# Red Hat Enterprise Linux 8

## Tuning Performance in Identity Management

Adjusting Identity Management services for better performance on Red Hat Enterprise Linux 8

Last Updated: 2021-03-25

# Red Hat Enterprise Linux 8 Tuning Performance in Identity Management

Adjusting Identity Management services for better performance on Red Hat Enterprise Linux 8

## Legal Notice



## Abstract

This documentation collection provides instructions for adjusting common performance settings in Identity Management on Red Hat Enterprise Linux 8.

# Table of Contents

# PROVIDING FEEDBACK ON RED HAT DOCUMENTATION

We appreciate your input on our documentation. Please let us know how we could make it better. To do so:

- For simple comments on specific passages:
  1. Make sure you are viewing the documentation in the *Multi-page HTML* format. In addition, ensure you see the **Feedback** button in the upper right corner of the document.
  2. Use your mouse cursor to highlight the part of text that you want to comment on.
  3. Click the **Add Feedback** pop-up that appears below the highlighted text.
  4. Follow the displayed instructions.
- For submitting more complex feedback, create a Bugzilla ticket:
  1. Go to the Bugzilla website.
  2. As the Component, use **Documentation**.
  3. Fill in the **Description** field with your suggestion for improvement. Include a link to the relevant part(s) of documentation.
  4. Click **Submit Bug**.

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

**NOTE**

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

# CHAPTER 3. 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.

## 3.1. CLIENT-SIDE FAILOVER CAPABILITY

- By default, the **SSSD** service on an IdM client is configured to use service (SRV) resource records from DNS to automatically determine the best IdM server to connect to. This behavior is controlled by the **_srv_** option in the **ipa_server** parameter of the **/etc/sssd/sssd.conf** file:

```
[root@client ~]# cat /etc/sssd/sssd.conf

[domain/example.com]
id_provider = ipa
ipa_server = _srv_, server.example.com
...
```

  If an IdM server goes offline, the SSSD service on the IdM client connects to another IdM server it has automatically discovered.

- 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:

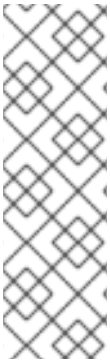
```
[root@client ~]# cat /etc/sssd/sssd.conf

[domain/example.com]
id_provider = ipa
ipa_server = server1.example.com, server2.example.com
...
```

## 3.2. 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.

# CHAPTER 4. 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.

## 4.1. DETERMINING THE APPROPRIATE NUMBER OF REPLICAS

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

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

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

One Identity Management (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 more 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.

## 4.2. CONNECTING THE REPLICAS IN A TOPOLOGY

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

Configuring additional replication agreements ensures that information is replicated not just between the initial replica and the 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.

## 4.3. REPLICA TOPOLOGY EXAMPLES

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

Figure 4.1, "Replica Topology Example 1" shows four data centers, each with four servers. The servers are connected with replication agreements.

**Figure 4.1. Replica Topology Example 1**

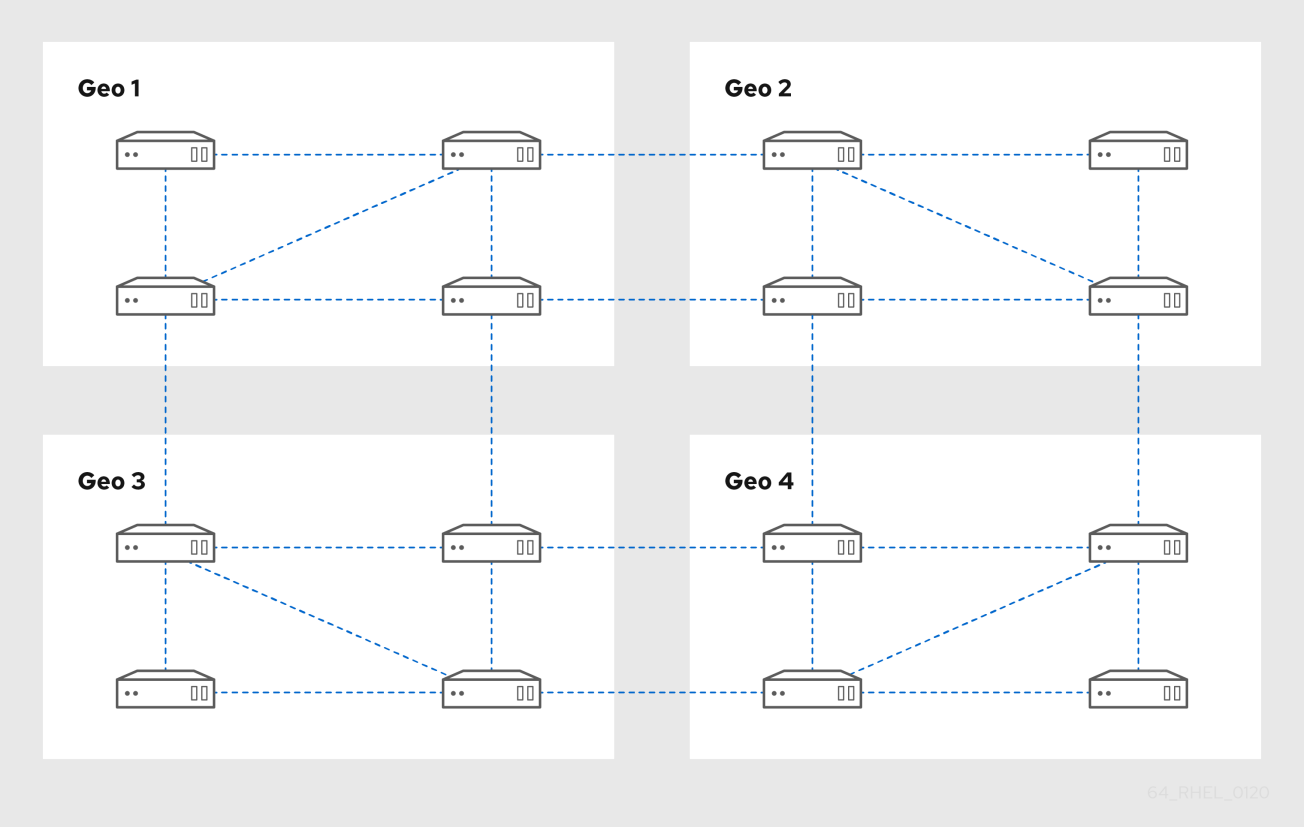
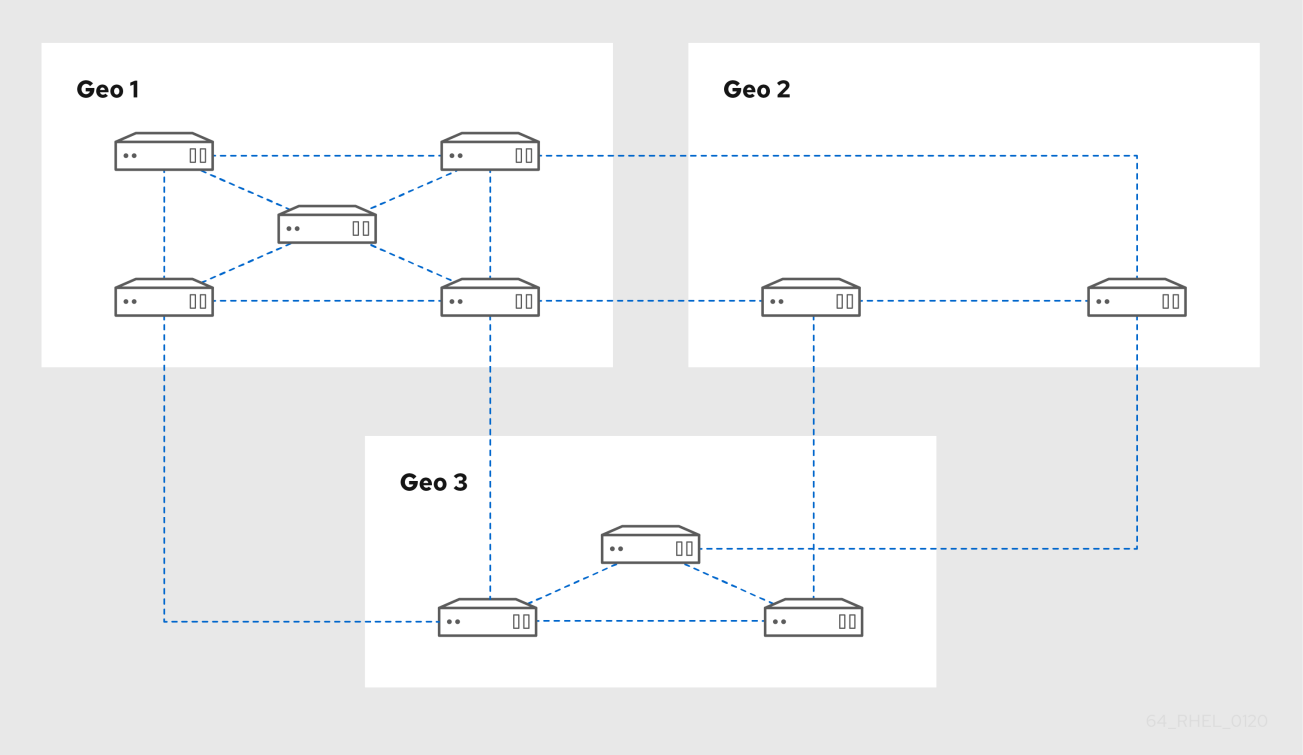



Figure 4.2, "Replica Topology Example 2" shows three data centers, each with a different number of servers. The servers are connected with replication agreements.

Figure 4.2. Replica Topology Example 2




## 4.4. ADDITIONAL RESOURCES

- For more details on establishing an effective replica topology, see Planning the replica topology.

# CHAPTER 5. 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.

## 5.1. ADJUSTING THE SEARCH SIZE AND TIME LIMIT IN THE COMMAND LINE

The following text describes adjusting search size and time limits in the command line:

- Globally
- For a specific entry

**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 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
```

## 5.2. ADJUSTING THE SEARCH SIZE AND TIME LIMIT IN THE WEB UI

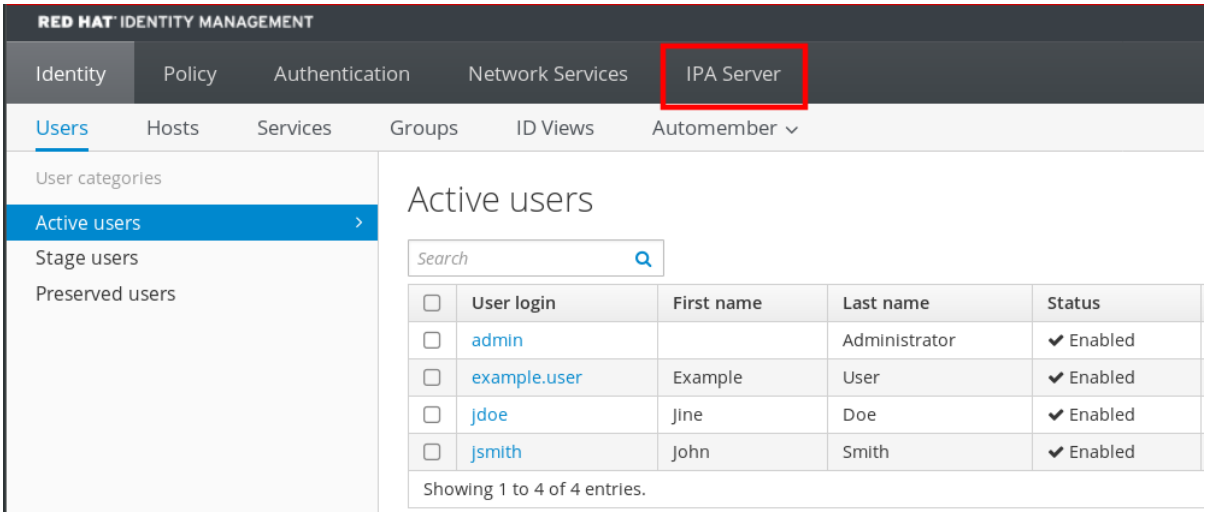
The following text describes adjusting search size and time limits in the IdM Web UI:

- Globally
- For a specific entry

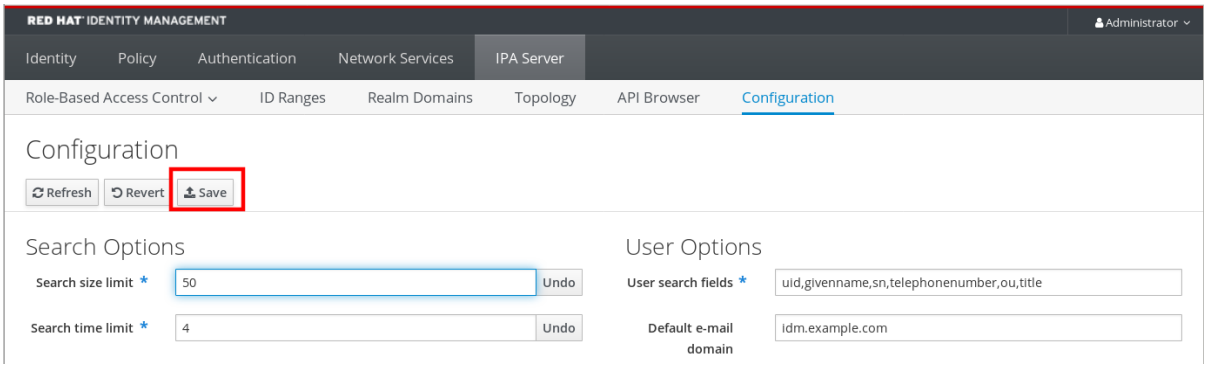
### Procedure

To adjust the limits globally for all queries:

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.




After saving the values, search an entry and verify the result.

# CHAPTER 6. 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.

To adjust how the Directory Server caches data, see the following procedures:

- Adjusting the entry cache size
- Adjusting the database index cache size
- Re-enabling entry and database cache auto-sizing
- Adjusting the DN cache size
- Adjusting the normalized DN cache size

To adjust the Directory Server's resource limits, see the following procedures:

- Adjusting the maximum message size
- Adjusting the maximum number of file descriptors
- Adjusting the connection backlog size
- Adjusting the maximum number of database locks

To adjust timeouts that have the most influence on performance, see the following procedures:

- Adjusting the input/output block timeout
- Adjusting the idle connection timeout
- Adjusting the replication release timeout

## 6.1. ADJUSTING THE ENTRY CACHE SIZE

**IMPORTANT**

Red Hat recommends using the built-in cache auto-sizing feature for optimized performance. Only change this value if you need to purposely deviate from the auto-tuned values.

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 at **/var/log/dirsrv/slapd-*INSTANCE-NAME*/errors**:

> 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.

| | |
|---|---|
| **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**

1. Disable automatic cache tuning.

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://server.example.com
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.example.com
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.example.com
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 change in a desirable way, repeat this procedure and adjust **cache-memsize** to a different value, or re-enable cache auto-sizing.

**Verification steps**

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

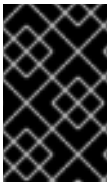
```
[root@server ~]# ldapsearch -D "cn=directory manager" -w DirectoryManagerPassword
-b "cn=userroot,cn=ldbm database,cn=plugins,cn=config" | grep nsslapd-
```

**cachememsize**
nsslapd-cachememsize: **2147483648**

### Additional resources

- For additional details about the **nsslapd-cachememsize** attribute, see nsslapd-cachememsize.
- To re-enable cache auto-sizing, see Re-enabling entry and database cache auto-sizing.

## 6.2. ADJUSTING THE DATABASE INDEX CACHE SIZE



IMPORTANT

Red Hat recommends using the built-in cache auto-sizing feature for optimized performance. Only change this value if you need to purposely deviate from the auto-tuned values.

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.

| | |
|---|---|
| **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.example.com
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 change in a desirable way, repeat this procedure and adjust **dbcachesize** to a different value, or re-enable cache auto-sizing.

### Verification steps

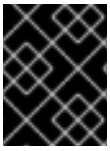
- Display the value of the **nsslapd-dbcachesize** attribute and verify it has been set to your desired value.

```
[root@server ~]# ldapsearch -D "cn=directory manager" -w DirectoryManagerPassword
-b "cn=config,cn=ldbm database,cn=plugins,cn=config" | grep nsslapd-dbcachesize
nsslapd-dbcachesize: 2147483648
```

### Additional resources

- For additional details about the **nsslapd-dbcachesize** attribute, see nsslapd-dbcachesize.
- To re-enable cache auto-sizing, see Re-enabling entry and database cache auto-sizing.

## 6.3. RE-ENABLING DATABASE AND ENTRY CACHE AUTO-SIZING



**IMPORTANT**

Red Hat recommends using the built-in cache auto-sizing feature for optimized performance. Red Hat does not recommend setting 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.

| | |
|---|---|
| **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** |

| | |
|---|---|
| **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 steps

- 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 DirectoryManagerPassword
-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

- For additional details about the **nsslapd-cache-autosize** and **nsslapd-cache-autosize-split** attributes, see Manually Re-enabling the Database and Entry Cache Auto-sizing in Directory Server 11 documentation.

## 6.4. ADJUSTING THE DN CACHE SIZE

### IMPORTANT

Red Hat recommends using the built-in cache auto-sizing feature for optimized performance. Only change this value if you need to purposely deviate from the auto-tuned values.

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.

| | |
|---|---|
| **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.example.com
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.example.com
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 change in a desirable way, repeat this procedure and adjust **dncache-memsize** to a different value, or back to the default of 10 MB.

### Verification steps

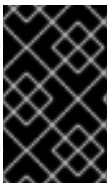
- Display the new value of the **nsslapd-dncachememsize** attribute and verify it has been set to your desired value.

```
[root@server ~]# ldapsearch -D "cn=directory manager" -w DirectoryManagerPassword
-b "cn=userroot,cn=ldbm database,cn=plugins,cn=config" | grep nsslapd-
dncachememsize
nsslapd-dncachememsize: 20971520
```

### Additional resources

- For additional details about the **nsslapd-dncachememsize** attribute, see nsslapd-dncachememsize.

## 6.5. ADJUSTING THE NORMALIZED DN CACHE SIZE



**IMPORTANT**

Red Hat recommends using the built-in cache auto-sizing feature for optimized performance. Only change this value if you need to purposely deviate from the auto-tuned values.

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.

| | |
|---|---|
| **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.example.com 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.example.com 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.example.com 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.example.com config
replace nsslapd-ndn-cache-max-size=41943040
```

4. Monitor the IdM directory server's performance. If it does not change in a desirable way, repeat this procedure and adjust **nsslapd-ndn-cache-max-size** to a different value, or re-enable cache auto-sizing.

### Verification steps

- 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.example.com 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

- For additional details about the **nsslapd-ndn-cache-max-size** attribute, see nsslapd-ndn-cache-max-size.

## 6.6. ADJUSTING THE MAXIMUM MESSAGE SIZE

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.

| | |
|---|---|
| **Default value** | **209715200** *(20 MB)* |
| **Valid range** | **0 - 2147483647** |
| **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.example.com config
get _nsslapd-maxbersize
Enter password for cn=Directory Manager on ldap://server.example.com:
nsslapd-maxdescriptors: 209715200
```

2. Modify the value of the **nsslapd-maxbersize** attribute. This example increases the value to **419430400**.

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://server.example.com config
replace nsslapd-maxbersize=419430400
```

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 change in a desirable way, repeat this procedure and adjust **nsslapd-maxbersize** to a different value, or back to the default of **209715200**.

### Verification steps

- 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.example.com config
get nsslapd-maxbersize
Enter password for cn=Directory Manager on ldap://server.example.com:
nsslapd-maxbersize: 419430400
```

### Additional resources

- For additional details about the **nsslapd-maxbersize** attribute, see nsslapd-maxbersize (Maximum Message Size).

## 6.7. ADJUSTING THE MAXIMUM NUMBER OF FILE DESCRIPTORS

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

If you set the **nsslapd-maxdescriptors** value higher than the total number of file descriptors that the operating system allows the **ns-slapd** process to use, the Directory Server queries the operating system for the maximum allowable value, and then uses that value.

| | |
|---|---|
| **Default value** | **4096** descriptors |
| **Valid range** | **1 - 65535** |
| **Entry DN location** | **cn=config** |

### 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.example.com 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.example.com 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 change in a desirable way, repeat this procedure and adjust **nsslapd-maxdescriptors** to a different value, or back to the default of **4096**.

### Verification steps

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

### Additional resources

- For additional details about the **nsslapd-maxdescriptors** attribute, see nsslapd-maxdescriptors (Maximum File Descriptors).

## 6.8. ADJUSTING THE CONNECTION BACKLOG SIZE

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**.

| | |
|---|---|
| **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.example.com 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.example.com 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 steps

- 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.example.com config
get nsslapd-listen-backlog-size
Enter password for cn=Directory Manager on ldap://server.example.com:
nsslapd-listen-backlog-size: 192
```

### Additional resources

- For additional details about the **nsslapd-listen-backlog-size** attribute, see nsslapd-listen-backlog-size).

## 6.9. ADJUSTING THE MAXIMUM NUMBER OF DATABASE LOCKS

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
```

| | |
|---|---|
| **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 DirectoryManagerPassword
-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.example.com
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 steps

- 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 DirectoryManagerPassword
-b "cn=bdb,cn=config,cn=ldbm database,cn=plugins,cn=config" | grep nsslapd-db-
locks
nsslapd-db-locks: 100000
```

### Additional resources

- For additional details about the **nsslapd-db-locks** attribute, see nsslapd-db-locks.

## 6.10. ADJUSTING THE INPUT/OUTPUT BLOCK TIMEOUT

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.

| | |
|---|---|
| **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.example.com 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.example.com 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 change in a desirable way, repeat this procedure and adjust **nsslapd-ioblocktimeout** to a different value, or back to the default of **10000**.

### Verification steps

- 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.example.com config
get nsslapd-ioblocktimeout
Enter password for cn=Directory Manager on ldap://server.example.com:
nsslapd-idletimeout: 8000
```

### Additional resources

- For additional details about the **nsslapd-ioblocktimeout** attribute, see nsslapd-ioblocktimeout (IO Block Time Out).

## 6.11. ADJUSTING THE IDLE CONNECTION TIMEOUT

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.

| | |
|---|---|
| **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.example.com 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.example.com 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 change in a desirable way, repeat this procedure and adjust **nsslapd-idletimeout** to a different value, or back to the default of **3600**.

**Verification steps**

- 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.example.com config
get nsslapd-idletimeout
Enter password for cn=Directory Manager on ldap://server.example.com:
nsslapd-idletimeout: 3600
```

**Additional resources**

- For additional details about the **nsslapd-idletimeout** attribute, see nsslapd-idletimeout (Default Idle Timeout).

## 6.12. 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.

| | |
|---|---|
| **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.example.com
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.example.com
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.example.com 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.

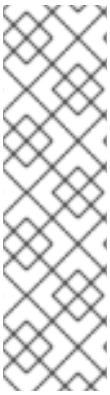
```
[root@server ~]# systemctl restart dirsrv.target
```

6. Monitor the IdM directory server's performance. If it does not change in a desirable way, repeat this procedure and adjust **repl-release-timeout** to a different value, or back to the default of **60** seconds.

### Verification steps

- 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 DirectoryManagerPassword
-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

- For additional details about the **nsds5ReplicaReleaseTimeout** attribute set by the **repl-release-timeout** option, see nsDS5ReplicaReleaseTimeout.

## 6.13. ADDITIONAL RESOURCES

- For a full list of tuning options in Directory Server 11, see the Directory Server 11 Performance Tuning Guide.

# CHAPTER 7. ADJUSTING THE PERFORMANCE OF THE KDC

The following sections describe how to adjust the performance of the Kerberos Key Distribution Center (KDC), which is responsible for authenticating users, hosts, and services.

## 7.1. OPTIONS CONTROLLING GENERAL KDC BEHAVIOR

You can adjust general KDC behavior by setting the following options in the **[kdcdefaults]** section of the **/var/kerberos/krb5kdc/kdc.conf** file.

**kdc_tcp_listen_backlog**

This option sets the size of the listen queue length for the KDC daemon. 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 will degrade performance.

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

## 7.2. ADJUSTING GENERAL KDC SETTINGS

This procedure adjusts general KDC behavior.

**Procedure**

1. Open the **/var/kerberos/krb5kdc/kdc.conf** file in a text editor.
2. Specify any options and their desired values within the **[kdcdefaults]** section. In this example, you are setting the TCP listen backlog to **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.

## 7.3. 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** |

## 7.4. ADJUSTING KDC SETTINGS PER REALM

This procedure adjusts KDC behavior per Kerberos realm.

**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 **EXAMPLE.COM** Kerberos realm.

```
[dbmodules]
    EXAMPLE.COM = {
        disable_last_success = true
    }
```

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

## 7.5. ADDITIONAL RESOURCES

- For more information on additional **kdc.conf** options, see MIT Kerberos Documentation - kdc.conf.

# CHAPTER 8. TUNING SSSD IN IDM SERVERS AND CLIENTS 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.

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

This procedure applies 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 Active Directory domain. If you do not already have a domain section for your AD domain, create one.

```
[domain/ad.example.com]
ignore_group_members = true
subdomain_inherit = ignore_group_members
...
```

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

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

This procedure applies 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 **client.example.com**.

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

   b. Measure how long it takes to log in as an AD user with the **time** command. In this example, from the IdM client **client.example.com**, log into the same host as the user **ad-user** from the **ad.example.com** AD domain.

   ```
   [root@client ~]# time ssh ad-user@ad.example.com@client.example.com
   ```

   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-user@ad.example.com@client ~]$
   ```

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

   ```
   [ad-user@ad.example.com@client /]$ 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/ad.example.com]
   pam_id_timeout = 9
   krb5_auth_timeout = 9
   ldap_deref_threshold = 0
   ...
   ```

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

5. 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

## 8.3. 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 by adding the following entry to **/etc/fstab** as a single line:

```
tmpfs /var/lib/sss/db/ tmpfs
size=300M,mode=0700,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
```

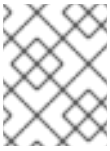
## 8.4. OPTIONS FOR TUNING SSSD IN IDM SERVERS AND CLIENTS FOR LARGE IDM-AD TRUST DEPLOYMENTS

The following options in the **/etc/sssd/sssd.conf** configuration file.

### 8.4.1. 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** |

**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** |

### 8.4.2. Tuning options for IdM Clients

**pam_id_timeout**

This parameter controls how long results from a PAM session will be cached, in order 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 in order 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** |

## 8.5. ADDITIONAL RESOURCES

- Performance tuning SSSD for large IdM-AD trust deployments .