust.

Trust controllers run more network-facing services than trust agents, and thus present a greater attack surface for potential intruders.

In addition to trust agents and controllers, the Identity Management domain can also include standard

Identity Management servers. However, these servers do not communicate with Active Directory. Therefore, clients that communicate with the standard servers cannot resolve Active Directory users and groups or authenticate and authorize Active Directory users.

**Table 4.1. Comparing the capabilities supported by trust controllers and trust agents**

Capability	Trust agent	Trust controller
Resolve Active Directory users and groups	Yes	Yes
Enroll Identity Management clients that run services accessible by users from trusted Active Directory forests	Yes	Yes
Manage the trust (for example, add trust agreements)	No	Yes

When planning the deployment of trust controllers and trust agents, consider these guidelines:

- Configure at least two trust controllers per Identity Management deployment.
- Configure at least two trust controllers in each data center.

If you ever want to create additional trust controllers or if an existing trust controller fails, create a new trust controller by promoting a trust agent or a standard server. To do this, use the **ipa-adtrust-install** utility on the Identity Management server.



### IMPORTANT

You cannot downgrade an existing trust controller to a trust agent.

## 4.3. ONE-WAY TRUSTS AND TWO-WAY TRUSTS

In one way trusts, Identity Management (IdM) trusts Active Directory (AD) but AD does not trust IdM. AD users can access resources in the IdM domain but users from IdM cannot access resources within the AD domain. The IdM server connects to AD using a special account, and reads identity information that is then delivered to IdM clients over LDAP.

In two way trusts, IdM users can authenticate to AD, and AD users can authenticate to IdM. AD users can authenticate to and access resources in the IdM domain as in the one way trust case. IdM users can authenticate but cannot access most of the resources in AD. They can only access those Kerberized services in AD forests that do not require any access control check.

To be able to grant access to the AD resources, IdM needs to implement the Global Catalog service. This service does not yet exist in the current version of the IdM server. Because of that, a two-way trust between IdM and AD is nearly functionally equivalent to a one-way trust between IdM and AD.

## 4.4. NON-POSIX EXTERNAL GROUPS AND SECURITY ID MAPPING

Identity Management uses LDAP for managing groups. Active Directory entries are not synchronized or copied over to Identity Management, which means that Active Directory users and groups have no LDAP objects in the LDAP server, so they cannot be directly used to express group membership in the Identity Management LDAP. For this reason, administrators in Identity Management need to create non-POSIX external groups, referenced as normal Identity Management LDAP objects to signify group membership for Active Directory users and groups in Identity Management.

Security IDs (SIDs) for non-POSIX external groups are processed by SSSD, which maps the SIDs of groups in Active Directory to POSIX groups in Identity Management. In Active Directory, SIDs are associated with user names. When an Active Directory user name is used to access Identity Management resources, SSSD uses the user's SID to build up a full group membership information for the user in the Identity Management domain.

## 4.5. SETTING UP DNS

These guidelines can help you achieve the right DNS configuration for establishing a cross-forest trust between Identity Management and Active Directory.

### Unique primary DNS domains

Ensure both Active Directory and Identity Management have their own unique primary DNS domains configured. For example:

- **ad.example.com** for Active Directory and **idm.example.com** for Identity Management
- **example.com** for Active Directory and **idm.example.com** for Identity Management

The most convenient management solution is an environment where each DNS domain is managed by integrated DNS servers, but you can also use any other standard-compliant DNS server.

### No overlap between Identity Management and Active Directory DNS Domains

Systems joined to Identity Management can be distributed over multiple DNS domains. Ensure the DNS domains that contain Identity Management clients do not overlap with DNS domains that contain systems joined to Active Directory.

### Proper SRV records

Ensure the primary Identity Management DNS domain has proper SRV records to support Active Directory trusts.

For other DNS domains that are part of the same Identity Management realm, the SRV records do not have to be configured when the trust to Active Directory is established. The reason is that Active Directory domain controllers do not use SRV records to discover Kerberos key distribution centers (KDCs) but rather base the KDC discovery on name suffix routing information for the trust.

### DNS records resolvable from all DNS domains in the trust

Ensure all machines can resolve DNS records from all DNS domains involved in the trust relationship:

- When configuring the Identity Management DNS, follow the instructions described in [Chapter 7, \*Installing an Identity Management server: With integrated DNS, with an external CA\*](#).
- If you are using Identity Management without integrated DNS, follow the instructions described in [Chapter 9, \*Installing an Identity Management server: Without integrated DNS, with an integrated CA\*](#).

### Kerberos realm names as upper-case versions of primary DNS domain names

Ensure Kerberos realm names are the same as the primary DNS domain names, with all letters uppercase. For example, if the domain names are **ad.example.com** for Active Directory and **idm.example.com** for Identity Management, the Kerberos realm names must be **AD.EXAMPLE.COM** and **IDM.EXAMPLE.COM**.

## 4.6. NETBIOS NAMES

The NetBIOS name is usually the far-left component of the domain name. For example:

- In the domain name **linux.example.com**, the NetBIOS name is **linux**.
- In the domain name **example.com**, the NetBIOS name is **example**.

#### **Different NetBIOS names for the Identity Management and Active Directory domains**

Ensure the Identity Management and Active Directory domains have different NetBIOS names.

The NetBIOS name is critical for identifying the Active Directory domain. If the Identity Management domain is within a subdomain of the Active Directory DNS, the NetBIOS name is also critical for identifying the Identity Management domain and services.

#### **Character limit for NetBIOS names**

The maximum length of a NetBIOS name is 15 characters.

## **4.7. CONFIGURING ACTIVE DIRECTORY SERVER DISCOVERY AND AFFINITY**

Server discovery and affinity configuration affects which Active Directory servers an Identity Management client communicates with. This section provides an overview of how discovery and affinity work in an environment with a cross-forest trust between Identity Management and Active Directory.

Configuring clients to prefer servers in the same geographical location helps prevent time lags and other problems that occur when clients contact servers from another, remote datacenter. To make sure clients communicate with local servers, you must ensure that:

- Clients communicate with local Identity Management servers over LDAP and over Kerberos
- Clients communicate with local Active Directory servers over Kerberos
- Embedded clients on Identity Management servers communicate with local Active Directory servers over LDAP and over Kerberos

### **Options for configuring LDAP and Kerberos on the Identity Management client for communication with local Identity Management servers**

#### **When using Identity Management with integrated DNS**

By default, clients use automatic service lookup based on the DNS records. In this setup, you can also use the *DNS locations* feature to configure DNS-based service discovery.

To override the automatic lookup, you can disable the DNS discovery in one of the following ways:

- During the Identity Management client installation by providing failover parameters from the command line
- After the client installation by modifying the System Security Services Daemon configuration

#### **When using Identity Management without integrated DNS**

You must explicitly configure clients in one of the following ways:

- During the Identity Management client installation by providing failover parameters from the command line
- After the client installation by modifying the System Security Services Daemon configuration

## Options for configuring Kerberos on the Identity Management client for communication with local Active Directory servers

Identity Management clients are unable to automatically discover which Active Directory servers to communicate with. To specify the Active Directory servers manually, modify the **krb5.conf** file:

- Add the Active Directory realm information
- Explicitly list the Active Directory servers to communicate with

For example:

```
[realms]
AD.EXAMPLE.COM = {
  kdc = server1.ad.example.com
  kdc = server2.ad.example.com
}
```

## Options for configuring embedded clients on Identity Management servers for communication with local Active Directory servers over Kerberos and LDAP

The embedded client on an Identity Management server works also as a client of the Active Directory server. It can automatically discover and use the appropriate Active Directory site.

When the embedded client performs the discovery, it might first discover an Active Directory server in a remote location. If the attempt to contact the remote server takes too long, the client might stop the operation without establishing the connection. Use the **dns\_resolver\_timeout** option in the **sssd.conf** file on the client to increase the amount of time for which the client waits for a reply from the DNS resolver. See the *sssd.conf(5)* man page for details.

Once the embedded client has been configured to communicate with the local Active Directory servers, the System Security Services Daemon (SSSD) remembers the Active Directory site the embedded client belongs to. Thanks to this, SSSD normally sends an LDAP ping directly to a local domain controller to refresh its site information. If the site no longer exists or the client has meanwhile been assigned to a different site, SSSD starts querying for SRV records in the forest and goes through a whole process of autodiscovery.

Using *trusted domain sections* in **sssd.conf**, you can also explicitly override some of the information that is discovered automatically by default.

## 4.8. OPERATIONS PERFORMED DURING INDIRECT INTEGRATION OF IDENTITY MANAGEMENT TO ACTIVE DIRECTORY

Table 4.2, “Operations performed from an Identity Management trust controller towards Active Directory domain controllers” shows which operations and requests are performed during the creation of an Identity Management to Active Directory trust from the Identity Management trust controller towards Active Directory domain controllers.

**Table 4.2. Operations performed from an Identity Management trust controller towards Active Directory domain controllers**

Operation	Protocol used	Purpose
DNS resolution against the Active Directory DNS resolvers configured on an Identity Management trust controller	DNS	To discover the IP addresses of Active Directory domain controllers
Requests to UDP/UDP6 port 389 on an Active Directory DC	Connectionless LDAP (CLDAP)	To perform Active Directory DC discovery
Requests to TCP/TCP6 ports 389 and 3268 on an Active Directory DC	LDAP	To query Active Directory user and group information
Requests to TCP/TCP6 ports 389 and 3268 on an Active Directory DC	DCE RPC and SMB	To set up and support cross-forest trust to Active Directory
Requests to TCP/TCP6 ports 135, 139, 445 on an Active Directory DC	DCE RPC and SMB	To set up and support cross-forest trust to Active Directory
Requests to dynamically opened ports on an Active Directory DC as directed by the Active Directory domain controller, likely in the range of 49152-65535 (TCP/TCP6)	DCE RPC and SMB	To respond to requests by DCE RPC End-point mapper (port 135 TCP/TCP6)
Requests to ports 88 (TCP/TCP6 and UDP/UDP6), 464 (TCP/TCP6 and UDP/UDP6), and 749 (TCP/TCP6) on an Active Directory DC	Kerberos	To obtain a Kerberos ticket; change a Kerberos password; administer Kerberos remotely

Table 4.3, “Operations performed from an Active Directory domain controller towards Identity Management trust controllers” shows which operations and requests are performed during the creation of an Identity Management to Active Directory trust from the Active Directory domain controller towards Identity Management trust controllers.

**Table 4.3. Operations performed from an Active Directory domain controller towards Identity Management trust controllers**

Operation	Protocol used	Purpose
DNS resolution against the Identity Management DNS resolvers configured on an Active Directory domain controller	DNS	To discover the IP addresses of Identity Management trust controllers
Requests to UDP/UDP6 port 389 on an Identity Management trust controller	Connectionless LDAP (CLDAP)	To perform Identity Management trust controller discovery



Operation	Protocol used	Purpose
Requests to TCP/TCP6 ports 135, 139, 445 on an Identity Management trust controller	DCE RPC and SMB	To verify the cross-forest trust to Active Directory
Requests to dynamically opened ports on an Identity Management trust controller as directed by the Identity Management trust controller, likely in the range of 49152-65535 (TCP/TCP6)	DCE RPC and SMB	To respond to requests by DCE RPC End-point mapper (port 135 TCP/TCP6)
Requests to ports 88 (TCP/TCP6 and UDP/UDP6), 464 (TCP/TCP6 and UDP/UDP6), and 749 (TCP/TCP6) on an Identity Management trust controller	Kerberos	To obtain a Kerberos ticket; change a Kerberos password; administer Kerberos remotely

## PART II. INSTALLING IDENTITY MANAGEMENT

## CHAPTER 5. PREPARING THE SYSTEM FOR IDENTITY MANAGEMENT SERVER INSTALLATION

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

### 5.1. 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 3 GB of RAM and 1 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.



#### NOTE

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

### 5.2. CUSTOM CONFIGURATION REQUIREMENTS FOR IDENTITY MANAGEMENT

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

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

#### 5.2.1. IPv6 requirements in Identity Management

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.

### 5.3. HOST NAME AND DNS REQUIREMENTS FOR IDENTITY MANAGEMENT

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 servers, those with integrated DNS and those without integrated DNS.



## WARNING

DNS records are vital for nearly all Identity Management 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 Identity Management servers with **and** without integrated DNS.

### Verify the server host name

The host name must be a fully qualified domain name, such as **server.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.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. 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
...
```

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

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

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

```
[root@server ~]# dig +short server.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.

2. 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, even though **dig +short server\_host\_name** in the previous step returned an IP address, it indicates that 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.example.com
```

- b. If the command **dig +short -x server.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.example.com
```



#### NOTE

If **dig +short server.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.

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

Ensure that all DNS forwarders you want to use with the Identity Management 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****IMPORTANT**

Do not modify the **/etc/hosts** file manually. If **/etc/hosts** has been modified manually before, make sure its contents conform to the following rules.

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

- It properly lists the IPv4 and IPv6 localhost entries for the host.
- These entries are followed by the Identity Management server IP address and host name as the first entry.
- Note that the Identity Management server host name cannot be part of the **localhost** entry.

```
127.0.0.1 localhost.localdomain localhost
::1 localhost6.localdomain6 localhost6
192.0.2.1 server.example.com server
```

```
2001:DB8::1111 server.example.com server
```

## 5.4. PORT REQUIREMENTS FOR IDENTITY MANAGEMENT

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

**Table 5.1. Identity Management 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 5.2. firewallld 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>

### Opening the required ports

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
```

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

- 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}
```

- 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.

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

- 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.

## 5.5. PACKAGES REQUIRED FOR AN IDENTITY MANAGEMENT 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 Identity Management environment of your choice.

- Enable the **idm:DL1** stream:

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

- 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 will have 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/default
```

or

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



## NOTE

The IdM client packages installed from the **idm:DL1** stream are likely to be of a lower version than the IdM client packages installed from the default **idm:client** 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 Identity Management, too.

## CHAPTER 6. INSTALLING AN IDENTITY MANAGEMENT SERVER: WITH INTEGRATED DNS, WITH AN INTEGRATED CA

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

- You can automate much of the maintenance and DNS record management using native Identity Management 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 Identity Management 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 Identity Management domain.

Installing Identity Management with integrated DNS has certain limitations:

- Identity Management 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 Identity Management server with an integrated 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 Identity Management server is installed with an integrated CA.

## 6.1. 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 administration system user account (**admin**).

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

5. The script prompts for DNS forwarders.

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

- To configure DNS forwarders, enter **yes**, and then follow the instructions on the command line. The installation process will add the forwarder IP addresses to the **/etc/named.conf** file on the installed Identity Management server.
  - 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, your environment will be isolated, and names from other DNS domains in your infrastructure will not be resolved.

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.
```



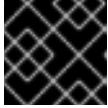
### NOTE

Using Identity Management 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, add DNS delegation from the parent domain to the Identity Management DNS domain. For example, if the Identity Management DNS domain is **ipa.example.com**, add a name server (NS) record to the **example.com** parent domain.



### IMPORTANT

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

## 6.2. NON-INTERACTIVE INSTALLATION



### 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 Identity Management administrator
  - **--unattended** to let the installation process select default options for the host name and domain name

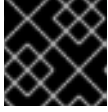
To install a server with integrated DNS, add also these options:

- **--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 Identity Management 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, add DNS delegation from the parent domain to the Identity Management DNS domain. For example, if the Identity Management DNS domain is **ipa.example.com**, add a name server (NS) record to the **example.com** parent domain.



### IMPORTANT

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

### Additional resources

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

## CHAPTER 7. INSTALLING AN IDENTITY MANAGEMENT SERVER: WITH INTEGRATED DNS, WITH AN EXTERNAL CA

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

- You can automate much of the maintenance and DNS record management using native Identity Management 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 Identity Management 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 Identity Management domain.

Installing Identity Management with integrated DNS has certain limitations:

- Identity Management 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 Identity Management server with an external CA as the root CA.

### 7.1. 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 9, \*Installing an Identity Management server: Without integrated DNS, with an integrated 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 administration system user account (**admin**).

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

5. The script prompts for DNS forwarders.

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

- To configure DNS forwarders, enter **yes**, and then follow the instructions on the command line. The installation process will add the forwarder IP addresses to the **/etc/named.conf** file on the installed Identity Management server.

- o 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, your environment will be isolated, and names from other DNS domains in your infrastructure will not be resolved.
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 Identity Management 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.

**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 [Section 7.2, “Troubleshooting: External CA installation fails”](#).

## 7.2. 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 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 server installation:

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

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

## CHAPTER 8. INSTALLING AN IDENTITY MANAGEMENT SERVER: WITH INTEGRATED DNS, WITHOUT A CA

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

- You can automate much of the maintenance and DNS record management using native Identity Management 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 Identity Management 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 Identity Management domain.

Installing Identity Management with integrated DNS has certain limitations:

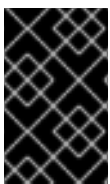
- Identity Management DNS is not meant to be used as a general-purpose DNS server. Some of the advanced DNS functions are not supported.

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

### 8.1. CERTIFICATES REQUIRED TO INSTALL AN IDENTITY MANAGEMENT SERVER WITHOUT A CA

This section lists:

- the certificates required to install an Identity Management 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

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

The files provided using **--dirsrv-cert-file** and **--http-cert-file** must contain exactly one server certificate and exactly one private key. The contents of the files provided using **--dirsrv-cert-file** and **--http-cert-file** are often identical.

### 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 **--dirsrv-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.

## 8.2. 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 [Section 8.1](#), “Certificates required to install an Identity Management server without a CA” for details on the provided certificates.

- 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 9, Installing an Identity Management server: Without integrated DNS, with an integrated CA](#) for details on the steps for installing a server without DNS.

- 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.

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

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

- The script prompts for DNS forwarders.

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

- To configure DNS forwarders, enter **yes**, and then follow the instructions on the command line. The installation process will add the forwarder IP addresses to the `/etc/named.conf` file on the installed Identity Management server.
    - 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, your environment will be isolated, and names from other DNS domains in your infrastructure will not be resolved.
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.
```



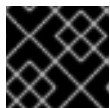
#### NOTE

Using Identity Management 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, add DNS delegation from the parent domain to the Identity Management DNS domain. For example, if the Identity Management DNS domain is **ipa.example.com**, add a name server (NS) record to the **example.com** parent domain.



#### IMPORTANT

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

## CHAPTER 9. INSTALLING AN IDENTITY MANAGEMENT SERVER: WITHOUT INTEGRATED DNS, WITH AN INTEGRATED CA

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

### 9.1. 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 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 Identity Management 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.

## 9.2. NON-INTERACTIVE INSTALLATION

This procedure installs a server:

- Without integrated DNS
- With integrated Identity Management 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 Identity Management 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
```

### Additional resources

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



## CHAPTER 10. UNINSTALLING AN IDENTITY MANAGEMENT SERVER

As an administrator, you can remove an Identity Management server from the topology.

This procedure describes how you can uninstall an example server named **server.example.com**.

### Prerequisites

- Before uninstalling a server that serves as a certificate authority (CA), key recovery authority (KRA), or DNS server, make sure these services are running on another server in the domain.



#### WARNING

Removing the last server that serves as a CA, KRA, or DNS server seriously disrupts the Identity Management functionality.

### Procedure

1. On all the servers in the topology that have a replication agreement with **server.example.com**, use the **ipa server-del** command to delete the replica from the topology:

```
[root@another_server ~]# ipa server-del server.example.com
```

2. On **server.example.com**, use the **ipa-server-install --uninstall** command:

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

3. Make sure all name server (NS) DNS records pointing to **server.example.com** are deleted from your DNS zones. This applies regardless of whether you use integrated DNS managed by Identity Management or external DNS.

## CHAPTER 11. RENAMING AN IDENTITY MANAGEMENT SERVER

You cannot change the host name of an existing Identity Management 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 [Chapter 19, \*Installing an Identity Management replica\*](#).



#### IMPORTANT

If the server you are uninstalling is a CRL master server, make another server the CRL master server before proceeding.

2. Stop the existing Identity Management server instance.

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

3. Uninstall the existing server as described in [Chapter 10, \*Uninstalling an Identity Management server\*](#).

## CHAPTER 12. PREPARING THE SYSTEM FOR IDENTITY MANAGEMENT CLIENT INSTALLATION

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

### 12.1. DNS REQUIREMENTS FOR IDENTITY MANAGEMENT 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 Identity Management 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 Identity Management clients are not required to be part of the primary DNS domain. If the client machine hostname is not in a subdomain of an Identity Management 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 Identity Management servers.

#### Additional resources

- For details on DNS requirements in Identity Management, see [Section 5.3, “Host name and DNS requirements for Identity Management”](#).

### 12.2. PORT REQUIREMENTS FOR IDENTITY MANAGEMENT CLIENTS

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

On Identity Management 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 [Section 5.4, “Port requirements for Identity Management”](#).

### 12.3. PACKAGES REQUIRED TO INSTALL AN IDENTITY MANAGEMENT CLIENT

In RHEL8, the packages necessary for installing an Identity Management client are shipped as a module. The `client` stream is the default stream of the `idm` module, and you do not need to enable the stream before downloading the packages.

You do need to enable the `idm:client` stream if you previously enabled the `idm:DL1` stream.

1. (Optional) To switch to the RPMs delivered through the default, `idm:client` stream if you previously enabled the `idm:DL1` stream:

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

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

```
# yum module install idm
```

## CHAPTER 13. INSTALLING AN IDENTITY MANAGEMENT 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.

Several options are available for a basic installation:

- For installing a client interactively using privileged user's credentials, see [Section 13.3, "Installing a client by using user credentials: Interactive installation"](#).
- For installing a client interactively using a one-time password, see [Section 13.4, "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 previous enrollment, see [Section 13.5, "Installing a client: Non-interactive installation"](#).

### 13.1. PREREQUISITES

Before you start installing the Identity Management client, make sure that you have met all the prerequisites. See [Chapter 12, \*Preparing the system for Identity Management client installation\*](#).

### 13.2. AN OVERVIEW OF THE IDENTITY MANAGEMENT CLIENT INSTALLATION OPTIONS

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

- The credentials of a user authorized to enroll clients. This is the default option expected by **ipa-client-install**.
  - To provide the credentials of an authorized user directly to **ipa-client-install**, use the **--principal** and **--password** options. See [Section 13.3, "Installing a client by using user credentials: Interactive installation"](#) for a detailed procedure.
- A random, one-time password pre-generated on the server:
  - To use this authentication method, add the **--random** option to **ipa-client-install** option. See [Section 13.4, "Installing a client by using a one-time password: Interactive installation"](#) for a detailed procedure.
- The client principal from the previous enrollment:
  - This option is available if the system was previously enrolled as an Identity Management client. To use this authentication method, add the **--keytab** option to **ipa-client-install**. See [Chapter 15, \*Re-enrolling an Identity Management client\*](#) for details.

#### Additional resources

- For details on the options accepted by **ipa-client-install**, see the **ipa-client-install(1)** man page.

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

This procedure describes installing an Identity Management 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 Identity Management 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 Identity Management client.

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

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 Identity Management 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
```

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.

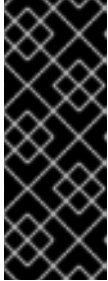
- If the SRV records are set properly in the IdM DNS zone, the script automatically discovers all the other required values and displays them. Enter **yes** to confirm.

```
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**
- If the script fails to obtain some settings automatically, it prompts you for the values.



### 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.

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.

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

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

### Prerequisites

1. On a server in the domain, add the future client system as an Identity Management 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 Identity Management 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 Identity Management 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
```

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 Identity Management 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
```

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.

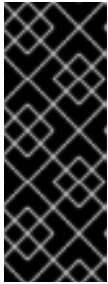
```
Client hostname: client.example.com
Realm: EXAMPLE.COM
DNS Domain: example.com
IPA Server: server.example.com
BaseDN: dc=example,dc=com
```

```
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**
- If the script fails to obtain some settings automatically, it prompts you for the values.



### 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.

3. 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.

## 13.5. 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

For details, see [Section 13.2, “An overview of the Identity Management client installation options”](#).

### The option for unattended installation

The **--unattended** 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 host name for the client machine



## 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.
- **--server** to specify the host name of the IdM server the client will be enrolled with
  - **--domain** to specify the DNS domain name of the IdM server the client will be enrolled with
  - **--realm** to specify the Kerberos realm name

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

```
# ipa-client-install --password 'W5YpARl=7M.n' --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 example.com --server server.example.com --unattended
```

## Additional resources

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

## 13.6. REMOVING PRE-IDENTITY MANAGEMENT CONFIGURATION AFTER INSTALLING A CLIENT

The **ipa-client-install** script does not remove any previous LDAP and 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 configuration values:

1. Open **/etc/openldap/ldap.conf** and **/etc/sss/sss.conf**.
2. Delete the previous configuration.
3. Uncomment the new Identity Management 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.

## 13.7. TESTING AN IDENTITY MANAGEMENT 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 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 ~]#
```

## 13.8. CONNECTIONS PERFORMED DURING AN IDENTITY MANAGEMENT CLIENT INSTALLATION

Table 13.1, “Requests performed during an Identity Management client installation” lists the operations performed by **ipa-client-install**, the Identity Management (IdM) client installation tool.

**Table 13.1. Requests performed during an Identity Management 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 masters; (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 masters	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

## 13.9. IDENTITY MANAGEMENT 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.

[Table 13.2, “CLI post-installation operations”](#) shows the operations performed by the CLI during an IdM client post-installation deployment. [Table 13.3, “webUI 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**. [Section 13.9.1, “SSSD communication patterns”](#) and [Section 13.9.2, “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 masters
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 masters	HTTPS	any <b>ipa</b> utility usage

**Table 13.3. webUI post-installation operations**

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

### 13.9.1. SSSD communication patterns

The System Security Services Daemon (SSSD) is a system service to access remote directories and authentication mechanisms. If configured on an 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 System Security Services Daemon (SSSD) can be configured to communicate with multiple servers. [Table 13.4, “Communication patterns of SSSD on IdM clients when talking to IdM servers”](#) and [Table 13.5, “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 masters
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
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 masters
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

Operation	Protocol used	Purpose
(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

### 13.9.2. Certmonger communication patterns

**Certmonger** is a daemon running on IdM masters 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 masters.

**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 masters
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 masters	HTTPS	To request new certificates
Access over port 8080 (TCP/TCP6) on the IdM master	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 master	HTTPS	To administer the Certificate Authority on the IdM master (only during IdM master and replica installation)

## CHAPTER 14. INSTALLING AN IDENTITY MANAGEMENT CLIENT WITH KICKSTART

A Kickstart enrollment automatically adds a new system to the Identity Management 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 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

1. Pre-create the host entry on the Identity Management 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.

2. 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.
3. Use the Kickstart file to install the Identity Management 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 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 Identity Management 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 Identity Management domain

...and request information to access and configure the Identity Management domain services

- The password which you set when pre-creating the client host on the Identity Management 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/sbin/sshd-keygen

# 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 IDENTITY MANAGEMENT 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 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. RE-ENROLLING AN IDENTITY MANAGEMENT CLIENT

If a client virtual machine has been destroyed and lost connection with the Identity Management (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 new certificates, 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 Identity Management, 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 [Chapter 17, Renaming Identity Management client systems](#).

## 15.1. WHAT HAPPENS DURING CLIENT RE-ENROLLMENT

During re-enrollment, Identity Management:

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

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

This procedure describes re-enrolling an Identity Management 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 [Section 13.3, "Installing a client by using user credentials: Interactive installation"](#).

## 15.3. 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 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.

- Re-create the client machine with the same host name.
- Copy the keytab file from the backup location to the `/etc/` directory on the re-created client machine.
- 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.

## 15.4. TESTING AN IDENTITY MANAGEMENT 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 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 16. UNINSTALLING AN IDENTITY MANAGEMENT CLIENT

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

### 16.1. UNINSTALLING AN IDENTITY MANAGEMENT CLIENT

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

#### Procedure

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

```
# ipa-client-install --uninstall
```

2. Remove the DNS entries for the client host manually from the 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"
```

## CHAPTER 17. RENAMING IDENTITY MANAGEMENT CLIENT SYSTEMS

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



### WARNING

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

Renaming an Identity Management client involves:

1. Preparing the host. For details, see [Section 17.1, “Prerequisites”](#)
2. Uninstalling the IdM client from the host. For details, see [Section 17.2, “Uninstalling an Identity Management client”](#)
3. Renaming the host. For details, see [Section 17.3, “Renaming the host system”](#)
4. Installing the IdM client on the host with the new name. For details, see [Section 17.4, “Re-installing an Identity Management client”](#)
5. Configuring the host after the IdM client installation. For details, see [Section 17.5, “Re-adding services, re-generating certificates, and re-adding host groups”](#)

### 17.1. PREREQUISITES

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
```

## 17.2. UNINSTALLING AN IDENTITY MANAGEMENT CLIENT

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

### Procedure

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

```
# ipa-client-install --uninstall
```

2. Remove the DNS entries for the client host manually from the 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"
```

## 17.3. 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 client to the Identity Management domain with the new host name.

## 17.4. RE-INSTALLING AN IDENTITY MANAGEMENT CLIENT

Install an client on your renamed host following the procedure described in [Chapter 13, \*Installing an Identity Management client: Basic scenario\*](#).

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

1. On the Identity Management server, add a new keytab for every service identified in [Section 17.1, “Prerequisites”](#).

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

2. Generate certificates for services that had a certificate assigned in [Section 17.1, “Prerequisites”](#). You can do this:

- Using the Identity Management administration tools

- Using the **certmonger** utility
3. Re-add the client to the host groups identified in [Section 17.1, “Prerequisites”](#).

## CHAPTER 18. PREPARING THE SYSTEM FOR IDENTITY MANAGEMENT REPLICA INSTALLATION

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

A system where you want to install a replica must meet the general requirements for servers:

- [Chapter 5, \*Preparing the system for Identity Management server installation\*](#)

For additional requirements specific to replicas, see:

- [Section 18.1, “Replica version requirements”](#)

### 18.1. REPLICA VERSION REQUIREMENTS

Red Hat Enterprise Linux (RHEL) 8 replicas only work with IdM masters 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 Identity Management. For example:

- The master server is installed on Red Hat Enterprise Linux 8 and uses the Identity Management 4.x packages.
- You must install the replica also on Red Hat Enterprise Linux 8 or later and use Identity Management version 4.x or later.

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



## CHAPTER 19. INSTALLING AN IDENTITY MANAGEMENT REPLICA

The following sections describe how to install an Identity Management replica based on an existing server. The replica installation process copies the configuration of the existing server, and installs the replica based on that configuration.



### NOTE

Install one Identity Management replica at a time. The installation of multiple replicas at the same time is not supported.

Before installing a replica, the target system must be authorized for enrollment in the Identity Management domain. See:

- [Section 19.1, “Prerequisites for installing a replica on an Identity Management client”](#)
- [Section 19.2, “Prerequisites for installing a replica on a system outside the Identity Management domain”](#)

For the replica installation procedures, see:

- [Section 19.3, “Installing an Identity Management replica with integrated DNS”](#)
- [Section 19.5, “Installing an Identity Management replica without a CA”](#)

After the installation, see:

- [Section 19.6, “Testing an Identity Management replica”](#)

### 19.1. PREREQUISITES FOR INSTALLING A REPLICA ON AN IDENTITY MANAGEMENT CLIENT

When installing a replica on an existing client, choose one of the following authorization methods.

#### A privileged user’s credentials

Choose this method to authorize the replica installation by providing a privileged user’s credentials:

- Log in as the privileged user before running the **ipa-replica-install** utility. The default privileged user is **admin**:

```
$ kinit admin
```

- Let Identity Management prompt you for the credentials interactively. This is the default behavior.

#### The ipaservers host group

Choose this method to authorize the replica installation by adding the client to the **ipaservers** host group. Membership in **ipaservers** grants the machine elevated privileges analogous to the administrator’s credentials.

To add the client as a member of **ipaservers**:

■

```
$ kinit admin
```

```
$ ipa hostgroup-add-member ipaservers --hosts replica.example.com
Host-group: ipaservers
Description: IPA server hosts
Member hosts: server.example.com, client.example.com
-----
Number of members added 1
-----
```

## 19.2. PREREQUISITES FOR INSTALLING A REPLICA ON A SYSTEM OUTSIDE THE IDENTITY MANAGEMENT DOMAIN

When you run the **ipa-replica-install** utility on a system that has not yet been enrolled in the Identity Management domain, **ipa-replica-install** first enrolls the system as a client and then installs the replica components.

When installing a replica on a system outside the Identity Management domain, choose one of the following authorization methods.

### A privileged user's credentials

Using this method, the replica installation is authorized by providing a privileged user's credentials. The default privileged user is **admin**.

To use this method, add the principal name and password options (**--principal admin --admin-password password**) to **ipa-replica-install** directly during the installation.

### A random password generated on an Identity Management server

Using this method, the replica installation is authorized by providing a random password for one-time enrollment.

To generate the random password for the future replica and add the future replica system to the **ipaservers** host group, use these commands on any server in the domain:

1. Log in as the administrator.

```
$ kinit admin
```

2. Add the new machine 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 replica installation.

```
$ ipa host-add replica.example2.com --random
-----
Added host "replica.example2.com"
-----
Host name: replica.example2.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 machine to the **ipaservers** host group.

```
$ ipa hostgroup-add-member ipaservers --hosts replica.example2.com
Host-group: ipaservers
Description: IPA server hosts
Member hosts: server.example.com, replica.example2.com
-----
Number of members added 1
-----
```

Membership in **ipaservers** grants the machine elevated privileges required to set up the necessary server services.

## 19.3. INSTALLING AN IDENTITY MANAGEMENT REPLICA WITH INTEGRATED DNS

This procedure describes installing a replica:

- With integrated DNS
- Without a certificate authority (CA) in an Identity Management (IdM) environment in which a CA is already installed. The replica will forward all certificate operations to the Identity Management IdM server with a CA installed.

### Procedure

1. Run **ipa-replica-install** with these options:

- **--setup-dns** to configure the replica as the DNS server
- **--forwarder** to specify a forwarder, or **--no-forwarder** if you do not want to use any forwarders. To specify multiple 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.

## 19.4. INSTALLING AN IDENTITY MANAGEMENT REPLICA WITH A CA

This procedure describes installing a 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 master server.

For example, if the server includes an integrated Identity Management CA as the root CA, the replica must also be installed with an integrated CA as the root CA. No other CA configuration is available in this case.

The inclusion of the `--setup-ca` option in the `ipa-replica-install` command takes care of copying the CA configuration of the initial server.

**Procedure**

1. Run `ipa-replica-install` with the `--setup-ca` option.

```
# ipa-replica-install --setup-ca
```

**19.5. INSTALLING AN IDENTITY MANAGEMENT REPLICA WITHOUT A CA**

This procedure describes installing a replica:

- Without integrated DNS
- Without a certificate authority (CA) by providing the required certificates manually. The assumption here is that the master server was also 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.

**Procedure**

- Run `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 8.1, “Certificates required to install an Identity Management 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 master server.

## 19.6. TESTING AN IDENTITY MANAGEMENT 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
```

## 19.7. CONNECTIONS PERFORMED DURING AN IDENTITY MANAGEMENT REPLICA INSTALLATION

Table 19.1, “Requests performed during an Identity Management replica installation” lists the operations performed by `ipa-replica-install`, the Identity Management (IdM) replica installation tool.

**Table 19.1. Requests performed during an Identity Management 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 masters
Requests to ports 88 (TCP/TCP6 and UDP/UDP6) on the discovered IdM masters	Kerberos	To obtain a Kerberos ticket
JSON-RPC calls to the IdM Apache-based web-service on the discovered or configured IdM masters	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

Operation	Protocol used	Purpose
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 master	HTTPS	To administer the Certificate Authority on the IdM master (only during IdM master and replica installation)

## CHAPTER 20. UNINSTALLING AN IDENTITY MANAGEMENT REPLICA

As an administrator, you can remove an Identity Management server from the topology.

This procedure describes how you can uninstall an example server named **server.example.com**.

### Prerequisites

- Before uninstalling a server that serves as a certificate authority (CA), key recovery authority (KRA), or DNS server, make sure these services are running on another server in the domain.



#### WARNING

Removing the last server that serves as a CA, KRA, or DNS server seriously disrupts the Identity Management functionality.

### Procedure

1. On all the servers in the topology that have a replication agreement with **server.example.com**, use the **ipa server-del** command to delete the replica from the topology:

```
[root@another_server ~]# ipa server-del server.example.com
```

2. On **server.example.com**, use the **ipa-server-install --uninstall** command:

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

3. Make sure all name server (NS) DNS records pointing to **server.example.com** are deleted from your DNS zones. This applies regardless of whether you use integrated DNS managed by Identity Management or external DNS.

## **PART III. MANAGING IDENTITY MANAGEMENT**



## CHAPTER 21. CONFIGURING IDENTITY MANAGEMENT FOR AUTHENTICATING WITH A CERTIFICATE

By configuring Identity Management (IdM), IdM system administrators can enable users to authenticate to the IdM web UI and command-line interface (CLI) using a certificate that a Certificate Authority (CA) has issued to the users. Authenticating using a certificate can be very convenient and fast, as the user is not prompted for any password.

The web browser can run on a system that is not part of the IdM domain.

This user story provides instructions on how to effectively configure and test logging into Identity Management web UI and CLI with a certificate. In following this user story,

- you can skip [Section 21.2, “Requesting a new user certificate and exporting it to the client”](#) if the user you want to authenticate using a certificate already has a certificate;
- you can skip [Section 21.3, “Making sure the certificate and user are linked together”](#) if the user’s certificate has been issued by the IdM CA.



### NOTE

Only Identity Management users can log into the web UI using a certificate. Active Directory users can log in with their user name and password.

## 21.1. CONFIGURING THE IDENTITY MANAGEMENT SERVER FOR CERTIFICATE AUTHENTICATION IN THE WEB UI

As an Identity Management (IdM) administrator, you can allow users to use certificates to authenticate to your IdM environment.

### Procedure

As the Identity Management administrator:

1. On an Identity Management server, obtain administrator privileges and create a shell script to configure the server.
  - a. Run the **ipa-adviser config-server-for-smart-card-auth** command, and save its output to a file, for example **server\_certificate\_script.sh**:

```
# kinit admin
# ipa-adviser config-server-for-smart-card-auth >
server_certificate_script.sh
```

- b. Add execute permissions to the file using the **chmod** utility:

```
# chmod +x server_certificate_script.sh
```

2. On all the servers in the Identity Management domain, run the **server\_certificate\_script.sh** script
  - a. with the path of the IdM Certificate Authority certificate, **/etc/ipa/ca.crt**, as input if the IdM CA is the only certificate authority that has issued the certificates of the users you want to enable certificate authentication for:

■

```
# ./server_certificate_script.sh /etc/ipa/ca.crt
```

- b. with the paths leading to the relevant CA certificates as input if different external CAs signed the certificates of the users who you want to enable certificate authentication for:

```
# ./server_certificate_script.sh /tmp/ca1.pem /tmp/ca2.pem
```



#### NOTE

Do not forget to run the script on each new replica that you add to the system in the future if you want to have certificate authentication for users enabled in the whole topology.

## 21.2. REQUESTING A NEW USER CERTIFICATE AND EXPORTING IT TO THE CLIENT

As an Identity Management (IdM) administrator, you can create certificates for users in your IdM environment and export them to the IdM clients on which you want to enable certificate authentication for users.



#### NOTE

You can skip this section if the user you want to authenticate using a certificate already has a certificate.

### Procedure

1. Optionally, create a new directory, for example `~/certdb/`, and make it a temporary certificate database. When asked, create an NSS Certificate DB password to encrypt the keys to the certificate to be generated in a subsequent step:

```
# mkdir ~/certdb/
# certutil -N -d ~/certdb/
Enter a password which will be used to encrypt your keys.
The password should be at least 8 characters long,
and should contain at least one non-alphabetic character.
```

```
Enter new password:
Re-enter password:
```

2. Create the certificate signing request (CSR) and redirect the output to a file. For example, to create a CSR with the name `certificate_request.csr` for a **4096** bit certificate for the `idm_user` user in the `IDM.EXAMPLE.COM` realm, setting the nickname of the certificate private keys to `idm_user` for easy findability, and setting the subject to `CN=idm_user,O=IDM.EXAMPLE.COM`:

```
# certutil -R -d ~/certdb/ -a -g 4096 -n idm_user -s
"CN=idm_user,O=IDM.EXAMPLE.COM" > certificate_request.csr
```

3. When prompted, enter the same password that you entered when using `certutil` to create the temporary database. Then continue typing randomly until told to stop:

```
Enter Password or Pin for "NSS Certificate DB":
```

A random seed must be generated that will be used in the creation of your key. One of the easiest ways to create a random seed is to use the timing of keystrokes on a keyboard.

To begin, type keys on the keyboard until this progress meter is full. DO NOT USE THE AUTOREPEAT FUNCTION ON YOUR KEYBOARD!

Continue typing until the progress meter is full:

4. Submit the certificate request file to the server. Specify the Kerberos principal to associate with the newly-issued certificate, the output file to store the certificate, and optionally the certificate profile. For example, to obtain a certificate of the **IECUserRoles** profile, a profile with added user roles extension, for the **idm\_user@IDM.EXAMPLE.COM** principal, and save it in the **~/idm\_user.pem** file:

```
# ipa cert-request certificate_request.csr --
principal=idm_user@IDM.EXAMPLE.COM --profile-id=IECUserRoles --
certificate-out=~/idm_user.pem
```

5. Add the certificate to the NSS database. Use the **-n** option to set the same nickname that you used when creating the CSR previously so that the certificate matches the private key in the NSS database. The **-t** option sets the trust level. For details, see the `certutil(1)` man page. The **-i** option specifies the input certificate file. For example, to add to the NSS database a certificate with the **idm\_user** nickname that is stored in the **~/idm\_user.pem** file in the **~/certdb/** database:

```
# certutil -A -d ~/certdb/ -n idm_user -t "P,," -i ~/idm_user.pem
```

6. Verify that the key in the NSS database does not show (**orphan**) as its nickname. For example, to verify that the certificate stored in the **~/certdb/** database is not orphaned:

```
# certutil -K -d ~/certdb/
< 0> rsa      5ad14d41463b87a095b1896cf0068ccc467df395    NSS
Certificate DB:[replaceable]idm_user
```

7. Use the **pk12util** command to export the certificate from the NSS database to the PKCS12 format. For example, to export the certificate with the **idm\_user** nickname from the **/root/certdb** NSS database into the **~/idm\_user.p12** file:

```
# pk12util -d ~/certdb -o ~/idm_user.p12 -n idm_user
Enter Password or Pin for "NSS Certificate DB":
Enter password for PKCS12 file:
Re-enter password:
pk12util: PKCS12 EXPORT SUCCESSFUL
```

8. Transfer the certificate to the host on which you want the certificate authentication for **idm\_user** to be enabled:

```
# scp ~/idm_user.p12 idm_user@client.idm.example.com:/home/idm_user/
```

9. On the host to which the certificate has been transferred, make the directory in which the .pkcs12 file is stored inaccessible to the 'other' group for security reasons:

```
# chmod o-rwx /home/idm_user/
```

10. For security reasons, remove the temporary NSS database and the .pkcs12 file from the server:

```
# rm ~/certdb/  
# rm ~/idm_user.p12
```

## 21.3. MAKING SURE THE CERTIFICATE AND USER ARE LINKED TOGETHER



### NOTE

You can skip this section if the user's certificate has been issued by the IdM CA.

For certificate authentication to work, you need to make sure that the certificate is linked to the user that will use it to authenticate to Identity Management (IdM).

- If the certificate is provided by a Certificate Authority that is not part of your Identity Management environment, link the user and the certificate following the procedure described in [Linking User Accounts to Certificates](#).
- If the certificate is provided by Identity Management CA, the certificate is already automatically added in the user entry and you do not have to link the certificate to the user account. For details on creating a new certificate in IdM, see [Section 21.2, “Requesting a new user certificate and exporting it to the client”](#).

## 21.4. CONFIGURING A BROWSER TO ENABLE CERTIFICATE AUTHENTICATION

For certificate authentication to work in your Identity Management web UI, you need to import the user and Certificate Authority (CA) certificates into the Mozilla Firefox or Google Chrome browser running on the host on which you want to enable certificate authentication. The host itself does not have to be part of the IdM domain.

Identity Management supports the following browsers for connecting to the web UI:

- Mozilla Firefox 38 and later
- Google Chrome 46 and later

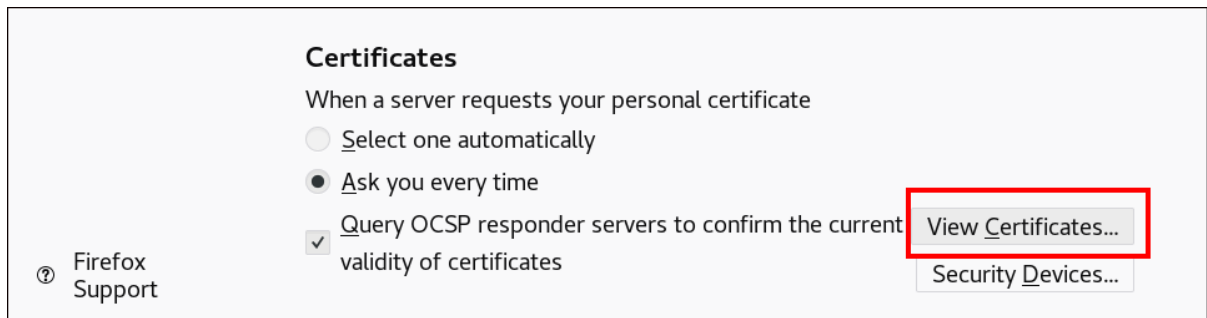
The following procedure shows how to configure the Mozilla Firefox 57.0.1 browser.

### Procedure

1. Open Firefox, then navigate to **Preferences** → **Privacy & Security**.



2. Click **View Certificates**.

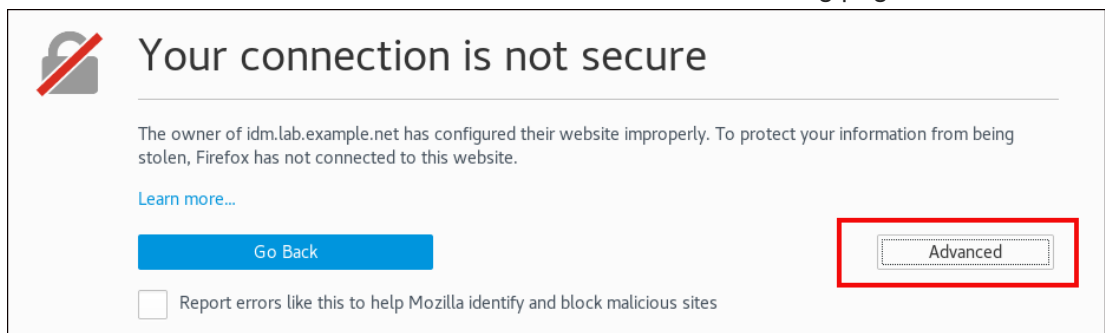


3. In the **Your Certificates** tab, click **Import**. Locate and open the certificate of the user in the PKCS12 format, then click **OK** and **OK**.

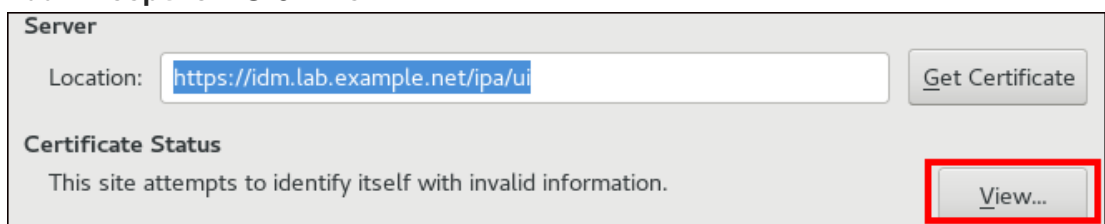
4. Make sure that the Identity Management Certificate Authority is recognized by Firefox as a trusted authority:

a. Save the IdM CA certificate locally:

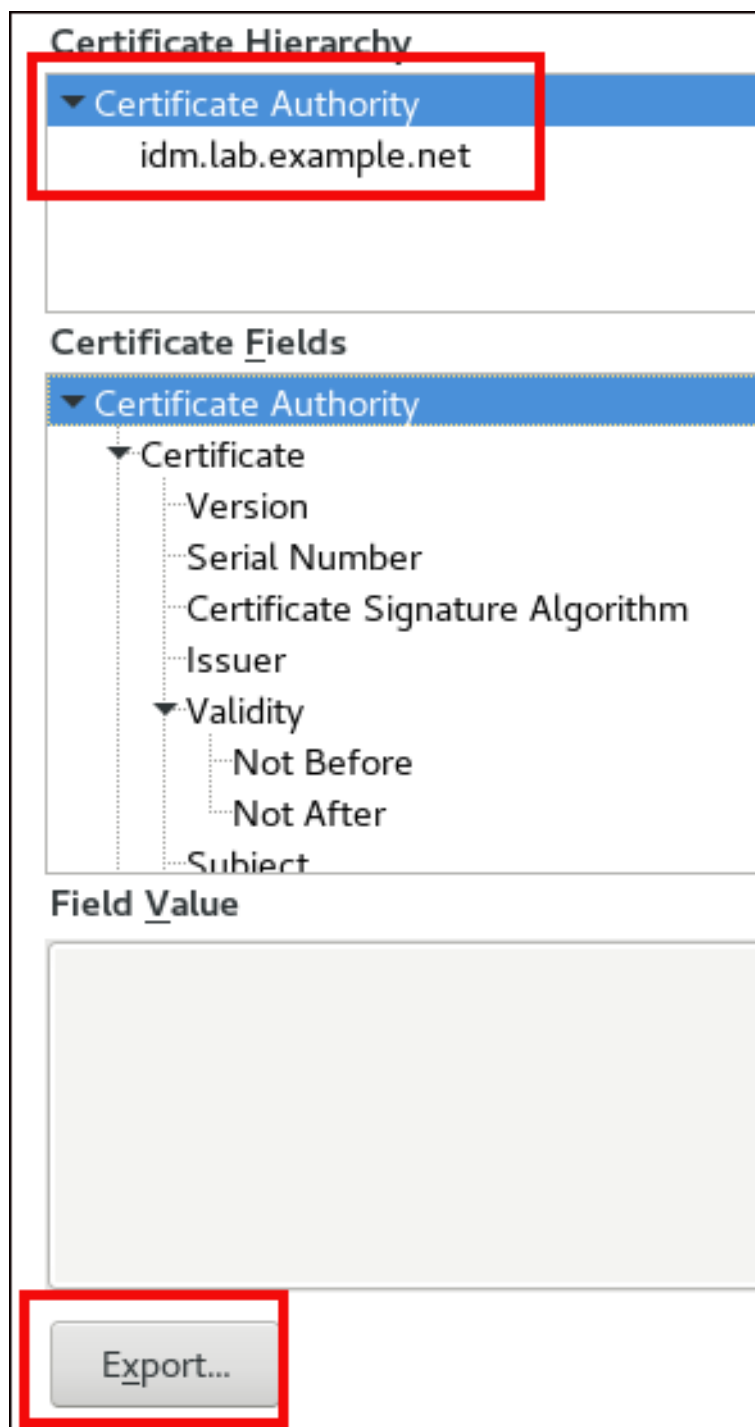
- Navigate to the IdM web UI by writing the name of your IdM server in the Firefox address bar. Click **Advanced** on the Insecure Connection warning page.



- **Add Exception.** Click **View**.



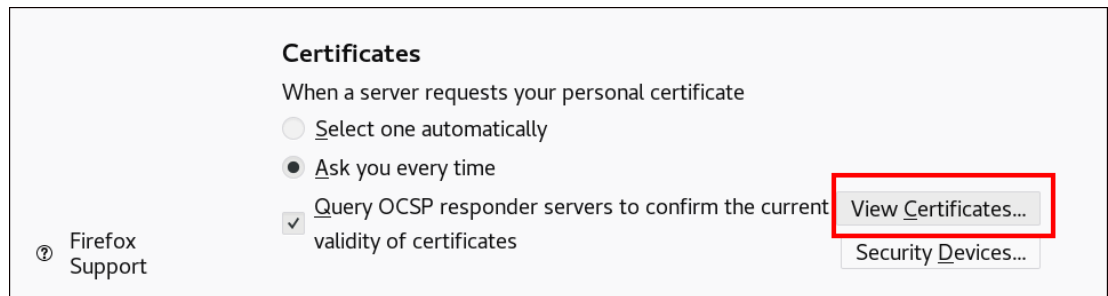
- In the **Details** tab, highlight the **Certificate Authority** fields.



- Click **Export**. Save the CA certificate, for example as the **CertificateAuthority.crt** file, then click **Close**, and **Cancel**.
- b. Import the IdM CA certificate to Firefox as a trusted certificate authority certificate:
- Open Firefox, navigate to Preferences and click **Privacy & Security**.



- Click **View Certificates**.



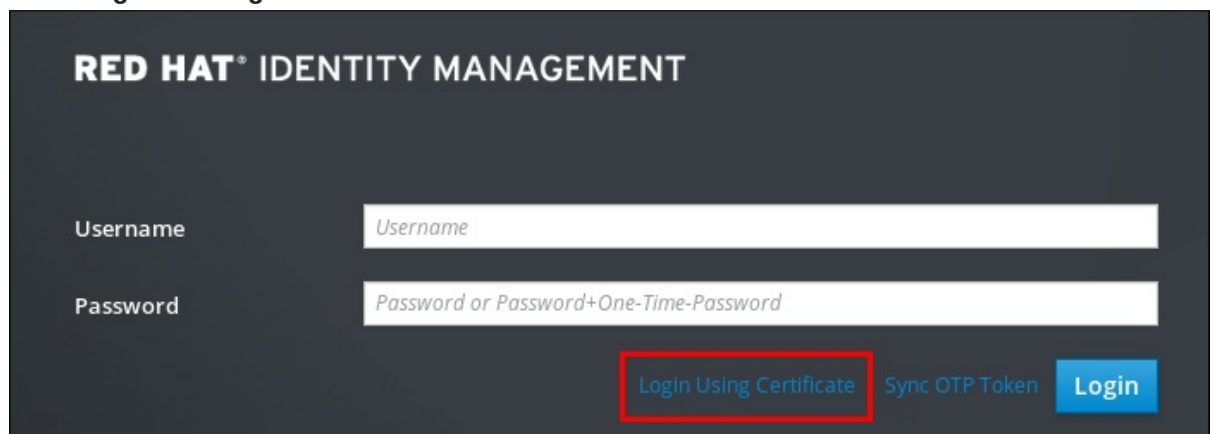
- In the **Authorities** tab, click **Import**. Locate and open the CA certificate that you saved in the previous step in the **CertificateAuthority.crt** file. Trust the certificate to identify websites, then click **OK** and **OK**.
5. Continue to [Section 21.5, “Authenticating to the Identity Management Web UI with a Certificate as an Identity Management User”](#).

## 21.5. AUTHENTICATING TO THE IDENTITY MANAGEMENT WEB UI WITH A CERTIFICATE AS AN IDENTITY MANAGEMENT USER

This procedure describes authenticating as a user to the Identity Management (IdM) web UI using a certificate stored on the desktop of an Identity Management client.

### Procedure

1. In the browser, navigate to the Identity Management web UI at, for example, **https://server.idm.example.com/ipa/ui**.
2. Click **Login Using Certificate**.



3. The user's certificate should already be selected. Uncheck **Remember this decision**, then click **OK**.

You are now authenticated as the user who corresponds to the certificate.

### Additional Resources

- If the authentication fails, see [Investigating Smart Card Authentication Failures](#) in the *Linux Domain Identity, Authentication, and Policy Guide*.
- For information about authenticating to the IdM web UI using a certificate stored on a smart card, see [Managing Smart Card Links in the Identity Management Server](#) in the *Linux Domain Identity, Authentication, and Policy Guide*.

## 21.6. CONFIGURING AN IDM CLIENT TO ENABLE AUTHENTICATING TO THE CLI USING A CERTIFICATE

To make certificate authentication work for an IdM user in the Command Line Interface (CLI) of your IdM client, import the IdM user's certificate and the private key to the IdM client. For details on creating and transferring the user certificate, see [Section 21.2, "Requesting a new user certificate and exporting it to the client"](#).

### Procedure

Log into the IdM client and have the .p12 file containing the user's certificate and the private key ready. To obtain and cache the Kerberos ticket granting ticket (TGT), run the **kinit** command with the user's principal, using the **-X** option with the **X509\_username:/path/to/file.p12** attribute to specify where to find the user's X509 identity information. For example, to obtain the TGT for **idm\_user** using the user's identity information stored in the **~/idm\_user.p12** file:

```
$ kinit -X X509_idm_user='PKCS12:~/idm_user.p12' idm_user
```



### NOTE

The command also supports the .pem file format: **kinit -X X509\_username='FILE:/path/to/cert.pem,/path/to/key' user\_principal**



## CHAPTER 22. ENABLING AD USERS TO ADMINISTER IDM

### 22.1. ID OVERRIDES FOR AD USERS

In Red Hat Enterprise Linux (RHEL) 7, external group membership allows AD users and groups to access IdM resources in a POSIX environment with the help of the System Security Services Daemon (SSSD).

The IdM LDAP server has its own mechanisms to grant access control. RHEL 8 introduces an update that allows adding an ID user override for an AD user as a member of an IdM group. An ID override is a record describing what a specific Active Directory user or group properties should look like within a specific ID view, in this case the Default Trust View. As a consequence of the update, the IdM LDAP server is able to apply access control rules for the IdM group to the AD user.

AD users are now able to use the self service features of IdM UI, for example to upload their SSH keys, or change their personal data. An AD administrator is able to fully administer IdM without having two different accounts and passwords.



#### NOTE

Currently, selected features in IdM may still be unavailable to AD users. For example, setting passwords for IdM users as an AD user from the IdM **admins** group might fail.

### 22.2. USING ID OVERRIDES TO ENABLE AD USERS TO ADMINISTER IDM

#### PREREQUISITES

- The **idm:DL1** stream is enabled on your IdM server and you have switched to the RPMs delivered through this stream:

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

- The **idm:DL1/adtrust** profile is installed on your IdM server.

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

The profile contains all the packages necessary for installing an IdM server that will have a trust agreement with Active Directory, including the **ipa-idoverride-memberof** package.

- A working Identity Management environment is set up. For details, see [Chapter 5, \*Preparing the system for Identity Management server installation\*](#).
- A working trust between your Identity Management environment and Active Directory is set up.

#### PROCEDURE

This procedure describes creating and using an ID override for an Active Directory (AD) user to give that user rights identical to those of an Identity Management (IdM) user. During this procedure, work on an IdM server that is configured as a trust controller or a trust agent. For details on trust controllers and trust agents, see [Section 4.2, “Trust controllers and trust agents”](#).

1. As an IdM administrator, create an ID override for an AD user in the Default Trust View. For example, to create an ID override for the **ad\_user@ad.example.com** user:

```
# kinit admin
# ipa idoverrideuser-add 'default trust view' ad_user@ad.example.com
```

2. Add the ID override from the Default Trust View as a member to an IdM group. If the group in question is a member of an IdM role, the AD user represented by the ID override will gain all permissions granted by the role when using the IdM API, including both the command line interface and the IdM web UI. For example, to add the ID override for the **ad\_user@ad.example.com** user to the **admins** group:

```
# ipa group-add-member admins --
  idoverrideusers=ad_user@ad.example.com
```

## 22.3. MANAGING IDM COMMAND-LINE INTERFACE (CLI) AS AN AD USER

This procedure checks that an Active Directory user can log into Identity Management CLI and run commands appropriate for his role.

1. Destroy the current Kerberos ticket of the IdM administrator:

```
# kdestroy -A
```



### NOTE

The destruction of the Kerberos ticket is required because the GSSAPI implementation in MIT Kerberos chooses credentials from the realm of the target service by preference, which in this case is the IdM realm. This means that if a credentials cache collection, namely the KCM:, KEYRING:, or DIR: type of credentials cache is in use, a previously obtained **admin** or any other IdM principal's credentials will be used to access the IdM API instead of the AD user's credentials.

2. Obtain the Kerberos credentials of the AD user for whom an ID override has been created:

```
# kinit ad_user@AD.EXAMPLE.COM
Password for ad_user@AD.EXAMPLE.COM:
```

3. Test that the ID override of the AD user enjoys the same privileges stemming from membership in the IdM group as any IdM user in that group. If the ID override of the AD user has been added to the **admins** group, the AD user can, for example, create groups in IdM:

```
# ipa group-add some-new-group
-----
Added group "some-new-group"
-----
Group name: some-new-group
GID: 1997000011
```

## **PART IV. CONFIGURING AUTHENTICATION ON A RED HAT ENTERPRISE LINUX HOST**

## CHAPTER 23. USING AUTHSELECT

### 23.1. EXPLAINING AUTHSELECT

**Authselect** is a utility that simplifies the configuration of user authentication on a Red Hat Enterprise Linux host. **Authselect** offers two ready-made profiles that can be universally used with all modern identity management systems:

- the **sssd** profile
- the **winbind** profile

For legacy compatibility reasons, the **nis** profile is also available.

Red Hat recommends using **authselect** in semi-centralized identity management environments, for example if your company utilizes the LDAP, winbind or nis databases to authenticate users to use services in your domain.



#### WARNING

Do not use **authselect** if your host is part of Red Hat Enterprise Linux Identity Management or Active Directory. The **ipa-client-install** command, called when joining your host to a Red Hat Identity Management domain, takes full care of configuring authentication on your host. Similarly the **realm join** command, called when joining your host to an Active Directory domain, takes full care of configuring authentication on your host.

The **authconfig** utility, used in previous Red Hat Enterprise Linux versions, created and modified many different configuration files, making troubleshooting a difficult task. **Authselect** makes testing and troubleshooting easy because it only modifies files in these directories:

- **/etc/nsswitch.conf**
- **/etc/pam.d/\*** files
- **/etc/dconf/db/distro.d/\*** files

The Name Service Switch (NSS) configuration file, **/etc/nsswitch.conf**, is used by the GNU C Library and certain other applications to determine the sources from which to obtain name-service information in a range of categories, and in what order. Each category of information is identified by a database name.

Linux-PAM is a system of libraries that handle the authentication tasks of applications (services) on the system. The nature of the authentication is dynamically configurable: the system administrator can choose how individual service-providing applications will authenticate users. This dynamic configuration is set by the contents of the configuration files in the **/etc/pam.d/** directory, which list the PAMs that will do the authentication tasks required by this service, and the appropriate behavior of the PAM-API in the event that individual PAMs fail.

Once an **authselect** profile is selected for a given host, the profile will be applied to every user logging into the host.

## 23.2. CHOOSING AN AUTHSELECT PROFILE

As a system administrator, you can select a profile for the **authselect** utility for a specific host. The profile will be applied to every user logging into the host.

### Procedure

1. Select the **authselect** profile that is appropriate for your authentication provider. For example, for logging into the network of a company that uses LDAP, choose **sssd**. Run the command as root:

```
# authselect select sssd
```

2. Optionally, review the contents of the **/etc/nsswitch.conf** file:

```
passwd:      sss files
group:       sss files
netgroup:    sss files
automount:   sss files
services:    sss files
...
```

The content of the **/etc/nsswitch.conf** file shows that selecting the **sssd** profile means that the system first uses **sssd** if information concerning one of the first five items is requested. Only if the requested information is not found in the **sssd** cache and on the server providing authentication, or if **sssd** is not running, the system looks at the local files, that is **/etc/\***.

For example, if information is requested about a user id, the user id is first searched in the **sssd** cache. If it is not found there, the local **/etc/passwd** file is consulted. Analogically, if a user's group affiliation is requested, it is first searched in the **sssd** cache and only if not found there, the **/etc/group** file is consulted.

In practice, the local **files** database does not normally get consulted at all. The only exception is the case of the **root** user, which is never handled by **sssd** but by **files**.

3. Optionally, review the contents of the **/etc/pam.d/system-auth** file:

```
# Generated by authselect on Tue Sep 11 22:59:06 2018
# Do not modify this file manually.

auth      required      pam_env.so
auth      required      pam_faildelay.so delay=2000000
auth      [default=1 ignore=ignore success=ok]
pam_succeed_if.so uid >= 1000 quiet
auth      [default=1 ignore=ignore success=ok]      pam_localuser.so
auth      sufficient    pam_unix.so nullok try_first_pass
auth      requisite     pam_succeed_if.so uid >= 1000
quiet_success
auth      sufficient    pam_sss.so forward_pass
auth      required      pam_deny.so
```

account	required	pam_unix.so
account	sufficient	pam_localuser.so
...		

Among other things, the `/etc/pam.d/system-auth` file contains information about:

- user password lockout condition
- the possibility to authenticate with a smart card
- the possibility to authenticate with fingerprints

You can modify the default profile settings by adding the following options to the **authselect select sssd** or **authselect select winbind** command, for example:

- **with-faillock**
- **with-smartcard**
- **with-fingerprint**

To see the full list of available options, see [Section 23.5, “Converting your scripts from authconfig to authselect”](#) or the `authselect-migration(7)` man page.



## NOTE

Make sure that the configuration files that are relevant for your profile are configured properly before finishing the **authselect select** procedure. For example, if the **sssd** daemon is not configured correctly and active, running **authselect select** results in only local users being able to authenticate, using `pam_unix`.

If adjusting a ready-made profile by adding one of the **authselect select** command-line options described above is not enough for your use case, you can:

- modify a ready-made profile by changing the `/etc/authselect/user-nsswitch.conf` file. For details, see [Section 23.3, “Modifying a ready-made authselect profile”](#).
- create your own custom profile. For details, see [Section 23.4, “Creating and deploying your own custom authselect profile”](#).

## 23.3. MODIFYING A READY-MADE AUTHSELECT PROFILE

As a system administrator, you can modify one of the default profiles, the **sssd**, **winbind**, or the **nis** profile, to suit your needs. You can modify any of the items in the `/etc/authselect/user-nsswitch.conf` file with the exception of:

- `passwd`
- `group`
- `netgroup`
- `automount`
- `services`

Running **authselect select profile\_name** afterwards will result in permissible changes to the profile being transferred from **/etc/authselect/user-nsswitch.conf** to the **/etc/nsswitch.conf** file but unacceptable changes being overwritten by the default profile configuration.



### IMPORTANT

Do not modify the **/etc/nsswitch.conf** file directly.

## Procedure

1. Select an **authselect** profile, for example:

```
# authselect select sssd
```

2. Edit the **/etc/authselect/user-nsswitch.conf** file.
3. Apply the changes from the **/etc/authselect/user-nsswitch.conf** file:

```
# authselect apply-changes
```

4. Optionally, review the **/etc/nsswitch.conf** file to verify that the changes from **/etc/authselect/user-nsswitch.conf** have been propagated there.

## 23.4. CREATING AND DEPLOYING YOUR OWN CUSTOM AUTHSELECT PROFILE

As a system administrator, you can create and deploy a custom profile by customizing one of the default profiles, the **sssd**, **winbind**, or the **nis** profile. This is particularly useful if [Section 23.3, “Modifying a ready-made authselect profile”](#) is not enough for your needs. When you deploy a custom profile, the profile is applied to every user logging into the given host.

## Procedure

1. Create your custom profile by using the **authselect create-profile** command. For example, to create a custom profile called **user-profile** based on the ready-made **sssd** profile but one in which you can configure the items in the **/etc/nsswitch.conf** file yourself:

```
# authselect create-profile user-profile -b sssd --symlink-meta --symlink-pam
New profile was created at /etc/authselect/custom/user-profile
```

Including the **--symlink-pam** option in the command means that PAM templates will be symbolic links to the origin profile files instead of their copy; including the **--symlink-meta** option means that meta files, such as README and REQUIREMENTS will be symbolic links to the origin profile files instead of their copy. This ensures that all future updates to the PAM templates and meta files in the original profile will be reflected in your custom profile, too.

The command has created a copy of the **/etc/nsswitch.conf** file in the **/etc/authselect/custom/user-profile/** directory.

2. Configure the **/etc/authselect/custom/user-profile/nsswitch.conf** file.

3. Select the custom profile by running the **authselect select** command, and adding `custom/name_of_the_profile` as a parameter. For example, to select the **user-profile** profile:

```
# authselect select custom/user-profile
```

Selecting the **user-profile** profile for your machine means that if the **sssd** profile is subsequently updated by Red Hat, you will benefit from all the updates with the exception of updates made to the `/etc/nsswitch.conf` file.

## Example

The following procedure shows how to create a profile based on the **sssd** profile which only consults the local static table lookup for hostnames in the `/etc/hosts` file, not in the **dns** or **myhostname** databases.

1. Edit the `/etc/nsswitch.conf` file by editing the following line:

```
hosts:      files
```

2. Create a custom profile based on **sssd** that excludes changes to `/etc/nsswitch.conf`:

```
# authselect create-profile user-profile -b sssd --symlink-meta --symlink-pam
```

3. Select the profile:

```
# authselect select custom/user-profile
```

4. Optionally, check that selecting the custom profile has

- created the `/etc/pam.d/system-auth` file according to the chosen **sssd** profile
- left the configuration in the `/etc/nsswitch.conf` unchanged:

```
hosts:      files
```



### NOTE

Running **authselect select sssd** would, in contrast, result in

```
hosts:      files dns myhostname
```

## 23.5. CONVERTING YOUR SCRIPTS FROM AUTHCONFIG TO AUTHSELECT

If you use **ipa-client-install** or **realm join** to join a domain, you can safely remove any **authconfig** call in your scripts. If this is not possible, replace each **authconfig** call with its equivalent **authselect** call. In doing that, select the correct profile and the appropriate options. In addition, edit the necessary configuration files:

- `/etc/krb5.conf`
- `/etc/sss/sss.conf` (for the **sssd** profile) or `/etc/samba/smb.conf` (for the **winbind**



profile)

Table 23.1, “Relation of authconfig options to authselect profiles” and Table 23.2, “Authselect profile option equivalents of authconfig options” show the **authselect** equivalents of **authconfig** options.

**Table 23.1. Relation of authconfig options to authselect profiles**

Authconfig options	Authselect profile
--enableldap --enableldapauth	<b>sssd</b>
--enablesssd --enablesssdauth	<b>sssd</b>
--enablekrb5	<b>sssd</b>
--enablewinbind --enablewinbindauth	<b>winbind</b>
--enablenis	<b>nis</b>

**Table 23.2. Authselect profile option equivalents of authconfig options**

Authconfig option	Authselect profile feature
--enablesmartcard	with-smartcard
--enablefingerprint	with-fingerprint
--enableecryptfs	with-ecryptfs
--enablemkhomedir	with-mkhomedir
--enablefaillock	with-faillock
--enablepamaccess	with-pamaccess
--enablewinbindkrb5	with-krb5
--enablepamaccess	with-pamaccess

Table 23.3, “Examples of authselect commands equivalents to authconfig commands” shows example transformations of Kickstart calls to **authconfig** into Kickstart calls to **authselect**.

**Table 23.3. Examples of authselect commands equivalents to authconfig commands**

authconfig command	authselect equivalent
authconfig --enableldap --enableldapauth --enablefaillock --updateall	authselect select sssd with-faillock

<code>authconfig --enablesssd --enablesssdauth --enablesmartcard --smartcardmodule=sssd --updateall</code>	<code>authselect select sssd with-smartcard</code>
<code>authconfig --enableecryptfs --enablepamaccess --updateall</code>	<code>authselect select sssd with-ecryptfs with-pamaccess</code>
<code>authconfig --enablewinbind --enablewinbindauth --winbindjoin=Administrator --updateall</code>	<code>realm join -U Administrator --client-software=winbind</code> <b>WINBINDDOMAIN</b>

## **PART V. STARTING TO USE THE SESSION RECORDING SOLUTION**

## CHAPTER 24. GETTING STARTED WITH SESSION RECORDING ON RED HAT ENTERPRISE LINUX

### 24.1. SESSION RECORDING IN RED HAT ENTERPRISE LINUX

This section introduces the session recording solution and its purpose.

The session recording solution is provided within Red Hat Enterprise Linux 8 and it is based on the **tlog** package. The **tlog** package and its associated Cockpit session player provide you with the ability to record and playback user terminal sessions. You can configure the recording to take place per user or user group via the SSSD service. All terminal input and output is captured and stored in a text-based format in the system journal.



#### IMPORTANT

Recording of the terminal input is turned off by default to not intercept raw passwords and other sensitive information.

The solution can be used for auditing user sessions on security-sensitive systems or, in the event of a security breach, reviewing recorded sessions as part of forensic analysis. System administrators are able to configure session recording locally on RHEL 8.0 systems. You can review the recorded sessions from the Cockpit web-based interface or in a terminal using the **tlog-play** command.

### 24.2. COMPONENTS OF SESSION RECORDING

There are three main components key to the session recording solution. The **tlog** utility, the SSSD service and a Cockpit embedded user interface.

#### **tlog**

The **tlog** utility is a terminal input/output (I/O) recording and playback program. It inserts itself (specifically the **tlog-rec-session** tool) between the user terminal and the user shell, and logs everything that passes through as JSON messages.

#### **SSSD**

The System Security Services Daemon (SSSD) service provides a set of daemons to manage access to remote directories and authentication mechanisms. When configuring session recording, you can use SSSD to specify, which users or user groups should **tlog** record. This can be done either from a command-line interface (CLI) or from the Cockpit web interface.

#### **Cockpit embedded web interface**

The Session Recording page is part of the Cockpit user interface. Cockpit embedded web interface for session recording enables you to manage recorded sessions.



#### IMPORTANT

You have to have administrator privileges to be able to access the recorded sessions.

## CHAPTER 25. DEPLOYING SESSION RECORDING ON RED HAT ENTERPRISE LINUX

In this section we cover how to deploy the session recording solution on a Red Hat Enterprise Linux system.

### Prerequisites

To be able to deploy the session recording solution you need to have the following packages installed: **tlog**, **SSSD**, **cockpit-session-recording**.

### 25.1. INSTALLING TLOG

To install the **tlog** packages, run:

```
# yum install tlog
```

### 25.2. INSTALLING COCKPIT-SESSION-RECORDING

The basic Cockpit packages are a part of Red Hat Enterprise Linux 8 by default. To be able to use the session recording solution, you have to install the **cockpit-session-recording** packages and start or enable Cockpit on your system:

1. Install **cockpit-session-recording**.

```
# yum install cockpit-session-recording
```

2. Start or enable Cockpit on your system:

```
# systemctl start cockpit.socket
```

or

```
# systemctl enable cockpit.socket --now
```

When you have all the necessary packages installed, you can move on to configuring your recording parameters.

### 25.3. CONFIGURING THE RECORDED USERS OR USER GROUPS WITH SSSD FROM THE CLI

If you choose to manage recorded users or user groups with SSSD, which is the recommended option, every user's original shell will be preserved.

1. To specify which users or user groups you want to record from the command-line interface (CLI), modify open the **sssd-session-recording.conf** configuration file:

```
# vi /etc/sss/conf.d/sss-session-recording.conf
```

**NOTE**

The **sssd-session-recording.conf** file is created automatically once you have opened the configuration page in the Cockpit interface.

2. Specify the scope of recorded users or user groups, either enter:
  - **none** to record no sessions.
  - **some** to record only specified sessions.
  - **all** to record all sessions.
3. In case you choose **some** as a scope of recorded users or groups, add their names divided by commas to the file.

**Example 25.1. SSSD configuration**

In the following example users **example1** and **example2**, and group **examples** have session recording enabled.

```
[session recording]
scope = some
users = example1, example2
groups = examples
```

## 25.4. CONFIGURING THE RECORDED USERS OR USER GROUPS WITH SSSD FROM WEB UI

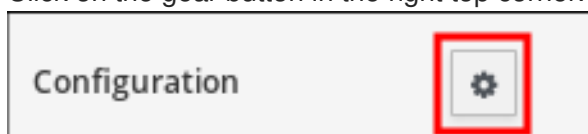
Second option for specifying recorded users or user groups using SSSD is to list them directly in the Cockpit web interface.

1. Connect to the Cockpit web interface locally by entering **localhost:9090** or by entering your IP address **<IP\_ADDRESS>:9090** to your browser.
2. Log in to the Cockpit web interface.

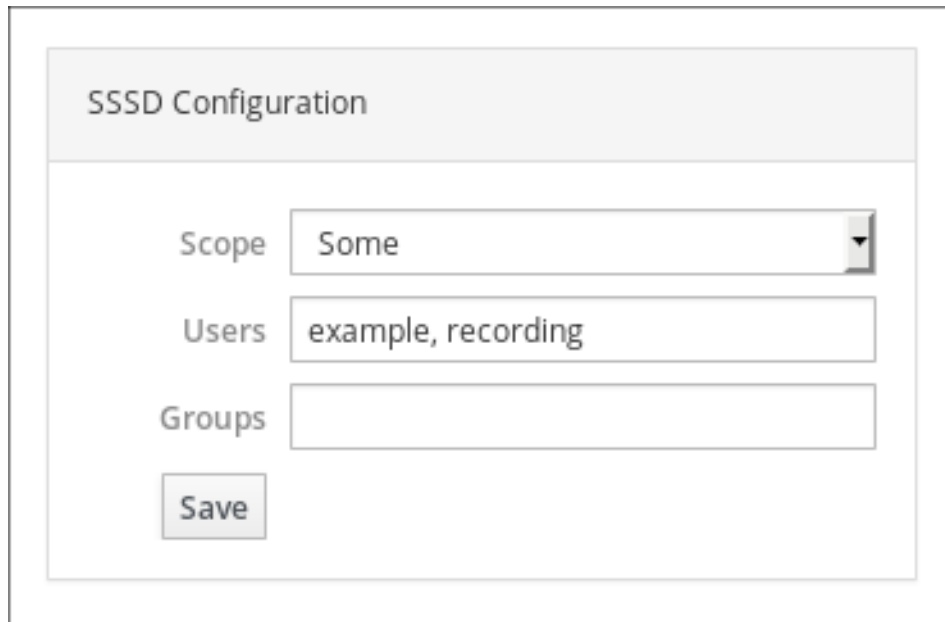
**IMPORTANT**

Your user has to have administrator privileges to be able to view te recorded sessions.

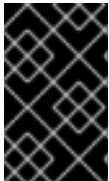
3. Go to the Session Recording page in the menu on the left of the interface.
4. Click on the gear button in the right top corner.



5. Set your parameters in the SSSD Configuration table. Names in the Users and Groups lists should be divided by commas.

**Example 25.2. Configuration of recorded users with SSSD**


The screenshot shows a window titled "SSSD Configuration". Inside, there are three input fields: "Scope" with a dropdown menu showing "Some", "Users" with a text box containing "example, recording", and "Groups" with an empty text box. Below these fields is a "Save" button.

**25.5. CONFIGURING RECORDED USERS OR USER GROUPS WITHOUT SSSD****IMPORTANT**

Be aware that this practice is not recommended to use. The preferred option is to configure your recorded users via SSSD either from command-line interface or directly from the Cockpit web interface.

If choose to manually change the user's shell, their working shell will be the one that is listed in the **tlog-rec-session.conf** configuration file.

If you do not want to use SSSD for specifying recorded user or user groups it is possible to directly change the shell of the user you want to record to **/usr/bin/tlog-rec-session**:

```
# chsh <user_name>
Changing shell for <user_name>.
New shell [</old/shell/location>]
```

## CHAPTER 26. PLAYING BACK RECORDED SESSIONS

There are two possibilities for replaying already recorded sessions. The first one is to use the **tlog-play** tool. The second option is to manage your recorded sessions from the Cockpit web interface.

### 26.1. PLAYBACK WITH COCKPIT

The Cockpit suite has a whole interface for managing recorded sessions. You can choose the session you want to review directly from the Session recording page, where the list of your recorded session is.

#### Example 26.1. Example list of recorded sessions

User	Start	End	Duration
example	2018-11-12 16:42:31	2018-11-12 16:43:09	00:38

#### Features

- The Cockpit player supports window resizing.
- Clicking a specific log under the player window moves you to the respective part of the video.
- You can slow down or speed up the playback video.
- You can change between grabbing and selection of text in the recording.
- You can change size of the recording.

### 26.2. PLAYBACK WITH **TLOG-PLAY**

Other option for playback of recorded sessions is using the **tlog-play** tool. The **tlog-play** tool is a playback program for terminal input and output recorded with the **tlog-rec** tool. It reproduces the recording of the terminal it is under, but cannot change its site. For this reason the playback terminal needs to match the recorded terminal size for proper playback. The **tlog-play** tool loads its parameters from the `/usr/local/etc/tlog/tlog-play.conf` configuration file. The parameters can be overridden with command line options described in the **tlog-play** manual pages.

### 26.3. PLAYING BACK RECORDED SESSIONS WITH **TLOG-PLAY**

Recorded sessions can be played back either from a simple file or from Systemd Journal.

#### Playing back from a file

You can play a session back from a file both during and after recording:



```
# tlog-play --reader=file --file-path=tlog.log
```

### Playing back from Journal

Generally, you can select Journal log entries for playback using Journal matches and timestamp limits, with the **-M** or **--journal-match**, **-S** or **--journal-since**, and **-U** or **--journal-until** options.

In practice however, playback from Journal is usually done with a single match against the **TLOG\_REC** Journal field. The **TLOG\_REC** field contains a copy of the **rec** field from the logged JSON data, which is a host-unique ID of the recording.

You can take the ID either from the **TLOG\_REC** field value directly, or from the **MESSAGE** field from the JSON **rec** field. Then you can play back the whole recording as follows:

```
# tlog-play -r journal -M TLOG-REC=<your-unique-host-id>
```

You can find further instructions and documentation in the **tlog-play** manual pages.