Red Hat Enterprise Linux 9

Tuning performance in Identity Management

Optimizing the IdM services, such as Directory Server, KDC, and SSSD, for better performance

Last Updated: 2025-05-20

Red Hat Enterprise Linux 9 Tuning performance in Identity Management

Optimizing the IdM services, such as Directory Server, KDC, and SSSD, for better performance

Legal Notice

Abstract

Red Hat tunes Identity Management (IdM) to perform well in most deployments. However, in specific scenarios, it can be beneficial to tune IdM components, such as replication agreements, the Directory Server, the Kerberos Key Distribution Center (KDC), or the System Security Services Daemon (SSSD).

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. IMPORTANT CONSIDERATIONS WHEN TUNING IDM

Identity Management's component services are tuned to work in an optimal way for most deployments. As a system administrator, you might want to adjust the performance of IdM services to suit the demands of your specific environment.

Important considerations

- Each IdM deployment is a unique combination of hardware, software, networking, data, workloads, and many other factors. Adjustments that benefit one environment may be detrimental to another.
- Performance-tuning is an iterative, experimental process. Red Hat recommends making adjustments to only one variable at a time and monitoring its impact in your environment. After achieving the desired result with one variable, adjust the next variable while continuing to monitor the performance of previous adjustments.

CHAPTER 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.

CHAPTER 3. IDM SERVER PERFORMANCE RECOMMENDATIONS

To ensure stable performance, Identity Management (IdM) enforces limits on the maximum number of users and clients that you can add or enroll to IdM server simultaneously.

Table 3.1. Limits for IdM operations

Action	Description	Number
Client enrollments	The maximum number of the IdM clients that you can enroll with an IdM server at the same time before enrollment starts failing.	130
Adding users	The maximum number of users you can add at the same time using the **ipa user-add[]** command from different IdM clients before failing to add a user. You can add more users over the same amount of time by using the IdM API **batch** command. We recommend adding users in batches of 100 users.	325
Client authentications	The maximum number of the IdM clients that can authenticate at the same time before authentications start to fail.	800
Adding a member to a user group	The recommended number of members in a group that you can add without exceeding time for adding a new member to a group. IdM has a two-seconds rule as a normal time frame for adding a member to a group. You can add more members, but the time of the action will progressively extend.	1500

Additional resources

- Using batches for executing IdM API commands

CHAPTER 4. FAILOVER, LOAD-BALANCING, AND HIGH-AVAILABILITY IN IDM

Identity Management (IdM) has built-in failover mechanisms for IdM clients, and load-balancing and high-availability features for IdM servers.

Client-side failover capability

By default, the SSSD service on an IdM client is configured to use DNS service (SRV) resource records so that the client can automatically determine the best IdM server to connect to.

Primary and backup server configuration

The server resolution behavior is controlled by the **_srv_** option in the **ipa_server** parameter of the **/etc/sssd/sssd.conf** file:

Example /etc/sssd/sssd.conf

```
[domain/<idm_domain_name>]
id_provider = ipa
ipa_server = _srv_, <primary_idm_server1>, <primary_idm_server2>
ipa_backup_server = <backup_idm_server1>, <backup_idm_server2>
...
```

With the **_srv_** option specified, SSSD retrieves a list of IdM servers ordered by preference. If a primary server goes offline, the SSSD service on the IdM client automatically connects to another available IdM server.

Primary servers are specified in the **ipa_server** parameter. SSSD attempts to connect to primary servers first and switches to backup servers only if no primary servers are available.

The **_srv_** option is not supported for backup servers.

NOTE

SSSD queries SRV records from the DNS server. By default, SSSD waits for **6** seconds for a reply from the DNS resolver before attempting to query another DNS server. If all DNS servers are unreachable, the domain will continue to operate in offline mode. You can use the **dns_resolver_timeout** option to increase the time the client waits for a reply from the DNS resolver.

If you prefer to bypass DNS lookups for performance reasons, remove the **_srv_** entry from the **ipa_server** parameter and specify which IdM servers the client should connect to, in order of preference:

Example /etc/sssd/sssd.conf

```
[domain/<idm_domain_name>]
id_provider = ipa
ipa_server = <primary_idm_server1>, <primary_idm_server2>
ipa_backup_server = <backup_idm_server1>, <backup_idm_server2>
...
```

Failover behavior for IdM servers and services

SSSD failover mechanism treats an IdM server and its services independently. If the hostname resolution

for a server succeeds, SSSD considers the machine is online and tries to connect to the required service on that machine. If the connection to the service fails, SSSD considers only that specific service as offline, not the entire machine or other services on it.

If hostname resolution fails, SSSD considers the entire machine as offline, and does not attempt to connect to any services on that machine.

When all primary servers are unavailable, SSSD attempts to connect to a configured backup server. While connected to a backup server, SSSD periodically attempts to reconnect to one of the primary servers and connects immediately once a primary server becomes available. The interval between these attempts is controlled by the **failover_primary_timeout** option , which defaults to 31 seconds.

If all IdM servers become unreachable, SSSD switches to offline mode. In this state, SSSD retries connections every 30 seconds until a server becomes available.

Server-side load-balancing and service availability

You can achieve load-balancing and high-availability in IdM by installing multiple IdM replicas:

- If you have a geographically dispersed network, you can shorten the path between IdM clients and the nearest accessible server by configuring multiple IdM replicas per data center.
- Red Hat supports environments with up to 60 replicas.
- The IdM replication mechanism provides active/active service availability: services at all IdM replicas are readily available at the same time.

NOTE

Red Hat recommends against combining IdM and other load-balancing or high-availability (HA) software.

Many third-party high availability solutions assume active/passive scenarios and cause unnecessary service interruption to IdM availability. Other solutions use virtual IPs or a single hostname per clustered service. All these methods do not typically work well with the type of service availability provided by the IdM solution. They also integrate very poorly with Kerberos, decreasing the overall security and stability of the deployment.

Additional resources

- **sssd.conf(5)** man pages on your system

CHAPTER 5. OPTIMIZING THE REPLICA TOPOLOGY

A robust replica topology distributes workloads and reduces replication delays. Follow these guidelines to optimize the layout of your replica topology.

5.1. GUIDELINES FOR DETERMINING THE APPROPRIATE NUMBER OF IDM REPLICAS IN A TOPOLOGY

Plan IdM topology to match your organization's requirements and ensure optimal performance and service availability.

Set up at least two replicas in each data center

Deploy at least two replicas in each data center to ensure that if one server fails, the replica can take over and handle requests.

Set up a sufficient number of servers to serve your clients

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

Set up a sufficient number of Certificate Authority (CA) replicas

Only replicas with the CA role installed can replicate certificate data. If you use the IdM CA, ensure your environment has at least two CA replicas with certificate replication agreements between them.

Set up a maximum of 60 replicas in a single IdM domain

Red Hat supports environments with up to 60 replicas.

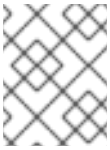
5.2. GUIDELINES FOR CONNECTING IDM REPLICAS IN A TOPOLOGY

Connect each replica to at least two other replicas

This ensures that information is replicated not just between the initial replica and the first server you installed, but between other replicas as well.

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

A large number of replication agreements per server does not add significant benefits. A receiving replica can only be updated by one other replica at a time and meanwhile, the other replication agreements are idle. More than four replication agreements per replica typically means a waste of resources.



NOTE

This recommendation applies to both certificate replication and domain replication agreements.

There are two exceptions to the limit of four replication agreements per replica:

- You want failover paths if certain replicas are not online or responding.
- In larger deployments, you want additional direct links between specific nodes.

Configuring a high number of replication agreements can have a negative impact on overall performance: when multiple replication agreements in the topology are sending updates, certain replicas can experience a high contention on the changelog database file between incoming updates and the outgoing updates.

If you decide to use more replication agreements per replica, ensure that you do not experience replication issues and latency. However, note that large distances and high numbers of intermediate nodes can also cause latency problems.

Connect the replicas in a data center with each other

This ensures domain replication within the data center.

Connect each data center to at least two other data centers

This ensures domain replication between data centers.

Connect data centers using at least a pair of replication agreements

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

5.3. REPLICA TOPOLOGY EXAMPLES

You can create a reliable replica topology by using one of the following examples.

Figure 5.1. Replica topology with four data centers, each with four servers that are connected with replication agreements

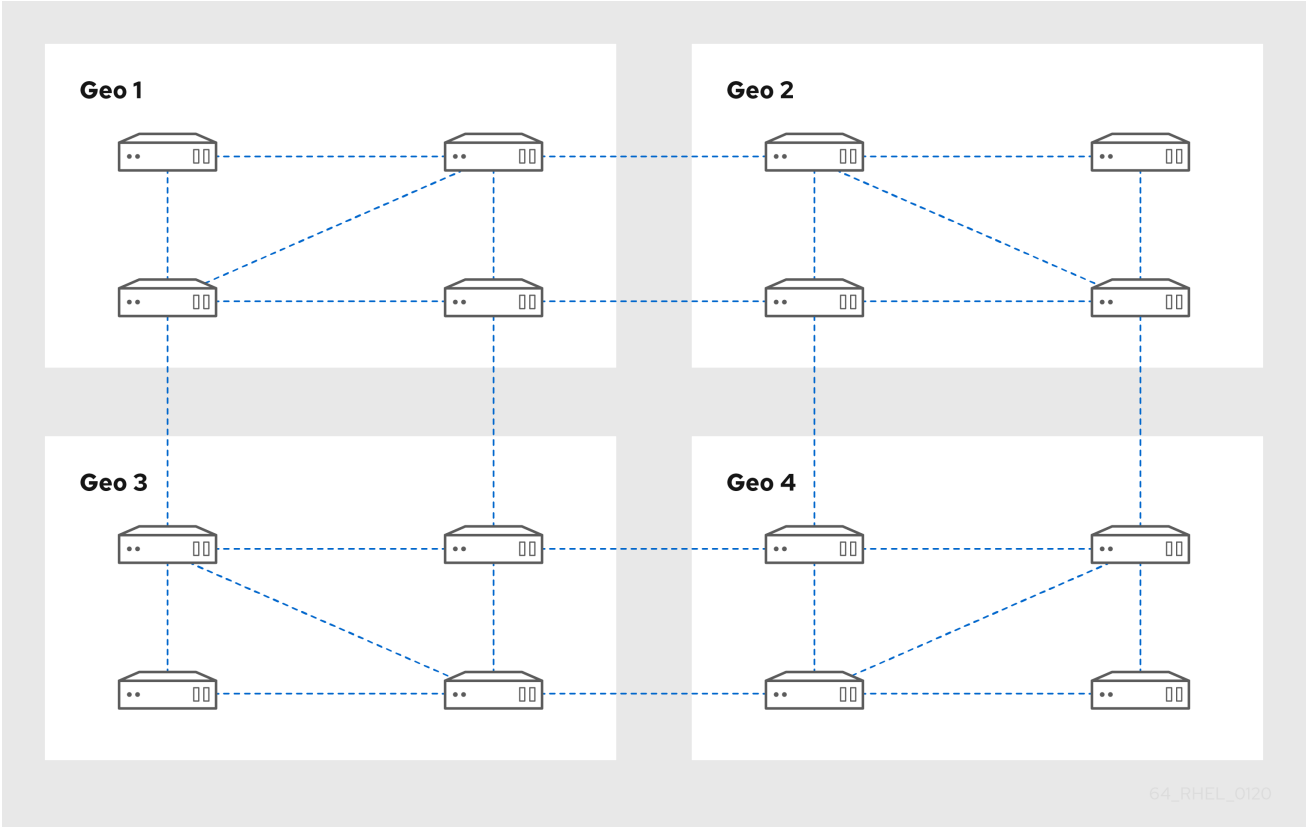
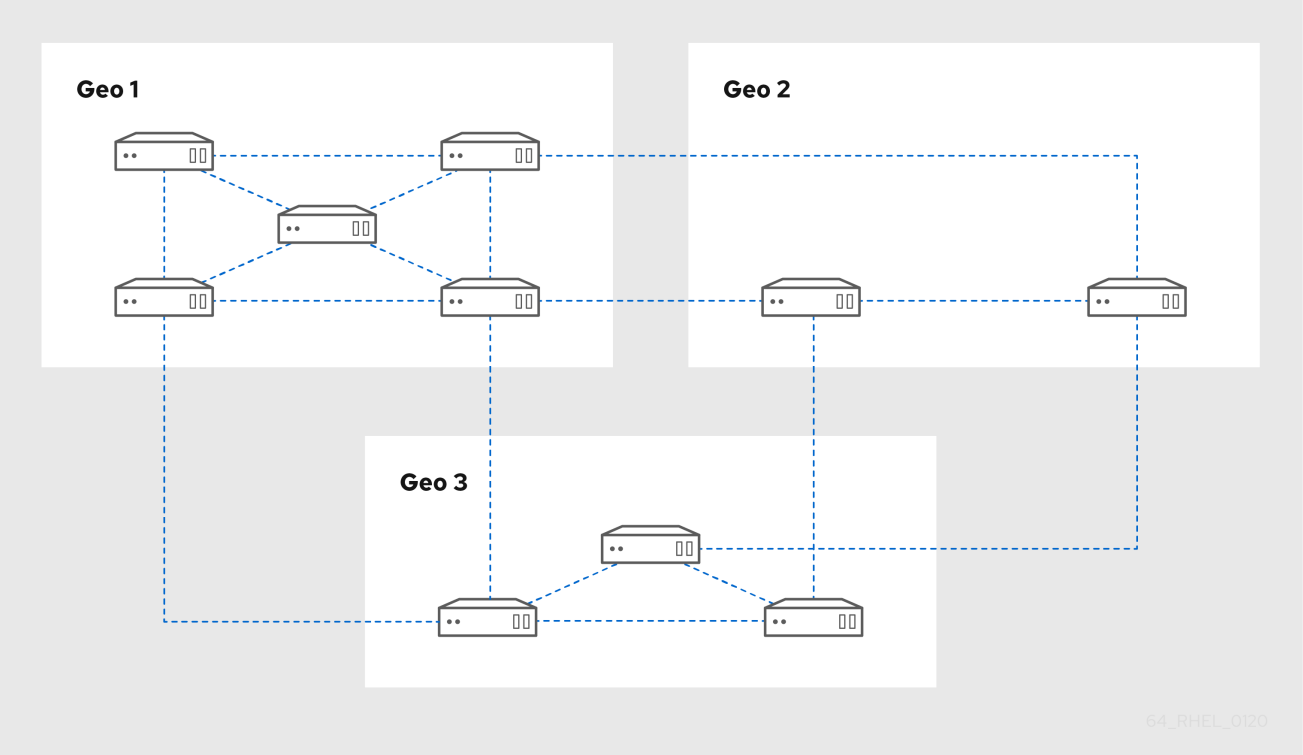
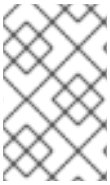


Figure 5.2. Replica topology with three data centers, each with a different number of servers that are all interconnected through replication agreements



5.4. 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.

5.5. ADDITIONAL RESOURCES

- Planning the replica topology
- Managing replication topology

CHAPTER 6. ADJUSTING THE SEARCH SIZE AND TIME LIMIT

Some queries, such as requesting a list of IdM users, can return a very large number of entries. By tuning these search operations, you can improve the overall server performance when running the **ipa *-find** commands, such as **ipa user-find**, and when displaying corresponding lists in the Web UI.

Search size limit

Defines the maximum number of entries returned for a request sent to the server from a client's CLI or from a browser accessing the IdM Web UI.
Default: 100 entries.

Search time limit

Defines the maximum time (in seconds) that the server waits for searches to run. Once the search reaches this limit, the server stops the search and returns the entries discovered in that time.
Default: 2 seconds.

If you set the values to **-1**, IdM will not apply any limits when searching.

IMPORTANT

Setting search size or time limits too high can negatively affect server performance.

6.1. ADJUSTING THE SEARCH SIZE AND TIME LIMIT IN THE COMMAND LINE

You can adjust the search size and time limits globally or for a specific entry to optimize search performance and responsiveness.

Procedure

1. To display current search time and size limits in CLI, use the **ipa config-show** command:

```
$ ipa config-show

Search time limit: 2
Search size limit: 100
```

2. To adjust the limits **globally** for all queries, use the **ipa config-mod** command and add the **--searchrecordslimit** and **--searchtimelimit** options. For example:

```
$ ipa config-mod --searchrecordslimit=500 --searchtimelimit=5
```

3. To **temporarily** adjust the limits only for a specific query, add the **--sizelimit** or **--timelimit** options to the command. For example:

```
$ ipa user-find --sizelimit=200 --timelimit=120
```

6.2. ADJUSTING THE SEARCH SIZE AND TIME LIMIT IN THE WEB UI

You can adjust global search size and time limits using the IdM Web UI to optimize search performance and responsiveness.

Procedure

1. Log in to the IdM Web UI.
2. Click **IPA Server**.
3. On the **IPA Server** tab, click **Configuration**.
4. Set the required values in the **Search Options** area.
   Default values are:
   - Search size limit: 100 entries
   - Search time limit: 2 seconds
5. Click **Save** at the top of the page.

CHAPTER 7. ADJUSTING IDM DIRECTORY SERVER PERFORMANCE

You can tune the performance of Identity Management's databases by adjusting LDAP attributes controlling the Directory Server's resources and behavior.

You can fine-tune the following:

- Adjust how the Directory Server **caches data**.
- Adjust the Directory Server's **resource limits**.
- Adjust **timeouts** that have the most influence on performance.
- Install an IdM server or replica with custom Directory Server settings from an LDIF file.

7.1. ADJUSTING THE ENTRY CACHE SIZE IN THE IDM DIRECTORY SERVER

IMPORTANT

Do not change this settings, unless you have a strong need to apply your custom values. IdM Directory Server uses the built-in cache auto-sizing feature for optimized performance.

The **nsslapd-cachememsize** attribute specifies the size, in bytes, for the available memory space for the entry cache. This attribute is one of the most important values for controlling how much physical RAM the Directory Server uses.

If the entry cache size is too small, you might see the following error in the Directory Server error logs in the **/var/log/dirsrv/slapd-*<instance_name>*/errors** log file:

> REASON: entry too large (*83886080* bytes) for the import buffer size (*67108864* bytes). **Try increasing nsslapd-cachememsize.**

Red Hat recommends fitting the entry cache and the database index entry cache in memory.

Table 7.1. nsslapd-cachememsize attribute values

Default value	**209715200** *(200 MiB)*
Valid range	**500000 - 18446744073709551615** *(500 kB - (2^{64}-1))*
Entry DN location	**cn=*<database_name>*,cn=ldbm database,cn=plugins,cn=config**

Prerequisites

- The LDAP Directory Manager password

Procedure

Procedure

1. Disable automatic cache tuning.

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://<server_fqdn> backend config set --cache-autosize=0
```

2. Display the database suffixes and their corresponding back ends.

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://<server_fqdn> backend suffix list
cn=changelog (changelog)
dc=example,dc=com (userroot)
o=ipaca (ipaca)
```

   This command displays the name of the back end database next to each suffix. Use the suffix's database name in the next step.

3. Set the entry cache size for the database. This example sets the entry cache for the userroot database to 2 gigabytes.

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://<server_fqdn> backend suffix set --cache-memsize=2147483648 userroot
```

4. Restart the Directory Server.

```
[root@server ~]# systemctl restart dirsrv.target
```

5. Monitor the IdM Directory Server's performance. If it does not improve, repeat this procedure and adjust **cache-memsize** to a different value, or re-enable cache auto-sizing.

Verification

- Display the value of the **nsslapd-cachememsize** attribute and verify it has been set to your desired value.

```
[root@server ~]# ldapsearch -D "cn=Directory Manager" -w <directory_manager_password> -b "cn=userroot,cn=ldbm database,cn=plugins,cn=config" | grep nsslapd-cachememsize
nsslapd-cachememsize: 2147483648
```

Additional resources

- nsslapd-cachememsize in Directory Server 11 documentation
- Re-enabling entry and database cache auto-sizing.

7.2. ADJUSTING THE DATABASE INDEX CACHE SIZE IN THE IDM DIRECTORY SERVER

IMPORTANT

Do not change this settings, unless you have a strong need to apply your custom values. IdM Directory Server uses the built-in cache auto-sizing feature for optimized performance.

The **nsslapd-dbcachesize** attribute controls the amount of memory the database indexes use. This cache size has less of an impact on Directory Server performance than the entry cache size does, but if there is available RAM after the entry cache size is set, Red Hat recommends increasing the amount of memory allocated to the database cache.

The database cache is limited to 1.5 GB RAM because higher values do not improve performance.

Table 7.2. nsslapd-dbcachesize attribute values

Default value	**10000000** *(10 MB)*
Valid range	**500000 - 1610611911** *(500 kB - 1.5GB)*
Entry DN location	**cn=config,cn=ldbm database,cn=plugins,cn=config**

Prerequisites

- The LDAP Directory Manager password

Procedure

1. Disable automatic cache tuning, and set the database cache size. This example sets the database cache to 256 megabytes.

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://<server_fqdn> backend config set --cache-autosize=0 --dbcachesize=268435456
```

2. Restart the Directory Server.

```
[root@server ~]# systemctl restart dirsrv.target
```

3. Monitor the IdM Directory Server's performance. If it does not improve, repeat this procedure and adjust **dbcachesize** to a different value, or re-enable cache auto-sizing.

Verification

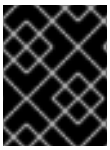
- Display the value of the **nsslapd-dbcachesize** attribute and verify it has been set to your desired value.

```
[root@server ~]# ldapsearch -D "cn=Directory Manager" -w <directory_manager_password> -b "cn=config,cn=ldbm database,cn=plugins,cn=config" | grep nsslapd-dbcachesize
nsslapd-dbcachesize: 2147483648
```

Additional resources

- nsslapd-dbcachesize in Directory Server 11 documentation
- Re-enabling entry and database cache auto-sizing.

7.3. RE-ENABLING DATABASE AND ENTRY CACHE AUTO-SIZING IN THE IDM DIRECTORY SERVER



IMPORTANT

Use the built-in cache auto-sizing feature for optimized performance. Do not set cache sizes manually.

By default, the IdM Directory Server automatically determines the optimal size for the database cache and entry cache. Auto-sizing sets aside a portion of free RAM and optimizes the size of both caches based on the hardware resources of the server when the instance starts.

Use this procedure to undo custom database cache and entry cache values and restore the cache auto-sizing feature to its default values.

Table 7.3. nsslapd-cache-autosize attribute values

nsslapd-cache-autosize	This settings controls how much free RAM is allocated for auto-sizing the database and entry caches. A value of **0** disables auto-sizing.
Default value	**10** *(10% of free RAM)*
Valid range	**0 - 100**
Entry DN location	**cn=config,cn=ldbm database,cn=plugins,cn=config**

Table 7.4. nsslapd-cache-autosize-split attribute values

nsslapd-cache-autosize-split	This value sets the percentage of free memory determined by **nsslapd-cache-autosize** that is used for the database cache. The remaining percentage is used for the entry cache.
Default value	**25** *(25% for the database cache, 60% for the entry cache)*
Valid range	**0 - 100**
Entry DN location	**cn=config,cn=ldbm database,cn=plugins,cn=config**

Prerequisites

- You have previously disabled database and entry cache auto-tuning.

Procedure

1. Stop the Directory Server.

```
[root@server ~]# systemctl stop dirsrv.target
```

2. Backup the **/etc/dirsrv/*slapd-<instance_name>*/dse.ldif** file before making any further modifications.

```
[root@server ~]# cp /etc/dirsrv/slapd-<instance_name>/dse.ldif \ /etc/dirsrv/slapd-<instance_name>/dse.ldif.bak.$(date "+%F_%H-%M-%S")
```

3. Edit the **/etc/dirsrv/*slapd-<instance_name>*/dse.ldif** file:

   a. Set the percentage of free system RAM to use for the database and entry caches back to the default of 10% of free RAM.

```
nsslapd-cache-autosize: 10
```

   b. Set the percentage used from the free system RAM for the database cache to the default of 25%:

```
nsslapd-cache-autosize-split: 25
```

4. Save your changes to the **/etc/dirsrv/*slapd-<instance_name>*/dse.ldif** file.

5. Start the Directory Server.

```
[root@server ~]# systemctl start dirsrv.target
```

Verification

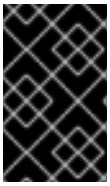
- Display the values of the **nsslapd-cache-autosize** and **nsslapd-cache-autosize-split** attributes and verify they have been set to your desired values.

```
[root@server ~]# ldapsearch -D "cn=Directory Manager" -w <directory_manager_password> -b "cn=config,cn=ldbm database,cn=plugins,cn=config" | grep nsslapd-cache-autosize nsslapd-cache-autosize: *10
nsslapd-cache-autosize-split: 25
```

Additional resources

- nsslapd-cache-autosize in Directory Server 11 documentation

7.4. ADJUSTING THE DN CACHE SIZE IN THE IDM DIRECTORY SERVER



IMPORTANT

Do not change this settings, unless you have a strong need to apply your custom values. IdM Directory Server uses the built-in cache auto-sizing feature for optimized performance.

The **nsslapd-dncachememsize** attribute specifies the size, in bytes, for the available memory space for the Distinguished Names (DN) cache. The DN cache is similar to the entry cache for a database, but its table stores only the entry ID and the entry DN, which allows faster lookups for **rename** and **moddn** operations.

Table 7.5. nsslapd-dncachememsize attribute values

Default value	**10485760** *(10 MB)*
Valid range	**500000 - 18446744073709551615** *(500 kB - (2^{64}-1))*
Entry DN location	**cn=*database-name*,cn=ldbm database,cn=plugins,cn=config**

Prerequisites

- The LDAP Directory Manager password

Procedure

1. Optional: Display the database suffixes and their corresponding database names.

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://<server_fqdn> backend suffix list
dc=example,dc=com (userroot)
```

   This command displays the name of the back end database next to each suffix. Use the suffix's database name in the next step.

2. Set the DN cache size for the database. This example sets the DN cache to 20 megabytes.

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://<server_fqdn> backend suffix set --dncache-memsize=20971520 userroot
```

3. Restart the Directory Server.

```
[root@server ~]# systemctl restart dirsrv.target
```

4. Monitor the IdM Directory Server's performance. If it does not improve, repeat this procedure and adjust **dncache-memsize** to a different value, or back to the default of 10 MB.

Verification

- Display the new value of the **nsslapd-dncachememsize** attribute and verify it has been set to your desired value.

```
[root@server ~]# ldapsearch -D "cn=Directory Manager" -w <directory_manager_password> -b "cn=userroot,cn=ldbm database,cn=plugins,cn=config" | grep nsslapd-dncachememsize
nsslapd-dncachememsize: 20971520
```

Additional resources

- nsslapd-dncachememsize in Directory Server 11 documentation

7.5. ADJUSTING THE NORMALIZED DN CACHE SIZE IN THE IDM DIRECTORY SERVER

IMPORTANT

Do not change this settings, unless you have a strong need to apply your custom values. IdM Directory Server uses the built-in cache auto-sizing feature for optimized performance.

The **nsslapd-ndn-cache-max-size** attribute controls the size, in bytes, of the cache that stores normalized distinguished names (NDNs). Increasing this value will retain more frequently used DNs in memory.

Table 7.6. nsslapd-ndn-cache-max-size attribute values

Default value	**20971520** *(20 MB)*
Valid range	**0 - 2147483647**
Entry DN location	**cn=config**

Prerequisites

- The LDAP Directory Manager password

Procedure

1. Ensure the NDN cache is enabled.

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://<server_fqdn> config get
nsslapd-ndn-cache-enabled
Enter password for cn=Directory Manager on ldap://server.example.com:
nsslapd-ndn-cache-enabled: on
```

   If the cache is **off**, enable it with the following command.

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://<server_fqdn> config
replace nsslapd-ndn-cache-enabled=on
Enter password for cn=Directory Manager on ldap://server.example.com:
Successfully replaced "nsslapd-ndn-cache-enabled"
```

2. Retrieve the current value of the **nsslapd-ndn-cache-max-size** parameter and make a note of it before making any adjustments, in case it needs to be restored. Enter the Directory Manager password when prompted.

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://<server_fqdn> config get
nsslapd-ndn-cache-max-size
```

```
Enter password for cn=Directory Manager on ldap://server.example.com:
nsslapd-ndn-cache-max-size: 20971520
```

3. Modify the value of the **nsslapd-ndn-cache-max-size** attribute. This example increases the value to **41943040** (40 MB).

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://<server_fqdn> config
replace nsslapd-ndn-cache-max-size=41943040
```

4. Monitor the IdM Directory Server's performance. If it does not improve, repeat this procedure and adjust **nsslapd-ndn-cache-max-size** to a different value, or re-enable cache auto-sizing.

Verification

- Display the new value of the **nsslapd-ndn-cache-max-size** attribute and verify it has been set to your desired value.

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://<server_fqdn> config get
nsslapd-ndn-cache-max-size
Enter password for cn=Directory Manager on ldap://server.example.com:
nsslapd-ndn-cache-max-size: 41943040
```

Additional resources

- nsslapd-ndn-cache-max-size in Directory Server 11 documentation

7.6. ADJUSTING THE MAXIMUM MESSAGE SIZE IN THE IDM DIRECTORY SERVER

The **nsslapd-maxbersize** attribute sets the maximum size in bytes allowed for an incoming message or LDAP request. Limiting the size of requests prevents some kinds of denial of service attacks.

If the maximum message size is too small, you might see the following error in the Directory Server error logs at **/var/log/dirsrv/slapd-*<instance_name>*/errors**:

```
Incoming BER Element was too long, max allowable is 2097152 bytes. Change the nsslapd-
maxbersize attribute in cn=config to increase.
```

The limit applies to the total size of the LDAP request. For example, if the request is to add an entry and if the entry in the request is larger than the configured value or the default, then the add request is denied. However, the limit is not applied to replication processes. Be cautious before changing this attribute.

Table 7.7. nsslapd-maxbersize attribute values

Default value	**2097152** *(2 MB)*
Valid range	**0 - 2147483647** *(0 to 2 GB)*
Entry DN location	**cn=config**

Prerequisites

- The LDAP Directory Manager password

Procedure

1. Retrieve the current value of the **nsslapd-maxbersize** parameter and make a note of it before making any adjustments, in case it needs to be restored. Enter the Directory Manager password when prompted.

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://<server_fqdn> config get
nsslapd-maxbersize
Enter password for cn=Directory Manager on ldap://server.example.com:
nsslapd-maxbersize: 2097152
```

2. Modify the value of the **nsslapd-maxbersize** attribute. This example increases the value to **4194304**, 4 MB.

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://<server_fqdn> config
replace nsslapd-maxbersize=4194304
```

3. Authenticate as the Directory Manager to make the configuration change.

```
Enter password for cn=Directory Manager on ldap://server.example.com:
Successfully replaced "nsslapd-maxbersize"
```

4. Monitor the IdM Directory Server's performance. If it does not improve, repeat this procedure and adjust **nsslapd-maxbersize** to a different value, or back to the default of **2097152**.

Verification

- Display the value of the **nsslapd-maxbersize** attribute and verify it has been set to your desired value.

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://<server_fqdn> config get
nsslapd-maxbersize
Enter password for cn=Directory Manager on ldap://server.example.com:
nsslapd-maxbersize: 4194304
```

Additional resources

- nsslapd-maxbersize (Maximum Message Size) in Directory Server 11 documentation

7.7. ADJUSTING THE MAXIMUM NUMBER OF FILE DESCRIPTORS IN THE IDM DIRECTORY SERVER

A value can be defined for the **DefaultLimitNOFILE** parameter in the **/etc/systemd/system.conf** file. An administrator with **root** privileges can set the **DefaultLimitNOFILE** parameter for the **ns-slapd** process to a lower value by using the **setrlimit** command. This value then takes precedence over what is in **/etc/systemd/system.conf** and is accepted by the Identity Management (IdM) Directory Server (DS) as the value for the **nsslapd-maxdescriptors** attribute.

The **nsslapd-maxdescriptors** attribute sets the maximum, platform-dependent number of file descriptors that the IdM LDAP uses. File descriptors are used for client connections, log files, sockets, and other resources.

If no value is defined in either **/etc/systemd/system.conf** or by **setrlimit**, then IdM DS sets the **nsslapd-maxdescriptors** attribute to 1048576.

If an IdM DS administrator later decides to set a new value for **nsslapd-maxdescriptors** manually, then IdM DS compares the new value with what is defined locally, by **setrlimit** or in **/etc/systemd/system.conf**, with the following result:

- If the new value for **nsslapd-maxdescriptors** is higher than what is defined locally, then the server rejects the new value setting and continues to enforce the local limit value as the high watermark value.
- If the new value is lower than what is defined locally, then the new value will be used.

This procedure describes how to set a new value for **nsslapd-maxdescriptors**.

Prerequisites

- The LDAP Directory Manager password

Procedure

1. Retrieve the current value of the **nsslapd-maxdescriptors** parameter and make a note of it before making any adjustments, in case it needs to be restored. Enter the Directory Manager password when prompted.

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://<server_fqdn> config get nsslapd-maxdescriptors
Enter password for cn=Directory Manager on ldap://server.example.com:
nsslapd-maxdescriptors: 4096
```

2. Modify the value of the **nsslapd-maxdescriptors** attribute. This example increases the value to **8192**.

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://<server_fqdn> config replace nsslapd-maxdescriptors=8192
```

3. Authenticate as the Directory Manager to make the configuration change.

```
Enter password for cn=Directory Manager on ldap://server.example.com:
Successfully replaced "nsslapd-maxdescriptors"
```

4. Monitor the IdM Directory Server's performance. If it does not improve, repeat this procedure and adjust **nsslapd-maxdescriptors** to a different value, or back to the default of **4096**.

Verification

- Display the value of the **nsslapd-maxdescriptors** attribute and verify it has been set to your desired value.

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://<server_fqdn> config get nsslapd-maxdescriptors
```

```
Enter password for cn=Directory Manager on ldap://server.example.com:
nsslapd-maxdescriptors: 8192
```

Additional resources

- nsslapd-maxdescriptors (Maximum File Descriptors) in Directory Server 12 documentation

7.8. ADJUSTING THE CONNECTION BACKLOG SIZE IN THE IDM DIRECTORY SERVER

The listen service sets the number of sockets available to receive incoming connections. The **nsslapd-listen-backlog-size** value sets the maximum length of the queue for the **sockfd** socket before refusing connections.

If your IdM environment handles a large amount of connections, consider increasing the value of **nsslapd-listen-backlog-size**.

Table 7.8. nsslapd-listen-backlog-size attribute values

Default value	**128** queue slots
Valid range	**0 - 9223372036854775807**
Entry DN location	**cn=config**

Prerequisites

- The LDAP Directory Manager password

Procedure

1. Retrieve the current value of the **nsslapd-listen-backlog-size** parameter and make a note of it before making any adjustments, in case it needs to be restored. Enter the Directory Manager password when prompted.

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://<server_fqdn> config get
nsslapd-listen-backlog-size
Enter password for cn=Directory Manager on ldap://server.example.com:
nsslapd-listen-backlog-size: 128
```

2. Modify the value of the **nsslapd-listen-backlog-size** attribute. This example increases the value to **192**.

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://<server_fqdn> config
replace nsslapd-listen-backlog-size=192
```

3. Authenticate as the Directory Manager to make the configuration change.

```
Enter password for cn=Directory Manager on ldap://server.example.com:
Successfully replaced "nsslapd-listen-backlog-size"
```

Verification

Verification

- Display the value of the **nsslapd-listen-backlog-size** attribute and verify it has been set to your desired value.

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://<server_fqdn> config get
nsslapd-listen-backlog-size
Enter password for cn=Directory Manager on ldap://server.example.com:
nsslapd-listen-backlog-size: 192
```

Additional resources

- nsslapd-listen-backlog-size) in Directory Server 11 documentation

7.9. ADJUSTING THE MAXIMUM NUMBER OF DATABASE LOCKS IN THE IDM DIRECTORY SERVER

Lock mechanisms control how many copies of Directory Server processes can run at the same time, and the **nsslapd-db-locks** parameter sets the maximum number of locks.

Increase the maximum number of locks if if you see the following error messages in the **/var/log/dirsrv/slapd-*<instance_name>*/errors** log file:

```
libdb: Lock table is out of available locks
```

Table 7.9. nsslapd-db-locks attribute values

Default value	**50000** locks
Valid range	**0 - 2147483647**
Entry DN location	**cn=bdb,cn=config,cn=ldbm database,cn=plugins,cn=config**

Prerequisites

- The LDAP Directory Manager password

Procedure

1. Retrieve the current value of the **nsslapd-db-locks** parameter and make a note of it before making any adjustments, in case it needs to be restored.

```
[root@server ~]# ldapsearch -D "cn=Directory Manager" -w
<directory_manager_password> -b "cn=bdb,cn=config,cn=ldbm
database,cn=plugins,cn=config" | grep nsslapd-db-locks
nsslapd-db-locks: 50000
```

2. Modify the value of the **locks** attribute. This example doubles the value to **100000** locks.

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://<server_fqdn> backend
config set --locks=100000
```

3. Authenticate as the Directory Manager to make the configuration change.

```
Enter password for cn=Directory Manager on ldap://server.example.com:
Successfully updated database configuration
```

4. Restart the Directory Server.

```
[root@server ~]# systemctl restart dirsrv.target
```

Verification

- Display the value of the **nsslapd-db-locks** attribute and verify it has been set to your desired value.

```
[root@server ~]# ldapsearch -D "cn=directory manager" -w
<directory_manager_password> -b "cn=bdb,cn=config,cn=ldbm
database,cn=plugins,cn=config" | grep nsslapd-db-locks
nsslapd-db-locks: 100000
```

Additional resources

- nsslapd-db-locks in Directory Server 11 documentation

7.10. DISABLING THE TRANSPARENT HUGE PAGES FEATURE IN THE IDM DIRECTORY SERVER

Transparent Huge Pages (THP) Linux memory management feature is enabled by default on RHEL. The THP feature can decrease the IdM Directory Server (DS) performance because DS has sparse memory access patterns.

How to disable the feature, see Disabling the Transparent Huge Pages feature in Red Hat Directory Server documentation.

Additional resources

- The negative effects of Transparent Huge Pages (THP) on RHDS

7.11. ADJUSTING THE INPUT/OUTPUT BLOCK TIMEOUT IN THE IDM DIRECTORY SERVER

The **nsslapd-ioblocktimeout** attribute sets the amount of time in milliseconds after which the connection to a stalled LDAP client is closed. An LDAP client is considered to be stalled when it has not made any I/O progress for read or write operations.

Lower the value of the **nsslapd-ioblocktimeout** attribute to free up connections sooner.

Table 7.10. nsslapd-ioblocktimeout attribute values

Default value	**10000** milliseconds
Valid range	**0 - 2147483647**

Entry DN location	**cn=config**

Prerequisites

- The LDAP Directory Manager password

Procedure

1. Retrieve the current value of the **nsslapd-ioblocktimeout** parameter and make a note of it before making any adjustments, in case it needs to be restored. Enter the Directory Manager password when prompted.

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://<server_fqdn> config get nsslapd-ioblocktimeout
Enter password for cn=Directory Manager on ldap://server.example.com:
nsslapd-ioblocktimeout: 10000
```

2. Modify the value of the **nsslapd-ioblocktimeout** attribute. This example lowers the value to **8000**.

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://<server_fqdn> config replace nsslapd-ioblocktimeout=8000
```

3. Authenticate as the Directory Manager to make the configuration change.

```
Enter password for cn=Directory Manager on ldap://server.example.com:
Successfully replaced "nsslapd-ioblocktimeout"
```

4. Monitor the IdM directory server's performance. If it does not improve, repeat this procedure and adjust **nsslapd-ioblocktimeout** to a different value, or back to the default of **10000**.

Verification

- Display the value of the **nsslapd-ioblocktimeout** attribute and verify it has been set to your desired value.

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://<server_fqdn> config get nsslapd-ioblocktimeout
Enter password for cn=Directory Manager on ldap://server.example.com:
nsslapd-idletimeout: 8000
```

Additional resources

- nsslapd-ioblocktimeout (IO Block Time Out) in Directory Server 11 documentation

7.12. ADJUSTING THE IDLE CONNECTION TIMEOUT IN THE IDM DIRECTORY SERVER

The **nsslapd-idletimeout** attribute sets the amount of time in seconds after which an idle LDAP client connection is closed by the IdM server. A value of **0** means that the server never closes idle connections.

Red Hat recommends adjusting this value so stale connections are closed, but active connections are not closed prematurely.

Table 7.11. nsslapd-idletimeout attribute values

Default value	**3600** seconds *(1 hour)*
Valid range	**0 - 2147483647**
Entry DN location	**cn=config**

Prerequisites

- The LDAP Directory Manager password

Procedure

1. Retrieve the current value of the **nsslapd-idletimeout** parameter and make a note of it before making any adjustments, in case it needs to be restored. Enter the Directory Manager password when prompted.

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://<server_fqdn> config get
nsslapd-idletimeout
Enter password for cn=Directory Manager on ldap://server.example.com:
nsslapd-idletimeout: 3600
```

2. Modify the value of the **nsslapd-idletimeout** attribute. This example lowers the value to **1800** (30 minutes).

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://<server_fqdn> config
replace nsslapd-idletimeout=1800
```

3. Authenticate as the Directory Manager to make the configuration change.

```
Enter password for cn=Directory Manager on ldap://server.example.com:
Successfully replaced "nsslapd-idletimeout"
```

4. Monitor the IdM directory server's performance. If it does not improve, repeat this procedure and adjust **nsslapd-idletimeout** to a different value, or back to the default of **3600**.

Verification

- Display the value of the **nsslapd-idletimeout** attribute and verify it has been set to your desired value.

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://<server_fqdn> config get
nsslapd-idletimeout
Enter password for cn=Directory Manager on ldap://server.example.com:
nsslapd-idletimeout: 3600
```

Additional resources

- nsslapd-idletimeout (Default Idle Timeout) in Directory Server 11 documentation

7.13. ADJUSTING THE REPLICATION RELEASE TIMEOUT

An IdM replica is exclusively locked during a replication session with another replica. In some environments, a replica is locked for a long time due to large updates or network congestion, which increases replication latency.

You can release a replica after a fixed amount of time by adjusting the **repl-release-timeout** parameter. Red Hat recommends setting this value between **30** and **120**:

- If the value is set too low, replicas are constantly reacquiring one another and replicas are not able to send larger updates.
- A longer timeout can improve high-traffic situations where it is best if a server exclusively accesses a replica for longer amounts of time, but a value higher than **120** seconds slows down replication.

Table 7.12. repl-release-timeout attribute values

Default value	**60** seconds
Valid range	**0 - 2147483647**
Recommended range	**30 - 120**

Prerequisites

- The LDAP Directory Manager password

Procedure

1. Display the database suffixes and their corresponding back ends.

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://<server_fqdn> backend suffix list
cn=changelog (changelog)
dc=example,dc=com (userroot)
o=ipaca (ipaca)
```

   This command displays the names of the back end databases next to their suffix. Use the suffix name in the next step.

2. Modify the value of the **repl-release-timeout** attribute for the main userroot database. This example increases the value to **90** seconds.

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://<server_fqdn> replication set --suffix="dc=example,dc=com" --repl-release-timeout=90
```

3. Authenticate as the Directory Manager to make the configuration change.

```
Enter password for cn=Directory Manager on ldap://server.example.com:
Successfully replaced "repl-release-timeout"
```

4. Optional: If your IdM environment uses the IdM Certificate Authority (CA), you can modify the value of the **repl-release-timeout** attribute for the CA database. This example increases the value to **90** seconds.

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://<server_fqdn> replication
set *--suffix="o=ipaca" --repl-release-timeout=90*
Enter password for cn=Directory Manager on ldap://server.example.com:
Successfully replaced "repl-release-timeout"
```

5. Restart the Directory Server.

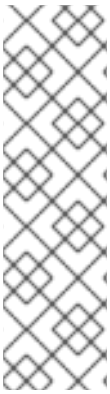
```
[root@server ~]# systemctl restart dirsrv.target
```

6. Monitor the IdM Directory Server's performance. If it does not improve, repeat this procedure and adjust **repl-release-timeout** to a different value, or back to the default of **60** seconds.

Verification

- Display the value of the **nsds5ReplicaReleaseTimeout** attribute and verify it has been set to your desired value.

```
[root@server ~]# ldapsearch -D "cn=Directory Manager" -w
<directory_manager_password> -b
"cn=replica,cn=dc\3Dexample\2Cdc\3Dcom,cn=mapping tree,cn=config" | grep
nsds5ReplicaReleaseTimeout
nsds5ReplicaReleaseTimeout: 90
```



NOTE

The Distinguished Name of the suffix in this example is **dc=example,dc=com**, but the equals sign (**=**) and comma (**,**) must be escaped in the **ldapsearch** command.

Convert the suffix DN to **cn=dc\3Dexample\2Cdc\3Dcom** with the following escape characters:

- **\3D** replacing **=**
- **\2C** replacing **,**

Additional resources

- nsDS5ReplicaReleaseTimeout in Directory Server 11 documentation

7.14. 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

7.15. ADDITIONAL RESOURCES

- Directory Server 11 Performance Tuning Guide

CHAPTER 8. ADJUSTING THE PERFORMANCE OF THE KDC

To optimize the performance of Kerberos Key Distribution Center (KDC), which is responsible for authenticating users, hosts, and services, adjust key parameters based on your deployment's traffic patterns.

8.1. ADJUSTING THE LENGTH OF THE KDC LISTEN QUEUE

You can adjust the size of the listen queue length for the KDC daemon by setting the **kdc_tcp_listen_backlog** option in the **[kdcdefaults]** section of the **/var/kerberos/krb5kdc/kdc.conf** file. The default value of **5** may be too low for some IdM deployments that experience high amounts of Kerberos traffic, but setting this value too high degrades performance.

Default value	**5**
Valid range	**1 - 10**

Procedure

1. Open the **/var/kerberos/krb5kdc/kdc.conf** file in a text editor.
2. Set the TCP listen backlog to your desired value, such as **7**.

```
[kdcdefaults]
 ...
 kdc_tcp_listen_backlog = 7
```

3. Save and close the **/var/kerberos/krb5kdc/kdc.conf** file.
4. Restart the KDC to load the new settings.

8.2. OPTIONS CONTROLLING KDC BEHAVIOR PER REALM

To track locking and unlocking user accounts for each Kerberos realm, the KDC writes to its database after each successful and failed authentication. By adjusting the following options in the **[dbmodules]** section of the **/etc/krb5.conf** file, you may be able to improve performance by minimizing how often the KDC writes information.

disable_last_success

If set to **true**, this option suppresses KDC updates to the **Last successful authentication** field of principal entries requiring preauthentication.

Default value	**false**
Valid range	**true** or **false**

disable_lockout

If set to **true**, this option suppresses KDC updates to the **Last failed authentication** and **Failed password attempts** fields of principal entries requiring preauthentication. Setting this flag may improve performance, but disabling account lockout may be considered a security risk.

Default value	**false**
Valid range	**true** or **false**

Additional resources

- Adjusting KDC settings per realm

8.3. ADJUSTING KDC SETTINGS PER REALM

Adjust KDC settings for a specific Kerberos realm by modifying the **/etc/krb5.conf** file.

Procedure

1. Open the **/etc/krb5.conf** file in a text editor.
2. Specify any options and their desired values within the **[dbmodules]** section, and in the respective Kerberos realm. In this example, you are setting the **disable_last_success** variable for the ***<kerberos_realm>***, for example EXAMPLE.COM.

```
[dbmodules]
    <kerberos_realm> = {
        disable_last_success = true
    }
```

3. Save and close the **/etc/krb5.conf** file.
4. Restart the KDC to load the new settings.

Additional resources

- Options controlling KDC behavior per realm

8.4. ADJUSTING THE NUMBER OF KRB5KDC PROCESSES

You can manually adjust the number of processes that the Key Distribution Center (KDC) starts to handle incoming connections.

By default, the IdM installer detects the number of CPU cores and enters the value in the **/etc/sysconfig/krb5kdc** file. For example, the file might contain the following entry:

```
KRB5KDC_ARGS='-w 2'
[...]
```

In this example, with the **KRB5KDC_ARGS** parameter set to **-w 2**, the KDC starts two separate processes to handle incoming connections from the main process. You might want to adjust this value, especially in virtual environments where you can easily add or remove the number of virtual CPUs based on your requirements. To prevent performance issues or even IdM servers becoming unresponsive due to an ever-increasing TCP/IP queue on port 88, simulate a higher number of processes by manually setting the **KRB5KDC_ARGS** parameter to a higher value.

Procedure

1. Open the **/etc/sysconfig/krb5kdc** file in a text editor.
2. Specify the value of the **KRB5KDC_ARGS** parameter. In this example, you are setting the number of processes to 10:

```
KRB5KDC_ARGS='-w 10'
[...]
```

3. Save and close the **/etc/sysconfig/krb5kdc** file.
4. Reload the systemd configuration:

```
# systemctl daemon-reload
```

5. Restart the **krb5kdc** service:

```
# systemctl restart krb5kdc.service
```

NOTE

You can use the IdM Healthcheck utility to verify that the KDC is configured to use the optimal number of worker processes. See Verifying the optimal number of KDC worker processes using IdM Healthcheck.

8.5. ADDITIONAL RESOURCES

- MIT Kerberos Documentation - kdc.conf .

CHAPTER 9. TUNING SSSD PERFORMANCE FOR LARGE IDM-AD TRUST DEPLOYMENTS

Retrieving user and group information is a very data-intensive operation for the System Security Services Daemon (SSSD), especially in an IdM deployment with a trust to a large Active Directory (AD) domain. You can improve this performance by adjusting which information SSSD retrieves from identity providers and for how long.

9.1. TUNING SSSD IN IDM SERVERS FOR LARGE IDM-AD TRUST DEPLOYMENTS

Apply tuning options to the configuration of the SSSD service in an IdM server to improve its response time when retrieving information from a large AD environment.

Prerequisites

- You need **root** permissions to edit the **/etc/sssd/sssd.conf** configuration file.

Procedure

1. Open the **/etc/sssd/sssd.conf** configuration file in a text editor.
2. Add the following options to the **[domain]** section for your Identity Management (IdM) domain:

```
[domain/<idm_domain_name>]
ignore_group_members = true
subdomain_inherit = ignore_group_members
...
```

NOTE

Settings listed in the **subdomain_inherit** options apply to both the main (IdM) domain as well as the trusted AD domain(s).

3. Save and close the **/etc/sssd/sssd.conf** file on the server.
4. Restart the SSSD service to load the configuration changes.

```
[root@client ~]# systemctl restart sssd
```

Additional resources

- Options for tuning SSSD in IdM servers and clients for large IdM-AD trust deployments

9.2. TUNING THE CONFIG TIMEOUT FOR THE IPA-EXTDOM PLUGIN ON IDM SERVERS

IdM clients cannot receive information about users and groups from Active Directory (AD) directly, so IdM servers use the **ipa-extdom** plugin to receive information about AD users and groups, and that information is forwarded to the requesting client.

The **ipa-extdom** plug-in sends a request to SSSD for the data about AD users. If the information is not in the SSSD cache, SSSD requests the data from an AD domain controller (DC). You can adjust the config timeout value, which defines how long the **ipa-extdom** plug-in waits for a reply from SSSD before the plug-in cancels the connection and returns a timeout error to the caller. The default value is 10000 milliseconds (10 seconds).

The following example adjusts the config timeout to 20 seconds (20000 milliseconds).

WARNING

Exercise caution when adjusting the config timeout:

- If you set a value that is too small, such as 500 milliseconds, SSSD might not have enough time to reply and requests will always return a timeout.
- If you set a value that is too large, such as 30000 milliseconds (30 seconds), a single request might block the connection to SSSD for this amount of time. Because only one thread can connect to SSSD at a time, all other requests from the plug-in have to wait.
- If there are many requests sent by IdM clients, they can block all available workers configured for the Directory Server on the IdM server. As a consequence, the server might not be able to reply to any kind of request for some time.

Only change the config timeout in the following situations:

- If IdM clients frequently receive timeout errors before their own search timeout is reached when requesting information about AD users and groups, the config timeout value is **too small**.
- If the Directory Server on the IdM server is often locked and the **pstack** utility reports that many or all worker threads are handling **ipa-extdom** requests at this time, the value is **too large**.

Prerequisites

- The LDAP Directory Manager password

Procedure

- Use the following command to adjust the config timeout to 20000 milliseconds:

```
# ldapmodify -D "cn=Directory Manager" -W
dn: cn=ipa_extdom_extop,cn=plugins,cn=config
changetype: modify
replace: ipaExtdomMaxNssTimeout
ipaExtdomMaxNssTimeout: 20000
```

9.3. TUNING THE MAXIMUM BUFFER SIZE FOR THE IPA-EXTDOM PLUGIN ON IDM SERVERS

IdM clients cannot receive information about users and groups from Active Directory (AD) directly, so IdM servers use the **ipa-extdom** plugin to receive information about AD users and groups, and that information is forwarded to the requesting client.

You can tune the maximum buffer size for the **ipa-extdom** plugin, which adjusts the size of the buffer where SSSD can store the data it receives. If the buffer is too small, SSSD returns an **ERANGE** error and the plug-in retries the request with a larger buffer. The default buffer size is 134217728 bytes (128 MB).

The following example adjusts the maximum buffer size to 256 MB (268435456 bytes).

Prerequisites

- The LDAP Directory Manager password

Procedure

- Use the following command to set the maximum buffer size to 268435456 bytes:

```
# ldapmodify -D "cn=Directory Manager" -W
dn: cn=ipa_extdom_extop,cn=plugins,cn=config
changetype: modify
replace: ipaExtdomMaxNssBufSize
ipaExtdomMaxNssBufSize: 268435456
```

9.4. TUNING THE MAXIMUM NUMBER OF INSTANCES FOR THE IPA-EXTDOM PLUGIN ON IDM SERVERS

As IdM clients cannot receive information about users and groups from Active Directory (AD) directly, IdM servers use the **ipa-extdom** plugin to receive information about AD users and groups and then forward this information to the requesting client.

By default, the **ipa-extdom** plugin is configured to use up to 80% of the LDAP worker threads to handle requests from IdM clients. If the SSSD service on an IdM client has requested a large amount of information about AD trust users and groups, this operation can halt the LDAP service if it uses most of the LDAP threads. If you experience these issues, you might see similar errors in the SSSD log file for your AD domain, **/var/log/sssd/sssd__<your-ad-domain-name.com>_.log**:

```
(2022-05-22  5:00:13): [be[ad.example.com]] [ipa_s2n_get_user_done] (0x0040): s2n exop request failed.
(2022-05-22  5:00:13): [be[ad.example.com]] [ipa_s2n_get_user_done] (0x0040): s2n exop request failed.
(2022-05-22  5:00:13): [be[ad.example.com]] [ipa_s2n_exop_done] (0x0040): ldap_extended_operation result: Server is busy(51), Too many extdom instances running.
```

You can adjust the maximum number of **ipa-extdom** instances by setting the value for the **ipaExtdomMaxInstances** option, which must be an integer larger than 0 and less than the total number of worker threads.

Prerequisites

- The LDAP Directory Manager password

Procedure

1. Retrieve the total number of worker threads:

```
# ldapsearch -xLLLD cn=Directory\ Manager -W -b cn=config -s base nsslapd-threadnumber
Enter LDAP Password:
dn: cn=config
nsslapd-threadnumber: 16
```

   This means that the current value for **ipaExtdomMaxInstances** is 13.

2. Adjust the maximum number of instances. This example changes the value to 14:

```
# ldapmodify -D "cn=Directory Manager" -W
dn: cn=ipa_extdom_extop,cn=plugins,cn=config
changetype: modify
replace: ipaExtdomMaxInstances
ipaExtdomMaxInstances: 14
```

3. Retrieve the current value of **ipaExtdomMaxInstances**:

```
# ldapsearch -xLLLD "cn=Directory Manager" -W -b "cn=ipa_extdom_extop,cn=plugins,cn=config" |grep ipaextdommaxinstances

Enter LDAP Password:

ipaextdommaxinstances: 14
```

4. Monitor the IdM Directory Server's performance and if it does not improve, repeat this procedure and adjust the value of the **ipaExtdomMaxInstances** variable.

9.5. TUNING SSSD IN IDM CLIENTS FOR LARGE IDM-AD TRUST DEPLOYMENTS

Apply tuning options to SSSD service configuration in an IdM client to improve its response time when retrieving information from a large AD environment.

Prerequisites

- You need **root** permissions to edit the **/etc/sssd/sssd.conf** configuration file.

Procedure

1. Determine the number of seconds a single un-cached login takes.

   a. Clear the SSSD cache on the IdM client.

```
[root@client_hostname ~]# sss_cache -E
```

b. Measure login time for an AD user using the **time** command. From the IdM client, authenticate locally as an AD user by logging into the same host.

```
[root@client_hostname ~]# time ssh <ad_username>@<ad_domain>@<client_fqdn>
```

c. Type in the password as soon as possible.

```
Password:
Last login: Sat Jan 23 06:29:54 2021 from 10.0.2.15
[ad_username@ad_domain@client_fqdn ~]$
```

d. Log out as soon as possible to display elapsed time. In this example, a single un-cached login takes about **9** seconds.

```
[ad_username@ad_domain@client_fqdn /]$ exit
logout
Connection to client.example.com closed.

real 0m8.755s
user    0m0.017s
sys     0m0.013s
```

2. Open the **/etc/sssd/sssd.conf** configuration file in a text editor.

3. Add the following options to the **[domain]** section for your Active Directory domain. Set the **pam_id_timeout** and **krb5_auth_timeout** options to the number of seconds an un-cached login takes. If you do not already have a domain section for your AD domain, create one.

```
[domain/<idm_domain>/<ad_domain>]
krb5_auth_timeout = 9
ldap_deref_threshold = 0
...
```

4. Add the following option to the **[pam]** section:

```
[pam]
pam_id_timeout = 9
```

5. Save and close the **/etc/sssd/sssd.conf** file on the server.

6. Restart the SSSD service to load the configuration changes.

```
[root@client_hostname ~]# systemctl restart sssd
```

Additional resources

- Options for tuning SSSD in IdM servers and clients for large IdM-AD trust deployments

9.6. MOUNTING THE SSSD CACHE IN TMPFS

The System Security Services Daemon (SSSD) constantly writes LDAP objects to its cache. These internal SSSD transactions write data to disk, which is much slower than reading and writing from Random-Access Memory (RAM).

To improve this performance, mount the SSSD cache in RAM.

Considerations

- Cached information does not persist after a reboot if the SSSD cache is in RAM.
- It is safe to perform this change on IdM servers, as the SSSD instance on an IdM server cannot lose connectivity with the Directory Server on the same host.
- If you perform this adjustment on an IdM client and it loses connectivity to IdM servers, users will not be able to authenticate after a reboot until you reestablish connectivity.

Prerequisites

- You need **root** permissions to edit the **/etc/fstab** configuration file.

Procedure

1. Create a **tmpfs** temporary filesystem:
   a. Confirm that the SSSD user owns the **config.ldb** file:

```
# ls -al /var/lib/sss/db/config.ldb
-rw-------. 1 sssd sssd 1286144 Jun  8 16:41 /var/lib/sss/db/config.ldb
```

   b. Add the following entry to the **/etc/fstab** file as a single line:

```
tmpfs /var/lib/sss/db/ tmpfs
size=300M,mode=0700,uid=sssd,gid=sssd,rootcontext=system_u:object_r:sssd_var_lib_
t:s0 0 0
```

   This example creates a 300MB cache. Tune the **size** parameter according to your IdM and AD directory size, estimating 100 MBs per 10,000 LDAP entries.
2. Mount the new SSSD cache directory.

```
[root@host ~]# mount /var/lib/sss/db/
```

3. Restart SSSD to reflect this configuration change.

```
[root@host ~]# systemctl restart sssd
```

9.7. OPTIONS IN SSSD.CONF FOR TUNING IDM SERVERS AND CLIENTS FOR LARGE IDM-AD TRUST DEPLOYMENTS

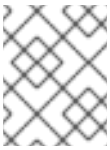
You can use the following options in the **/etc/sssd/sssd.conf** configuration file to tune the performance of SSSD in IdM servers and clients when you have a large IdM-AD trust deployment.

9.7.1. Tuning options for IdM servers

ignore_group_members

Knowing which groups a user belongs to, as opposed to all the users that belong to a group, is

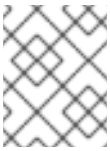
important when authenticating and authorizing a user. When **ignore_group_members** is set to **true**, SSSD only retrieves information about the group objects themselves and not their members, providing a significant performance boost.



NOTE

The **id user@ad-domain.com** command still returns the correct list of groups, but **getent group ad-group@ad-domain.com** returns an empty list.

Default value	**false**
Recommended value	**true**



NOTE

You should not set this option to **true** when the deployment involves an IdM server with the compat tree.

subdomain_inherit

With the **subdomain_inherit** option, you can apply the **ignore_group_members** setting to the trusted AD domains' configuration. Settings listed in the **subdomain_inherit** options apply to both the main (IdM) domain as well as the AD subdomain.

Default value	**none**
Recommended value	**subdomain_inherit = ignore_group_members**

9.7.2. Tuning options for IdM clients

pam_id_timeout

This parameter controls how long results from a PAM session are cached, to avoid excessive round-trips to the identity provider during an identity lookup. The default value of **5** seconds might not be enough in environments where complex group memberships are populated on the IdM Server and IdM client side. Red Hat recommends setting **pam_id_timeout** to the number of seconds a single un-cached login takes.

Default value	**5**
Recommended value	**the number of seconds a single un-cached login takes**

krb5_auth_timeout

Increasing **krb5_auth_timeout** allows more time to process complex group information in environments where users are members of a large number of groups. Red Hat recommends setting this value to the number of seconds a single un-cached login takes.

Default value	**6**
Recommended value	**the number of seconds a single un-cached login takes**

ldap_deref_threshold

A dereference lookup is a means of fetching all group members in a single LDAP call. The **ldap_deref_threshold** value specifies the number of group members that must be missing from the internal cache to trigger a dereference lookup. If less members are missing, they are looked up individually. Dereference lookups may take a long time in large environments and decrease performance. To disable dereference lookups, set this option to **0**.

Default value	**10**
Recommended value	**0**

9.8. ADDITIONAL RESOURCES

- Performance tuning SSSD for large IdM-AD trust deployments

CHAPTER 10. TUNING THE WSGI PROCESSES

If you are seeing request failures due to long-running API processes, those API processes can benefit from tuning.

By default, IPA allocates 4 Web Server Gateway Interface (WSGI) processes for the API service on 64-bit systems. This default limitation to 4 processes is implemented for memory conservation purposes. Increasing the number of the WSGI processes allows more requests to be accepted at the expense of higher CPU use and memory consumption. By default, IPA uses approximately 100 to 110MB of resident memory for the API per WSGI process. With tuning this to 16 processes, which is the recommended limit, the amount is approximately 1.3GB.

Procedure

- Modify the processes value in the **/etc/httpd/conf.d/ipa.conf** file:

```
WSGIDaemonProcess ipa processes=<4> threads=1 maximum-requests=500 \
```

Any of the longer-running API endpoints can benefit from tuning. This tuning decision is for the user to make.

For example, an OpenStack installation consists of several controllers containing multiple services. Each service requests a certificate so that all internal communication occurs over Transport Layer Security (TLS). When installing or refreshing a controller or compute node, these certificates can be requested or refreshed. In scenarios involving multiple controllers or compute nodes, the volume of certificate requests can become considerable. These requests are automated, so they happen at or nearly at the same time. Increasing the number of the WSGI threads allows the installation to complete.

10.1. OPTIMIZING CPU USAGE FOR IMPROVED IPA SERVER PERFORMANCE

When encountering performance limitations during high-volume certificate issuance tasks, tuning the CPU and Web Server Gateway Interface (WSGI) process counts can significantly enhance an IPA server's capability to handle simultaneous requests.

With a server configured with 4 CPUs and 70 clients requesting 7 certificates each (490 certificates in total), server timeouts occured as the request volume exceeded the server's processing capacity.

Increasing the CPU count to 8 and matching the WSGI process count to 8 raised the certificate handling capacity to 630 certificates, a 28% increase over the 4 CPU configuration, despite a 100% increase in CPU count. Further increasing the CPU count to 16 showed no additional performance gains with only 8 WSGI processes. However, by increasing the WSGI process count to 16, the server processed 770 certificates with 110 clients, reflecting a 22% improvement over the 8 CPU setup.

On average, doubling the number of CPUs resulted in a 25% increase in certificate issuance capacity, as long as the WSGI processes were tuned accordingly. This emphasizes the need to scale both CPU and WSGI processes together to prevent bottlenecks and optimize server performance.