Red Hat Enterprise Linux 9

Installing Identity Management

Methods of installing IdM servers and clients

Last Updated: 2025-06-13

Red Hat Enterprise Linux 9 Installing Identity Management

Methods of installing IdM servers and clients

Legal Notice

Abstract

Depending on your environment, you can install Red Hat Identity Management (IdM) to provide DNS and Certificate Authority (CA) services, or you configure IdM to use an existing DNS and CA infrastructure. You can install IdM servers, replicas, and clients manually or by using Ansible Playbooks. Additionally, you can use a Kickstart file to automatically join a client to an IdM domain during the system installation.

Table of Contents

PROVIDING FEEDBACK ON RED HAT DOCUMENTATION

We appreciate your feedback on our documentation. Let us know how we can improve it.

Submitting feedback through Jira (account required)

1. Log in to the Jira website.
2. Click **Create** in the top navigation bar
3. Enter a descriptive title in the **Summary** field.
4. Enter your suggestion for improvement in the **Description** field. Include links to the relevant parts of the documentation.
5. Click **Create** at the bottom of the dialogue.

CHAPTER 1. PREPARING THE SYSTEM FOR IDM SERVER INSTALLATION

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

1.1. PREREQUISITES

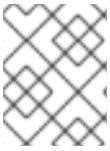
- You need **root** privileges to install an Identity Management (IdM) server on your host.

1.2. HARDWARE RECOMMENDATIONS

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

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

For larger deployments, increasing RAM is more effective than increasing disk space because much of the data is stored in cache. In general, adding more RAM leads to better performance for larger deployments due to caching. In virtualized environments, memory ballooning must be disabled or the complete RAM must be reserved for the guest IdM servers.



NOTE

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

1.3. CUSTOM CONFIGURATION REQUIREMENTS FOR IDM

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

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

1.3.1. IPv6 requirements in IdM

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

NOTE

IPv6 does not have to be enabled on the network.

1.3.2. Support for encryption types in IdM

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

List of supported encryption types

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

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

RC4 encryption types are disabled by default

The following RC4 encryption types have been disabled by default in RHEL 9, as they are considered less secure than the newer AES-128 and AES-256 encryption types:

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

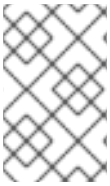
For more information about manually enabling RC4 support for compatibility with legacy Active Directory environments, see Ensuring support for common encryption types in AD and RHEL .

Support for DES and 3DES encryption has been removed

Due to security reasons, support for the DES algorithm was deprecated in RHEL 7. Single-DES (DES) and triple-DES (3DES) encryption types were removed from RHEL 8 and are not used in RHEL 9.

1.3.3. Support for system-wide cryptographic policies in IdM

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



NOTE

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

Additional Resources

- System-wide cryptographic policies

1.3.4. FIPS compliance

You can install a new IdM server or replica on a system with the Federal Information Processing Standard (FIPS) mode enabled. The only exception is a system on which the **FIPS:OSPP** cryptographic subpolicy is enabled.

To install IdM with FIPS, first enable FIPS mode on the host, then install IdM. The IdM installation script detects if FIPS is enabled and configures IdM to only use encryption types that are compliant with FIPS 140-3:

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

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

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

Migration to FIPS-compliant IdM

You cannot migrate an existing IdM installation from a non-FIPS environment to a FIPS-compliant installation. This is not a technical problem but a legal and regulatory restriction.

To operate a FIPS-compliant system, all cryptographic key material must be created in FIPS mode. Furthermore, the cryptographic key material must never leave the FIPS environment unless it is securely wrapped and never unwrapped in non-FIPS environments.

If your scenario requires a migration of a non-FIPS IdM realm to a FIPS-compliant one, you must:

1. create a new IdM realm in FIPS mode
2. perform data migration from the non-FIPS realm to the new FIPS-mode realm with a filter that blocks all key material

The migration filter must block:

- KDC master key, keytabs, and all related Kerberos key material
- user passwords
- all certificates including CA, service, and user certificates
- OTP tokens
- SSH keys and fingerprints
- DNSSEC KSK and ZSK

- all vault entries
- AD trust-related key material

Effectively, the new FIPS installation is a different installation. Even with rigorous filtering, such a migration may not pass a FIPS 140 certification. Your FIPS auditor may flag this migration.

Support for cross-forest trust with FIPS mode enabled

To establish a cross-forest trust with an Active Directory (AD) domain while FIPS mode is enabled, you must authenticate with an AD administrative account. You cannot establish a trust using a shared secret while FIPS mode is enabled.

IMPORTANT

RADIUS authentication is not FIPS compliant. Do not install IdM on a server with FIPS mode enabled if you require RADIUS authentication.

Additional Resources

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

1.4. TIME SERVICE REQUIREMENTS FOR IDM

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

1.4.1. How IdM uses chronyd for synchronization

You can use **chronyd** to keep your IdM hosts in sync with a central time source as described here.

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

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

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

NOTE

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

Additional resources

- Implementation of NTP
- Using the Chrony suite to configure NTP

1.4.2. List of NTP configuration options for IdM installation commands

You can use **chronyd** to keep your IdM hosts in sync with a central time source.

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

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

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

Additional resources

- Implementation of NTP
- Using the Chrony suite to configure NTP

1.4.3. Ensuring IdM can reference your NTP time server

You can verify if you have the necessary configurations in place for IdM to be able to synchronize with your Network Time Protocol (NTP) time server.

Prerequisites

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

Procedure

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

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

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

Verification

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

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

Additional resources

- Implementation of NTP
- Using the Chrony suite to configure NTP

1.5. MEETING DNS HOST NAME AND DNS REQUIREMENTS FOR IDM

The host name and DNS requirements for server and replica systems are outlined below and also how to verify that the systems meet the requirements.

WARNING

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

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

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

Verify the server host name

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

IMPORTANT

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

The fully qualified domain name must meet the following conditions:

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

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

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

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

Verify the forward and reverse DNS configuration

1. Obtain the IP address of the server.

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

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

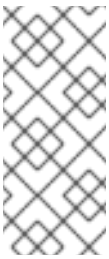
2. Verify the forward DNS configuration using the **dig** utility.

   a. Run the command **dig +short *server.idm.example.com* A**. The returned IPv4 address must match the IP address returned by **ip addr show**:

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

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

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

   NOTE

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

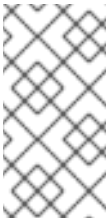
3. Verify the reverse DNS configuration (PTR records). Use the **dig** utility and add the IP address.
   If the commands below display a different host name or no host name, the reverse DNS configuration is incorrect.

   a. Run the command **dig +short -x *IPv4_address***. The output must display the server host name. For example:

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

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

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



NOTE

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

WARNING

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

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

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

```
$ dig @IP_address_of_the_DNS_forwarder . SOA
```

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

- Status: **NOERROR**
- Flags: **ra**

If either of these items is missing from the output, inspect the documentation for your DNS forwarder and verify that EDNS0 is supported and enabled.

Determine your DNS Security Extensions (DNSSEC) policy (required for integrated DNS only)

WARNING

DNSSEC is only available as Technology Preview in IdM.

DNSSEC validation is enabled in the IdM-integrated DNS server by default. If you do not require the DNSSEC feature in your IdM deployment, add the **--no-dnssec-validation** option to the **ipa-server-install --setup-dns** and **ipa-replica-install --setup-dns** commands when installing the primary IdM server and the IdM replicas.

If you do want to use DNSSEC, ensure that all DNS forwarders you want to use with the IdM DNS server comply with the DNSSEC standard. 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 DNSSEC is 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 +dnssec**:

```
;; ->>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 [...]
```

NOTE

On already deployed IdM servers, you can check whether DNSSEC validation is enabled by searching for the **dnssec-validation** boolean option in the **/etc/named/ipa-options-ext.conf** file.

Verify the /etc/hosts file

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

- The file does not contain an entry for the host. It only lists the IPv4 and IPv6 localhost entries for the host.
- The file contains an entry for the host and the file fulfills all the following conditions:
  - The first two entries are the IPv4 and IPv6 localhost entries.

- The next entry specifies the IdM server IPv4 address and host name.
- The **FQDN** of the IdM server comes before the short name of the IdM server.
- The IdM server host name is not part of the localhost entry.

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

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

1.6. PORT REQUIREMENTS FOR IDM

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

Table 1.2. IdM ports

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

NOTE

IdM uses ports 80 and 389. This is a secure practice because of the following safeguards:

- IdM normally redirects requests that arrive on port 80 to port 443. Port 80 (HTTP) is only used to provide Online Certificate Status Protocol (OCSP) responses and Certificate Revocation Lists (CRL). Both are digitally signed and therefore secured against man-in-the-middle attacks.
- Port 389 (LDAP) uses STARTTLS and Generic Security Services API (GSSAPI) for encryption.

In addition, ports 8080 and 8443 are used internally by **pki-tomcat** and leave them blocked in the firewall to prevent their use by other services. Port 749 is used for remote management of the Kerberos server and only open it if you intend to use remote management.

Table 1.3. firewalld services

Service name	For details, see:
freeipa-4	**/usr/lib/firewalld/services/freeipa-4.xml**
dns	**/usr/lib/firewalld/services/dns.xml**

1.7. OPENING THE PORTS REQUIRED BY IDM

You can open the required ports that IdM uses to communicate with its services.

Procedure

1. Verify that the **firewalld** service is running.
   - To find out if **firewalld** is currently running:

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

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

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

2. Open the required ports using the **firewall-cmd** utility. Choose one of the following options:
   a. Add the individual ports to the firewall by using the **firewall-cmd --add-port** command. For example, to open the ports in the default zone:

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

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

      ```
      # firewall-cmd --permanent --add-service={freeipa-4,dns}
      ```

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

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

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

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

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

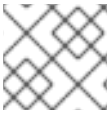
Verification

- Log in to a host on the client subnet and use the **nmap** or **nc** utilities to connect to the opened ports or run a port scan.
  - For example, to scan the ports that are required for TCP traffic:

```
$ nmap -p 80,443,389,636,88,464,53 server.idm.example.com
[...]
PORT    STATE SERVICE
53/tcp  open  domain
80/tcp  open  http
88/tcp  open  kerberos-sec
389/tcp open  ldap
443/tcp open  https
464/tcp open  kpasswd5
636/tcp open  ldapssl
```

  - To scan the ports that are required for UDP traffic:

```
# nmap -sU -p 88,464,53 server.idm.example.com
[...]
PORT    STATE         SERVICE
53/udp  open          domain
88/udp  open|filtered kerberos-sec
464/udp open|filtered kpasswd5
```



NOTE

You also have to open network-based firewalls for both incoming and outgoing traffic.

1.8. INSTALLING PACKAGES REQUIRED FOR AN IDM SERVER

The following procedure shows how to download the packages necessary for setting up the IdM environment of your choice.

Prerequisites

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

```
# subscription-manager repos --enable=rhel-9-for-x86_64-baseos-rpms
# subscription-manager repos --enable=rhel-9-for-x86_64-appstream-rpms
```

    For details on how to enable and disable specific repositories using RHSM, see Subscription Central.
  - If your RHEL system is running in the cloud, skip the registration. The required repositories are already available via the Red Hat Update Infrastructure (RHUI).

Procedure

- 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:

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

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

```
# dnf install ipa-server ipa-server-dns
```

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

```
# dnf install ipa-server ipa-server-trust-ad samba-client
```

1.9. SETTING THE CORRECT FILE MODE CREATION MASK FOR IDM INSTALLATION

The Identity Management (IdM) installation process requires that the file mode creation mask (**umask**) is set to **0022** for the **root** account. This allows users other than **root** to read files created during the installation. If a different **umask** is set, the installation of an IdM server will display a warning. If you continue with the installation, some functions of the server will not perform properly. For example, you will be unable to install an IdM replica from this server. After the installation, you can set the **umask** back to its original value.

Prerequisites

- You have **root** privileges.

Procedure

1. Optional: Display the current **umask**:

```
# umask
0027
```

2. Set the **umask** to **0022**:

```
# umask 0022
```

3. Optional: After the IdM installation is complete, set the **umask** back to its original value:

```
# umask 0027
```

1.10. ENSURING THAT FAPOLICYD RULES DO NOT BLOCK IDM INSTALLATION

If you are using the **fapolicyd** software framework on your RHEL host to control the execution of applications based on a user-defined policy, the installation of the Identity Management (IdM) server can fail. As the installation and operation requires the Java program to complete successfully, ensure

that Java and Java classes are not blocked by any **fapolicyd** rules.

For more information, see the Red Hat Knowledgebase solution fapolicy restrictions causing IdM installation failures.

1.11. OPTIONS FOR THE IDM INSTALLATION COMMANDS

Commands such as **ipa-server-install**, **ipa-replica-install**, **ipa-dns-install** and **ipa-ca-install** have numerous options you can use to supply additional information for an interactive installation. You can also use these options to script an unattended installation.

The following tables display some of the most common options for different components. Options for a specific component are shared across multiple commands. For example, you can use the **--ca-subject** option with both the **ipa-ca-install** and **ipa-server-install** commands.

For an exhaustive list of options, see the **ipa-server-install(1)**, **ipa-replica-install(1)**, **ipa-dns-install(1)** and **ipa-ca-install(1)** man pages.

Table 1.4. General options: available for ipa-server-install and ipa-replica-install

Argument	Description
-d, **--debug**	Enables debug logging for more verbose output.
-U, **--unattended**	Enables an unattended installation session that does not prompt for user input.
--hostname=*server.idm.example.com*	The fully-qualified domain name of the IdM server machine. Only numbers, lowercase alphabetic characters, and hyphens (-) are allowed.
--ip-address *127.0.0.1*	Specifies the IP address of the server. This option only accepts IP addresses associated with the local interface.
--dirsrv-config-file *<LDIF_file_name>*	The path to an LDIF file used to modify the configuration of the directory server instance.
-n *example.com*	The name of the LDAP server domain to use for the IdM domain. This is usually based on the IdM server's hostname.
-p *<directory_manager_password>*	The password of the superuser, **cn=Directory Manager**, for the LDAP service.
-a *<ipa_admin_password>*	The password for the **admin** IdM administrator account to authenticate to the Kerberos realm. For **ipa-replica-install**, use **-w** instead.
-r *<KERBEROS_REALM_NAME>*	The name of the Kerberos realm to create for the IdM domain in uppercase, such as **EXAMPLE.COM**. For **ipa-replica-install**, this specifies the name of a Kerberos realm of an existing IdM deployment.

Argument	Description
--setup-dns	Tells the installation script to set up a DNS service within the IdM domain.
--setup-ca	Install and configure a CA on this replica. If a CA is not configured, certificate operations are forwarded to another replica with a CA installed. For **ipa-server-install**, a CA is installed by default and you do not need to use this option.

Table 1.5. CA options: available for **ipa-ca-install** and **ipa-server-install**

Argument	Description
--random-serial-numbers	Enables Random Serial Numbers version 3 (RSNv3) for the IdM CA. When enabled, the CA generates fully random serial numbers for certificates and requests in PKI without range management. IMPORTANT: RSNv3 is supported only for new IdM CA installations. If enabled, it is required to use RSNv3 on all PKI services.
--ca-subject=*<SUBJECT>*	Specifies the CA certificate subject Distinguished Name (default: CN=Certificate Authority,O=REALM.NAME). Relative Distinguished Names (RDN) are in LDAP order, with the most specific RDN first.
--subject-base=*<SUBJECT>*	Specifies the subject base for certificates issued by IdM (default O=REALM.NAME). Relative Distinguished Names (RDN) are in LDAP order, with the most specific RDN first.
--external-ca	Generates a certificate signing request to be signed by an external CA.
--ca-signing-algorithm=*<ALGORITHM>*	Specifies the signing algorithm of the IdM CA certificate. Possible values are SHA1withRSA, SHA256withRSA, SHA512withRSA. The default is SHA256withRSA. Use this option with **--external-ca** if the external CA does not support the default signing algorithm.

Table 1.6. DNS options: available for **ipa-dns-install**, or for **ipa-server-install** and **ipa-replica-install** when using **--setup-dns**

Argument	Description
--forwarder=*192.0.2.1*	Specifies a DNS forwarder to use with the DNS service. To specify more than one forwarder, use this option multiple times.
--no-forwarders	Uses root servers with the DNS service instead of forwarders.

Argument	Description
--no-reverse	Does not create a reverse DNS zone when the DNS domain is set up. If a reverse DNS zone is already configured, then that existing reverse DNS zone is used. If this option is not used, then the default value is **true**. This instructs the installation script to configure reverse DNS.

Additional resources

- **ipa-server-install(1)** and **ipa-replica-install(1)** man pages on your system
- **ipa-dns-install(1)** and **ipa-ca-install(1)** man pages on your system

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

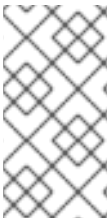
Installing a new Identity Management (IdM) server with integrated DNS has the following advantages:

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

Installing IdM with integrated DNS has certain limitations:

- IdM DNS is not meant to be used as a general-purpose DNS server. Some of the advanced DNS functions are not supported. For more information, see DNS services available in an IdM server .

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



NOTE

The default configuration for the **ipa-server-install** command is an integrated CA as the root CA. If you do not provide HTTP and LDAP server certificates using **--http-cert-file** and **--dirsrv-cert-file**, the IdM server is installed with an integrated CA. The CA is either self-signed by default or externally signed if you specify **--external-ca**).

2.1. INTERACTIVE INSTALLATION OF AN IDM SERVER WITH INTEGRATED DNS AND WITH AN INTEGRATED CA AS THE ROOT CA

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.idm.example.com]:
Please confirm the domain name [idm.example.com]:
Please provide a realm name [IDM.EXAMPLE.COM]:
```

WARNING

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

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

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

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

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

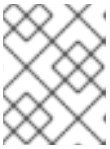
- To configure per-server DNS forwarders, enter **yes**, and then follow the instructions on the command line. The installation process will add the forwarder IP addresses to the IdM LDAP.
  - For the forwarding policy default settings, see the **--forward-policy** description in the **ipa-dns-install**(1)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

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

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

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

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



NOTE

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

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

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

8. The installation script now configures the server. Wait for the operation to complete.

9. After the installation script completes, update your DNS records in the following way:

   a. Add DNS delegation from the parent domain to the IdM DNS domain. For example, if the IdM DNS domain is ***idm.example.com***, add a name server (NS) record to the **example.com** parent domain.

   IMPORTANT

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

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

2.2. NON-INTERACTIVE INSTALLATION OF AN IDM SERVER WITH INTEGRATED DNS AND WITH AN INTEGRATED CA AS THE ROOT CA

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

Procedure

1. Run the **ipa-server-install** utility with the options to supply all the required information. The minimum required options for non-interactive installation are:

   - **--realm** to provide the Kerberos realm name
   - **--ds-password** to provide the password for the Directory Manager (DM), the Directory Server super user
   - **--admin-password** to provide the password for **admin**, the Identity Management (IdM) administrator
   - **--unattended** to let the installation process select default options for the host name and domain name

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

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

   For example:

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

2. After the installation script completes, update your DNS records in the following way:

   a. Add DNS delegation from the parent domain to the IdM DNS domain. For example, if the IdM DNS domain is ***idm.example.com***, add a name server (NS) record to the **example.com** parent domain.

   IMPORTANT

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

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

Additional resources

- For a complete list of options accepted by **ipa-server-install**, run the **ipa-server-install --help** command.
- Failover, load-balancing, and high-availability in IdM .

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

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

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

Installing IdM with integrated DNS has certain limitations:

- IdM DNS is not meant to be used as a general-purpose DNS server. Some of the advanced DNS functions are not supported. For more information, see DNS services available in an IdM server .

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

3.1. INTERACTIVE INSTALLATION OF AN IDM SERVER WITH INTEGRATED DNS AND WITH AN EXTERNAL CA AS THE ROOT CA

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.

Follow this procedure to install a server:

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

Prerequisites

- You have determined the type of the external CA to specify with the **--external-ca-type** option. See the **ipa-server-install**(1) man page for details.
- If you are using a Microsoft Certificate Services certificate authority (MS CS CA) as your external CA: you have determined the certificate profile or template to specify with the **--external-ca-profile** option. By default, the **SubCA** template is used.
  For more information about the **--external-ca-type** and **--external-ca-profile** options, see Options used when installing an IdM CA with an external CA as the root CA .

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 (MS CS) CA, also use the **--external-ca-type** option and, optionally, the **--external-ca-profile** option:

```
[root@server ~]# ipa-server-install --external-ca --external-ca-type=ms-cs --external-ca-profile=<oid>/<name>/default
```

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

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

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

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

NOTE

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

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

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

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

WARNING

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

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

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

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

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

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

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

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

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

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

NOTE

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

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

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

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

```
...

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

When this happens:

a. Submit the CSR located in **/root/ipa.csr** to the external CA. The process differs depending on the service to be used as the external CA.

b. Retrieve the issued certificate and the CA certificate chain for the issuing CA in a base 64-encoded blob (either a PEM file or a Base_64 certificate from a Windows CA). Again, the

process differs for every certificate service. Usually, a download link on a web page or in the notification email allows the administrator to download all the required certificates.

IMPORTANT

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

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

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

9. The installation script now configures the server. Wait for the operation to complete.

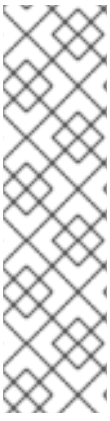
10. After the installation script completes, update your DNS records in the following way:

a. Add DNS delegation from the parent domain to the IdM DNS domain. For example, if the IdM DNS domain is ***idm.example.com***, add a name server (NS) record to the **example.com** parent domain.

IMPORTANT

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

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



NOTE

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

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

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

3.2. TROUBLESHOOTING: EXTERNAL CA INSTALLATION FAILS

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

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

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

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

What this means:
The ***_proxy** environmental variables are preventing the server from being installed.

To fix the problem:

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

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

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

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

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

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

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

Additional resources

- Failover, load-balancing, and high-availability in IdM

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

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

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

Installing IdM with integrated DNS has certain limitations:

- IdM DNS is not meant to be used as a general-purpose DNS server. Some of the advanced DNS functions are not supported. For more information, see DNS services available in an IdM server .

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

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

You need to provide the certificates required to install an Identity Management (IdM) server without a certificate authority (CA). By using the command-line options described, you can 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

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

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

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

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

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

The files provided using **--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

- If you have a PKINIT certificate, use the following 2 options:
  - **--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**
- If you do not have a PKINIT certificate and want to configure the IdM server with a local KDC with a self-signed certificate, use the following option:
  - **--no-pkinit** for disabling pkinit setup steps

Additional resources

- For details on what the certificate file formats these options accept, see the **ipa-server-install**(1) man page.
- RHEL IdM PKINIT KDC certificate and extensions (Red Hat Knowledgebase)

4.2. INTERACTIVE INSTALLATION OF AN IDM SERVER WITH INTEGRATED DNS AND WITHOUT A CA

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

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

Procedure

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

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

   See Certificates required to install an IdM server without a CA for details on the provided certificates.

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

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

NOTE

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

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

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

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

WARNING

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

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

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

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

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

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

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

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

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

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

NOTE

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

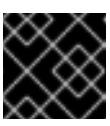
7. Enter **yes** to confirm the server configuration.

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

8. The installation script now configures the server. Wait for the operation to complete.

9. After the installation script completes, update your DNS records in the following way:

   a. Add DNS delegation from the parent domain to the IdM DNS domain. For example, if the IdM DNS domain is ***idm.example.com***, add a name server (NS) record to the **example.com** parent domain.



IMPORTANT

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

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

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

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

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

NOTE

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

For more details, see Planning your DNS services and host names .

5.1. INTERACTIVE INSTALLATION OF AN IDM SERVER WITHOUT INTEGRATED DNS AND WITH AN INTEGRATED CA AS THE ROOT CA

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

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

This procedure installs a server:

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

Procedure

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

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

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

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

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

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

WARNING

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

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

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

5. 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.

```
NetBIOS domain name [EXAMPLE]:
Do you want to configure chrony with NTP server or pool address? [no]:
```

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

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

7. The installation script now configures the server. Wait for the operation to complete.

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

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

IMPORTANT

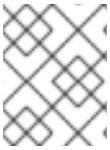
The server installation is not complete until you add the DNS records to the existing DNS servers.

Additional resources

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

5.2. NON-INTERACTIVE INSTALLATION OF AN IDM SERVER WITHOUT INTEGRATED DNS AND WITH AN INTEGRATED CA AS THE ROOT CA

You can install a server without integrated DNS or with integrated Identity Management (IdM) certificate authority (CA) as the root CA, which is the default CA configuration.



NOTE

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

Procedure

1. Run the **ipa-server-install** utility with the options to supply all the required information. The minimum required options for non-interactive installation are:

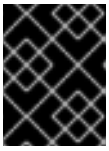
   - **--realm** to provide the Kerberos realm name
   - **--ds-password** to provide the password for the Directory Manager (DM), the Directory Server super user
   - **--admin-password** to provide the password for **admin**, the IdM administrator
   - **--unattended** to let the installation process select default options for the host name and domain name

   For example:

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

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

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



IMPORTANT

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

Additional resources

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

5.3. IDM DNS RECORDS FOR EXTERNAL DNS SYSTEMS

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

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

```
Please add records in this file to your DNS system: /tmp/ipa.system.records.6zdjqxh3.db
```

This is an example of the contents of the file:

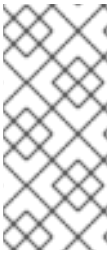
```
_kerberos-master._tcp.example.com. 86400 IN SRV 0 100 88 server.example.com.
_kerberos-master._udp.example.com. 86400 IN SRV 0 100 88 server.example.com.
_kerberos._tcp.example.com. 86400 IN SRV 0 100 88 server.example.com.
_kerberos._udp.example.com. 86400 IN SRV 0 100 88 server.example.com.
_kerberos.example.com.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

NOTE

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

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

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



NOTE

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

For more details, see Planning your DNS services and host names .

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

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

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

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

--external-ca-type=*TYPE*

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

--external-ca-profile=*PROFILE_SPEC*

Specify the certificate profile or template that you want the MS CS to apply when issuing the certificate for your IdM CA.
Note that the **--external-ca-profile** option can only be used if **--external-ca-type** is ms-cs.

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

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

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

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

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

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

6.2. INTERACTIVE INSTALLATION OF AN IDM SERVER WITHOUT INTEGRATED DNS AND WITH AN EXTERNAL CA AS THE ROOT CA

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.

Follow this procedure to install a server:

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

Prerequisites

- You have determined the type of the external CA to specify with the **--external-ca-type** option. See the **ipa-server-install**(1) man page for details.
- If you are using a Microsoft Certificate Services certificate authority (MS CS CA) as your external CA: you have determined the certificate profile or template to specify with the **--external-ca-profile** option. By default, the **SubCA** template is used.
  For more information about the **--external-ca-type** and **--external-ca-profile** options, see Options used when installing an IdM CA with an external CA as the root CA .

Procedure

1. Run the **ipa-server-install** utility with the **--external-ca** option.
   - If you are using the Microsoft Certificate Services (MS CS) CA, also use the **--external-ca-type** option and, optionally, the **--external-ca-profile** option:

     ```
     [root@server ~]# ipa-server-install --external-ca --external-ca-type=ms-cs --external-ca-profile=<oid>/<name>/default
     ```

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

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

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

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

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

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

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

   WARNING

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

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

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

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

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

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

   ```
   ...

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

   When this happens:

   a. Submit the CSR located in **/root/ipa.csr** to the external CA. The process differs depending on the service to be used as the external CA.

   b. Retrieve the issued certificate and the CA certificate chain for the issuing CA in a base 64-encoded blob (either a PEM file or a Base_64 certificate from a Windows CA). Again, the process differs for every certificate service. Usually, a download link on a web page or in the notification email allows the administrator to download all the required certificates.

IMPORTANT

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

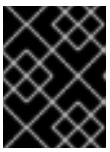
c. Run **ipa-server-install** again, this time specifying the locations and names of the newly-issued CA certificate and the CA chain files. For example:

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

7. The installation script now configures the server. Wait for the operation to complete.

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

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



IMPORTANT

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

Additional resources

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

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

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

6.3. NON-INTERACTIVE INSTALLATION OF AN IDM SERVER WITHOUT INTEGRATED DNS AND WITH AN EXTERNAL CA AS THE ROOT CA

You can install a server:

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

NOTE

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

Prerequisites

- You have determined the type of the external CA to specify with the **--external-ca-type** option. See the **ipa-server-install**(1) man page for details.
- If you are using a Microsoft Certificate Services certificate authority (MS CS CA) as your external CA: you have determined the certificate profile or template to specify with the **--external-ca-profile** option. By default, the **SubCA** template is used.
  For more information about the **--external-ca-type** and **--external-ca-profile** options, see Options used when installing an IdM CA with an external CA as the root CA .

Procedure

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

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

   If you are using a Microsoft Certificate Services (MS CS) CA, also use the **--external-ca-type** option and, optionally, the **--external-ca-profile** option. For more information, see Options used when installing an IdM CA with an external CA as the root CA.

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

   ```
   ...

   Configuring certificate server (pki-tomcatd). Estimated time: 3 minutes
     [1/11]: configuring certificate server instance
   The next step is to get /root/ipa.csr signed by your CA and re-run /usr/sbin/ipa-server-install
   as:
   /usr/sbin/ipa-server-install --external-cert-file=/path/to/signed_certificate --external-cert-
   file=/path/to/external_ca_certificate
   The ipa-server-install command was successful
   ```

When this happens:

a. Submit the CSR located in **/root/ipa.csr** to the external CA. The process differs depending on the service to be used as the external CA.

b. Retrieve the issued certificate and the CA certificate chain for the issuing CA in a base 64-encoded blob (either a PEM file or a Base_64 certificate from a Windows CA). Again, the process differs for every certificate service. Usually, a download link on a web page or in the notification email allows the administrator to download all the required certificates.

IMPORTANT

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

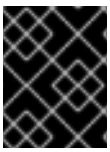
c. Run **ipa-server-install** again, this time specifying the locations and names of the newly-issued CA certificate and the CA chain files. For example:

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

3. The installation script now configures the server. Wait for the operation to complete.

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

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



IMPORTANT

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

Additional resources

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

6.4. IDM DNS RECORDS FOR EXTERNAL DNS SYSTEMS

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

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

-

```
Please add records in this file to your DNS system: /tmp/ipa.system.records.6zdjqxh3.db
```

This is an example of the contents of the file:

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

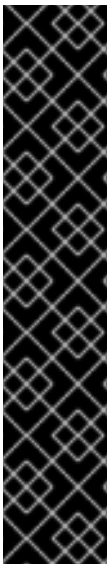
NOTE

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

CHAPTER 7. INSTALLING AN IDM DEPLOYMENT WITH KEYS AND CERTIFICATES STORED ON AN HSM

A hardware security module (HSM) provides a hardened, tamper-resistant environment for secure cryptographic processing, key generation, and encryption. You can store your key pairs and certificates for your IdM Certificate Authority (CA) and Key Recovery Authority (KRA) on an HSM. This adds physical security to the private key material.

IdM relies on the networking features of the HSM to share the keys between machines to create replicas. The HSM provides additional security without visibly affecting most IdM operations. When you use low-level tooling, the system handles certificates and keys differently, but this is seamless for most users.



IMPORTANT

Note the following:

- The HSM must be connected to a network.
- The private keys cannot leave the device.
- You cannot mix what is stored on an HSM. For example, you cannot install the KRA private keys on an HSM without also installing the CA private keys on it.
- If you use an HSM on the initial installation, then all replicas and KRAs must also use the same HSM.
- You cannot upgrade an existing installation where the keys were not generated on an HSM to an HSM-based install.

Using an HSM is largely invisible to users and administrators beyond passing additional options during the installation. The options required and any pre-installation work are HSM-specific.

7.1. SUPPORTED HARDWARE SECURITY MODULES

The following table lists hardware security modules (HSMs) supported by Identity Management (IdM):

HSM	Firmware	Appliance Software	Client Software
nCipher nShield Connect XC (High)	nShield_HSM_Firmware-12.72.1	12.71.0	SecWorld_Lin64-12.71.0
Thales TCT Luna Network HSM Luna-T7	lunafw_update-7.11.1-4	7.11.0-25	610-500244-001_LunaClient-7.11.1-5

7.2. INSTALLING AN IDM SERVER WITH AN INTEGRATED CA WITH KEYS AND CERTIFICATES STORED ON AN HSM

The default configuration for the **ipa-server-install** command is an integrated CA as the root CA.

During the installation, you must 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.

Prerequisites

- A supported networked HSM installed set up according to its vendors instructions. See Supported HSMs.
- The HSM PKCS #11 library path, **/opt/nfast/toolkits/pkcs11/libcknfast.so**.
- An available slot, token, and the token password.

Procedure

1. Run the install command, ensuring you specify the location of the PKCS #11 library, the token name, and the token password:

```
ipa-server-install -a <password> -p <dmpassword>-r <IDM.EXAMPLE.COM> -U --setup-dns
--allow-zone-overlap --no-forwarders -N --auto-reverse --random-serial-numbers –-token-
name=<HSM-TOKEN> --token-library-path=/opt/nfast/toolkits/pkcs11/libcknfast.so
```

2. Specify the token password when prompted.

Verification

1. Run **certutil** to display CA certificate information:

```
certutil -L -d /etc/pki/pki-tomcat/alias

Certificate Nickname                    Trust Attributes
                                        SSL,S/MIME,JAR/XPI

caSigningCert cert-pki-ca               CT,C,C
ocspSigningCert cert-pki-ca             ,,
Server-Cert cert-pki-ca                 u,u,u
subsystemCert cert-pki-ca               ,,
auditSigningCert cert-pki-ca            ,,P
```

Note that where there is no **u** listed under Trust Attributes for a certificate, it indicates the private keys are stored on the token. In this case, only the **Server-Cert cert-pki-ca** has the **u** flags as it is not installed on the HSM for performance reasons.

2. Verify that the keys and certificates are stored on the HSM:

```
certutil -L -d /etc/pki/pki-tomcat/alias -h <HSM-TOKEN>

Certificate Nickname                              Trust Attributes
   SSL,S/MIME,JAR/XPI

Enter Password or Pin for "<HSM-TOKEN>":
<HSM-TOKEN>:subsystemCert cert-pki-ca                u,u,u
```

```
<HSM-TOKEN>:ocspSigningCert cert-pki-ca          u,u,u
<HSM-TOKEN>:caSigningCert cert-pki-ca          CTu,Cu,Cu
<HSM-TOKEN>:auditSigningCert cert-pki-ca          u,u,Pu
```

The certificate name is prefixed with the HSM token name, which indicates that the private keys and certificates are stored on the token.

Where the keys are stored does not affect how users obtain or use certificates.

Additional resources

- Installing an IdM server: With integrated DNS, with an integrated CA as the root CA
- Installing an IdM server: Without integrated DNS, with an integrated CA as the root CA

7.3. INSTALLING AN IDM SERVER WITH AN EXTERNAL CA WITH KEYS AND CERTIFICATES STORED ON AN HSM

You can install a new Identity Management (IdM) server that uses an external certificate authority (CA) as a root CA.

During the installation, you must 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.

Prerequisites

- A supported networked HSM installed set up according to its vendors instructions. See Supported HSMs.
- The HSM PKCS #11 library path, **/opt/nfast/toolkits/pkcs11/libcknfast.so**.
- An available slot, token, and the token password.
- If you install a server without an integrated IdM CA, you must request the following certificates from a third-party authority:
  - An LDAP server certificate
  - An Apache server certificate
  - A PKINIT certificate
  - Full CA certificate chain of the CA that issued the LDAP and Apache server certificates

Procedure

1. Run the install command, ensuring you specify that you are using an external CA.

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

During the installation process, 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
```

2. To complete the certificate process, using the CSR generated by the installation utility, complete the following steps:

   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

Obtain the full certificate chain for the CA, not just the CA certificate.

3. Run the **ipa-server-install** utility again to specify the path and names of the newly-issued CA certificate and the CA chain files and the location of the PKCS #11 library, the token name, and the token password:

```
# ipa-server-install --external-cert-file=</tmp/servercert20170601.pem> --external-cert-
file=</tmp/cacert.pem> –-token-name=<HSM-TOKEN> --token-library-
path=/opt/nfast/toolkits/pkcs11/libcknfast.so
```

4. Specify the token password when prompted.

5. The installation script now configures the server. Wait for the operation to complete.

Verification

1. Run **certutil** to display CA certificate information:

```
certutil -L -d /etc/pki/pki-tomcat/alias

Certificate Nickname                    Trust Attributes
                                        SSL,S/MIME,JAR/XPI

caSigningCert cert-pki-ca               CT,C,C
ocspSigningCert cert-pki-ca             ,,
Server-Cert cert-pki-ca                 u,u,u
subsystemCert cert-pki-ca               ,,
auditSigningCert cert-pki-ca            ,,P
```

   You can see the certificates but the **,,** indicates no private keys as they are stored on the token.

2. Verify that the keys and certificates are stored on the HSM:

```
certutil -L -d /etc/pki/pki-tomcat/alias - h <HSM-TOKEN>

Certificate Nickname                    Trust Attributes
   SSL,S/MIME,JAR/XPI

Enter Password or Pin for "<HSM-TOKEN>":
<HSM-TOKEN>:subsystemCert cert-pki-ca              u,u,u
<HSM-TOKEN>:ocspSigningCert cert-pki-ca            u,u,u
<HSM-TOKEN>:caSigningCert cert-pki-ca             CTu,Cu,Cu
<HSM-TOKEN>:auditSigningCert cert-pki-ca           u,u,Pu
```

The certificate name is prefixed with the HSM token name, which indicates that the private keys and certificates are stored on the token.

Where the keys are stored does not affect how users obtain or use certificates.

Additional resources

- Installing an IdM server: With integrated DNS, with an external CA as the root CA
- Installing an IdM server: Without integrated DNS, with an external CA as the root CA

7.4. INSTALLING AN IDM REPLICA SERVER WITH KEYS AND CERTIFICATES STORED ON AN HSM

The replica installation process copies the configuration of the existing server and installs the replica based on that configuration.

Prerequisites

- A supported HSM installed and the CA keys and certificates installed on that HSM. See Installing an IdM server with an integrated CA with keys and certificates stored on an HSM .
- An available slot, token, and the token password.

Procedure

1. Run the install command, ensuring you specify the token name:

```
# ipa-replica-install --token-name=<HSM-TOKEN> --setup-ca -P admin -w <password> -U
```

2. Specify the token password when prompted.

Verification

- Verify that the keys and certificates are stored on the HSM:

```
certutil -L -d /etc/pki/pki-tomcat/alias - h <HSM-TOKEN>

Certificate Nickname                    Trust Attributes
   SSL,S/MIME,JAR/XPI

Enter Password or Pin for "<HSM-TOKEN>":
<HSM-TOKEN>:subsystemCert cert-pki-ca              u,u,u
```

```
<HSM-TOKEN>:ocspSigningCert cert-pki-ca            u,u,u
<HSM-TOKEN>:caSigningCert cert-pki-ca             CTu,Cu,Cu
<HSM-TOKEN>:auditSigningCert cert-pki-ca           u,u,Pu
```

The certificate name is prefixed with the HSM token name, which indicates that the private keys and certificates are stored on the token.

Where the keys are stored does not affect how users obtain or use certificates.

Additional resources

- Installing an IdM replica

7.5. INSTALLING A KRA SERVER WITH KEYS AND CERTIFICATES STORED ON AN HSM

To enable vaults in RHEL Identity Management (IdM), install the Key Recovery Authority (KRA) Certificate System (CS) component on a specific IdM server.

Prerequisites

- The token password.

Procedure

1. Run the install command, ensuring you specify the token name and the token password:

```
# ipa-kra-install -p <password>
```

2. Specify the token password when prompted.

Verification

- Verify that the keys and certificates are stored on the HSM:

```
certutil -L -d /etc/pki/pki-tomcat/alias - h <HSM-TOKEN>

Certificate Nickname                         Trust Attributes
   SSL,S/MIME,JAR/XPI

Enter Password or Pin for "<HSM-TOKEN>":
<HSM-TOKEN>:subsystemCert cert-pki-ca            u,u,u
<HSM-TOKEN>:ocspSigningCert cert-pki-ca          u,u,u
<HSM-TOKEN>:caSigningCert cert-pki-ca           CTu,Cu,Cu
<HSM-TOKEN>:auditSigningCert cert-pki-ca         u,u,Pu
<HSM-TOKEN>:storageCert cert-pki-kra               u,u,u
<HSM-TOKEN>:transportCert cert-pki-kra             u,u,u
<HSM-TOKEN>:auditSigningCert cert-pki-kra            u,u,Pu
```

The certificate name is prefixed with the HSM token name, which indicates that the private keys and certificates are stored on the token.

Where the keys are stored does not affect how users obtain or use certificates.

Additional resources

- Installing the Key Recovery Authority in IdM

7.6. INSTALLING A KRA CLONE WITH KEYS AND CERTIFICATES STORED ON AN HSM

By default an IdM replica does not have a KRA, unless you specified the **--setup-kra** option during the IdM client promotion.

Prerequisites

- The token password.
- A KRA server installed.

Procedure

1. To install a KRA clone, execute the following command on the replica:

```
# ipa-kra-install -p <Secret.123 >
```

2. Specify the token password when prompted.

Verification

- Verify that the keys and certificates are stored on the HSM:

```
certutil -L -d /etc/pki/pki-tomcat/alias - h <HSM-TOKEN>

Certificate Nickname                          Trust Attributes
   SSL,S/MIME,JAR/XPI

Enter Password or Pin for "<HSM-TOKEN>":
<HSM-TOKEN>:subsystemCert cert-pki-ca              u,u,u
<HSM-TOKEN>:ocspSigningCert cert-pki-ca            u,u,u
<HSM-TOKEN>:caSigningCert cert-pki-ca             CTu,Cu,Cu
<HSM-TOKEN>:auditSigningCert cert-pki-ca           u,u,Pu
<HSM-TOKEN>:storageCert cert-pki-kra                 u,u,u
<HSM-TOKEN>:transportCert cert-pki-kra               u,u,u
<HSM-TOKEN>:auditSigningCert cert-pki-kra              u,u,Pu
```

The certificate name is prefixed with the HSM token name, which indicates that the private keys and certificates are stored on the token.

Where the keys are stored does not affect how users obtain or use certificates.

Additional resources

- Installing the Key Recovery Authority in IdM

CHAPTER 8. INSTALLING AN IDM SERVER OR REPLICA WITH CUSTOM DATABASE SETTINGS FROM AN LDIF FILE

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

Prerequisites

- You have determined custom Directory Server settings that improve the performance of your IdM environment. See Adjusting IdM Directory Server performance .

Procedure

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

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

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

   a. To install an IdM server:

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

   b. To install an IdM replica:

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

Additional resources

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

CHAPTER 9. TROUBLESHOOTING IDM SERVER INSTALLATION

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

9.1. REVIEWING IDM SERVER INSTALLATION ERROR LOGS

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

- **/var/log/ipaserver-install.log**
- **/var/log/httpd/error_log**
- **/var/log/dirsrv/slapd-*INSTANCE-NAME*/access**
- **/var/log/dirsrv/slapd-*INSTANCE-NAME*/errors**

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

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

Prerequisites

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

Procedure

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

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

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

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

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

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

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

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

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

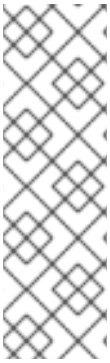
Additional resources

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

9.2. CA INSTALLATION ERROR LOG FILES ON THE FIRST IDM CA SERVER

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

Location	Description
/var/log/pki/pki-ca-spawn.*$TIME_OF_INSTALLATION*.log	High-level issues and Python traces for the **pkispawn** installation process
journalctl -u pki-tomcatd@pki-tomcat output	Errors from the **pki-tomcatd@pki-tomcat** service
/var/log/pki/pki-tomcat/ca/debug.*$DATE*.log	Large JAVA stacktraces of activity in the core of the Public Key Infrastructure (PKI) product
/var/log/pki/pki-tomcat/ca/signedAudit/ca_audit log file	Audit log of the PKI product
• **/var/log/pki/pki-tomcat/ca/system** • **/var/log/pki/pki-tomcat/ca/transactions** • **/var/log/pki/pki-tomcat/catalina.*$DATE*.log**	Low-level debug data of certificate operations for service principals, hosts, and other entities that use certificates



NOTE

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

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

Additional resources

- Reviewing CA installation errors on the first IdM CA server

9.3. REVIEWING CA INSTALLATION ERRORS ON THE FIRST IDM CA SERVER

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

Prerequisites

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

Procedure

1. To review a log file interactively, open the end of the log file using the **less** utility and use the kbd:[] arrow keys to navigate, while searching for **ScriptError** entries. The following example opens **/var/log/pki/pki-ca-spawn.*$TIME_OF_INSTALLATION*.log**.

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

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

Additional resources

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

9.4. REMOVING A PARTIAL IDM SERVER INSTALLATION

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

Example system with existing partial IdM configuration

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

The log file for this installation can be found in /var/log/ipaserver-install.log
IPA server is already configured on this system.
If you want to reinstall the IPA server, please uninstall it first using 'ipa-server-install --uninstall'.
The ipa-server-install command failed. See /var/log/ipaserver-install.log for more information
```

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

Prerequisites

- You must have **root** privileges.

Procedure

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

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

2. If you continue to experience difficulty installing an IdM server because of repeated failed installations, reinstall the operating system.
   One of the requirements for installing an IdM server is a clean system without any customization. Failed installations may have compromised the integrity of the host by unexpectedly modifying system files.

Additional resources

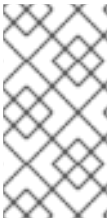
- For additional details on uninstalling an IdM server, see Uninstalling an IdM server.
- If installation attempts fail after repeated uninstallation attempts, and you have a Red Hat Technical Support subscription, open a Technical Support case at the Red Hat Customer Portal and provide an **sosreport**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

9.5. ADDITIONAL RESOURCES

- Troubleshooting IdM replica installation
- Troubleshooting IdM client installation
- Backing up and restoring IdM

CHAPTER 10. UNINSTALLING AN IDM SERVER

Follow this procedure to uninstall an Identity Management (IdM) server named **server123.idm.example.com** (server123). In the procedure, you first ensure that other servers are running critical services and that the topology will continue to be redundant before performing the uninstallation.



NOTE

If your keys and certificates are stored on a hardware security module (HSM), they are not deleted as part of the uninstall. You should refer to the documentation that came with your HSM for information on how to clear or reset your HSM to ensure that the public and private keys are deleted from the HSM.

Prerequisites

- You have **root** access to server123.
- You have an IdM administrator's credentials.

Procedure

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

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

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

2. If your IdM environment uses an integrated certificate authority (CA):

   a. Ensure that server123 is not the only **enabled** CA server:

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

  Server name: r8server.idm.example.com
  Role name: CA server
  Role status: enabled
```

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

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

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

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

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

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

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

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

If server123 is the CA renewal server, see Changing and resetting IdM CA renewal server for more information about how to move the CA renewal server role to another server.

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

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

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

3. Connect to another IdM server in the topology:

```
$ ssh idm_user@server456
```

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

```
[idm_user@server456 ~]$ kinit admin
```

5. View the DNA ID ranges assigned to the servers in the topology:

```
[idm_user@server456 ~]$ ipa-replica-manage dnarange-show
server123.idm.example.com: 1001-1500
server456.idm.example.com: 1501-2000
[...]
```

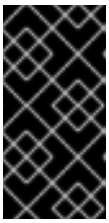
The output shows that a DNA ID range is assigned to both server123 and server456.

6. If server123 is the only IdM server in the topology with a DNA ID range assigned, create a test IdM user on server456 to ensure that the server has a DNA ID range assigned:

```
[idm_user@server456 ~]$ ipa user-add test_idm_user
```

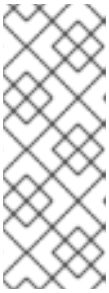
7. Delete server123.idm.example.com from the topology:

```
[idm_user@server456 ~]$ ipa server-del server123.idm.example.com
```



IMPORTANT

If deleting server123 would lead to a disconnected topology, the script warns you about it. For information about how to create a replication agreement between the remaining replicas so that the deletion can proceed, see Setting up replication between two servers using the CLI.



NOTE

Running the **ipa server-del** command removes all replication data and agreements related to server123 for both the **domain** and **ca** suffixes. This is in contrast to Domain Level 0 IdM topologies, where you initially had to remove these data by using the **ipa-replica-manage del *server123*** command. Domain Level 0 IdM topologies are those running on RHEL 7.2 and earlier. Use the **ipa domainlevel-get** command to view the current domain level.

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

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

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

Additional resources

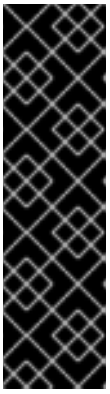
- Displaying and raising the domain level in RHEL 7 documentation
- Planning the replica topology
- Explanation of IdM CA renewal server Generating CRL on an IdM CA server

CHAPTER 11. RENAMING AN IDM SERVER

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

Procedure

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



IMPORTANT

If the server you are uninstalling is the certificate revocation list (CRL) publisher server, make another server the CRL publisher server before proceeding.

For details on how this is done in the context of a migration procedure, see the following sections:

- Stopping CRL generation on a RHEL 8 IdM CA server
- Starting CRL generation on the new RHEL 9 IdM CA server

2. Stop the existing IdM server instance.

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

3. Uninstall the existing server as described in Uninstalling an IdM server .

CHAPTER 12. UPDATING AND DOWNGRADING IDM

12.1. UPDATING IDM PACKAGES

You can use the **dnf** utility to update the Identity Management (IdM) packages on the system.

Prerequisites

- Ensure you have applied all previously released errata relevant to the RHEL system. For more information, see the How do I apply package updates to my RHEL system? KCS article.

Procedure

- Select one of the following options:
  - To update all IdM packages that are relevant for your profile and that have updates available:

```
# dnf upgrade ipa-*
```

  - To install or update packages to match the latest version available for your profile from any enabled repository:

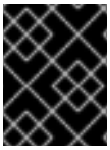
```
# dnf distro-sync ipa-*
```

After you update the IdM packages on at least one server, all other servers in the topology receive the updated schema, even if you do not update their packages. This ensures that any new entries which use the new schema can be replicated among the other servers.

WARNING

When updating multiple IdM servers, wait at least 10 minutes after updating one server before updating another server. However, the actual time required for a server's successful update depends on the topology deployed, the latency of the connections, and the number of changes generated by the update.

When two or more servers are updated simultaneously or with only short intervals between the upgrades, there is not enough time to replicate the post-upgrade data changes throughout the topology, which can result in conflicting replication events.



IMPORTANT

Upgrade to the next version only. For example, if you want to upgrade to IdM for RHEL 9.4, upgrade from IdM for RHEL 9.3. Upgrading from earlier versions can cause problems.

Additional resources

- **dnf(8)** man page on your system

12.2. DOWNGRADING IDM PACKAGES

Red Hat does not support downgrading Identity Management.

CHAPTER 13. PREPARING THE SYSTEM FOR IDM CLIENT INSTALLATION

Ensure your system meets the following conditions before you install an Identity Management (IdM) client.

13.1. SUPPORTED VERSIONS OF RHEL FOR INSTALLING IDM CLIENTS

An Identity Management deployment in which IdM servers are running on the latest minor version of Red Hat Enterprise Linux 9 supports clients that are running on the latest minor versions of:

- RHEL 7
- RHEL 8
- RHEL 9

NOTE

While other client systems, for example Ubuntu, can work with IdM 9 servers, Red Hat does not provide support for these clients.

13.2. DNS REQUIREMENTS FOR IDM CLIENTS

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

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

Additional resources

- For details on DNS requirements in IdM, see Host name and DNS requirements for IdM .

13.3. PORT REQUIREMENTS FOR IDM CLIENTS

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

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

Additional resources

- For information about which specific ports are used, see Port requirements for IdM.

13.4. IPV6 REQUIREMENTS FOR IDM CLIENTS

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

```
lookup_family_order = ipv4_only
```

Additional resources

- For more information about the **lookup_family_order** option, see the **sssd.conf(5)** man page on your system.

13.5. INSTALLING PACKAGES REQUIRED FOR AN IDM CLIENT

Installing the **ipa-client** package automatically installs other required packages as dependencies, such as the System Security Services Daemon (SSSD) packages.

Procedure

- Install the **ipa-client** package:

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

CHAPTER 14. INSTALLING AN IDM CLIENT

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

To install an Identity Management (IdM) client successfully, you must provide credentials that can be used to enroll the client.

14.1. PREREQUISITES

- You have prepared the system for IdM client installation. For details, see Preparing the system for IdM client installation.

14.2. INSTALLING A CLIENT BY USING USER CREDENTIALS: INTERACTIVE INSTALLATION

Follow this procedure to install an Identity Management (IdM) client interactively by using the credentials of an authorized user to enroll the system into the domain.

Prerequisites

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

Procedure

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

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

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

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

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

   Enabling DNS updates is useful if your client:

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

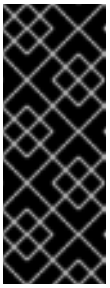
2. The installation script attempts to obtain all the required settings, such as DNS records, automatically.

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

     ```
     Client hostname: client.example.com
     ```

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

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

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



IMPORTANT

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

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

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

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

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

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

```
Client configuration complete.
```

Additional resources

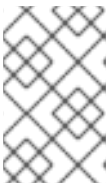
- For details on how the client installation script searches for the DNS records, see the **DNS Autodiscovery** section in the **ipa-client-install**(1) man page.

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

Follow this procedure to install an Identity Management (IdM) client interactively by using a one-time password to enroll the system into the domain.

Prerequisites

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



NOTE

The **ipa host-add <client_fqdn>** command requires that the client FQDN is resolvable through DNS. If it is not resolvable, provide the IdM client system's IP address using the **--ip address** option or alternatively, use the **--force** option.

```
$ ipa host-add client.example.com --random
 ----------------------------------------------------
 Added host "client.example.com"
 ----------------------------------------------------
  Host name: client.example.com
  Random password: W5YpARl=7M.n
  Password: True
  Keytab: False
  Managed by: server.example.com
```

NOTE

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

Procedure

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

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

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

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

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

Enabling DNS updates is useful if your client:

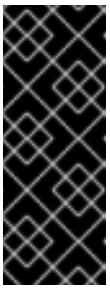
- has a dynamic IP address issued using the Dynamic Host Configuration Protocol
- has a static IP address but it has just been allocated and the IdM server does not know about it

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

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

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

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

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



IMPORTANT

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

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

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

3. The 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.

14.4. INSTALLING A CLIENT: NON-INTERACTIVE INSTALLATION

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

Options for the intended authentication method for client enrollment

The available options are:

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

The option for unattended installation

The **--unattended** option lets the installation run without requiring user confirmation.
If the SRV records are set properly in the IdM DNS zone, the script automatically discovers all the other required values. If the script cannot discover the values automatically, provide them using command-line options, such as:

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

IMPORTANT

The FQDN must be a valid DNS name:

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

- **--domain** to specify the primary DNS domain of an existing IdM deployment, such as **example.com**. The name is a lowercase version of the IdM Kerberos realm name.
- **--server** to specify the FQDN of the IdM server to connect to. When this option is used, DNS autodiscovery for Kerberos is disabled and a fixed list of KDC and Admin servers is configured. Under normal circumstances, this option is not needed as the list of servers is retrieved from the primary IdM DNS domain.
- **--realm** to specify the Kerberos realm of an existing IdM deployment. Usually it is an uppercase version of the primary DNS domain used by the IdM installation. Under normal circumstances, this option is not needed as the realm name is retrieved from the IdM server.

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

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

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

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

Additional resources

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

14.5. REMOVING PRE-IDM CONFIGURATION AFTER INSTALLING A CLIENT

The **ipa-client-install** script does not remove any previous LDAP and System Security Services Daemon (SSSD) configuration from the **/etc/openldap/ldap.conf** and **/etc/sssd/sssd.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
```

Procedure

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

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

14.6. TESTING AN IDM CLIENT

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

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

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

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

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

14.7. CONNECTIONS PERFORMED DURING AN IDM CLIENT INSTALLATION

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

Table 14.1. Requests performed during an IdM client installation

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

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

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

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

CLI post-installation operations shows the operations performed by the CLI during an IdM client post-installation deployment. Web UI post-installation operations shows the operations performed by the Web UI during an IdM client post-installation deployment.

Table 14.2. CLI post-installation operations

Operation	Protocol used	Purpose
DNS resolution against the DNS resolvers configured on the client system	DNS	To discover the IP addresses of IdM servers
Requests to ports 88 (TCP/TCP6 and UDP/UDP6) and 464 (TCP/TCP6 and UDP/UDP6) on an IdM replica	Kerberos	To obtain a Kerberos ticket; change a Kerberos password; authenticate to the IdM Web UI
JSON-RPC calls to the IdM Apache-based web-service on discovered or configured IdM servers	HTTPS	any **ipa** utility usage

Table 14.3. Web UI post-installation operations

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

Additional resources

- SSSD communication patterns for more information about how the **SSSD** daemon communicates with the services available on the IdM and Active Directory servers.
- Certmonger communication patterns for more information about how the **certmonger** daemon communicates with the services available on the IdM and Active Directory servers.

14.9. SSSD COMMUNICATION PATTERNS

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

The SSSD can be configured to communicate with multiple servers. The tables below show common communication patterns for SSSD in IdM.

Table 14.4. Communication patterns of SSSD on IdM clients when talking to IdM servers

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

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

Table 14.5. Communication patterns of SSSD on IdM servers acting as trust agents when talking to Active Directory Domain Controllers

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

Additional resources

- IdM client's communications with the server during post-installation deployment

14.10. CERTMONGER COMMUNICATION PATTERNS

Certmonger is a daemon running on Identity Management (IdM) servers and IdM clients to allow a timely renewal of SSL certificates associated with the services on the host. The Table 14.6, "Certmonger communication patterns" shows the operations performed by the **certmonger** utility on IdM servers.

Table 14.6. Certmonger communication patterns

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

Additional resources

- IdM client's communications with the server during post-installation deployment

CHAPTER 15. INSTALLING AN IDM CLIENT WITH KICKSTART

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

15.1. INSTALLING A CLIENT WITH KICKSTART

Follow this procedure to use a Kickstart file to install an Identity Management (IdM) client.

Prerequisites

- Do not start the **sshd** service prior to the kickstart enrollment. Starting **sshd** before enrolling the client generates the SSH keys automatically, but the Kickstart file in Section 15.2, "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 IdM server, and set a temporary password for the entry:

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

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

2. Create a Kickstart file with the contents described in Section 15.2, "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 IdM client.

15.2. KICKSTART FILE FOR CLIENT INSTALLATION

You can use a Kickstart file to install an Identity Management (IdM) client. The contents of the Kickstart file must meet certain requirements as outlined here.

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

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

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

Post-installation instructions for the IdM client

The post-installation instructions must include:

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

- The password which you set when pre-creating the client host on the IdM server. in Section 15.1, “Installing a client with Kickstart” .

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

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

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

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

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

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

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

15.3. TESTING AN IDM CLIENT

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

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

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

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

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

CHAPTER 16. TROUBLESHOOTING IDM CLIENT INSTALLATION

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

16.1. REVIEWING IDM CLIENT INSTALLATION ERRORS

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

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

Prerequisites

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

Procedure

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

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

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

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

Additional resources

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

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

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

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

Prerequisites

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

Procedure

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

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

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

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

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

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

Additional resources

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

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

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

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

Installation failed. Rolling back changes.
```

This failure can be caused by an empty Kerberos keytab.

Prerequisites

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

Procedure

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

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

2. Retry the IdM client installation.

Additional resources

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

16.4. RESOLVING ISSUES IF THE CLIENT INSTALLATION FAILS TO CONFIGURE AUTOMOUNT

In RHEL 7, you could configure an automount location for your client during the client installation. In RHEL 8, running the **ipa-client-install** command with the **--automount-location <raleigh>** fails to configure the automount location. However, as the rest of the installation is successful, running **/usr/sbin/ipa-client-automount <raleigh>** after the installation configures an automount location for the client correctly.

Prerequisites

- With the exception of configuring an automount location, the IdM client installation proceeded correctly. The CLI output was:

```
The ipa-client-install command was successful.
```

Procedure

- Configure the automount location:

```
/usr/sbin/ipa-client-automount -U --location <raleigh>
```

Additional resources

- man **ipa-client-automount**

16.5. ADDITIONAL RESOURCES

- To troubleshoot installing the first IdM server, see Troubleshooting IdM server installation.
- To troubleshoot installing an IdM replica, see Troubleshooting IdM replica installation .

CHAPTER 17. RE-ENROLLING AN IDM CLIENT

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

17.1. CLIENT RE-ENROLLMENT IN IDM

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

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

IMPORTANT

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

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

What happens during client re-enrollment

During re-enrollment, IdM:

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

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

Re-enroll an Identity Management (IdM) client interactively by using the credentials of an authorized user.

Procedure

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

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

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

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

Additional resources

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

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

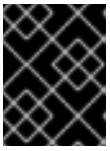
You can re-enroll an Identity Management (IdM) client non-interactively by using the **krb5.keytab** keytab file of the client system from the previous deployment. For example, re-enrollment using the client keytab is appropriate for an automated installation.

Prerequisites

- You have backed up the keytab of the client from the previous deployment on another system.

Procedure

1. Re-create the client machine with the same host name.
2. Copy the keytab file from the backup location to the re-created client machine, for example its **/tmp/** directory.



IMPORTANT

Do not put the keytab in the **/etc/krb5.keytab** file as old keys are removed from this location during the execution of the **ipa-client-install** installation script.

3. Use the **ipa-client-install** utility to re-enroll the client. Specify the keytab location with the **--keytab** option:

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

NOTE

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

17.4. TESTING AN IDM CLIENT

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

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

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

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

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

CHAPTER 18. UNINSTALLING AN IDM CLIENT

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

18.1. UNINSTALLING AN IDM CLIENT

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

Procedure

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

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

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

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

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

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

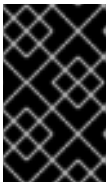
```
[root@client ~]# ipa-rmkeytab -k /path/to/keytab -r EXAMPLE.COM
```

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

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

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

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



IMPORTANT

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

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

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

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

Prerequisites

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

Procedure

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

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

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

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

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

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

     ```
     # dnf reinstall krb5-libs
     ```

     As a dependency, this command will also re-install the **krb5-workstation** package and the original version of the **/etc/krb5.conf** file.
2. Remove the **var/log/ipaclient-install.log** file if present.

Verification

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

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

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

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

CHAPTER 19. RENAMING IDM CLIENT SYSTEMS

You can change the host name of an Identity Management (IdM) client system.

WARNING

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

Renaming an Identity Management client involves:

1. Preparing the host. For details, see Preparing an IdM client for its renaming .
2. Uninstalling the IdM client from the host. For details, see Uninstalling an Identity Management client.
3. Renaming the host. For details, see Renaming the host system.
4. Installing the IdM client on the host with the new name. For details, see Installing an Identity Management client in *Installing Identity Management*.
5. Configuring the host after the IdM client installation. For details, see Re-adding services, re-generating certificates, and re-adding host groups.

19.1. PREPARING AN IDM CLIENT FOR ITS RENAMING

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

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

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

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

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

  Each service on the client system has a Kerberos principal in the form *service_name/host_name@REALM*, such as **ldap/*old-client-name.example.com@EXAMPLE.COM***.
- Identify all host groups to which the machine belongs.

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

19.2. UNINSTALLING AN IDM CLIENT

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

Procedure

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

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

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

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

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

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

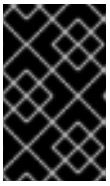
```
[root@client ~]# ipa-rmkeytab -k /path/to/keytab -r EXAMPLE.COM
```

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

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

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

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



IMPORTANT

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

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

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

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

Prerequisites

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

Procedure

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

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

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

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

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

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

     ```
     # dnf reinstall krb5-libs
     ```

     As a dependency, this command will also re-install the **krb5-workstation** package and the original version of the **/etc/krb5.conf** file.
2. Remove the **var/log/ipaclient-install.log** file if present.

Verification

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

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

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

19.4. RENAMING THE HOST SYSTEM

Rename the machine as required. For example:

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

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

19.5. RE-INSTALLING AN IDM CLIENT

Install a client on your renamed host following the procedure described in Installing a client .

19.6. RE-ADDING SERVICES, RE-GENERATING CERTIFICATES, AND RE-ADDING HOST GROUPS

Procedure

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

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

2. Generate certificates for services that had a certificate assigned in the Preparing an IdM client for its renaming. You can do this:

   - Using the IdM administration tools. See Managing Certificates for Users, Hosts, and Services.
   - Using the **certmonger** utility

3. Re-add the client to the host groups identified in the Preparing an IdM client for its renaming .

CHAPTER 20. PREPARING THE SYSTEM FOR AN IDM REPLICA INSTALLATION

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

1. Ensure the target system meets the general requirements for IdM server installation .
2. Ensure the target system meets the additional, version requirements for IdM replica installation .
3. Optional: If you are adding a RHEL 9 Identity Management (IdM) replica on which FIPS mode is enabled to a RHEL 8 IdM deployment in FIPS mode, ensure that the replica has the correct encryption types enabled.
4. Authorize the target system for enrollment into the IdM domain. For more information, see one of the following sections that best fits your needs:
   - Authorizing the installation of a replica on an IdM client
   - Authorizing the installation of a replica on a system that is not enrolled into IdM

Additional resources

- Planning the replica topology

20.1. REPLICA VERSION REQUIREMENTS

An IdM replica must be running the same or later version of IdM as other servers. For example:

- You have an IdM server installed on Red Hat Enterprise Linux 9 and it uses IdM 4.x packages.
- You must install the replica also on Red Hat Enterprise Linux 9 and use IdM version 4.x or later.

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

For details on how to display the IdM software version, see Methods for displaying IdM software version .

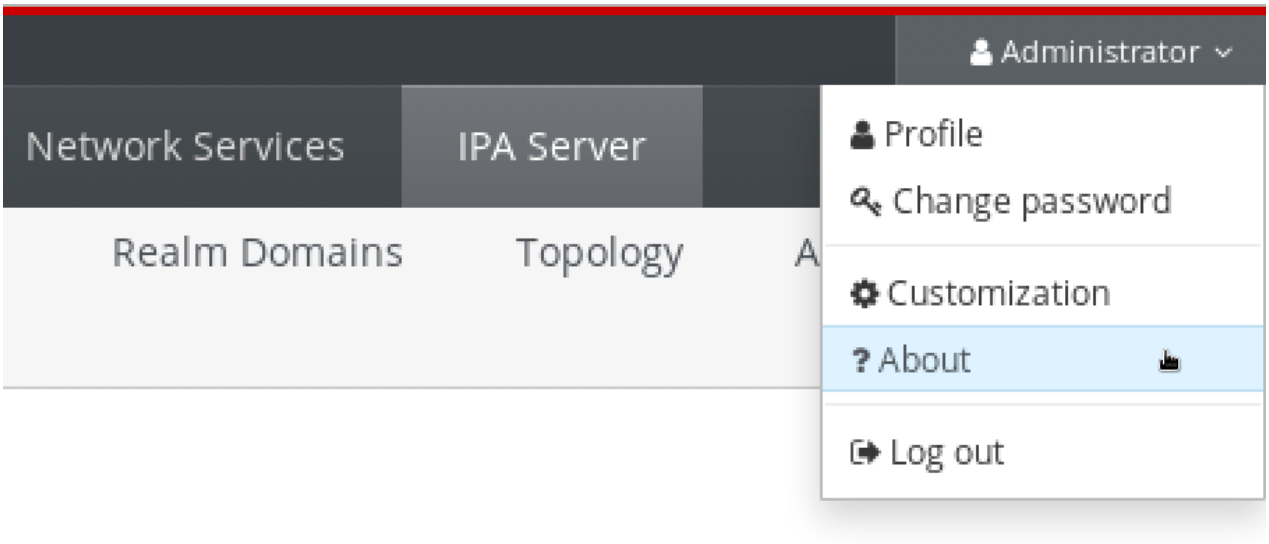
20.2. METHODS FOR DISPLAYING IDM SOFTWARE VERSION

You can display the IdM version number with:

- The IdM WebUI
- **ipa** commands
- **rpm** commands

Displaying version through the WebUI

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



Displaying version with ipa commands

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

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

Displaying version with rpm commands

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

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

20.3. ENSURING FIPS COMPLIANCE FOR A RHEL 9 REPLICA JOINING A RHEL 8 IDM ENVIRONMENT

If RHEL Identity Management (IdM) was originally installed on a RHEL 8.6 or earlier system, then the **AES HMAC-SHA1** encryption types it uses are not supported by RHEL 9 in FIPS mode by default. To add a RHEL 9 replica in FIPS mode to the deployment, you must enable these encryption keys on the RHEL 9. For more information, see the AD Domain Users unable to login in to the FIPS-compliant environment KCS solution.

20.4. AUTHORIZING THE INSTALLATION OF A REPLICA ON AN IDM CLIENT

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

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

NOTE

As of RHEL 9.5, during the installation of an IdM replica, checking if the provided Kerberos principal has the required privilege also extends to checking user ID overrides. As a result, you can deploy a replica using the credentials of an AD administrator that is configured to act as an IdM administrator.

Method 1: the ipaservers host group

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

```
$ kinit admin
```

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

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

NOTE

Membership in the **ipaservers** group grants the machine elevated privileges similar to the administrator's credentials. Therefore, in the next step, the **ipa-replica-install** utility can be run on the host successfully by a junior system administrator.

Method 2: a privileged user's credentials

Choose one of the following methods to authorize the replica installation by providing a privileged user's credentials:

- Let Identity Management (IdM) prompt you for the credentials interactively after you start the **ipa-replica-install** utility. This is the default behavior.
- Log in to the client as a privileged user immediately before running the **ipa-replica-install** utility. The default privileged user is **admin**:

```
$ kinit admin
```

Additional resources

- To start the installation procedure, see Installing an IdM replica .
- You can use an Ansible playbook to install IdM replicas. For more information, see Installing an Identity Management replica using an Ansible playbook.

20.5. AUTHORIZING THE INSTALLATION OF A REPLICA ON A SYSTEM THAT IS NOT ENROLLED INTO IDM

When installing a replica on a system that is not enrolled in the Identity Management (IdM) domain, the **ipa-replica-install** utility first enrolls the system as a client and then installs the replica components. For this scenario, choose **Method 1** or **Method 2** below to authorize the replica installation. Choose **Method 1** if one of the following applies:

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

NOTE

As of RHEL 9.5, during the installation of an IdM replica, checking if the provided Kerberos principal has the required privilege also extends to checking user ID overrides. As a result, you can deploy a replica using the credentials of an AD administrator that is configured to act as an IdM administrator.

Method 1: a random password generated on an IdM server

Enter the following commands on any server in the domain:

1. Log in as the administrator.

```
$ kinit admin
```

2. Add the external system as an IdM host. Use the **--random** option with the **ipa host-add** command to generate a random one-time password to be used for the subsequent replica installation.

```
$ ipa host-add replica.example.com --random
--------------------------------------------------
Added host "replica.example.com"
--------------------------------------------------
  Host name: replica.example.com
  Random password: W5YpARl=7M.n
  Password: True
  Keytab: False
  Managed by: server.example.com
```

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

3. Add the system to the **ipaservers** host group.

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

NOTE

Membership in the **ipaservers** group grants the machine elevated privileges similar to the administrator's credentials. Therefore, in the next step, the **ipa-replica-install** utility can be run on the host successfully by a junior system administrator that provides the generated random password.

Method 2: a privileged user's credentials

Using this method, you authorize the replica installation by providing a privileged user's credentials. The default privileged user is **admin**.
No action is required prior to running the IdM replica installation utility. Add the principal name and password options (**--principal *admin* --admin-password *password***) to the **ipa-replica-install** command directly during the installation.

Additional resources

- To start the installation procedure, see Installing an IdM replica .
- You can use an Ansible playbook to install IdM replicas. For more information, see Installing an Identity Management replica using an Ansible playbook.

CHAPTER 21. INSTALLING AN IDM REPLICA

The following sections describe how to install an Identity Management (IdM) replica interactively, by using the command line (CLI). The replica installation process copies the configuration of the existing server and installs the replica based on that configuration.

NOTE

See Installing an Identity Management server using an Ansible playbook . Use Ansible roles to consistently install and customize multiple replicas.

Interactive and non-interactive methods that do not use Ansible are useful in topologies where, for example, the replica preparation is delegated to a user or a third party. You can also use these methods in geographically distributed topologies where you do not have access from the Ansible controller node.

Prerequisites

- You are installing one IdM replica at a time. The installation of multiple replicas at the same time is not supported.
- Ensure your system is prepared for IdM replica installation .

IMPORTANT

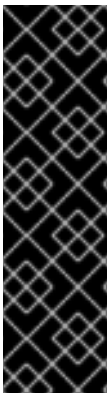
If this preparation is not performed, installing an IdM replica will fail.

21.1. INSTALLING AN IDM REPLICA WITH INTEGRATED DNS AND A CA

Follow this procedure to install an Identity Management (IdM) replica:

- With integrated DNS
- With a certificate authority (CA)

You can do this to, for example, replicate the CA service for resiliency after installing an IdM server with an integrated CA.



IMPORTANT

When configuring a replica with a CA, the CA configuration of the replica must mirror the CA configuration of the other server.

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

Including the **--setup-ca** option in the **ipa-replica-install** command copies the CA configuration of the initial server.

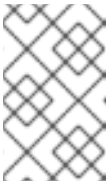
Prerequisites

- Ensure your system is prepared for an IdM replica installation .

Procedure

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

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

     NOTE

     The **ipa-replica-install** utility accepts a number of other options related to DNS settings, such as **--no-reverse** or **--no-host-dns**. For more information about them, see the **ipa-replica-install**(1) man page.

   - **--setup-ca** to include a CA on the replica

   For example, to set up a replica with an integrated DNS server and a CA that forwards all DNS requests not managed by the IdM servers to the DNS server running on IP 192.0.2.1:

   ```
   # ipa-replica-install --setup-dns --forwarder 192.0.2.1 --setup-ca
   ```

2. After the installation completes, add a DNS delegation from the parent domain to the IdM DNS domain. For example, if the IdM DNS domain is **idm.example.com**, add a name server (NS) record to the **example.com** parent domain.

   IMPORTANT

   Repeat this step each time after you install an IdM DNS server.

Next steps

- In large deployments, you might want to tune specific parameters of IdM replicas for better performance. Consult the Tuning Performance in Identity Management title to find tuning instructions to best suit your scenario.

21.2. INSTALLING AN IDM REPLICA WITH INTEGRATED DNS AND NO CA

Follow this procedure to install an Identity Management (IdM) replica:

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

Prerequisites

- Ensure your system is prepared for an IdM replica installation .

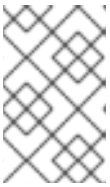
Procedure

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

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

For example, to set up a replica with an integrated DNS server that forwards all DNS requests not managed by the IdM servers to the DNS server running on IP 192.0.2.1:

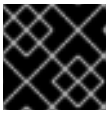
```
# ipa-replica-install --setup-dns --forwarder 192.0.2.1
```



NOTE

The **ipa-replica-install** utility accepts a number of other options related to DNS settings, such as **--no-reverse** or **--no-host-dns**. For more information about them, see the **ipa-replica-install**(1) man page.

2. After the installation completes, add a DNS delegation from the parent domain to the IdM DNS domain. For example, if the IdM DNS domain is **idm.example.com**, add a name server (NS) record to the **example.com** parent domain.



IMPORTANT

Repeat this step each time after you install an IdM DNS server.

Next steps

- In large deployments, you might want to tune specific parameters of IdM replicas for better performance. Consult the Tuning Performance in Identity Management title to find tuning instructions to best suit your scenario.

21.3. INSTALLING AN IDM REPLICA WITHOUT INTEGRATED DNS AND WITH A CA

Follow this procedure to install an Identity Management (IdM) replica:

- Without integrated DNS
- With a certificate authority (CA)

IMPORTANT

When configuring a replica with a CA, the CA configuration of the replica must mirror the CA configuration of the other server.

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

Including the **--setup-ca** option in the **ipa-replica-install** command copies the CA configuration of the initial server.

Prerequisites

- Ensure your system is prepared for an IdM replica installation .

Procedure

1. Enter **ipa-replica-install** with the **--setup-ca** option.

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

2. Add the newly created IdM DNS service records to your DNS server:
   a. Export the IdM DNS service records into a file in the **nsupdate** format:

```
$ ipa dns-update-system-records --dry-run --out dns_records_file.nsupdate
```

   b. Submit a DNS update request to your DNS server using the **nsupdate** utility and the **dns_records_file.nsupdate** file. For more information, see Updating External DNS Records Using nsupdate in RHEL 7 documentation. Alternatively, refer to your DNS server documentation for adding DNS records.

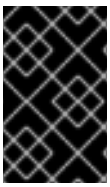
Next steps

- In large deployments, you might want to tune specific parameters of IdM replicas for better performance. Consult the Tuning Performance in Identity Management title to find tuning instructions to best suit your scenario.

21.4. INSTALLING AN IDM REPLICA WITHOUT INTEGRATED DNS AND WITHOUT A CA

Follow this procedure to install an Identity Management (IdM) replica:

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



IMPORTANT

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

Prerequisites

- Ensure your system is prepared for an IdM replica installation .

Procedure

- Enter **ipa-replica-install**, and provide the required certificate files by adding these options:
  - **--dirsrv-cert-file**
  - **--dirsrv-pin**
  - **--http-cert-file**

- **--http-pin**

For details about the files that are provided using these options, see Section 4.1, "Certificates required to install an IdM server without a CA".

For example:

```
# ipa-replica-install \
   --dirsrv-cert-file /tmp/server.crt \
   --dirsrv-cert-file /tmp/server.key \
   --dirsrv-pin secret \
   --http-cert-file /tmp/server.crt \
   --http-cert-file /tmp/server.key \
   --http-pin secret
```

NOTE

Do not add the **--ca-cert-file** option. The **ipa-replica-install** utility takes this part of the certificate information automatically from the first server you installed.

Next steps

- In large deployments, you might want to tune specific parameters of IdM replicas for better performance. Consult the Tuning Performance in Identity Management title to find tuning instructions to best suit your scenario.

21.5. INSTALLING AN IDM HIDDEN REPLICA

A hidden (unadvertised) replica is an Identity Management (IdM) server that has all services running and available. However, it has no SRV records in DNS, and LDAP server roles are not enabled. Therefore, clients cannot use service discovery to detect these hidden replicas.

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

Prerequisites

- Ensure your system is prepared for an IdM replica installation .

Procedure

- To install a hidden replica, use the following command:

```
ipa-replica-install --hidden-replica
```

Note that the command installs a replica without DNS SRV records and with disabled LDAP server roles.

You can also change the mode of existing replica to hidden. For details, see Demotion and promotion of hidden replicas

21.6. TESTING AN IDM REPLICA

After creating a replica, check if the replica replicates data as expected. You can use the following procedure.

Procedure

1. Create a user on the new replica:

```
[admin@new_replica ~]$ ipa user-add test_user
```

2. Make sure the user is visible on another replica:

```
[admin@another_replica ~]$ ipa user-show test_user
```

21.7. CONNECTIONS PERFORMED DURING AN IDM REPLICA INSTALLATION

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

Table 21.1. Requests performed during an IdM replica installation

Operation	Protocol used	Purpose
DNS resolution against the DNS resolvers configured on the client system	DNS	To discover the IP addresses of IdM servers
Requests to ports 88 (TCP/TCP6 and UDP/UDP6) on the discovered IdM servers	Kerberos	To obtain a Kerberos ticket
JSON-RPC calls to the IdM Apache-based web-service on the discovered or configured IdM servers	HTTPS	IdM client enrollment; replica keys retrieval and certificate issuance if required
Requests over TCP/TCP6 to port 389 on the IdM server, using SASL GSSAPI authentication, plain LDAP, or both	LDAP	IdM client enrollment; CA certificate chain retrieval; LDAP data replication
Requests over TCP/TCP6 to port 22 on IdM server	SSH	To check if the connection is working
(optionally) Access over port 8443 (TCP/TCP6) on the IdM servers	HTTPS	To administer the Certificate Authority on the IdM server (only during IdM server and replica installation)

CHAPTER 22. TROUBLESHOOTING IDM REPLICA INSTALLATION

The following sections describe the process for gathering information about a failing IdM replica installation, and how to resolve some common installation issues.

22.1. IDM REPLICA INSTALLATION ERROR LOG FILES

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

- **/var/log/ipareplica-install.log**
- **/var/log/ipareplica-conncheck.log**
- **/var/log/ipaclient-install.log**
- **/var/log/httpd/error_log**
- **/var/log/dirsrv/slapd-*INSTANCE-NAME*/access**
- **/var/log/dirsrv/slapd-*INSTANCE-NAME*/errors**
- **/var/log/ipaserver-install.log**

The replica installation process also appends debugging information to the following log files on the IdM **server** the replica is contacting:

- **/var/log/httpd/error_log**
- **/var/log/dirsrv/slapd-*INSTANCE-NAME*/access**
- **/var/log/dirsrv/slapd-*INSTANCE-NAME*/errors**

The last line of each log file reports success or failure, and **ERROR** and **DEBUG** entries provide additional context.

Additional resources

- Reviewing IdM replica installation errors

22.2. REVIEWING IDM REPLICA INSTALLATION ERRORS

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

Prerequisites

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

Procedure

1. Use the **tail** command to display the latest errors from the primary log file **/var/log/ipareplica-install.log**. The following example displays the last 10 lines.

```
[user@replica ~]$ sudo tail -n 10 /var/log/ipareplica-install.log
[sudo] password for user:
  func(installer)
File "/usr/lib/python3.6/site-packages/ipaserver/install/server/replicainstall.py", line 424, in
decorated
  func(installer)
File "/usr/lib/python3.6/site-packages/ipaserver/install/server/replicainstall.py", line 785, in
promote_check
  ensure_enrolled(installer)
File "/usr/lib/python3.6/site-packages/ipaserver/install/server/replicainstall.py", line 740, in
ensure_enrolled
  raise ScriptError("Configuration of client side components failed!")

2020-05-28T18:24:51Z DEBUG The ipa-replica-install command failed, exception:
ScriptError: Configuration of client side components failed!
2020-05-28T18:24:51Z ERROR Configuration of client side components failed!
2020-05-28T18:24:51Z ERROR The ipa-replica-install command failed. See
/var/log/ipareplica-install.log for more information
```

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

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

3. Optional: While **/var/log/ipareplica-install.log** is the primary log file for a replica installation, you can gather additional troubleshooting information by repeating this review process with additional files on the replica and the server.

   On the replica:

```
[user@replica ~]$ sudo less -N +G /var/log/ipareplica-conncheck.log
[user@replica ~]$ sudo less -N +G /var/log/ipaclient-install.log
[user@replica ~]$ sudo less -N +G /var/log/httpd/error_log
[user@replica ~]$ sudo less -N +G /var/log/dirsrv/slapd-INSTANCE-NAME/access
[user@replica ~]$ sudo less -N +G /var/log/dirsrv/slapd-INSTANCE-NAME/errors
[user@replica ~]$ sudo less -N +G /var/log/ipaserver-install.log
```

   On the server:

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

Additional resources

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

22.3. CA INSTALLATION ERROR LOG FILES ON AN IDM REPLICA

Installing the Certificate Authority (CA) service on an Identity Management (IdM) replica appends debugging information to several locations on the replica and the IdM server the replica communicates with.

Table 22.1. On the replica (in order of recommended priority):

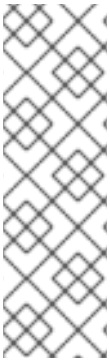
Location	Description
/var/log/pki/pki-ca-spawn.*$TIME_OF_INSTALLATION*.log	High-level issues and Python traces for the **pkispawn** installation process
journalctl -u pki-tomcatd@pki-tomcat output	Errors from the **pki-tomcatd@pki-tomcat** service
/var/log/pki/pki-tomcat/ca/debug.*$DATE*.log	Large JAVA stacktraces of activity in the core of the Public Key Infrastructure (PKI) product
/var/log/pki/pki-tomcat/ca/signedAudit/ca_audit	Audit log of the PKI product
• **/var/log/pki/pki-tomcat/ca/system** • **/var/log/pki/pki-tomcat/ca/transactions** • **/var/log/pki/pki-tomcat/catalina.*$DATE*.log**	Low-level debug data of certificate operations for service principals, hosts, and other entities that use certificates

On the server contacted by the replica:

- **/var/log/httpd/error_log** log file

Installing the CA service on an existing IdM replica also writes debugging information to the following log file:

- **/var/log/ipareplica-ca-install.log** log file



NOTE

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

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

Additional resources

- Reviewing IdM CA installation errors

22.4. REVIEWING CA INSTALLATION ERRORS ON AN IDM REPLICA

To troubleshoot a failing IdM CA installation, review the errors at the ends of the CA installation error log files on an IdM replica and use this information to resolve any corresponding issues.

Prerequisites

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

Procedure

1. To review a log file interactively, open the end of the log file using the **less** utility and use the kbd:[] arrow keys to navigate, while searching for **ScriptError** entries. The following example opens **/var/log/pki/pki-ca-spawn.*$TIME_OF_INSTALLATION*.log**.

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

2. Gather additional troubleshooting information by repeating this review process with all the CA installation error log files.

Additional resources

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

22.5. REMOVING A PARTIAL IDM REPLICA INSTALLATION

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

Example system with existing partial IdM configuration

```
[root@server ~]# ipa-replica-install
Your system may be partly configured.
Run /usr/sbin/ipa-server-install --uninstall to clean up.

IPA server is already configured on this system.
If you want to reinstall the IPA server, please uninstall it first using 'ipa-server-install --uninstall'.
The ipa-replica-install command failed. See /var/log/ipareplica-install.log for more information
```

To resolve this issue, uninstall IdM software from the replica, remove the replica from the IdM topology, and retry the installation process.

Prerequisites

- You must have **root** privileges.

Procedure

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

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

2. On all other servers in the topology, use the **ipa server-del** command to delete any references to the replica that did not install properly.

```
[root@other-replica ~]# ipa server-del replica.idm.example.com
```

3. Attempt installing the replica.

4. If you continue to experience difficulty installing an IdM replica because of repeated failed installations, reinstall the operating system.
One of the requirements for installing an IdM replica is a clean system without any customization. Failed installations may have compromised the integrity of the host by unexpectedly modifying system files.

Additional resources

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

22.6. RESOLVING INVALID CREDENTIAL ERRORS

If an IdM replica installation fails with an **Invalid credentials** error, the system clocks on the hosts might be out of sync with each other:

```
[27/40]: setting up initial replication
Starting replication, please wait until this has completed.
Update in progress, 15 seconds elapsed
[ldap://server.example.com:389] reports: Update failed! Status: [49 - LDAP error: Invalid credentials]

[error] RuntimeError: Failed to start replication
Your system may be partly configured.
Run /usr/sbin/ipa-server-install --uninstall to clean up.

ipa.ipapython.install.cli.install_tool(CompatServerReplicaInstall): ERROR    Failed to start replication
ipa.ipapython.install.cli.install_tool(CompatServerReplicaInstall): ERROR    The ipa-replica-install command failed. See /var/log/ipareplica-install.log for more information
```

If you use the **--no-ntp** or **-N** options to attempt the replica installation while clocks are out of sync, the installation fails because services are unable to authenticate with Kerberos.

To resolve this issue, synchronize the clocks on both hosts and retry the installation process.

Prerequisites

- You must have **root** privileges to change system time.

Procedure

1. Synchronize the system clocks manually or with **chronyd**.

   Synchronizing manually

   Display the system time on the server and set the replica's time to match.

   ```
   [user@server ~]$ date
   Thu May 28 21:03:57 EDT 2020

   [user@replica ~]$ sudo timedatectl set-time '2020-05-28 21:04:00'
   ```

   - **Synchronizing with chronyd**:
     See Using the Chrony suite to configure NTP to configure and set system time with **chrony** tools.

2. Attempt the IdM replica installation again.

Additional resources

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

22.7. ADDITIONAL RESOURCES

- Troubleshooting the first IdM server installation
- Troubleshooting IdM client installation
- Backing up and restoring IdM

CHAPTER 23. UNINSTALLING AN IDM REPLICA

As an IdM administrator, you can remove an Identity Management (IdM) replica from the topology. For more information, see Uninstalling an IdM server.

CHAPTER 24. MANAGING REPLICATION TOPOLOGY

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

Additional resources

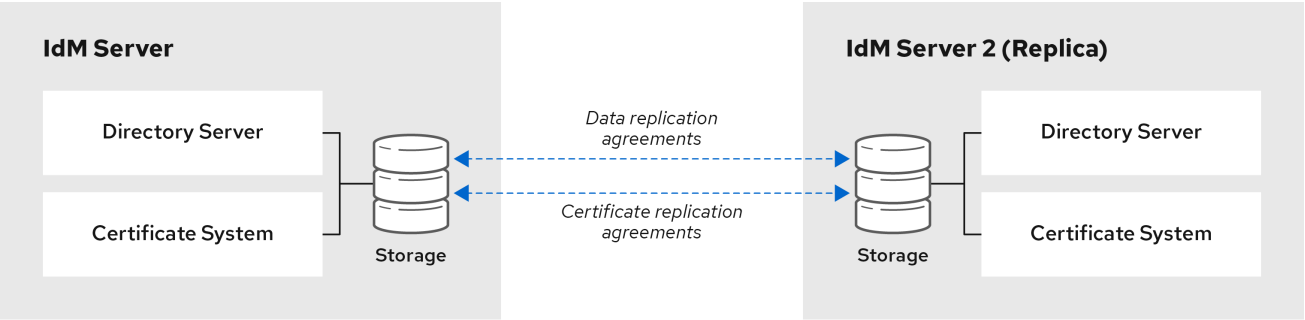
- Planning the replica topology
- Uninstalling an IdM server
- Failover, load-balancing, and high-availability in IdM
- Tuning performance in Identity Management

24.1. REPLICATION AGREEMENTS BETWEEN IDM REPLICAS

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

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

Figure 24.1. Server and replica agreements



IdM uses two types of replication agreements:

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

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

24.2. TOPOLOGY SUFFIXES

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

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

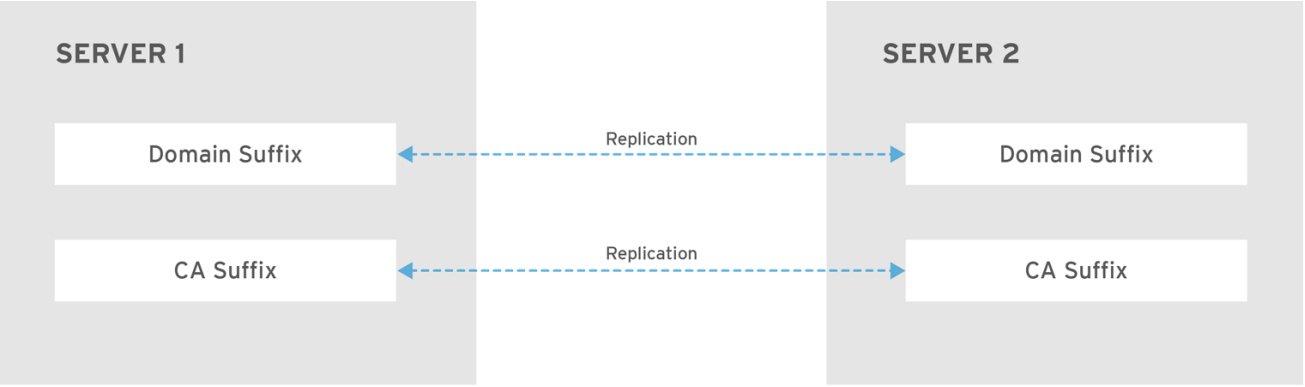
The domain suffix: dc=*example*,dc=*com*

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

The ca suffix: o=ipaca

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

Figure 24.2. Topology suffixes



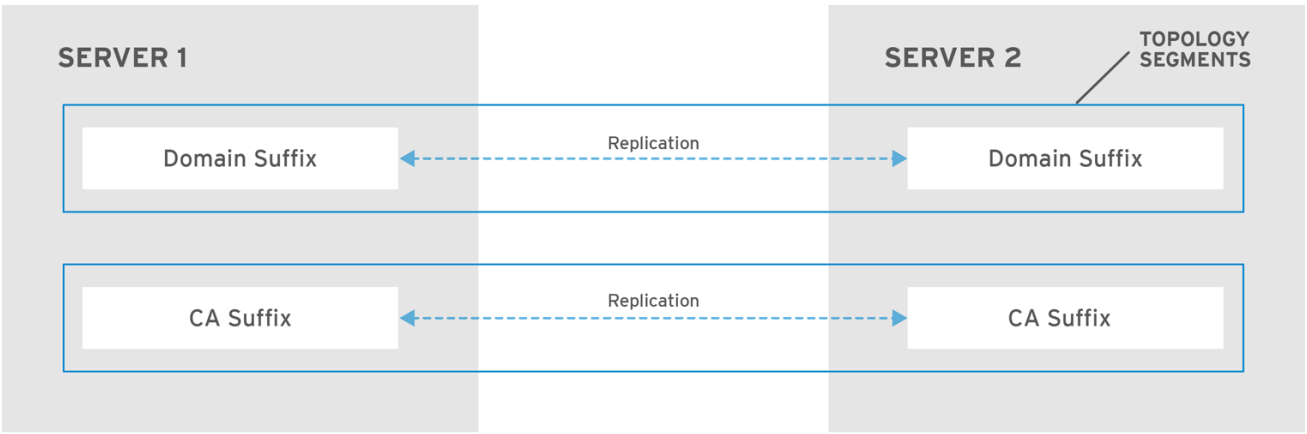
An initial topology replication agreement is set up between two servers by the **ipa-replica-install** script when installing a new replica.

24.3. TOPOLOGY SEGMENTS

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

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

Figure 24.3. Topology segments



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

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

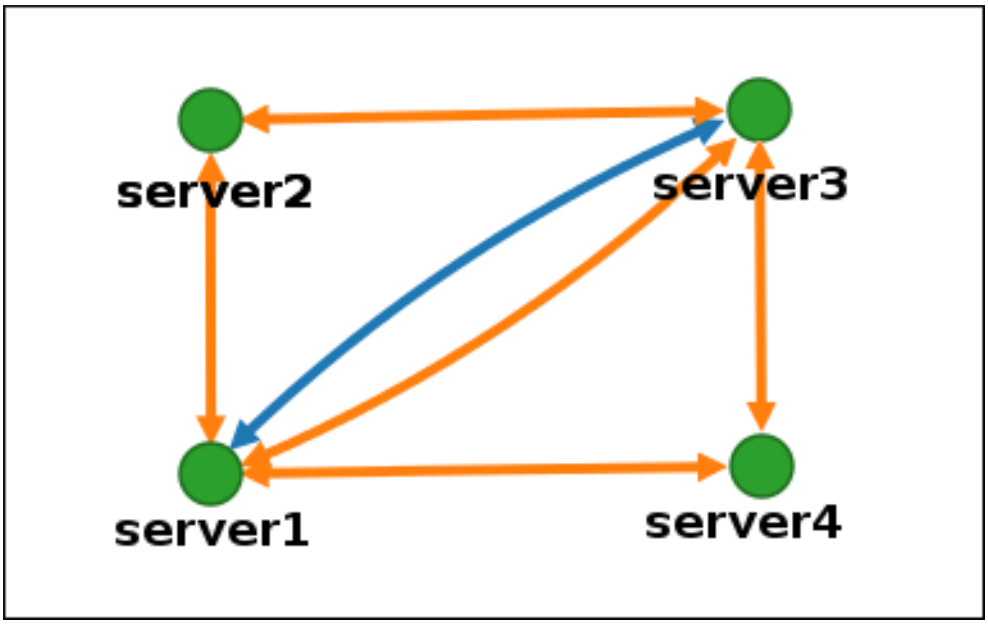
Interpreting the topology graph

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

Topology graph example: recommended topology

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

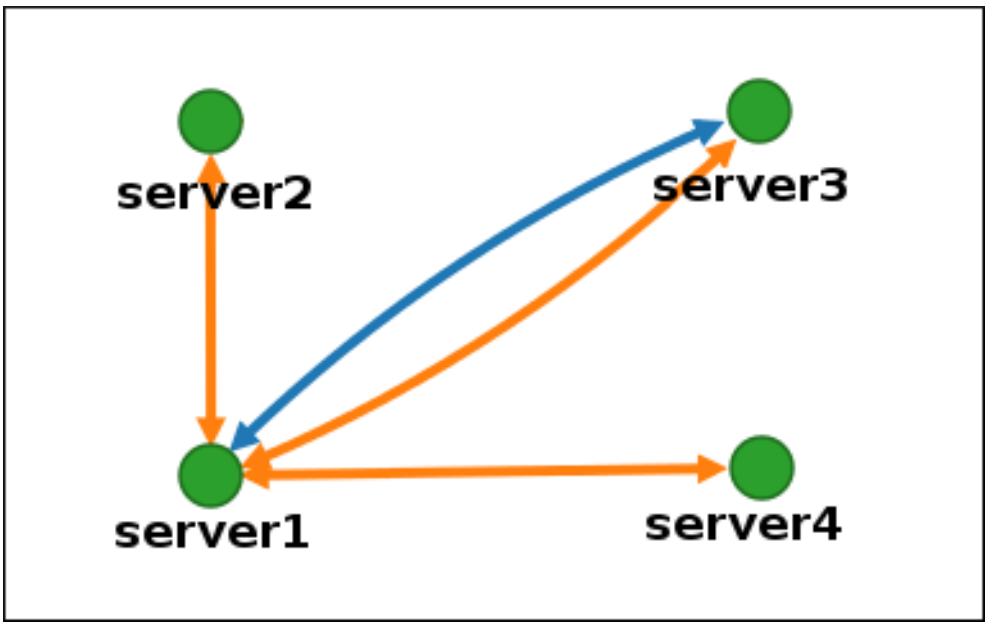
Figure 24.4. Recommended topology example



Topology graph example: discouraged topology

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

Figure 24.5. Discouraged topology example: Single Point of Failure

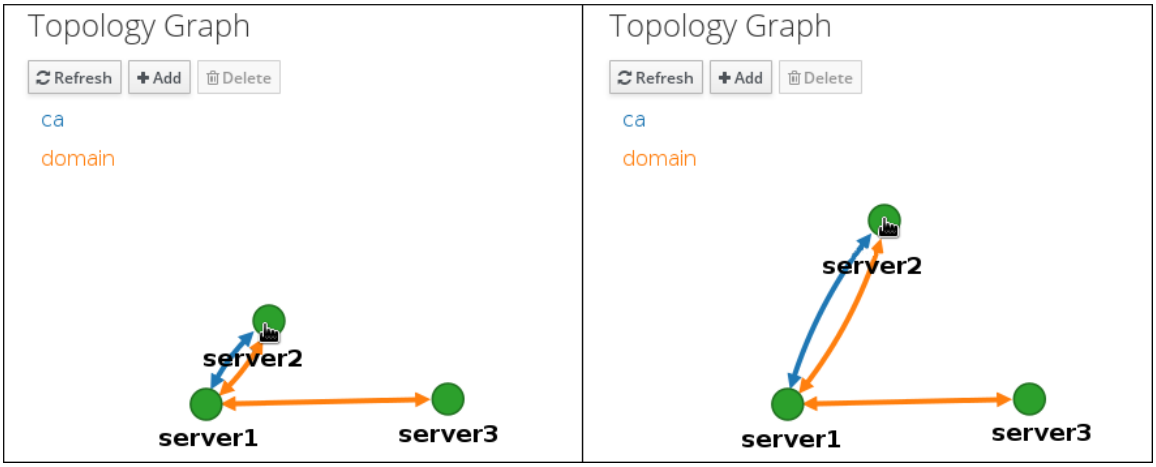


Prerequisites

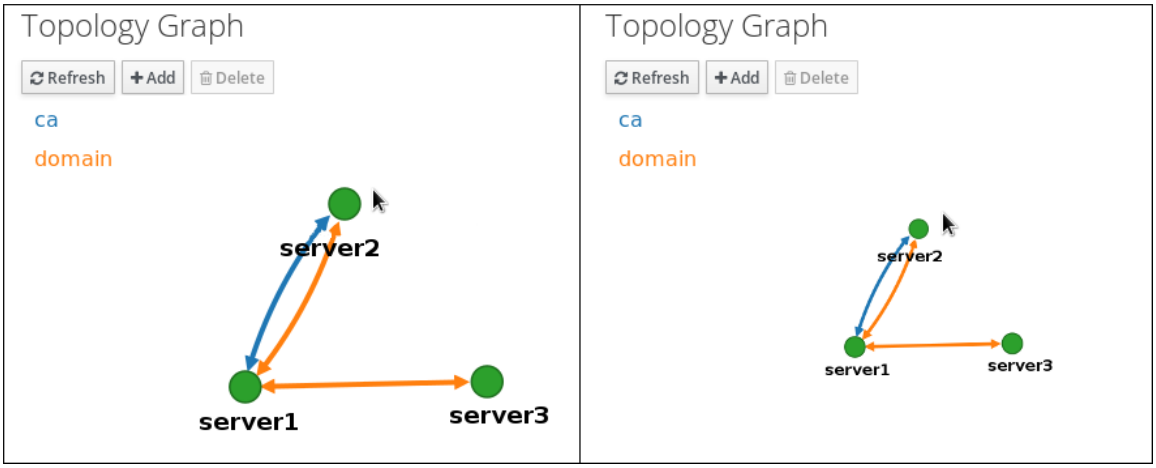
- You are logged in as an IdM administrator.

Procedure

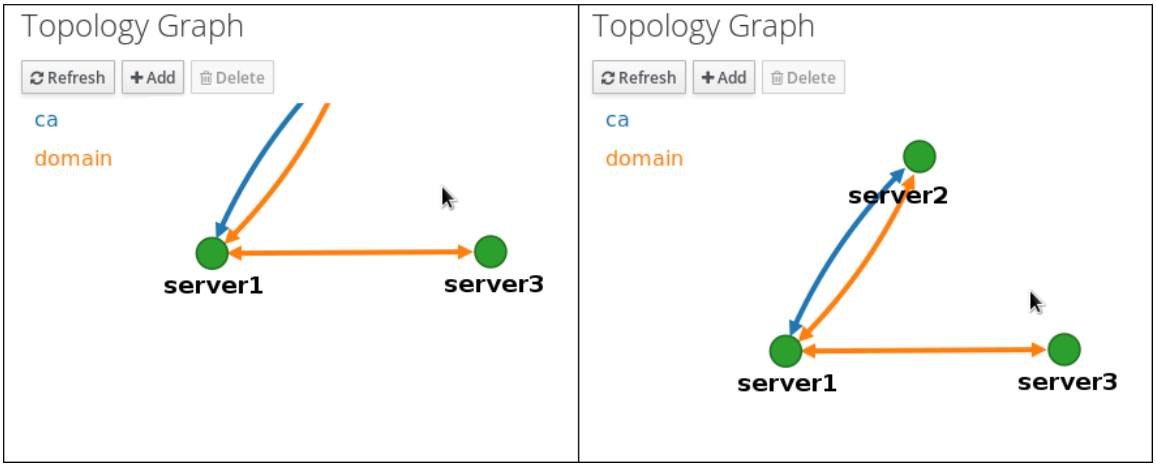
1. Select **IPA Server → Topology → Topology Graph**.
2. Make changes to the topology:
   - You can move the topology graph nodes using the left mouse button:



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



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



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

24.5. VIEWING TOPOLOGY SUFFIXES USING THE CLI

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

Procedure

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

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

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

Additional resources

- Topology suffixes

24.6. VIEWING TOPOLOGY SEGMENTS USING THE CLI

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

Procedure

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

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

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

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

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

Additional resources

- Topology segments

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

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

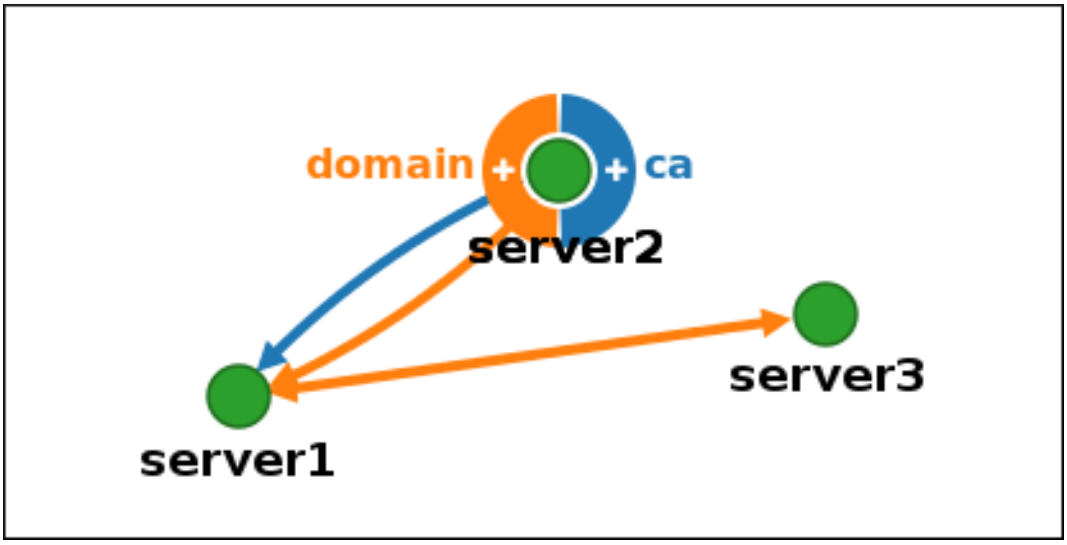
Prerequisites

- You are logged in as an IdM administrator.

Procedure

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

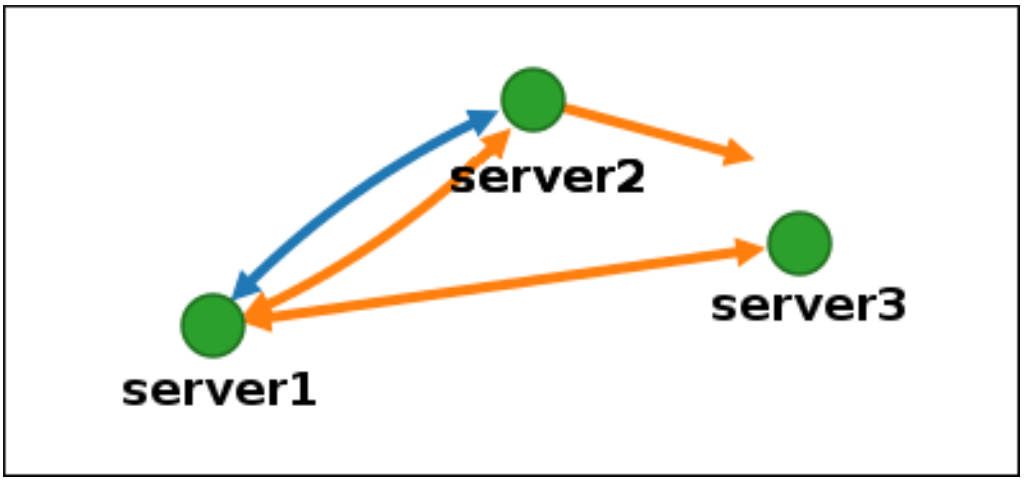
   Figure 24.6. Domain or CA options



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

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

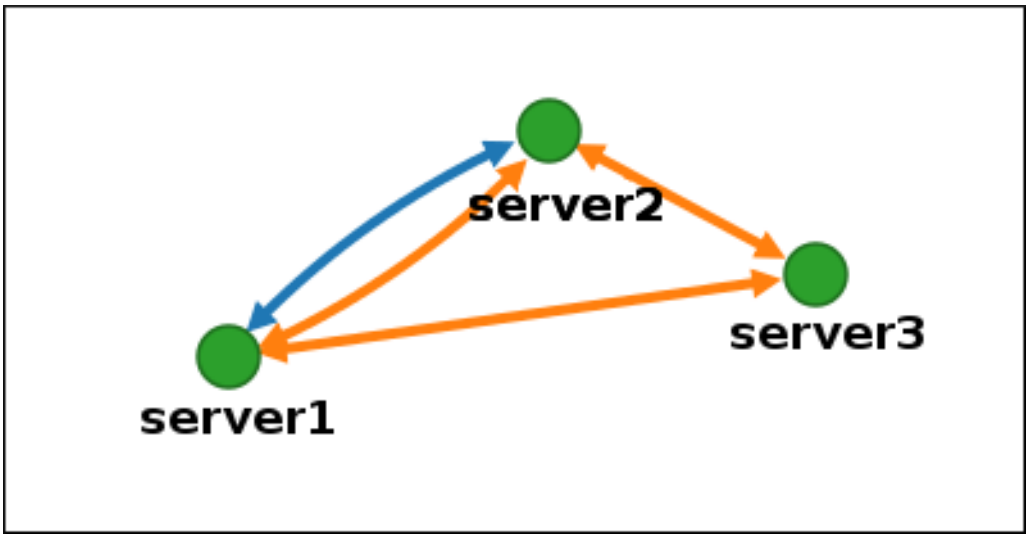
   Figure 24.7. Creating a new segment



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

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

Figure 24.8. New segment created



24.8. STOPPING REPLICATION BETWEEN TWO SERVERS USING THE WEB UI

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

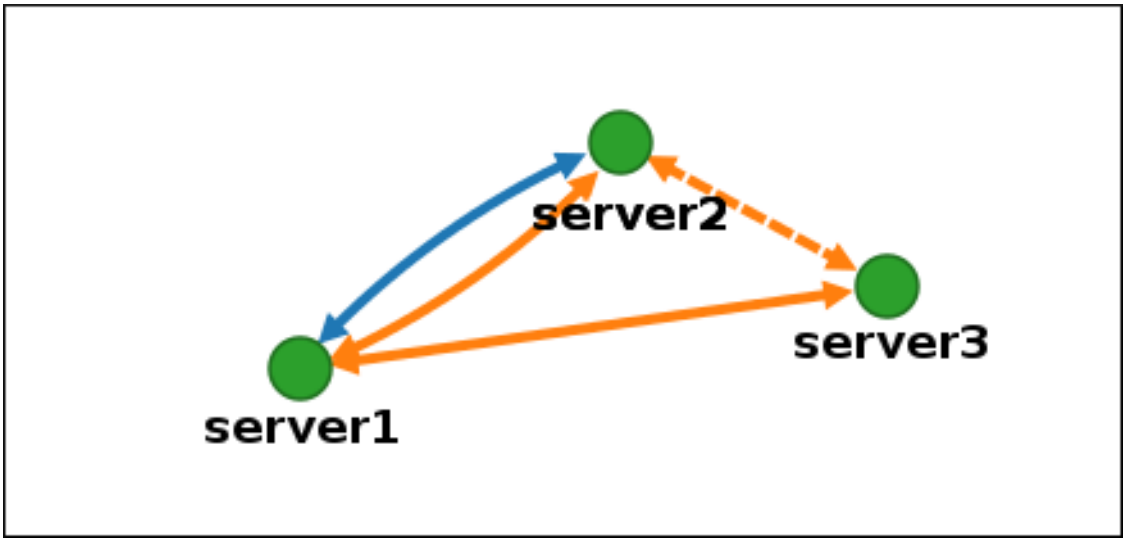
Prerequisites

- You are logged in as an IdM administrator.

Procedure

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

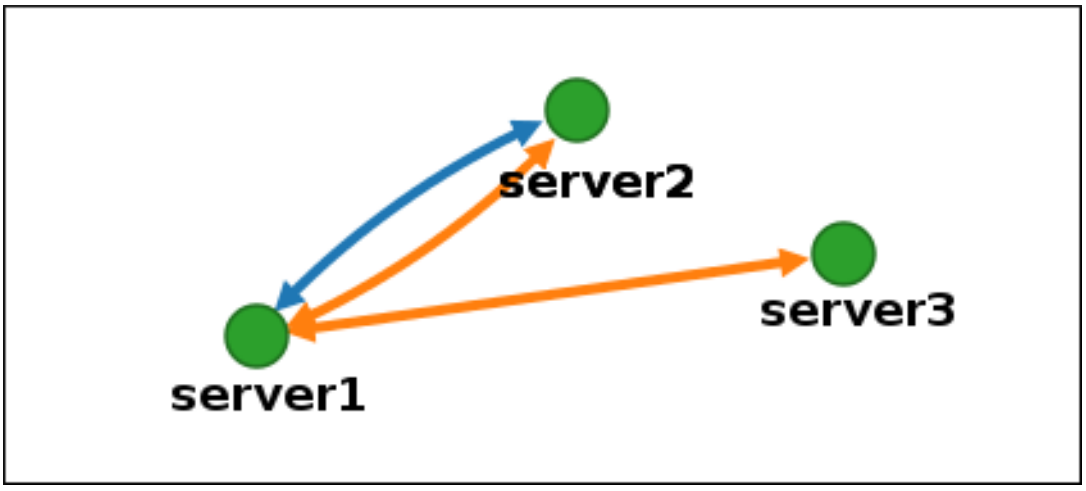
   Figure 24.9. Topology segment highlighted

2. Click **Delete**.

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

Figure 24.10. Topology segment deleted



24.9. SETTING UP REPLICATION BETWEEN TWO SERVERS USING THE CLI

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

Prerequisites

- You have the IdM administrator credentials.

Procedure

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

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

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

Verification

- Verify that the new segment is configured:

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

24.10. STOPPING REPLICATION BETWEEN TWO SERVERS USING THE CLI

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

Prerequisites

- You have the IdM administrator credentials.

Procedure

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

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

...

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

   Locate the required segment in the output.

2. Remove the topology segment joining the two servers:

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

   Deleting the segment removes the replication agreement.

Verification

- Verify that the segment is no longer listed:

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

...

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

24.11. REMOVING SERVER FROM TOPOLOGY USING THE WEB UI

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

Prerequisites

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

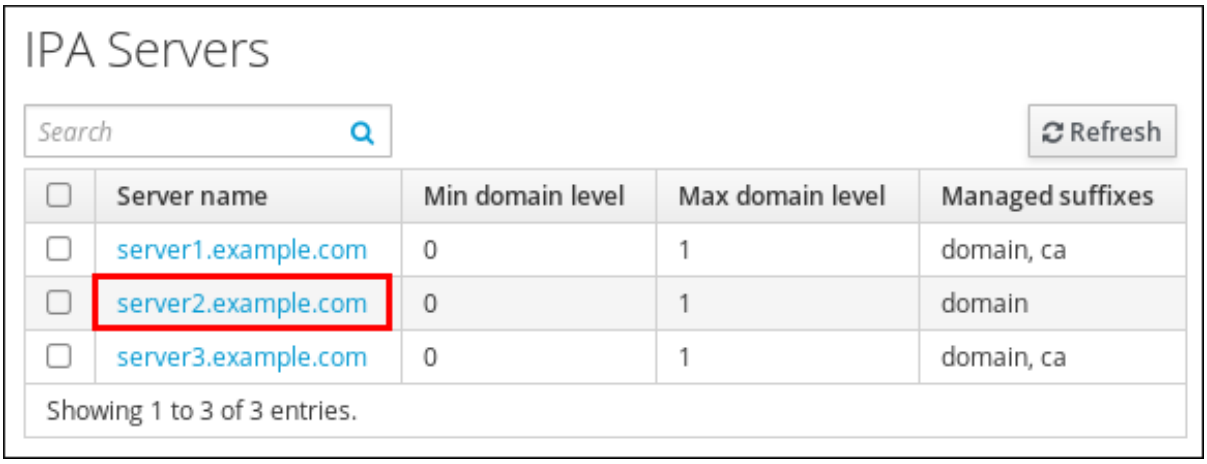
WARNING

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

Procedure

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

Figure 24.11. Selecting a server



3. Click **Delete Server**.

Additional resources

- Uninstalling an IdM server

24.12. REMOVING SERVER FROM TOPOLOGY USING THE CLI

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

Prerequisites

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

IMPORTANT

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

Procedure

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

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

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

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

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

24.13. REMOVING OBSOLETE RUV RECORDS

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

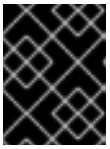
Prerequisites

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

Procedure

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

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

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



IMPORTANT

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

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

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

WARNING

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

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

Verification

1. Run **ipa-replica-manage list-ruv** again. If the command no longer displays any corrupt RUVs, the records have been successfully cleaned.

2. If the command still displays corrupt RUVs, clear them manually using this task:

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

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

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

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

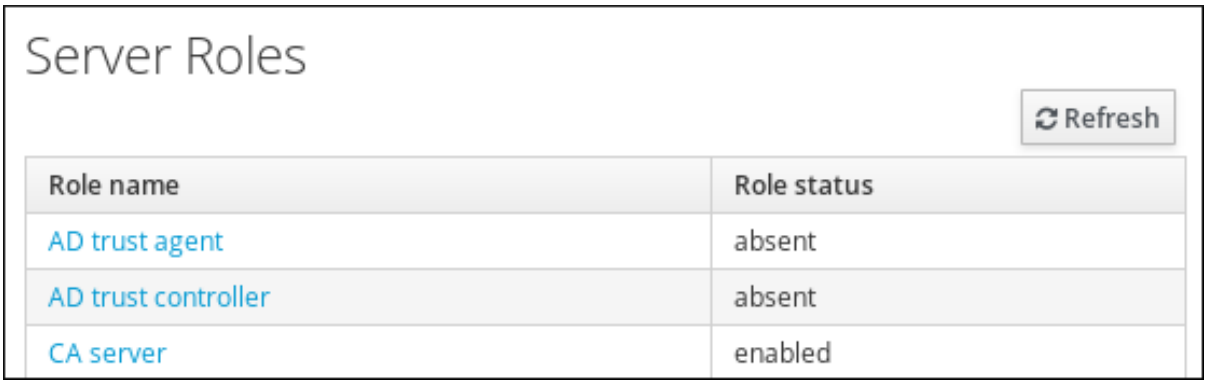
Procedure

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

NOTE

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

Figure 24.12. Server roles in the web UI



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

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

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

Procedure

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

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

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

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

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

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

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

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

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

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

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

Prerequisites

- You have the IdM administrator credentials.

Procedure

- Change the current CA renewal server.
- Configure a replica to generate CRLs.

24.17. DEMOTING OR PROMOTING HIDDEN REPLICAS

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

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

Prerequisites

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

Changing and resetting IdM CA renewal server

Procedure

- To hide a replica:

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

- To make a replica visible again:

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

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

```
# ipa config-show
```

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

CHAPTER 25. INSTALLING AND RUNNING THE IDM HEALTHCHECK TOOL

Learn more about the IdM Healthcheck tool and how to install and run it.

25.1. HEALTHCHECK IN IDM

The **Healthcheck** command line tool in Identity Management (IdM) helps find issues that can impact the performance of your IdM environment. Using Healthcheck, you can identify an issue in advance so that you can correct it before it becomes critical.

NOTE

You can use Healthcheck without obtaining a Kerberos ticket.

Modules are independent

Healthcheck consists of independent modules which check for:

- Replication issues
- Certificate validity
- Certificate authority infrastructure issues
- IdM and Active Directory trust issues
- Correct file permissions and ownership settings

Output formats and destination

You can set the following types of output for Healthcheck to generate by using the **output-type** option:

- **json**: Machine-readable output in JSON format (default)
- **human**: Human-readable output

You can specify a file to store the output by using the **--output-file** option.

Results

Each Healthcheck module returns one of the following results:

SUCCESS
: The system is configured as expected.

WARNING
: It is advisable to monitor or evaluate the configuration.

ERROR
: The system is not configured as expected.

CRITICAL
: The configuration is not as expected, with a significant potential to impact the functioning of your IdM deployment.

Additional resources

- **man ipa-healthcheck**

25.2. INSTALLING IDM HEALTHCHECK

Learn how you can install the IdM Healthcheck tool.

Prerequisites

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

Procedure

- Install the **ipa-healthcheck** package:

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

Verification

- Perform a basic Healthcheck test:

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

The empty square brackets **[]** indicate a fully-functioning IdM installation.

Additional resources

- Run **ipa-healthcheck --help** to see all supported arguments.

25.3. RUNNING IDM HEALTHCHECK

You can execute Healthcheck tests in one of the following ways:

- Manually
- Automatically by using log rotation.

This section describes how to execute the tests manually.

Prerequisites

- The Healthcheck tool is installed. See Installing IdM Healthcheck.

Procedure

1. [Optional] To display a list of all available Healthcheck tests, enter:

```
[root@server ~]# ipa-healthcheck --list-sources
```

2. To run the Healthcheck utility, enter:

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

Additional resources

- **man ipa-healthcheck**

25.4. ADDITIONAL RESOURCES

- See the following sections of the Using IdM Healthcheck to monitor your IdM environment guide for examples of using IdM Healthcheck.
  - Checking services
  - Verifying your IdM and AD trust configuration
  - Verifying certificates
  - Verifying system certificates
  - Checking disk space
  - Verifying permissions of IdM configuration files
  - Checking replication

CHAPTER 26. INSTALLING AN IDENTITY MANAGEMENT SERVER USING AN ANSIBLE PLAYBOOK

The following sections describe how to configure a system as an IdM server by using Ansible. Configuring a system as an IdM server establishes an IdM domain and enables the system to offer IdM services to IdM clients. The deployment is managed by the **ipaserver** Ansible role.

Prerequisites

- You understand Ansible and IdM concepts:
  - Ansible roles
  - Ansible nodes
  - Ansible inventory
  - Ansible tasks
  - Ansible modules
  - Ansible plays and playbooks

26.1. ANSIBLE AND ITS ADVANTAGES FOR INSTALLING IDM

Ansible is an automation tool used to configure systems, deploy software, and perform rolling updates. Ansible includes support for Identity Management (IdM), and you can use Ansible modules to automate installation tasks such as the setup of an IdM server, replica, client, or an entire IdM topology.

Advantages of using Ansible to install IdM

The following list presents advantages of installing Identity Management using Ansible in contrast to manual installation.

- You do not need to log into the managed node.
- You do not need to configure settings on each host to be deployed individually. Instead, you can have one inventory file to deploy a complete cluster.
- You can reuse an inventory file later for management tasks, for example to add users and hosts. You can reuse an inventory file even for such tasks as are not related to IdM.

Additional resources

- Automating Red Hat Enterprise Linux Identity Management installation
- Planning Identity Management
- Preparing the system for IdM server installation

26.2. INSTALLING THE ANSIBLE-FREEIPA PACKAGE

The following procedure describes how to install the the **ansible-freeipa** roles.

Prerequisites

- Ensure that the controller is a Red Hat Enterprise Linux system with a valid subscription. If this is not the case, see the official Ansible documentation Installation guide for alternative installation instructions.
- Ensure that you can reach the managed node over the **SSH** protocol from the controller. Check that the managed node is listed in the **/root/.ssh/known_hosts** file of the controller.

Procedure

Run the following procedure on the Ansible controller.

1. Enable the required repository:

```
# subscription-manager repos --enable rhel-9-for-x86_64-appstream-rpms
```

2. Install the IdM Ansible roles:

```
# dnf install ansible-freeipa
```

The roles are installed to the **/usr/share/ansible/roles/** directory.

26.3. ANSIBLE ROLES LOCATION IN THE FILE SYSTEM

By default, the **ansible-freeipa** roles are installed to the **/usr/share/ansible/roles/** directory. The structure of the **ansible-freeipa** package is as follows:

- The **/usr/share/ansible/roles/** directory stores the **ipaserver**, **ipareplica**, and **ipaclient** roles on the Ansible controller. Each role directory stores examples, a basic overview, the license and documentation about the role in a **README.md** Markdown file.

```
[root@server]# ls -1 /usr/share/ansible/roles/
ipaclient
ipareplica
ipaserver
```

- The **/usr/share/doc/ansible-freeipa/** directory stores the documentation about individual roles and the topology in **README.md** Markdown files. It also stores the **playbooks/** subdirectory.

```
[root@server]# ls -1 /usr/share/doc/ansible-freeipa/
playbooks
README-client.md
README.md
README-replica.md
README-server.md
README-topology.md
```

- The **/usr/share/doc/ansible-freeipa/playbooks/** directory stores the example playbooks:

```
[root@server]# ls -1 /usr/share/doc/ansible-freeipa/playbooks/
install-client.yml
install-cluster.yml
install-replica.yml
install-server.yml
uninstall-client.yml
```

```
uninstall-cluster.yml
uninstall-replica.yml
uninstall-server.yml
```

26.4. SETTING THE PARAMETERS FOR A DEPLOYMENT WITH AN INTEGRATED DNS AND AN INTEGRATED CA AS THE ROOT CA

Complete this procedure to configure the inventory file for installing an IdM server with an integrated CA as the root CA in an environment that uses the IdM integrated DNS solution.

NOTE

The inventory in this procedure uses the **INI** format. You can, alternatively, use the **YAML** or **JSON** formats.

Procedure

1. Create a ~/**MyPlaybooks**/ directory:

```
$ mkdir MyPlaybooks
```

2. Create a ~/**MyPlaybooks**/**inventory** file.
3. Open the inventory file for editing. Specify the fully-qualified domain names (**FQDN**) of the host you want to use as an IdM server. Ensure that the **FQDN** meets the following criteria:
   - Only alphanumeric characters and hyphens (-) are allowed. For example, underscores are not allowed and can cause DNS failures.
   - The host name must be all lower-case.
4. Specify the IdM domain and realm information.
5. Specify that you want to use integrated DNS by adding the following option:

```
ipaserver_setup_dns=true
```

6. Specify the DNS forwarding settings. Choose one of the following options:
   - Use the **ipaserver_auto_forwarders=true** option if you want the installer to use forwarders from the /**etc**/**resolv.conf** file. Do not use this option if the nameserver specified in the /**etc**/**resolv.conf** file is the localhost 127.0.0.1 address or if you are on a virtual private network and the DNS servers you are using are normally unreachable from the public internet.
   - Use the **ipaserver_forwarders** option to specify your forwarders manually. The installation process adds the forwarder IP addresses to the /**etc**/**named.conf** file on the installed IdM server.
   - Use the **ipaserver_no_forwarders=true** option to configure root DNS servers to be used instead.

NOTE

With no DNS forwarders, your environment is isolated, and names from other DNS domains in your infrastructure are not resolved.

7. Specify the DNS reverse record and zone settings. Choose from the following options:

   - Use the **ipaserver_allow_zone_overlap=true** option to allow the creation of a (reverse) zone even if the zone is already resolvable.
   - Use the **ipaserver_reverse_zones** option to specify your reverse zones manually.
   - Use the **ipaserver_no_reverse=true** option if you do not want the installer to create a reverse DNS zone.

NOTE

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

8. Specify the passwords for **admin** and for the **Directory Manager**. Use the Ansible Vault to store the password, and reference the Vault file from the playbook file. Alternatively and less securely, specify the passwords directly in the inventory file.

9. Optional: Specify a custom **firewalld** zone to be used by the IdM server. If you do not set a custom zone, IdM will add its services to the default **firewalld** zone. The predefined default zone is **public**.

IMPORTANT

The specified **firewalld** zone must exist and be permanent.

Example of an inventory file with the required server information (excluding the passwords)

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

[ipaserver:vars]
ipaserver_domain=idm.example.com
ipaserver_realm=IDM.EXAMPLE.COM
ipaserver_setup_dns=true
ipaserver_auto_forwarders=true
[...]
```

Example of an inventory file with the required server information (including the passwords)

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

[ipaserver:vars]
ipaserver_domain=idm.example.com
ipaserver_realm=IDM.EXAMPLE.COM
```

```
ipaserver_setup_dns=true
ipaserver_auto_forwarders=true
ipaadmin_password=MySecretPassword123
ipadm_password=MySecretPassword234

[...]
```

Example of an inventory file with a custom firewalld zone

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

[ipaserver:vars]
ipaserver_domain=idm.example.com
ipaserver_realm=IDM.EXAMPLE.COM
ipaserver_setup_dns=true
ipaserver_auto_forwarders=true
ipaadmin_password=MySecretPassword123
ipadm_password=MySecretPassword234
ipaserver_firewalld_zone=custom zone
```

Example playbook to set up an IdM server using admin and Directory Manager passwords stored in an Ansible Vault file

```
---
- name: Playbook to configure IPA server
  hosts: ipaserver
  become: true
  vars_files:
  - playbook_sensitive_data.yml

  roles:
  - role: ipaserver
    state: present
```

Example playbook to set up an IdM server using admin and Directory Manager passwords from an inventory file

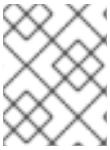
```
---
- name: Playbook to configure IPA server
  hosts: ipaserver
  become: true

  roles:
  - role: ipaserver
    state: present
```

Additional resources

- man **ipa-server-install(1)**
- **/usr/share/doc/ansible-freeipa/README-server.md**

26.5. SETTING THE PARAMETERS FOR A DEPLOYMENT WITH EXTERNAL DNS AND AN INTEGRATED CA AS THE ROOT CA

Complete this procedure to configure the inventory file for installing an IdM server with an integrated CA as the root CA in an environment that uses an external DNS solution.



NOTE

The inventory file in this procedure uses the **INI** format. You can, alternatively, use the **YAML** or **JSON** formats.

Procedure

1. Create a ~/**MyPlaybooks**/ directory:

```
$ mkdir MyPlaybooks
```

2. Create a ~/**MyPlaybooks**/**inventory** file.
3. Open the inventory file for editing. Specify the fully-qualified domain names (**FQDN**) of the host you want to use as an IdM server. Ensure that the **FQDN** meets the following criteria:
   - Only alphanumeric characters and hyphens (-) are allowed. For example, underscores are not allowed and can cause DNS failures.
   - The host name must be all lower-case.
4. Specify the IdM domain and realm information.
5. Make sure that the **ipaserver_setup_dns** option is set to **no** or that it is absent.
6. Specify the passwords for **admin** and for the **Directory Manager**. Use the Ansible Vault to store the password, and reference the Vault file from the playbook file. Alternatively and less securely, specify the passwords directly in the inventory file.
7. Optional: Specify a custom **firewalld** zone to be used by the IdM server. If you do not set a custom zone, IdM will add its services to the default **firewalld** zone. The predefined default zone is **public**.

IMPORTANT

The specified **firewalld** zone must exist and be permanent.

Example of an inventory file with the required server information (excluding the passwords)

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

[ipaserver:vars]
ipaserver_domain=idm.example.com
ipaserver_realm=IDM.EXAMPLE.COM
ipaserver_setup_dns=no
[...]
```

Example of an inventory file with the required server information (including the passwords)

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

[ipaserver:vars]
ipaserver_domain=idm.example.com
ipaserver_realm=IDM.EXAMPLE.COM
ipaserver_setup_dns=no
ipaadmin_password=MySecretPassword123
ipadm_password=MySecretPassword234

[...]
```

Example of an inventory file with a custom firewalld zone

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

[ipaserver:vars]
ipaserver_domain=idm.example.com
ipaserver_realm=IDM.EXAMPLE.COM
ipaserver_setup_dns=no
ipaadmin_password=MySecretPassword123
ipadm_password=MySecretPassword234
ipaserver_firewalld_zone=custom zone
```

Example playbook to set up an IdM server using admin and Directory Manager passwords stored in an Ansible Vault file

```
---
- name: Playbook to configure IPA server
  hosts: ipaserver
  become: true
  vars_files:
  - playbook_sensitive_data.yml

  roles:
  - role: ipaserver
    state: present
```

Example playbook to set up an IdM server using admin and Directory Manager passwords from an inventory file

```
---
- name: Playbook to configure IPA server
  hosts: ipaserver
  become: true

  roles:
  - role: ipaserver
    state: present
```

Additional resources

- man **ipa-server-install(1)**
- **/usr/share/doc/ansible-freeipa/README-server.md**

26.6. DEPLOYING AN IDM SERVER WITH AN INTEGRATED CA AS THE ROOT CA USING AN ANSIBLE PLAYBOOK

Complete this procedure to deploy an IdM server with an integrated certificate authority (CA) as the root CA using an Ansible playbook.

Prerequisites

- The managed node is a Red Hat Enterprise Linux 9 system with a static IP address and a working package manager.
- You have set the parameters that correspond to your scenario by choosing one of the following procedures:
  - Procedure with integrated DNS
  - Procedure with external DNS

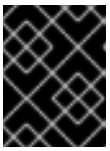
Procedure

1. Run the Ansible playbook:

```
$ ansible-playbook -i ~/MyPlaybooks/inventory ~/MyPlaybooks/install-server.yml
```

2. Choose one of the following options:
   - If your IdM deployment uses external DNS: add the DNS resource records contained in the **/tmp/ipa.system.records.UFRPto.db** file to the existing external DNS servers. The process of updating the DNS records varies depending on the particular DNS solution.

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



IMPORTANT

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

   - If your IdM deployment uses integrated DNS:
     - Add DNS delegation from the parent domain to the IdM DNS domain. For example, if the IdM DNS domain is ***idm.example.com***, add a name server (NS) record to the **example.com** parent domain.

IMPORTANT

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

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

26.7. SETTING THE PARAMETERS FOR A DEPLOYMENT WITH AN INTEGRATED DNS AND AN EXTERNAL CA AS THE ROOT CA

Complete this procedure to configure the inventory file for installing an IdM server with an external CA as the root CA in an environment that uses the IdM integrated DNS solution.

NOTE

The inventory file in this procedure uses the **INI** format. You can, alternatively, use the **YAML** or **JSON** formats.

Procedure

1. Create a ~/**MyPlaybooks**/ directory:

```
$ mkdir MyPlaybooks
```

2. Create a ~/**MyPlaybooks**/**inventory** file.

3. Open the inventory file for editing. Specify the fully-qualified domain names (**FQDN**) of the host you want to use as an IdM server. Ensure that the **FQDN** meets the following criteria:

   - Only alphanumeric characters and hyphens (-) are allowed. For example, underscores are not allowed and can cause DNS failures.
   - The host name must be all lower-case.

4. Specify the IdM domain and realm information.

5. Specify that you want to use integrated DNS by adding the following option:

```
ipaserver_setup_dns=true
```

6. Specify the DNS forwarding settings. Choose one of the following options:

   - Use the **ipaserver_auto_forwarders=true** option if you want the installation process to use forwarders from the /**etc**/**resolv.conf** file. This option is not recommended if the nameserver specified in the /**etc**/**resolv.conf** file is the localhost 127.0.0.1 address or if you are on a virtual private network and the DNS servers you are using are normally unreachable from the public internet.
   - Use the **ipaserver_forwarders** option to specify your forwarders manually. The installation process adds the forwarder IP addresses to the /**etc**/**named.conf** file on the installed IdM server.

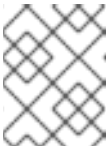
- Use the **ipaserver_no_forwarders=true** option to configure root DNS servers to be used instead.

NOTE

With no DNS forwarders, your environment is isolated, and names from other DNS domains in your infrastructure are not resolved.

7. Specify the DNS reverse record and zone settings. Choose from the following options:

   - Use the **ipaserver_allow_zone_overlap=true** option to allow the creation of a (reverse) zone even if the zone is already resolvable.
   - Use the **ipaserver_reverse_zones** option to specify your reverse zones manually.
   - Use the **ipaserver_no_reverse=true** option if you do not want the installation process to create a reverse DNS zone.



NOTE

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

8. Specify the passwords for **admin** and for the **Directory Manager**. Use the Ansible Vault to store the password, and reference the Vault file from the playbook file. Alternatively and less securely, specify the passwords directly in the inventory file.

9. Optional: Specify a custom **firewalld** zone to be used by the IdM server. If you do not set a custom zone, IdM adds its services to the default **firewalld** zone. The predefined default zone is **public**.

IMPORTANT

The specified **firewalld** zone must exist and be permanent.

Example of an inventory file with the required server information (excluding the passwords)

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

[ipaserver:vars]
ipaserver_domain=idm.example.com
ipaserver_realm=IDM.EXAMPLE.COM
ipaserver_setup_dns=true
ipaserver_auto_forwarders=true
[...]
```

Example of an inventory file with the required server information (including the passwords)

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

```
[ipaserver:vars]
ipaserver_domain=idm.example.com
ipaserver_realm=IDM.EXAMPLE.COM
ipaserver_setup_dns=true
ipaserver_auto_forwarders=true
ipaadmin_password=MySecretPassword123
ipadm_password=MySecretPassword234

[...]
```

Example of an inventory file with a custom firewalld zone

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

[ipaserver:vars]
ipaserver_domain=idm.example.com
ipaserver_realm=IDM.EXAMPLE.COM
ipaserver_setup_dns=true
ipaserver_auto_forwarders=true
ipaadmin_password=MySecretPassword123
ipadm_password=MySecretPassword234
ipaserver_firewalld_zone=custom zone

[...]
```

10. Create a playbook for the first step of the installation. Enter instructions for generating the certificate signing request (CSR) and copying it from the controller to the managed node.

```
---
- name: Playbook to configure IPA server Step 1
  hosts: ipaserver
  become: true
  vars_files:
  - playbook_sensitive_data.yml
  vars:
    ipaserver_external_ca: true

  roles:
  - role: ipaserver
    state: present

  post_tasks:
  - name: Copy CSR /root/ipa.csr from node to "{{ groups.ipaserver[0] + '-ipa.csr' }}"
    fetch:
      src: /root/ipa.csr
      dest: "{{ groups.ipaserver[0] + '-ipa.csr' }}"
      flat: true
```

11. Create another playbook for the final step of the installation.

```
---
- name: Playbook to configure IPA server Step 2
  hosts: ipaserver
```

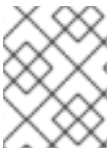
```
become: true
vars_files:
- playbook_sensitive_data.yml
vars:
  ipaserver_external_cert_files:
    - "/root/servercert20240601.pem"
    - "/root/cacert.pem"

pre_tasks:
- name: Copy "{{ groups.ipaserver[0] }}-{{ item }}" to "/root/{{ item }}" on node
  ansible.builtin.copy:
    src: "{{ groups.ipaserver[0] }}-{{ item }}"
    dest: "/root/{{ item }}"
    force: true
  with_items:
  - servercert20240601.pem
  - cacert.pem

roles:
- role: ipaserver
  state: present
```

Additional resources

- man **ipa-server-install(1)**
- **/usr/share/doc/ansible-freeipa/README-server.md**

26.8. SETTING THE PARAMETERS FOR A DEPLOYMENT WITH EXTERNAL DNS AND AN EXTERNAL CA AS THE ROOT CA

Complete this procedure to configure the inventory file for installing an IdM server with an external CA as the root CA in an environment that uses an external DNS solution.



NOTE

The inventory file in this procedure uses the **INI** format. You can, alternatively, use the **YAML** or **JSON** formats.

Procedure

1. Create a ~/**MyPlaybooks**/ directory:

```
$ mkdir MyPlaybooks
```

2. Create a ~/**MyPlaybooks**/**inventory** file.
3. Open the inventory file for editing. Specify the fully-qualified domain names (**FQDN**) of the host you want to use as an IdM server. Ensure that the **FQDN** meets the following criteria:
   - Only alphanumeric characters and hyphens (-) are allowed. For example, underscores are not allowed and can cause DNS failures.
   - The host name must be all lower-case.

4. Specify the IdM domain and realm information.

5. Make sure that the **ipaserver_setup_dns** option is set to **no** or that it is absent.

6. Specify the passwords for **admin** and for the **Directory Manager**. Use the Ansible Vault to store the password, and reference the Vault file from the playbook file. Alternatively and less securely, specify the passwords directly in the inventory file.

7. Optional: Specify a custom **firewalld** zone to be used by the IdM server. If you do not set a custom zone, IdM will add its services to the default **firewalld** zone. The predefined default zone is **public**.

IMPORTANT

The specified **firewalld** zone must exist and be permanent.

Example of an inventory file with the required server information (excluding the passwords)

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

[ipaserver:vars]
ipaserver_domain=idm.example.com
ipaserver_realm=IDM.EXAMPLE.COM
ipaserver_setup_dns=no
[...]
```

Example of an inventory file with the required server information (including the passwords)

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

[ipaserver:vars]
ipaserver_domain=idm.example.com
ipaserver_realm=IDM.EXAMPLE.COM
ipaserver_setup_dns=no
ipaadmin_password=MySecretPassword123
ipadm_password=MySecretPassword234

[...]
```

Example of an inventory file with a custom firewalld zone

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

[ipaserver:vars]
ipaserver_domain=idm.example.com
ipaserver_realm=IDM.EXAMPLE.COM
ipaserver_setup_dns=no
ipaadmin_password=MySecretPassword123
ipadm_password=MySecretPassword234
```

```
ipaserver_firewalld_zone=custom zone

[...]
```

8. Create a playbook for the first step of the installation. Enter instructions for generating the certificate signing request (CSR) and copying it from the controller to the managed node.

```
---
- name: Playbook to configure IPA server Step 1
  hosts: ipaserver
  become: true
  vars_files:
  - playbook_sensitive_data.yml
  vars:
    ipaserver_external_ca: true

  roles:
  - role: ipaserver
    state: present

  post_tasks:
  - name: Copy CSR /root/ipa.csr from node to "{{ groups.ipaserver[0] + '-ipa.csr' }}"
    fetch:
      src: /root/ipa.csr
      dest: "{{ groups.ipaserver[0] + '-ipa.csr' }}"
      flat: true
```

9. Create another playbook for the final step of the installation.

```
---
- name: Playbook to configure IPA server Step 2
  hosts: ipaserver
  become: true
  vars_files:
  - playbook_sensitive_data.yml
  vars:
    ipaserver_external_cert_files:
      - "/root/servercert20240601.pem"
      - "/root/cacert.pem"

  pre_tasks:
  - name: Copy "{{ groups.ipaserver[0] }}-{{ item }}" to "/root/{{ item }}" on node
    ansible.builtin.copy:
      src: "{{ groups.ipaserver[0] }}-{{ item }}"
      dest: "/root/{{ item }}"
      force: true
    with_items:
    - servercert20240601.pem
    - cacert.pem

  roles:
  - role: ipaserver
    state: present
```

Additional resources

- Installing an IdM server: Without integrated DNS, with an external CA as the root CA
- man **ipa-server-install(1)**
- **/usr/share/doc/ansible-freeipa/README-server.md**

26.9. DEPLOYING AN IDM SERVER WITH AN EXTERNAL CA AS THE ROOT CA USING AN ANSIBLE PLAYBOOK

Complete this procedure to deploy an IdM server with an external certificate authority (CA) as the root CA using an Ansible playbook.

Prerequisites

- The managed node is a Red Hat Enterprise Linux 9 system with a static IP address and a working package manager.
- You have set the parameters that correspond to your scenario by choosing one of the following procedures:
  - Procedure with integrated DNS
  - Procedure with external DNS

Procedure

1. Run the Ansible playbook with the instructions for the first step of the installation, for example **install-server-step1.yml**:

```
$ ansible-playbook --vault-password-file=password_file -v -i ~/MyPlaybooks/inventory
~/MyPlaybooks/install-server-step1.yml
```

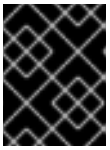
2. Locate the **ipa.csr** certificate signing request file on the controller and submit it to the external CA.
3. Place the IdM CA certificate signed by the external CA in the controller file system so that the playbook in the next step can find it.
4. Run the Ansible playbook with the instructions for the final step of the installation, for example **install-server-step2.yml**:

```
$ ansible-playbook -v -i ~/MyPlaybooks/inventory ~/MyPlaybooks/install-server-
step2.yml
```

5. Choose one of the following options:
   - If your IdM deployment uses external DNS: add the DNS resource records contained in the **/tmp/ipa.system.records.UFRPto.db** file to the existing external DNS servers. The process of updating the DNS records varies depending on the particular DNS solution.

```
...
Restarting the KDC
Please add records in this file to your DNS system:
```

```
/tmp/ipa.system.records.UFRBto.db
Restarting the web server
...
```



IMPORTANT

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

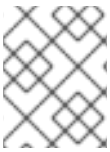
- If your IdM deployment uses integrated DNS:
  - Add DNS delegation from the parent domain to the IdM DNS domain. For example, if the IdM DNS domain is ***idm.example.com***, add a name server (NS) record to the **example.com** parent domain.

IMPORTANT

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

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

26.10. UNINSTALLING AN IDM SERVER USING AN ANSIBLE PLAYBOOK



NOTE

In an existing Identity Management (IdM) deployment, **replica** and **server** are interchangeable terms.

Complete this procedure to uninstall an IdM replica using an Ansible playbook. In this example:

- IdM configuration is uninstalled from **server123.idm.example.com**.
- **server123.idm.example.com** and the associated host entry are removed from the IdM topology.

Prerequisites

- On the control node:
  - You are using Ansible version 2.14 or later.
  - You have installed the **ansible-freeipa** package.
  - You have created an Ansible inventory file with the fully-qualified domain name (FQDN) of the IdM server in the ***~/MyPlaybooks/*** directory.
  - You have stored your **ipaadmin_password** in the **secret.yml** Ansible vault.
  - For the **ipaserver_remove_from_topology** option to work, the system must be running on RHEL 9.3 or later.

- On the managed node:
  - The system is running on RHEL 9.

Procedure

1. Create your Ansible playbook file **uninstall-server.yml** with the following content:

```
---
- name: Playbook to uninstall an IdM replica
  hosts: ipaserver
  become: true

  roles:
  - role: ipaserver
    ipaserver_remove_from_domain: true
    state: absent
```

   The **ipaserver_remove_from_domain** option unenrolls the host from the IdM topology.

NOTE

If the removal of server123.idm.example.com should lead to a disconnected topology, the removal will be aborted. For more information, see Using an Ansible playbook to uninstall an IdM server even if this leads to a disconnected topology.

2. Uninstall the replica:

```
$ ansible-playbook --vault-password-file=password_file -v -i
<path_to_inventory_directory>/inventory <path_to_playbooks_directory>/uninstall-
server.yml
```

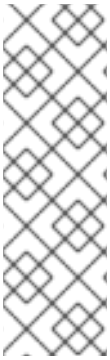
3. Ensure that all name server (NS) DNS records pointing to **server123.idm.example.com** are deleted from your DNS zones. This applies regardless of whether you use integrated DNS managed by IdM or external DNS. For more information on how to delete DNS records from IdM, see Deleting DNS records in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

26.11. USING AN ANSIBLE PLAYBOOK TO UNINSTALL AN IDM SERVER EVEN IF THIS LEADS TO A DISCONNECTED TOPOLOGY

NOTE

In an existing Identity Management (IdM) deployment, **replica** and **server** are interchangeable terms.

Complete this procedure to uninstall an IdM replica using an Ansible playbook even if this results in a disconnected IdM topology. In the example, **server456.idm.example.com** is used to remove the replica and the associated host entry with the FQDN of **server123.idm.example.com** from the topology, leaving certain replicas disconnected from **server456.idm.example.com** and the rest of the topology.



NOTE

If removing a replica from the topology using only the **remove_server_from_domain** does not result in a disconnected topology, no other options are required. If the result is a disconnected topology, you must specify which part of the domain you want to preserve. In that case, you must do the following:

- Specify the **ipaserver_remove_on_server** value.
- Set **ipaserver_ignore_topology_disconnect** to True.

Prerequisites

- On the control node:
  - You are using Ansible version 2.14 or later.
  - The system is running on RHEL 9.3 or later.
  - You have installed the **ansible-freeipa** package.
  - You have created an Ansible inventory file with the fully-qualified domain name (FQDN) of the IdM server in the ***~/MyPlaybooks/*** directory.
  - You have stored your **ipaadmin_password** in the **secret.yml** Ansible vault.
- On the managed node:
  - The system is running on 9 or later.

Procedure

1. Create your Ansible playbook file **uninstall-server.yml** with the following content:

```
---
- name: Playbook to uninstall an IdM replica
  hosts: ipaserver
  become: true

  roles:
  - role: ipaserver
    ipaserver_remove_from_domain: true
    ipaserver_remove_on_server: server456.idm.example.com
    ipaserver_ignore_topology_disconnect: true
    state: absent
```

NOTE

Under normal circumstances, if the removal of server123 does not result in a disconnected topology: if the value for **ipaserver_remove_on_server** is not set, the replica on which server123 is removed is automatically determined using the replication agreements of server123.

2. Uninstall the replica:

```
$ ansible-playbook --vault-password-file=password_file -v -i
<path_to_inventory_directory>/hosts <path_to_playbooks_directory>/uninstall-
server.yml
```

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

Additional resources

- Inventory basics: formats, hosts, and groups
- You can see sample Ansible playbooks for installing an IdM server and a list of possible variables in the **ansible-freeipa** upstream documentation.

CHAPTER 27. INSTALLING AN IDENTITY MANAGEMENT REPLICA USING AN ANSIBLE PLAYBOOK

Configuring a system as an IdM replica by using Ansible enrolls it into an IdM domain and enables the system to use IdM services on IdM servers in the domain.

The deployment is managed by the **ipareplica** Ansible role. The role can use the autodiscovery mode for identifying the IdM servers, domain and other settings. However, if you deploy multiple replicas in a tier-like model, with different groups of replicas being deployed at different times, you must define specific servers or replicas for each group.

Prerequisites

- You have installed the ansible-freeipa package on the Ansible control node.
- You understand the general Ansible and IdM concepts.
- You have planned the replica topology in your deployment .

27.1. SPECIFYING THE BASE, SERVER AND CLIENT VARIABLES FOR INSTALLING THE IDM REPLICA

Complete this procedure to configure the inventory file for installing an IdM replica.

Prerequisites

- You have configured your Ansible control node to meet the following requirements:
  - You are using Ansible version 2.14 or later.
  - You have installed the **ansible-freeipa** package on the Ansible controller.

Procedure

1. Open the inventory file for editing. Specify the fully-qualified domain names (FQDN) of the hosts to become IdM replicas. The FQDNs must be valid DNS names:
   - Only numbers, alphabetic characters, and hyphens (**-**) are allowed. For example, underscores are not allowed and can cause DNS failures.
   - The host name must be all lower-case.

     Example of a simple inventory hosts file with only the replicas' FQDN defined

     ```
     [ipareplicas]
     replica1.idm.example.com
     replica2.idm.example.com
     replica3.idm.example.com
     [...]
     ```

     If the IdM server is already deployed and the SRV records are set properly in the IdM DNS zone, the script automatically discovers all the other required values.

2. Optional: Provide additional information in the inventory file based on how you have designed your topology:

Scenario 1

If you want to avoid autodiscovery and have all replicas listed in the **[ipareplicas]** section use a specific IdM server, set the server in the **[ipaservers]** section of the inventory file.

Example inventory hosts file with the FQDN of the IdM server and replicas defined

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

[ipareplicas]
replica1.idm.example.com
replica2.idm.example.com
replica3.idm.example.com
[...]
```

Scenario 2

Alternatively, if you want to avoid autodiscovery but want to deploy specific replicas with specific servers, set the servers for specific replicas individually in the **[ipareplicas]** section in the inventory file.

Example inventory file with a specific IdM server defined for a specific replica

```
[ipaservers]
server.idm.example.com
replica1.idm.example.com

[ipareplicas]
replica2.idm.example.com
replica3.idm.example.com ipareplica_servers=replica1.idm.example.com
```

In the example above, **replica3.idm.example.com** uses the already deployed **replica1.idm.example.com** as its replication source.

Scenario 3

If you are deploying several replicas in one batch and time is a concern to you, multitier replica deployment can be useful for you. Define specific groups of replicas in the inventory file, for example **[ipareplicas_tier1]** and **[ipareplicas_tier2]**, and design separate plays for each group in the **install-replica.yml** playbook.

Example inventory file with replica tiers defined

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

[ipareplicas_tier1]
replica1.idm.example.com
```

```
[ipareplicas_tier2]
replica2.idm.example.com \
ipareplica_servers=replica1.idm.example.com,server.idm.example.com
```

The first entry in **ipareplica_servers** will be used. The second entry will be used as a fallback option. When using multiple tiers for deploying IdM replicas, you must have separate tasks in the playbook to first deploy replicas from tier1 and then replicas from tier2:

Example of a playbook file with different plays for different replica groups

```
---
- name: Playbook to configure IPA replicas (tier1)
  hosts: ipareplicas_tier1
  become: true

  roles:
  - role: ipareplica
    state: present

- name: Playbook to configure IPA replicas (tier2)
  hosts: ipareplicas_tier2
  become: true

  roles:
  - role: ipareplica
    state: present
```

3. Optional: Provide additional information regarding **firewalld** and DNS:

Scenario 1

If you want the replica to use a specified **firewalld** zone, for example an internal one, you can specify it in the inventory file. If you do not set a custom zone, IdM will add its services to the default **firewalld** zone. The predefined default zone is **public**.

IMPORTANT

The specified **firewalld** zone must exist and be permanent.

Example of a simple inventory hosts file with a custom firewalld zone

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

[ipareplicas]
replica1.idm.example.com
replica2.idm.example.com
replica3.idm.example.com
[...]

[ipareplicas:vars]
ipareplica_firewalld_zone=custom zone
```

Scenario 2

If you want the replica to host the IdM DNS service, add the **ipareplica_setup_dns=true** line to the **[ipareplicas:vars]** section. Additionally, specify if you want to use per-server DNS forwarders:

- To configure per-server forwarders, add the **ipareplica_forwarders** variable and a list of strings to the **[ipareplicas:vars]** section, for example: **ipareplica_forwarders=192.0.2.1,192.0.2.2**
- To configure no per-server forwarders, add the following line to the **[ipareplicas:vars]** section: **ipareplica_no_forwarders=true**.
- To configure per-server forwarders based on the forwarders listed in the **/etc/resolv.conf** file of the replica, add the **ipareplica_auto_forwarders** variable to the **[ipareplicas:vars]** section.

Example inventory file with instructions to set up DNS and per-server forwarders on the replicas

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

[ipareplicas]
replica1.idm.example.com
replica2.idm.example.com
replica3.idm.example.com
[...]

[ipareplicas:vars]
ipareplica_setup_dns=true
ipareplica_forwarders=192.0.2.1,192.0.2.2
```

Scenario 3

Specify the DNS resolver using the **ipaclient_configure_dns_resolve** and **ipaclient_dns_servers** options (if available) to simplify cluster deployments. This is especially useful if your IdM deployment is using integrated DNS:

An inventory file snippet specifying a DNS resolver:

```
[...]
[ipaclient:vars]
ipaclient_configure_dns_resolver=true
ipaclient_dns_servers=192.168.100.1
```

NOTE

The **ipaclient_dns_servers** list must contain only IP addresses. Host names are not allowed.

Additional resources

- **/usr/share/ansible/roles/ipareplica/README.md**

27.2. SPECIFYING THE CREDENTIALS FOR INSTALLING THE IDM REPLICA USING AN ANSIBLE PLAYBOOK

Complete this procedure to configure the authorization for installing the IdM replica.

Prerequisites

- You have configured your Ansible control node to meet the following requirements:
  - You are using Ansible version 2.14 or later.
  - You have installed the **ansible-freeipa** package on the Ansible controller.

Procedure

1. Specify the **password of a user authorized to deploy replicas**, for example the IdM **admin**.
   - Use the Ansible Vault to store the password, and reference the Vault file from the playbook file, for example **install-replica.yml**:

     Example playbook file using principal from inventory file and password from an Ansible Vault file

     ```
     - name: Playbook to configure IPA replicas
       hosts: ipareplicas
       become: true
       vars_files:
       - playbook_sensitive_data.yml

       roles:
       - role: ipareplica
         state: present
     ```

     For details how to use Ansible Vault, see the official Ansible Vault documentation.

   - Less securely, provide the credentials of **admin** directly in the inventory file. Use the **ipaadmin_password** option in the **[ipareplicas:vars]** section of the inventory file. The inventory file and the **install-replica.yml** playbook file can then look as follows:

     Example inventory hosts.replica file

     ```
     [...]
     [ipareplicas:vars]
     ipaadmin_password=Secret123
     ```

     Example playbook using principal and password from inventory file

     ```
     - name: Playbook to configure IPA replicas
       hosts: ipareplicas
       become: true

       roles:
       - role: ipareplica
         state: present
     ```

- Alternatively but also less securely, provide the credentials of another user authorized to deploy a replica directly in the inventory file. To specify a different authorized user, use the **ipaadmin_principal** option for the user name, and the **ipaadmin_password** option for the password. The inventory file and the **install-replica.yml** playbook file can then look as follows:

  Example inventory hosts.replica file

  ```
  [...]
  [ipareplicas:vars]
  ipaadmin_principal=my_admin
  ipaadmin_password=my_admin_secret123
  ```

  Example playbook using principal and password from inventory file

  ```
  - name: Playbook to configure IPA replicas
    hosts: ipareplicas
    become: true

    roles:
    - role: ipareplica
      state: present
  ```

NOTE

As of RHEL 9.5, during the installation of an IdM replica, checking if the provided Kerberos principal has the required privilege also extends to checking user ID overrides. As a result, you can deploy a replica using the credentials of an AD administrator that is configured to act as an IdM administrator.

Additional resources

- **/usr/share/ansible/roles/ipareplica/README.md**

27.3. DEPLOYING AN IDM REPLICA USING AN ANSIBLE PLAYBOOK

Complete this procedure to use an Ansible playbook to deploy an IdM replica.

Prerequisites

- The managed node is a Red Hat Enterprise Linux 9 system with a static IP address and a working package manager.
- You have configured the inventory file for installing an IdM replica .
- You have configured the authorization for installing the IdM replica .

Procedure

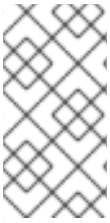
- Run the Ansible playbook:

  ```
  $ ansible-playbook -i ~/MyPlaybooks/inventory ~/MyPlaybooks/install-replica.yml
  ```

Next steps

- In large deployments, you might want to tune specific parameters of IdM replicas for better performance. Consult the Tuning Performance in Identity Management title to find tuning instructions to best suit your scenario.

27.4. UNINSTALLING AN IDM REPLICA USING AN ANSIBLE PLAYBOOK



NOTE

In an existing Identity Management (IdM) deployment, **replica** and **server** are interchangeable terms. For information on how to uninstall an IdM server, see Uninstalling an IdM server using an Ansible playbook or Using an Ansible playbook to uninstall an IdM server even if this leads to a disconnected topology.

Additional resources

- Introduction to IdM servers and clients

CHAPTER 28. INSTALLING AN IDENTITY MANAGEMENT CLIENT USING AN ANSIBLE PLAYBOOK

Learn more about how to configure a system as an Identity Management (IdM) client by using Ansible. Configuring a system as an IdM client enrolls it into an IdM domain and enables the system to use IdM services on IdM servers in the domain.

The deployment is managed by the **ipaclient** Ansible role. By default, the role uses the autodiscovery mode for identifying the IdM servers, domain and other settings. The role can be modified to have the Ansible playbook use the settings specified, for example in the inventory file.

Prerequisites

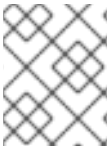
- You have installed the ansible-freeipa package on the Ansible control node.
- You are using Ansible version 2.15 or later.
- You understand the general Ansible and IdM concepts.

28.1. SETTING THE PARAMETERS OF THE INVENTORY FILE FOR THE AUTODISCOVERY CLIENT INSTALLATION MODE

To install an Identity Management (IdM) client using an Ansible playbook, configure the target host parameters in an inventory file, for example **inventory**:

- The information about the host
- The authorization for the task

The inventory file can be in one of many formats, depending on the inventory plugins you have. The **INI-like** format is one of Ansible's defaults and is used in the examples below.



NOTE

To use smart cards with the graphical user interface in RHEL, ensure that you include the **ipaclient_mkhomedir** variable in your Ansible playbook.

Procedure

1. Open your **inventory** file for editing.
2. Specify the fully-qualified hostname (FQDN) of the host to become an IdM client. The fully qualified domain name must be a valid DNS name:
   - Only numbers, alphabetic characters, and hyphens (**-**) are allowed. For example, underscores are not allowed and can cause DNS failures.
   - The host name must be all lower-case. No capital letters are allowed.

   If the SRV records are set properly in the IdM DNS zone, the script automatically discovers all the other required values.

   Example of a simple inventory hosts file with only the client FQDN defined

```
[ipaclients]
client.idm.example.com
[...]
```

3. Specify the credentials for enrolling the client. The following authentication methods are available:

- The **password of a user authorized to enroll clients** This is the default option.
  - Use the Ansible Vault to store the password, and reference the Vault file from the playbook file, for example **install-client.yml**, directly:

    Example playbook file using principal from inventory file and password from an Ansible Vault file

    ```
    - name: Playbook to configure IPA clients with username/password
      hosts: ipaclients
      become: true
      vars_files:
      - playbook_sensitive_data.yml

      roles:
      - role: ipaclient
        state: present
    ```

  - Less securely, provide the credentials of **admin** using the **ipaadmin_password** option in the **[ipaclients:vars]** section of the **inventory/hosts** file. Alternatively, to specify a different authorized user, use the **ipaadmin_principal** option for the user name, and the **ipaadmin_password** option for the password. The **inventory/hosts** inventory file and the **install-client.yml** playbook file can then look as follows:

    Example inventory hosts file

    ```
    [...]
    [ipaclients:vars]
    ipaadmin_principal=my_admin
    ipaadmin_password=Secret123
    ```

    Example Playbook using principal and password from inventory file

    ```
    - name: Playbook to unconfigure IPA clients
      hosts: ipaclients
      become: true

      roles:
      - role: ipaclient
        state: true
    ```

- The **client keytab** from the previous enrollment if it is still available.
  This option is available if the system was previously enrolled as an Identity Management client. To use this authentication method, uncomment the **#ipaclient_keytab** option, specifying the path to the file storing the keytab, for example in the **[ipaclient:vars]** section of **inventory/hosts**.

- A **random, one-time password** (OTP) to be generated during the enrollment. To use this authentication method, use the **ipaclient_use_otp=true** option in your inventory file. For example, you can uncomment the **ipaclient_use_otp=true** option in the **[ipaclients:vars]** section of the **inventory/hosts** file. Note that with OTP you must also specify one of the following options:
  - The **password of a user authorized to enroll clients**, for example by providing a value for **ipaadmin_password** in the **[ipaclients:vars]** section of the **inventory/hosts** file.
  - The **admin keytab**, for example by providing a value for **ipaadmin_keytab** in the **[ipaclients:vars]** section of **inventory/hosts**.

4. Optional: Specify the DNS resolver using the **ipaclient_configure_dns_resolve** and **ipaclient_dns_servers** options (if available) to simplify cluster deployments. This is especially useful if your IdM deployment is using integrated DNS:

   An inventory file snippet specifying a DNS resolver:

   ```
   [...]
   [ipaclients:vars]
   ipaadmin_password: "{{ ipaadmin_password }}"
   ipaclient_domain=idm.example.com
   ipaclient_configure_dns_resolver=true
   ipaclient_dns_servers=192.168.100.1
   ```

   NOTE

   The **ipaclient_dns_servers** list must contain only IP addresses. Host names are not allowed.

5. Starting with RHEL 9.3, you can also specify the **ipaclient_subid: true** option to have subid ranges configured for IdM users on the IdM level.

Additional resources

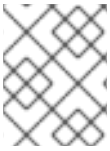
- **/usr/share/ansible/roles/ipaclient/README.md**
- Managing subID ranges manually

28.2. SETTING THE PARAMETERS OF THE INVENTORY FILE WHEN AUTODISCOVERY IS NOT POSSIBLE DURING CLIENT INSTALLATION

To install an Identity Management client using an Ansible playbook, configure the target host parameters in an inventory file, for example **inventory/hosts**:

- The information about the host, the IdM server and the IdM domain or the IdM realm
- The authorization for the task

The inventory file can be in one of many formats, depending on the inventory plugins you have. The **INI-like** format is one of Ansible's defaults and is used in the examples below.



NOTE

To use smart cards with the graphical user interface in RHEL, ensure that you include the **ipaclient_mkhomedir** variable in your Ansible playbook.

Procedure

1. Specify the fully-qualified hostname (FQDN) of the host to become an IdM client. The fully qualified domain name must be a valid DNS name:
   - Only numbers, alphabetic characters, and hyphens (**-**) are allowed. For example, underscores are not allowed and can cause DNS failures.
   - The host name must be all lower-case. No capital letters are allowed.
2. Specify other options in the relevant sections of the **inventory/hosts** file:
   - The FQDN of the servers in the **[ipaservers]** section to indicate which IdM server the client will be enrolled with
   - One of the two following options:
     - The **ipaclient_domain** option in the **[ipaclients:vars]** section to indicate the DNS domain name of the IdM server the client will be enrolled with
     - The **ipaclient_realm** option in the **[ipaclients:vars]** section to indicate the name of the Kerberos realm controlled by the IdM server

     Example of an inventory hosts file with the client FQDN, the server FQDN and the domain defined

     ```
     [ipaclients]
     client.idm.example.com

     [ipaservers]
     server.idm.example.com

     [ipaclients:vars]
     ipaclient_domain=idm.example.com
     [...]
     ```
3. Specify the credentials for enrolling the client. The following authentication methods are available:
   - The **password of a user authorized to enroll clients** This is the default option.
     - Use the Ansible Vault to store the password, and reference the Vault file from the playbook file, for example **install-client.yml**, directly:

     Example playbook file using principal from inventory file and password from an Ansible Vault file

     ```
     - name: Playbook to configure IPA clients with username/password
       hosts: ipaclients
       become: true
       vars_files:
     ```

```
    - playbook_sensitive_data.yml

    roles:
    - role: ipaclient
      state: present
```

- Less securely, the credentials of **admin** to be provided using the **ipaadmin_password** option in the **[ipaclients:vars]** section of the **inventory/hosts** file. Alternatively, to specify a different authorized user, use the **ipaadmin_principal** option for the user name, and the **ipaadmin_password** option for the password. The **install-client.yml** playbook file can then look as follows:

  Example inventory hosts file

  ```
  [...]
  [ipaclients:vars]
  ipaadmin_principal=my_admin
  ipaadmin_password=Secret123
  ```

  Example Playbook using principal and password from inventory file

  ```
  - name: Playbook to unconfigure IPA clients
    hosts: ipaclients
    become: true

    roles:
    - role: ipaclient
      state: true
  ```

- The **client keytab** from the previous enrollment if it is still available:
  This option is available if the system was previously enrolled as an Identity Management client. To use this authentication method, uncomment the **ipaclient_keytab** option, specifying the path to the file storing the keytab, for example in the **[ipaclient:vars]** section of **inventory/hosts**.

- A **random, one-time password** (OTP) to be generated during the enrollment. To use this authentication method, use the **ipaclient_use_otp=true** option in your inventory file. For example, you can uncomment the **#ipaclient_use_otp=true** option in the **[ipaclients:vars]** section of the **inventory/hosts** file. Note that with OTP you must also specify one of the following options:

  - The **password of a user authorized to enroll clients**, for example by providing a value for **ipaadmin_password** in the **[ipaclients:vars]** section of the **inventory/hosts** file.

  - The **admin keytab**, for example by providing a value for **ipaadmin_keytab** in the **[ipaclients:vars]** section of **inventory/hosts**.

4. Starting with RHEL 9.3, you can also specify the **ipaclient_subid: true** option to have subid ranges configured for IdM users on the IdM level.

Additional resources

- **/usr/share/ansible/roles/ipaclient/README.md**
- Managing subID ranges manually

28.3. AUTHORIZATION OPTIONS FOR IDM CLIENT ENROLLMENT USING AN ANSIBLE PLAYBOOK

You can authorize IdM client enrollment by using any of the following methods:

- A random, one-time password (OTP) + administrator password
- A random, one-time password (OTP) + an admin keytab
- The client keytab from the previous enrollment
- The password of a user authorized to enroll a client (**admin**) stored in an inventory file
- The password of a user authorized to enroll a client (**admin**) stored in an Ansible vault

It is possible to have the OTP generated by an IdM administrator before the IdM client installation. In that case, you do not need any credentials for the installation other than the OTP itself.

The following are sample inventory files for these methods:

Table 28.1. Sample inventory files

Authorization option	Inventory file

A random, one-time password (OTP) + administrator password	`[ipaclients:vars]`
	`ipaadmin_password=Secret123`
	`ipaclient_use_otp=true`

A random, one-time password (OTP)	`[ipaclients:vars]`
	`ipaclient_otp=<W5YpARl=7M.>`
	This scenario assumes that the OTP was already generated by an IdM **admin** before the installation.

A random, one-time password (OTP) + an admin keytab	`[ipaclients:vars]`
	`ipaadmin_keytab=/root/admin.keytab`
	`ipaclient_use_otp=true`

The client keytab from the previous enrollment	`[ipaclients:vars]` `ipaclient_keytab=/root/krb5.keytab`
Password of an **admin** user stored in an inventory file	`[ipaclients:vars]` `ipaadmin_password=Secret123`

Authorization option	Inventory file
Password of an **admin** user stored in an Ansible vault file	`[ipaclients:vars]` `[...]`

If you are using the password of an **admin** user stored in an Ansible vault file, the corresponding playbook file must have an additional **vars_files** directive:

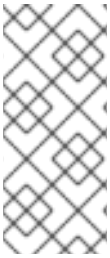
Table 28.2. User password stored in an Ansible vault

Inventory file	Playbook file

`[ipaclients:vars]`
`[...]`
	`- name: Playbook to configure IPA clients`
	`  hosts: ipaclients`
	`  become: true`
	`  vars_files:`
	`  - ansible_vault_file.yml`
	
	`  roles:`
	`  - role: ipaclient`
	`  state: present`

In all the other authorization scenarios described above, a basic playbook file could look as follows:

```
- name: Playbook to configure IPA clients
  hosts: ipaclients
  become: true

  roles:
  - role: ipaclient
    state: true
```



NOTE

As of RHEL 9.2, in the two OTP authorization scenarios described above, the requesting of the administrator's TGT by using the **kinit** command occurs on the first specified or discovered IdM server. Therefore, no additional modification of the Ansible control node is required. Before RHEL 9.2, the **krb5-workstation** package was required on the control node.

28.4. DEPLOYING AN IDM CLIENT USING AN ANSIBLE PLAYBOOK

Complete this procedure to use an Ansible playbook to deploy an IdM client in your IdM environment.

Prerequisites

- The managed node is a Red Hat Enterprise Linux 9 system with a static IP address and a working package manager.
- You have set the parameters of the IdM client deployment to correspond to your deployment scenario:
  - Setting the parameters of the inventory file for the autodiscovery client installation mode
  - Setting the parameters of the inventory file when autodiscovery is not possible during client installation

Procedure

- Run the Ansible playbook:

```
$ ansible-playbook -v -i ~/MyPlaybooks/inventory ~/MyPlaybooks/install-client.yml
```

28.5. USING THE ONE-TIME PASSWORD METHOD IN ANSIBLE TO INSTALL AN IDM CLIENT

You can generate a one-time password (OTP) for a new host in Identity Management (IdM) and use it to enroll a system into the IdM domain. This procedure describes how to use Ansible to install an IdM client after generating an OTP for it on another IdM host.

This method of installing an IdM client is convenient if two system administrators with different privileges exist in your organisation:

- One that has the credentials of an IdM administrator.
- Another that has the required Ansible credentials, including **root** access to the host to become an IdM client.

The IdM administrator performs the first part of the procedure in which the OTP password is generated. The Ansible administrator performs the remaining part of the procedure in which the OTP is used to install an IdM client.

Prerequisites

- You have the IdM **admin** credentials or at least the **Host Enrollment** privilege and a permission to add DNS records in IdM.
- You have configured a user escalation method on the Ansible managed node to allow you to install an IdM client.
- If your Ansible control node is running on RHEL 8.7 or earlier, you must be able to install packages on your Ansible control node.
- You have configured your Ansible control node to meet the following requirements:
  - You are using Ansible version 2.14 or later.
  - You have installed the **ansible-freeipa** package on the Ansible controller.
  - You have created an Ansible inventory file with the fully-qualified domain name (FQDN) of the IdM server.

- The managed node is a Red Hat Enterprise Linux 9 system with a static IP address and a working package manager.

Procedure

1. **SSH** to an IdM host as an IdM user with a role that has the **Host Enrollment** privilege and a permission to add DNS records:

```
$ ssh admin@server.idm.example.com
```

2. Generate an OTP for the new client:

```
[admin@server ~]$ ipa host-add client.idm.example.com --ip-address=172.25.250.11 --random
 ----------------------------------------------------
 Added host "client.idm.example.com"
 ----------------------------------------------------
  Host name: client.idm.example.com
  Random password: W5YpARl=7M.n
  Password: True
  Keytab: False
  Managed by: server.idm.example.com
```

   The **--ip-address=*<your_host_ip_address>*** option adds the host to IdM DNS with the specified IP address.

3. Exit the IdM host:

```
$ exit
logout
Connection to server.idm.example.com closed.
```

4. On the ansible controller, update the inventory file to include the random password:

```
[...]
[ipaclients]
client.idm.example.com

[ipaclients:vars]
ipaclient_domain=idm.example.com
ipaclient_otp=W5YpARl=7M.n
[...]
```

5. If your ansible controller is running RHEL ⇐ 9.1, install the **kinit** utility provided by the **krb5-workstation** package:

```
$ sudo dnf install krb5-workstation
```

6. Run the playbook to install the client:

```
$ ansible-playbook -i inventory install-client.yml
```

28.6. TESTING AN IDENTITY MANAGEMENT CLIENT AFTER ANSIBLE INSTALLATION

The command line (CLI) informs you that the **ansible-playbook** command was successful, but you can also do your own test.

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

```
[user@client1 ~]$ id admin
uid=1254400000(admin) gid=1254400000(admins) groups=1254400000(admins)
```

To test that authentication works correctly, **su -** as another already existing IdM user:

```
[user@client1 ~]$ su - idm_user
Last login: Thu Oct 18 18:39:11 CEST 2018 from 192.168.122.1 on pts/0
[idm_user@client1 ~]$
```

28.7. UNINSTALLING AN IDM CLIENT USING AN ANSIBLE PLAYBOOK

Complete this procedure to use an Ansible playbook to uninstall your host as an IdM client.

Prerequisites

- IdM administrator credentials.
- The managed node is a Red Hat Enterprise Linux 9 system with a static IP address.

Procedure

- Run the Ansible playbook with the instructions to uninstall the client, for example **uninstall-client.yml**:

```
$ ansible-playbook -v -i ~/MyPlaybooks/inventory ~/MyPlaybooks/uninstall-client.yml
```

IMPORTANT

The uninstallation of the client only removes the basic IdM configuration from the host but leaves the configuration files on the host in case you decide to re-install the client. In addition, the uninstallation has the following limitations:

- It does not remove the client host entry from the IdM LDAP server. The uninstallation only unenrolls the host.
- It does not remove any services residing on the client from IdM.
- It does not remove the DNS entries for the client from the IdM server.
- It does not remove the old principals for keytabs other than **/etc/krb5.keytab**.

Note that the uninstallation does remove all certificates that were issued for the host by the IdM CA.

Additional resources

- Uninstalling an IdM client

CHAPTER 29. SECURING DNS WITH DOT IN IDM

You can secure DNS traffic in Identity Management (IdM) deployments by enabling encrypted DNS (eDNS) that uses DNS-over-TLS (DoT). You can encrypt all DNS queries and responses between DNS clients and IdM DNS servers.

IMPORTANT

Encrypted DNS in IdM is a Technology Preview feature only. Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.

For more information about the support scope of Red Hat Technology Preview features, see Technology Preview Features Support Scope .

29.1. ENCRYPTED DNS IN IDM

Encrypted DNS (eDNS), using DNS over TLS (DoT), encrypts all DNS queries and responses between IdM DNS clients and servers. IdM configures the **unbound** service as a local caching resolver on clients and uses the BIND service to receive DoT requests on servers.

By default, IdM uses the **relaxed** DNS policy, which allows fallback to unencrypted DNS if DoT is unavailable. When using the **relaxed** policy, IdM clients and replicas automatically detect DoT-capable DNS servers during installation.

For encrypted-only communication, you can configure the **--dns-policy enforced** option. This setting strictly requires DoT for all DNS resolution and rejects any unencrypted requests. Before installation, you must manually preconfigure both client and replica systems to trust the IdM server's DoT certificate and use it for eDNS resolution.

IdM provides an optional integrated DNS server. When you use the integrated DNS server, IdM automatically manages SRV and other service records as you modify the topology. If you require advanced features such as DNS views, you can manage your DNS records manually on an external DNS server. The integrated IdM DNS is not a general-purpose DNS solution.

When setting up eDNS for your IdM servers, replicas, and clients, you can either use the IdM Certificate Authority (CA) service for certificate management or provide your own certificates. If you don't provide a certificate, IdM CA automatically generates and assigns TLS certificates for the DNS service.

Additional resources

- DNS services available in an IdM server

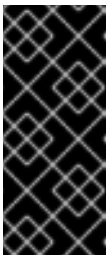
29.2. INSTALLING AN IDM SERVER CONFIGURED TO USE EDNS

You can install an IdM server with DoT enabled by performing a non-interactive installation using the **ipa-server-install** utility. This procedure describes how to configure DoT with the **enforced** policy using the integrated DNS service. If you require the **relaxed** policy instead, you can skip the steps for pre-configuring clients and replicas for DoT exclusively, as the IdM server automatically discovers them.

You can use a certificate issued by the integrated IdM Certificate Authority (CA) or provide a custom certificate issued by an external CA. If you do not provide a certificate, the IdM CA automatically issues a DoT certificate during the installation.

Prerequisites

- Review the steps outlined in Preparing the system for IdM server installation .
- For **enforced** DoT, complete the steps in Configuring client and replica systems to use DoT exclusively.
- Ensure the following packages are installed:
  - **ipa-server**
  - **ipa-server-dns**
  - **ipa-server-encrypted-dns**
  - **ipa-client-encrypted-dns**



IMPORTANT

The **ipa-server-encrypted-dns** package requires a newer version of the **bind-utils** package than the version installed by default on RHEL. Run **sudo dnf install ipa-server ipa-server-encrypted-dns --allowerasing** to allow the package manager to remove the older **bind-utils** package and install the version required by **ipa-server-encrypted-dns**.

Procedure

1. Add the **dns-over-tls** service to the system **firewall** to open port 853/TCP for DoT traffic:

```
# firewall-cmd --add-service=dns-over-tls
```

2. Optional: To use a custom PEM-formatted certificate and key for DoT issued by an external certificate authority, create the files:

```
$ openssl req \
  -newkey rsa:2048 \
  -nodes \
  -keyout /etc/pki/tls/certs/privkey.pem \
  -x509 \
  -days 36500 \
  -out /etc/pki/tls/certs/certificate.pem \
  -subj
"/C=<country_code>/ST=<state>/L=<location>/O=<organization>/OU=<organizational_
unit>/CN=<idm_server_fqdn>/emailAddress=<email>" && \
  chown named:named /etc/pki/tls/certs/privkey.pem /etc/pki/tls/certs/certificate.pem
```

3. Install the IdM server with integrated DNS:

NOTE

If you do not strictly require DoT, you can omit the **--dns-policy** option. The installer then uses the default **relaxed** policy.

- To install the IdM server with externally issued keys and certificates, specify the certificate and key paths:

```
# ipa-server-install --setup-dns --dns-over-tls --dot-forwarder
"<server_ip>#<dns_server_hostname>" --dns-policy enforced --dns-over-tls-cert
/etc/pki/tls/certs/certificate.pem --dns-over-tls-key /etc/pki/tls/certs/privkey.pem --no-dnssec-
validation --auto-reverse --domain <domain_name> --realm <realm_name> --hostname
<idm_server_fqdn> -p <admin_password> -a <admin_password> -U
```

- To install the IdM server with the integrated IdM CA, run the following command:

```
# ipa-server-install --setup-dns --dns-over-tls --dot-forwarder
"<server_ip>#<dns_server_hostname>" --dns-policy enforced --no-dnssec-validation --auto-
reverse --domain <domain_name> --realm <realm_name> --hostname <idm_server_fqdn> -
p <admin_password> -a <admin_password> -U
```

Troubleshooting

1. Enable detailed logging for the **unbound** service:

```
# unbound-control verbosity 3
```

2. Restart the **unbound** service to apply the updated configuration:

```
# systemctl restart unbound
```

3. Monitor real-time logs of the **unbound** service:

```
$ journalctl -u unbound -f
```

Additional resources

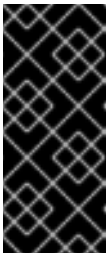
- Creating and managing TLS keys and certificates
- **firewall-cmd(1)**, **ipa-server-install(1)**, and **ipa-dns-install(1)** man pages on your system
- DoT configuration options for **ipa-server-install** and **ipa-dns-install**

29.3. CONFIGURING CLIENT AND REPLICA SYSTEMS TO USE DOT EXCLUSIVELY

To enforce DoT communication, you must configure clients and replica systems to use a DoT-capable resolver. You must update the DNS settings in NetworkManager to enable eDNS communication. This configuration is only required when the **--dns-policy** is set to **enforced**.

Prerequisites

- Review the steps outlined in Preparing the system for IdM client installation and Preparing the system for an IdM replica installation.
- Ensure the following packages are installed:
  - **ipa-server-encrypted-dns**
  - **ipa-client-encrypted-dns**



IMPORTANT

The **ipa-server-encrypted-dns** package requires a newer version of the **bind-utils** package than the version installed by default on RHEL. Run **sudo dnf install ipa-server ipa-server-encrypted-dns --allowerasing** to allow the package manager to remove the older **bind-utils** package and install the version required by **ipa-server-encrypted-dns**.

Procedure

1. Copy the IdM server's DoT certificate to the client and replica system.

```
$ scp /etc/pki/tls/certs/bind_dot.crt <username>@<ip>:/etc/pki/ca-trust/source/anchors/
```

2. Update the system-wide trust store configuration:

```
# update-ca-trust extract
```

3. On the client and replica system, install the **dnsconfd** package:

```
# dnf install dnsconfd
```

4. Generate the default configuration files for DoT on your system:

```
dnsconfd config install
```

5. Enable the **dnsconfd** service:

```
# systemctl enable --now dnsconfd
```

6. Reload NetworkManager to apply the configuration:

```
# nmcli g reload
```

7. Configure the system's DNS settings in NetworkManager.

```
# nmcli device modify <device_name> ipv4.dns dns+tls://<idm_server_ip>

Connection successfully reapplied to device '<device_name>'.
```

Additional resources

- DoT configuration options for **ipa-server-install** and **ipa-dns-install**

29.4. INSTALLING AN IDM CLIENT CONFIGURED TO USE EDNS

You can install an IdM client with DNS-over-TLS (DoT) enabled by performing the non-interactive installation. This setup applies the **enforced** DoT policy and requires the client to use eDNS queries exclusively.

Prerequisites

- Review the steps outlined in the Preparing the system for IdM client installation .
- For **enforced** DoT, complete the steps in Configuring client and replica systems to use DoT exclusively.
- Ensure the **ipa-client** and **ipa-client-encrypted-dns** packages are installed.

Procedure

- Install an IdM client with DoT enabled:

```
# ipa-client-install --domain <domain_name> --dns-over-tls -p admin --password
<admin_password> -U
```

Verification

1. On the IdM client, review cat /**etc/unbound/unbound.conf**:

```
$ cat /etc/unbound/unbound.conf
```

2. Verify that the configuration contains the IP address and hostname of the IdM server.

Troubleshooting

1. On the IdM client, run a DNS query to trigger traffic:

```
$ dig <domain_name>
```

2. Review the logs on the IdM server to verify that the query was routed through DoT.

Additional resources

- **ipa-client-install(1)** man page on your system

29.5. INSTALLING AN IDM REPLICA CONFIGURED TO USE EDNS

You can install an IdM replica with eDNS in an environment where the IdM server has DoT enabled.

When you install the replica with the integrated DNS service, the replica uses the same configuration as the IdM server. It runs BIND to handle incoming DNS queries, including encrypted queries, and uses **unbound** for outgoing encrypted DNS traffic.

When you install the replica without the integrated DNS service, the replica inherits the client-side configuration. It uses **unbound** with a DoT forwarder to send encrypted DNS queries to the IdM DNS server.

Prerequisites

- Review the steps outlined in Preparing the system for an IdM replica installation .
- For **enforced** DoT, complete the steps in Configuring client and replica systems to use DoT exclusively.
- Ensure the **ipa-client-encrypted-dns** and **ipa-server-encrypted-dns** packages are installed.

Procedure

1. Add the **dns-over-tls** service to the system **firewall** to open port 853/TCP for DoT traffic:

```
# firewall-cmd --add-service=dns-over-tls
```

2. Depending on whether you want the replica to manage DNS records, choose one of the following:
   - To install an IdM replica with integrated DNS:

```
# ipa-replica-install --setup-dns --dns-over-tls --dot-forwarder
<server_ip>#<dns_server_hostname>
```

   - To install an IdM replica without integrated DNS:

```
# ipa-replica-install --dns-over-tls
```

Verification

- On the IdM server, list all replicas in the topology:

```
# ipa-replica-manage list-ruv
```

Additional resources

- **ipa-replica-install(1)** man page on your system
- DoT configuration options for **ipa-server-install** and **ipa-dns-install**

29.6. CONFIGURING AN EXISTING IDM DNS SERVER TO USE EDNS

You can enable DNS-over-TLS (DoT) on an existing Identity Management (IdM) server by reconfiguring the integrated DNS service. Use the **ipa-dns-install** utility with DoT-specific options to update the DNS configuration without reinstalling the server.

Prerequisites

- You have root access to the IdM server.
- DNS is already installed on the IdM server.

Procedure

1. Optional: Verify that your IdM server uses integrated DNS:

```
$ ipa server-role-find --role 'DNS server'
----------------------
1 server role matched
----------------------
  Server name: server.idm.example.com
  Role name: DNS server
  Role status: enabled
---------------------------
Number of entries returned 1
---------------------------
```

2. Update the integrated DNS service to enable DoT and configure DoT policy and forwarders:

```
# ipa-dns-install --dns-over-tls --dot-forwarder "<server_ip>#<dns_server_hostname>" --dns-policy enforced -U
```

3. Add the **dns-over-tls** service to the system **firewall** to open port 853/TCP for DoT traffic:

```
# firewall-cmd --add-service=dns-over-tls
```

Verification

- Verify that the firewall allows DoT traffic:

```
# firewall-cmd --list-services
```

Additional resources

- Installing DNS on an existing IdM server
- **ipa-dns-install(1)** man page on your system
- DoT configuration options for **ipa-server-install** and **ipa-dns-install**

29.7. DOT CONFIGURATION OPTIONS FOR IPA-SERVER-INSTALL AND IPA-DNS-INSTALL

Learn about the available configuration options for enabling eDNS in your IdM deployment. You can use the same options to enable eDNS during a new IdM server installation using the **ipa-server-install** or to modify an existing installation using the **ipa-dns-install** command.

- **--dns-over-tls** enables DoT
- **--dot-forwarder** specifies upstream DoT servers using the **--dot-forwarder *<server_ip_1><dns_server_hostname_1>* --dot-forwarder *<server_ip_2><dns_server_hostname_2>*** format
- **--dns-over-tls-key** and **--dns-over-tls-cert** to configure custom keys and certificates
- **--dns-policy** sets the DNS security policy

- **--dns-policy=relaxed** allows both encrypted (DoT) and unencrypted DNS queries. The system attempts to use DoT but falls back to unencrypted DNS if DoT is unavailable. This is the default policy.
- **--dns-policy=enforced** requires only encrypted DNS communication. The system strictly enforces DoT, and rejects any DNS resolution that does not support encryption, including discovery from IdM clients and replicas.

Additional resources

- **ipa-dns-install(1)** man page on your system
- DNS OPTIONS section in the **ipa-server-install(1)** man page on your system

CHAPTER 30. INSTALLING DNS ON AN EXISTING IDM SERVER

Install the DNS service on an Identity Management (IdM) server that was originally installed without it.

Prerequisites

- You understand the advantages and limitations of using IdM with integrated DNS as described in Installing an IdM server: With integrated DNS, with an integrated CA as the root CA .
- You have **root** access to the IdM server.

Procedure

1. Optional: Verify that DNS is not already installed on the IdM server.

```
[root@r8server ~]# ipa server-role-show r8server.idm.example.com
Role name: DNS server
  Server name: r8server.idm.example.com
  Role name: DNS server
  Role status: absent
```

   The output confirms that IdM DNS is not available on the server.

2. Download the **ipa-dns-server** package and its dependencies:

```
[root@r8server ~]# dnf install ipa-server-dns
```

3. Start the script to install DNS on the server:

```
[root@r8server ~]# ipa-dns-install
```

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

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

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

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

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

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

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

NOTE

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

Additional resources

- **man ipa-dns-install(1)**

CHAPTER 31. UNINSTALLING THE INTEGRATED IDM DNS SERVICE FROM AN IDM SERVER

If you have more than one server with integrated DNS in an Identity Management (IdM) deployment, you might decide to remove the integrated DNS service from one of the servers. To do this, you must first decommission the IdM server completely before re-installing IdM on it, this time without the integrated DNS.

NOTE

While you can add the DNS role to an IdM server, IdM does not provide a method to remove only the DNS role from an IdM server: the **ipa-dns-install** command does not have an **--uninstall** option.

Prerequisites

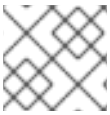
- You have integrated DNS installed on an IdM server.
- This is not the last integrated DNS service in your IdM topology.

Procedure

1. Identify the redundant DNS service and follow the procedure in Uninstalling an IdM server on the IdM replica that hosts this service.
2. On the same host, follow the procedure in either Without integrated DNS, with an integrated CA as the root CA or Without integrated DNS, with an external CA as the root CA , depending on your use case.

CHAPTER 32. ADDING THE IDM CA SERVICE TO AN IDM SERVER IN A DEPLOYMENT WITHOUT A CA

If you previously installed an Identity Management (IdM) domain without the certificate authority (CA) component, you can add the IdM CA service to the domain by using the **ipa-ca-install** command. Depending on your requirements, you can select one of the following options:



NOTE

For details on the supported CA configurations, see Planning your CA services.

32.1. INSTALLING THE FIRST IDM CA AS THE ROOT CA INTO AN EXISTING IDM DOMAIN

If you previously installed Identity Management (IdM) without the certificate authority (CA) component, you can install the CA on an IdM server subsequently. Follow this procedure to install, on the *idmserver* server, an IdM CA that is not subordinate to any external root CA.

Prerequisites

- You have **root** permissions on *idmserver*.
- The IdM server is installed on *idmserver*.
- Your IdM deployment has no CA installed.
- You know the IdM **Directory Manager** password.

Procedure

1. On *idmserver*, install the IdM Certificate Server CA:

```
[root@idmserver ~] ipa-ca-install
```

2. On each IdM host in the topology, run the **ipa-certupdate** utility to update the host with the information about the new certificate from the IdM LDAP.

IMPORTANT

If you do not run **ipa-certupdate** after generating the IdM CA certificate, the certificate will not be distributed to the other IdM machines.

32.2. INSTALLING THE FIRST IDM CA WITH AN EXTERNAL CA AS THE ROOT CA INTO AN EXISTING IDM DOMAIN

If you previously installed Identity Management (IdM) without the certificate authority (CA) component, you can install the CA on an IdM server subsequently. Follow this procedure to install, on the *idmserver* server, an IdM CA that is subordinate to an external root CA, with zero or several intermediate CAs in between.

Prerequisites

- You have **root** permissions on *idmserver*.
- The IdM server is installed on *idmserver*.
- Your IdM deployment has no CA installed.
- You know the IdM **Directory Manager** password.

Procedure

1. Start the installation:

```
[root@idmserver ~] ipa-ca-install --external-ca
```

2. Wait till the command line informs you that a certificate signing request (CSR) has been saved.
3. Submit the CSR to the external CA.
4. Copy the issued certificate to the IdM server.
5. Continue the installation by adding the certificates and full path to the external CA files to **ipa-ca-install**:

```
[root@idmserver ~]# ipa-ca-install --external-cert-file=/root/master.crt --external-cert-file=/root/ca.crt
```

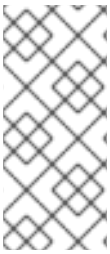
6. On each IdM host in the topology, run the **ipa-certupdate** utility to update the host with the information about the new certificate from the IdM LDAP.

IMPORTANT

Failing to run **ipa-certupdate** after generating the IdM CA certificate means that the certificate will not be distributed to the other IdM machines.

CHAPTER 33. ADDING THE IDM CA SERVICE TO AN IDM SERVER IN A DEPLOYMENT WITH A CA

If your Identity Management (IdM) environment already has the IdM certificate authority (CA) service installed but a particular IdM server, *idmserver*, was installed as an IdM replica without a CA, you can add the CA service to *idmserver* by using the **ipa-ca-install** command.



NOTE

This procedure is identical for both the following scenarios:

- The IdM CA is a root CA.
- The IdM CA is subordinate to an external, root CA.

Prerequisites

- You have **root** permissions on *idmserver*.
- The IdM server is installed on *idmserver*.
- Your IdM deployment has a CA installed on another IdM server.
- You know the IdM **Directory Manager** password.

Procedure

- On *idmserver*, install the IdM Certificate Server CA:

```
[root@idmserver ~] ipa-ca-install
```

CHAPTER 34. UNINSTALLING THE IDM CA SERVICE FROM AN IDM SERVER

If you have more than four Identity Management (IdM) replicas with the **CA role** in your topology and you run into performance problems due to redundant certificate replication, remove redundant CA service instances from IdM replicas. To do this, you must first decommission the affected IdM replicas completely, then reinstall IdM on them without the CA service.

NOTE

While you can **add** the CA role to an IdM replica, IdM does not provide a method to **remove** only the CA role from an IdM replica: the **ipa-ca-install** command does not have an **--uninstall** option.

Prerequisites

- You have the IdM CA service installed on more than four IdM servers in your topology.

Procedure

1. Identify the redundant CA service and follow the procedure in Uninstalling an IdM server on the IdM replica that hosts this service.
2. On the same host, follow the procedure in Installing an IdM server: With integrated DNS, without a CA.