Monitoring

This guide covers monitoring and alerts.

What to Monitor

Things to key an eye on.

HTTP

Monitor DS over HTTP.

LDAP

Monitor DS over LDAP.

Status/Tasks

About status and tasks.

Alerts

Manage alerts.

Metrics

Reference for DS metrics.

ForgeRock® Identity Platform serves as the basis for our simple and comprehensive Identity and Access Management solution. We help our customers deepen their relationships with their customers, and improve the productivity and connectivity of their employees and partners. For more information about ForgeRock and about the platform, see https://www.forgerock.com.

What to Monitor

Monitor the directory service for the following reasons:

- Noticing availability problems as they occur.

  If a server becomes unresponsive, goes offline, or crashes, you discover the problem quickly, and take corrective action.

- Identifying how client applications use the directory service.

  You can parse directory access logs to determine what client applications do. This information helps you understand what is most important, and make decisions about indexing, for example.

  Access log messages can also provide evidence of security threats, and traces of insecure client application behavior.

- Spotting performance problems, where the directory service does not meet habitual, expected, or formally defined functional, throughput, or response time characteristics.

  For example, if it suddenly becomes impossible to perform updates, the directory service has a performance problem. Alternatively, if a search that regularly completes in 500 milliseconds now takes 15 seconds, the directory service has a performance problem.

  A performance problem could also be evidence of a security threat.

Monitoring directory security is thus part of an overall monitoring strategy. Aim to answer at least the following questions when monitoring specifically for security problems:

- What insecure client behaviors do you observe?

  Examples:

  - Attempts to send simple bind credentials over insecure connections
  - Attempts to change passwords over insecure connections
  - Attempts to change configuration over insecure connections

- What unusual or unexpected usage patterns do you observe?

  Examples:

  - Search requests that perform unindexed searches
  - Requests that hit resource limits
  - Unusually large numbers of bind requests that fail
  - Unusual large numbers of password change requests that fail
  - Unusual large numbers of account lockout events

- Are you observing any sudden or hard-to-explain performance problems?

  Examples:

- Unusual increases in throughput
- Unusual increases in response times for typical requests
- Servers suddenly starved for system resources

Keep in mind when you see evidence of what looks like a security problem that it might be explained by a mistake made by an administrator or an application developer. Whether the problem is due to malice or user error, you can nevertheless use monitoring information to guide corrective actions.

HTTP-Based Monitoring

DS servers publish monitoring information at these HTTP endpoints:

`/alive`

Whether the server is currently *alive*, meaning that its internal checks have not found any errors that would require administrative action.

`/healthy`

Whether the server is currently *healthy*, meaning that it is alive and any replication delays are below a configurable threshold.

`/metrics/api`

Read-only, JSON-based view of `cn=monitor` and the monitoring backend.

Each LDAP entry maps to a resource under `/metrics/api`.

`/metrics/prometheus`

Monitoring information for Prometheus monitoring software.

For details, see Prometheus Metrics Reference.

The following example command accesses the Prometheus endpoint:

```
$ curl --cacert ca-cert.pem --user monitor:password
https://localhost:8443/metrics/prometheus
```

To give a regular user privileges to read monitoring data, see Monitor Privilege.

Server is Alive (HTTP)

The following example reads the `/alive` endpoint anonymously. If the DS server's internal tests do not find errors that require administrative action, then it returns HTTP 200 OK:

```
$ curl --cacert ca-cert.pem --head https://localhost:8443/alive

HTTP/1.1 200 OK
Content-Length: 0
Date: <date>
```

If the server finds that it is subject to errors requiring administrative action, it returns HTTP 503 Service Unavailable.

If there are errors, anonymous users receive only the 503 error status. Error strings for diagnosis are returned as an array of `"alive-errors"` in the response body, but the response body is only returned to a user with the `monitor-read` privilege.

When a server returns `"alive-errors"`, diagnose and fix the problem, and then either restart or replace the server.

Server Health (HTTP)

The following example reads the `/healthy` endpoint anonymously. If the DS server is alive, as described in Server is Alive (HTTP), and any replication delay is below the threshold configured as `max-replication-delay-health-check` (default: 5 seconds), then it returns HTTP 200 OK:

```
$ curl --cacert ca-cert.pem --head https://localhost:8443/healthy

HTTP/1.1 200 OK
Content-Length: 0
Date: <date>
```

If the server is subject to a replication delay above the threshold, then it returns HTTP 503 Service Unavailable. This result only indicates a problem if the replication delay is steadily high and increasing for the long term.

If there are errors, anonymous users receive only the 503 error status. Error strings for diagnosis are returned as an array of `"ready-errors"` in the response body, but the response body is only returned to a user with the `monitor-read` privilege.

When a server returns `"ready-errors"`, route traffic to another server until the current server is ready again.

Server Health (Prometheus)

In addition to the examples above, you can monitor whether a server is alive and able to handle requests as Prometheus metrics:

```
$ curl --cacert ca-cert.pem --user monitor:password
https://localhost:8443/metrics/prometheus 2>/dev/null | grep
health_status

# HELP ds_health_status_alive Indicates whether the server is
alive
# TYPE ds_health_status_alive gauge
ds_health_status_alive 1.0
# HELP ds_health_status_healthy Indicates whether the server is
able to handle requests
# TYPE ds_health_status_healthy gauge
ds_health_status_healthy 1.0
```

Replication Delay (Prometheus)

The following example reads a metric to check the delay in replication:

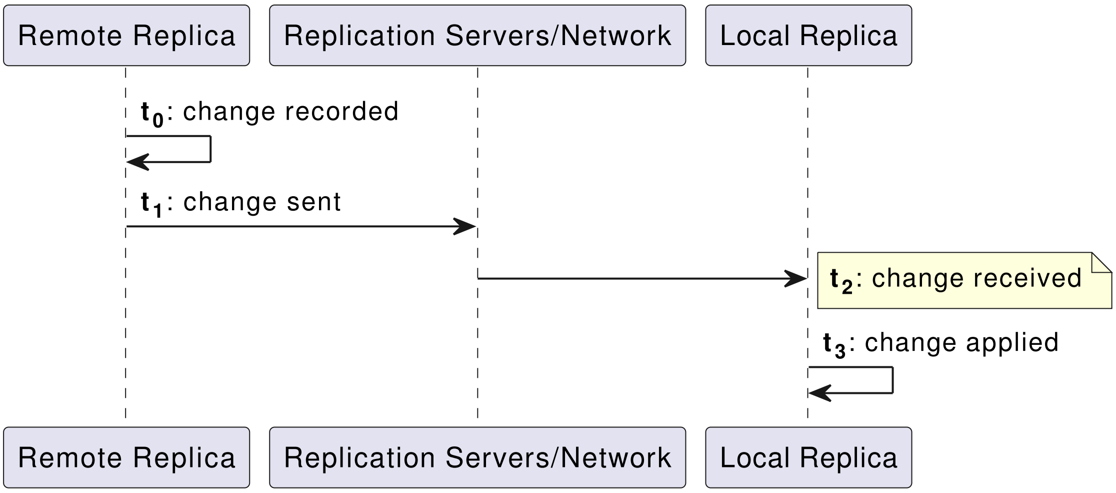
```
$ curl --cacert ca-cert.pem --user monitor:password
https://localhost:8443/metrics/prometheus 2>/dev/null | grep
receive_delay

# HELP
ds_replication_replica_remote_replicas_receive_delay_seconds
Current local delay in receiving replicated operations
# TYPE
ds_replication_replica_remote_replicas_receive_delay_seconds gauge
ds_replication_replica_remote_replicas_receive_delay_seconds{<labe
ls>} <delay>
```

DS replicas measure replication delay as the local delay when receiving and replaying changes. A replica calculates these local delays based on changes received from other replicas. Therefore, a replica can only calculate delays based on changes it has received. Network outages cause inaccuracy in delay metrics.

A replica calculates delay metrics based on times reflecting the following events:

- **t_0**: the remote replica records the change in its data
- **t_1**: the remote replica sends the change to a replica server
- **t_2**: the local replica receives the change from a replica server
- **t_3**: the local replica applies the change to its data

This figure illustrates when these events occur:



Replication keeps track of changes using change sequence numbers (CSNs), opaque and unique identifiers for each change that indicate when and where each change first occurred. The $\mathbf{t_n}$ values are CSNs.

When the CSNs for the last change received and the last change replayed are identical, the replica has applied all the changes it has received. In this case, there is no known delay. The receive and replay delay metrics are set to 0 (zero).

When the last received and last replayed CSNs differ:

- Receive delay is set to the time $\mathbf{t_2}$ - $\mathbf{t_0}$ for the last change received.

  Another name for receive delay is current delay.

- Replay delay is approximately $\mathbf{t_3}$ - $\mathbf{t_2}$ for the last change replayed. In other words, it is an approximation of how long it took the last change to be replayed.

As long as replication delay tends toward zero regularly and over the long term, temporary spikes and increases in delay measurements are normal. When all replicas remain connected and yet replication delay remains high and increases over the long term, the high replication delay indicates a problem. Steadily high and increasing replication delay shows that replication is not converging, and the service is failing to achieve eventual consistency.

For a current snapshot of replication delays, you can also use the **`dsrepl status`** command. For details, see Replication Status.

Disk Space (Prometheus)

The following example shows monitoring metrics you can use to check whether the server is running out of disk space:

```
$ curl --cacert ca-cert.pem --user monitor:password
https://localhost:8443/metrics/prometheus 2>/dev/null | grep disk
```

```
# HELP ds_disk_free_space_bytes The amount of free disk space (in bytes)
# TYPE ds_disk_free_space_bytes gauge
ds_disk_free_space_bytes{disk="<partition>",} <bytes>
# HELP ds_disk_free_space_full_threshold_bytes The effective full disk space threshold (in bytes)
# TYPE ds_disk_free_space_full_threshold_bytes gauge
ds_disk_free_space_full_threshold_bytes{disk="<partition>",} <bytes>
# HELP ds_disk_free_space_low_threshold_bytes The effective low disk space threshold (in bytes)
# TYPE ds_disk_free_space_low_threshold_bytes gauge
ds_disk_free_space_low_threshold_bytes{disk="<partition>",} <bytes>
```

In your monitoring software, compare free space with the disk low and disk full thresholds. For database backends, these thresholds are set using the configuration properties: disk-low-threshold and disk-full-threshold.

When you read from `cn=monitor` instead ,as described in LDAP-Based Monitoring, the relevant data are exposed on child entries of `cn=disk space monitor,cn=monitor`.

Certificate Expiration (Prometheus)

The following example shows how you can use monitoring metrics to check whether the server certificate is due to expire soon:

```
$ curl --cacert ca-cert.pem --user monitor:password https://localhost:8443/metrics/prometheus 2>/dev/null | grep cert

# HELP ds_certificates_certificate_expires_at_seconds Certificate expiration date and time
# TYPE ds_certificates_certificate_expires_at_seconds gauge
ds_certificates_certificate_expires_at_seconds{alias="ssl-key-pair",key_manager="PKCS12",} <sec_since_epoch>
```

In your monitoring software, compare the expiration date with the current date.

When you read from `cn=monitor` instead, as described in LDAP-Based Monitoring, the relevant data are exposed on child entries of `cn=certificates,cn=monitor`.

Request Statistics (Prometheus)

DS server connection handlers respond to client requests. The following example uses the default monitor user account to read statistics about client operations on each of the available connection handlers:

```
$ curl --cacert ca-cert.pem --user monitor:password
https://localhost:8443/metrics/prometheus 2>/dev/null | grep
connection_handlers
```

Work Queue (Prometheus)

DS servers have a work queue to track request processing by worker threads, and whether the server has rejected any requests due to a full queue. If enough worker threads are available, then no requests are rejected. The following example uses the default monitor user account to read statistics about the work queue:

```
$ curl --cacert ca-cert.pem --user monitor:password
https://localhost:8443/metrics/prometheus 2>/dev/null | grep
work_queue
```

To adjust the number of worker threads, see the settings for Traditional Work Queue.

Database Size (Prometheus)

DS servers maintain counts of the number of entries in each backend. The following example uses the default monitor user account to read the counts:

```
$ curl --cacert ca-cert.pem --user monitor:password
https://localhost:8443/metrics/prometheus 2>/dev/null | grep
backend_entry_count
```

Active Users (Prometheus)

DS server connection handlers respond to client requests. The following example uses the default monitor user account to read active connections on each connection handler:

```
$ curl --cacert ca-cert.pem --user monitor:password
https://localhost:8443/metrics/prometheus 2>/dev/null | grep
"active_[cp]"
```

Filtering results (Prometheus)

By default, DS servers return all Prometheus metrics. To limit what the server returns, set one of these HTTP endpoint properties for the `/metrics/prometheus`:

- `excluded-metric-pattern`
- `included-metric-pattern`

Set these properties to valid Java regular expression patterns.

The following configuration change causes the server to return only metrics whose names contain `connection`:

```
$ dsconfig \
 set-http-endpoint-prop \
 --endpoint-name /metrics/prometheus \
 --set included-metric-pattern:'.*connection.*' \
 --hostname localhost \
 --port 4444 \
 --bindDN uid=admin \
 --bindPassword password \
 --usePkcs12TrustStore /path/to/opendj/config/keystore \
 --trustStorePassword:file /path/to/opendj/config/keystore.pin \
 --no-prompt
```

The following configuration change causes the server to exclude metrics whose names start with `ds_jvm_`. As mentioned in the reference documentation, "The metric name prefix must not be included in the filter." Notice that the example uses the regular expression `jvm_.*`:

```
$ dsconfig \
 set-http-endpoint-prop \
 --endpoint-name /metrics/prometheus \
 --set excluded-metric-pattern:'jvm_.*' \
 --hostname localhost \
 --port 4444 \
 --bindDN uid=admin \
 --bindPassword password \
 --usePkcs12TrustStore /path/to/opendj/config/keystore \
 --trustStorePassword:file /path/to/opendj/config/keystore.pin \
 --no-prompt
```

LDAP-Based Monitoring

DS servers publish whether the server is alive and able to handle requests in the root DSE. They publish monitoring information over LDAP under the entry `cn=monitor` .

The following example reads all available monitoring entries:

```
$ ldapsearch \
 --hostname localhost \
 --port 1636 \
 --useSsl \
 --usePkcs12TrustStore /path/to/opendj/config/keystore \
 --trustStorePassword:file /path/to/opendj/config/keystore.pin \
 --bindDN uid=monitor \
 --bindPassword password \
 --baseDN cn=monitor \
 "(&)"
```

The monitoring entries under `cn=monitor` reflect activity since the server started.

Many different types of metrics are exposed. For details, see LDAP Metrics Reference.

Monitor Privilege

The following example assigns the required privilege to Kirsten Vaughan’s entry to read monitoring data, and shows monitoring information for the backend holding Example.com data:

```
$ ldapmodify \
 --hostname localhost \
 --port 1636 \
 --useSsl \
 --usePkcs12TrustStore /path/to/opendj/config/keystore \
 --trustStorePassword:file /path/to/opendj/config/keystore.pin \
 --bindDN uid=admin \
 --bindPassword password << EOF
dn: uid=kvaughan,ou=People,dc=example,dc=com
changetype: modify
add: ds-privilege-name
ds-privilege-name: monitor-read
EOF

$ ldapsearch \
 --hostname localhost \
 --port 1636 \
 --useSsl \
 --usePkcs12TrustStore /path/to/opendj/config/keystore \
 --trustStorePassword:file /path/to/opendj/config/keystore.pin \
 --bindDN uid=kvaughan,ou=People,dc=example,dc=com \
 --bindPassword bribery \
 --baseDN cn=monitor \
 "(ds-cfg-backend-id=dsEvaluation)"

dn: ds-cfg-backend-id=dsEvaluation,cn=backends,cn=monitor
ds-mon-backend-is-private: false
ds-mon-backend-entry-count: <count>
ds-mon-backend-writability-mode: enabled
ds-mon-backend-degraded-index-count: <count>
ds-mon-backend-ttl-is-running: <boolean>
ds-mon-backend-ttl-last-run-time: <timestamp>
ds-mon-backend-ttl-thread-count: <count>
ds-mon-backend-ttl-queue-size: <size>
ds-mon-backend-ttl-entries-deleted: <summary>
ds-mon-backend-filter-use-start-time: <timestamp>
ds-mon-backend-filter-use-indexed: <count>
ds-mon-backend-filter-use-unindexed: <count>
ds-mon-db-version: <version>
ds-mon-db-cache-evict-internal-nodes-count: <count>
ds-mon-db-cache-evict-leaf-nodes-count: <count>
ds-mon-db-cache-total-tries-internal-nodes: <count>
ds-mon-db-cache-total-tries-leaf-nodes: <count>
ds-mon-db-cache-misses-internal-nodes: <count>
ds-mon-db-cache-misses-leaf-nodes: <count>
ds-mon-db-cache-size-active: <size>
ds-mon-db-log-size-active: <size>
ds-mon-db-log-cleaner-file-deletion-count: <count>
ds-mon-db-log-utilization-min: <percentage>
ds-mon-db-log-utilization-max: <percentage>
ds-mon-db-log-size-total: <size>
ds-mon-db-log-files-open: <count>
ds-mon-db-log-files-opened: <count>
ds-mon-db-checkpoint-count: <count>
objectClass: top
objectClass: ds-monitor
objectClass: ds-monitor-backend
objectClass: ds-monitor-backend-pluggable
objectClass: ds-monitor-backend-db
ds-cfg-backend-id: dsEvaluation
```

Server Health (LDAP)

Anonymous clients can monitor the health status of the DS server by reading the `alive` attribute of the root DSE:

```
$ ldapsearch \
 --hostname localhost \
 --port 1636 \
 --useSsl \
 --usePkcs12TrustStore /path/to/opendj/config/keystore \
 --trustStorePassword:file /path/to/opendj/config/keystore.pin \
 --baseDN "" \
 --searchScope base \
 "(&)" \
 alive

dn:
alive: true
```

When `alive` is `true`, the server's internal tests have not found any errors requiring administrative action. When it is `false`, fix the errors and either restart or replace the server.

If the server returns `false` for this attribute, get error information, as described in Server Health Details (LDAP).

Server Health Details (LDAP)

The default monitor user can check whether the server is alive and able to handle requests on `cn=health status,cn=monitor`:

```
$ ldapsearch \
 --hostname localhost \
 --port 1636 \
 --useSsl \
 --usePkcs12TrustStore /path/to/opendj/config/keystore \
 --trustStorePassword:file /path/to/opendj/config/keystore.pin \
 --bindDN uid=monitor \
 --bindPassword password \
 --baseDN "cn=health status,cn=monitor" \
 --searchScope base \
 "(&)"

dn: cn=health status,cn=monitor
ds-mon-alive: true
ds-mon-healthy: true
```

```
objectClass: top
objectClass: ds-monitor
objectClass: ds-monitor-health-status
cn: health status
```

When the server is either not alive or not able to handle requests, this entry includes error diagnostics as strings on the `ds-mon-alive-errors` and `ds-mon-healthy-errors` attributes.

Replication Delay (LDAP)

The following example uses the default monitor user account to check the delay in replication:

```
$ ldapsearch \
 --hostname localhost \
 --port 1636 \
 --useSsl \
 --usePkcs12TrustStore /path/to/opendj/config/keystore \
 --trustStorePassword:file /path/to/opendj/config/keystore.pin \
 --bindDN uid=monitor \
 --bindPassword password \
 --baseDN cn=monitor \
 "(ds-mon-receive-delay=*)" \
 ds-mon-receive-delay

dn: ds-mon-domain-
name=dc=example\,dc=com,cn=replicas,cn=replication,cn=monitor
ds-mon-receive-delay: <delay>

dn: ds-mon-server-id=<id>,cn=remote replicas,ds-mon-domain-
name=dc=example\,dc=com,cn=replicas,cn=replication,cn=monitor
ds-mon-receive-delay: <delay>
```

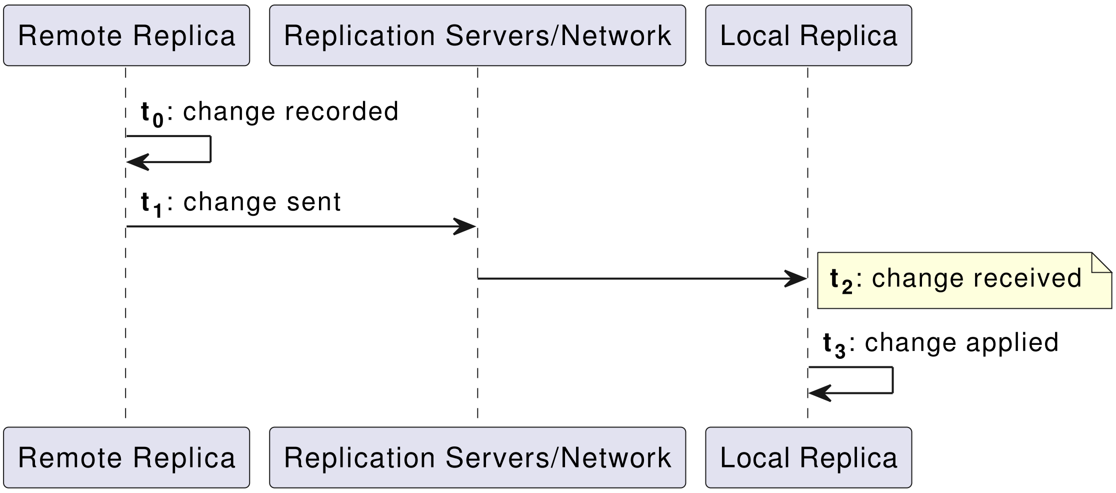
DS replicas measure replication delay as the local delay when receiving and replaying changes. A replica calculates these local delays based on changes received from other replicas. Therefore, a replica can only calculate delays based on changes it has received. Network outages cause inaccuracy in delay metrics.

A replica calculates delay metrics based on times reflecting the following events:

- **t_0**: the remote replica records the change in its data
- **t_1**: the remote replica sends the change to a replica server
- **t_2**: the local replica receives the change from a replica server

- **t_3**: the local replica applies the change to its data

This figure illustrates when these events occur:



Replication keeps track of changes using change sequence numbers (CSNs), opaque and unique identifiers for each change that indicate when and where each change first occurred. The **t_n** values are CSNs.

When the CSNs for the last change received and the last change replayed are identical, the replica has applied all the changes it has received. In this case, there is no known delay. The receive and replay delay metrics are set to 0 (zero).

When the last received and last replayed CSNs differ:

- Receive delay is set to the time **t_2** - **t_0** for the last change received.

  Another name for receive delay is current delay.

- Replay delay is approximately **t_3** - **t_2** for the last change replayed. In other words, it is an approximation of how long it took the last change to be replayed.

As long as replication delay tends toward zero regularly and over the long term, temporary spikes and increases in delay measurements are normal. When all replicas remain connected and yet replication delay remains high and increases over the long term, the high replication delay indicates a problem. Steadily high and increasing replication delay shows that replication is not converging, and the service is failing to achieve eventual consistency.

For a current snapshot of replication delays, you can also use the **`dsrepl status`** command. For details, see Replication Status.

Replication Status (LDAP)

The following example uses the default monitor user account to check the replication status of the local replica:

```
$ ldapsearch \
 --hostname localhost \
 --port 1636 \
 --useSsl \
 --usePkcs12TrustStore /path/to/opendj/config/keystore \
 --trustStorePassword:file /path/to/opendj/config/keystore.pin \
 --bindDN uid=monitor \
 --bindPassword password \
 --baseDN cn=monitor \
 "(ds-mon-status=*)" \
 ds-mon-status

dn: ds-mon-domain-name=dc=example\,dc=com,cn=replicas,cn=replication,cn=monitor
ds-mon-status: Normal
```

If the status is not `Normal`, how you react depends on the value of the `ds-mon-status` attribute:

Status	Explanation	Actions to Take
`Bad generation id`	Replication is broken. Internally, DS replicas store a shorthand form of the initial state called a generation ID. The generation ID is a hash of the first 1000 entries in a backend. If the replicas' generation IDs match, the servers can replicate data without user intervention. If the replicas' generation IDs do not match for a given backend, you must manually initialize replication between them to force the same initial state on all replicas. This status arises for one of the following reasons: • The replica and the replication server have different generation IDs for the data because the replica was not initialized with the same data as its peer replicas. • The replica has fallen further behind the replication server than allowed by the replication-purge-delay. In other words, the replica is missing too many changes, and lacks the historical information required to synchronize with peer replicas. • The fractional replication configuration for this replica does not match the backend data. For	Whenever you see this status: 1. If fractional replication is configured, make sure the configuration is compatible on all peer replicas. For details, see Fractional Replication. 2. Reinitialize replication to fix the bad generation IDs. For details, see Manual Initialization.

Status	Explanation	Actions to Take
	example, you reconfigured fractional replication to include or exclude different attributes, or you configured fractional replication in an incompatible way on different peer replicas.	
`Degraded`	Unless this status is persistent, replication is operating normally. The replica has fallen further behind peer replicas than the degraded-status-threshold. By default, the threshold is 5000, meaning this state is triggered if the replica falls 5000 or more changes behind. Additionally, the number of pending changes to apply is an *approximation* calculated internally using change sequence numbers that are not necessarily sequential. This status can arise periodically during normal operation when, for example, replication absorbs a burst of updates. In a directory service that sustains 5000 updates a second, a temporary `Degraded` status can represent a one-second delay.	If the `Degraded` status persists: 1. Make sure peer replica systems are sized appropriately. If some replicas are on more powerful systems with faster I/O than others, the replicas on the smaller systems can fall behind as load increases. 2. Consider raising the `degraded-status-threshold` setting.

Status	Explanation	Actions to Take
`Full update`	Replication is operating normally. You have chosen to initialize replication over the network. The time to complete the operation depends on the network bandwidth and volume of data to synchronize.	Monitor the server output and wait for initialization to complete.
`Invalid`	This status arises for one of the following reasons: • The replica has encountered a replication protocol error. This status can arise due to faulty network communication between the replica and the replication server. • The replica has just started, and is initializing.	If this status happens during normal operation: 1. Review the replica and replication server error logs, described in About Logs, for network-related replication error messages. 2. Independently verify network communication between the replica and the replication server systems.
`Normal`	Replication is operating normally.	Nothing to do.
`Not connected`	This status arises for one of the following reasons: • The replica has just started and is not yet connected to the replication server. • The replica cannot connect to a replication server.	If this status happens during normal operation: 1. Review the replica and replication server error logs for network-related replication error messages. 2. Independently verify network communication between the replica and the replication server systems.

Request Statistics (LDAP)

DS server connection handlers respond to client requests. The following example uses the default monitor user account to read statistics about client operations on each of the available connection handlers:

```
$ ldapsearch \
 --hostname localhost \
 --port 1636 \
 --useSsl \
 --usePkcs12TrustStore /path/to/opendj/config/keystore \
 --trustStorePassword:file /path/to/opendj/config/keystore.pin \
 --bindDN uid=monitor \
 --bindPassword password \
 --baseDN "cn=connection handlers,cn=monitor" \
 "(&)"
```

For details about the content of metrics returned, see Metric Types Reference.

Work Queue (LDAP)

DS servers have a work queue to track request processing by worker threads, and whether the server has rejected any requests due to a full queue. If enough worker threads are available, then no requests are rejected. The following example uses the default monitor user account to read statistics about the work queue:

```
$ ldapsearch \
 --hostname localhost \
 --port 1636 \
 --useSsl \
 --usePkcs12TrustStore /path/to/opendj/config/keystore \
 --trustStorePassword:file /path/to/opendj/config/keystore.pin \
 --bindDN uid=monitor \
 --bindPassword password \
 --baseDN "cn=work queue,cn=monitor" \
 "(&)"
```

For details about the content of metrics returned, see Metric Types Reference. To adjust the number of worker threads, see the settings for Traditional Work Queue.

Database Size (LDAP)

DS servers maintain counts of the number of entries in each backend and under each base DN. The following example uses the default monitor user account to read the counts:

```
$ ldapsearch \
 --hostname localhost \
 --port 1636 \
 --useSsl \
 --usePkcs12TrustStore /path/to/opendj/config/keystore \
 --trustStorePassword:file /path/to/opendj/config/keystore.pin \
 --bindDN uid=monitor \
 --bindPassword password \
 --baseDN cn=monitor \
 "(|(ds-mon-backend-entry-count=*)(ds-mon-base-dn-entry-count=*))"
\
 ds-mon-backend-entry-count ds-mon-base-dn-entry-count
```

Active Users (LDAP)

DS server connection handlers respond to client requests. The following example uses the default monitor user account to read the metrics about active connections on each connection handler:

```
$ ldapsearch \
 --hostname localhost \
 --port 1636 \
 --useSsl \
 --usePkcs12TrustStore /path/to/opendj/config/keystore \
 --trustStorePassword:file /path/to/opendj/config/keystore.pin \
 --bindDN uid=monitor \
 --bindPassword password \
 --baseDN cn=monitor \
 "(objectClass=ds-monitor-connection*)" \
 ds-mon-active-connections-count ds-mon-active-persistent-searches
ds-mon-connection ds-mon-listen-address
```

For details about the content of metrics returned, see Metric Types Reference.

SNMP-Based Monitoring

NOTE

This legacy feature is deprecated.

DS servers support SNMP, including the Management Information Base described in RFC 2605: Directory Server Monitoring MIB.

SNMP is not enabled by default. SNMP-based monitoring depends on an OpenDMK library. The OpenDMK binary bundle containing this library ships with DS servers as `snmp/opendmk.jar`. Installation requires that you accept the OpenDMK Binary License. OpenDMK installation is a separate step that you must perform before you can use SNMP.

1. Run the OpenDMK installer and accept the license, use the self-extracting .jar:

   ```
   $ java -jar /path/to/opendj/snmp/opendmk.jar
   ```

2. Install OpenDMK, and then copy the libraries to the `/path/to/opendj/extlib` directory. For example, if you install OpenDMK in the `/path/to` directory, copy the libraries from the `/path/to/OpenDMK-bin/lib` directory:

   ```
   $ cp /path/to/OpenDMK-bin/lib/* /path/to/opendj/extlib/
   ```

3. Set up an SNMP connection handler:

   ```
   $ dsconfig \
    set-connection-handler-prop \
    --handler-name SNMP \
    --set enabled:true \
    --hostname localhost \
    --port 4444 \
    --bindDN uid=admin \
    --bindPassword password \
    --usePkcs12TrustStore /path/to/opendj/config/keystore \
    --trustStorePassword:file
   /path/to/opendj/config/keystore.pin \
    --no-prompt
   ```

4. If the server does not have access to the default ports, change them.

   By default, the SNMP connection handler listens on port `161`, and uses port `162` for traps. On UNIX and Linux systems, only root can normally open these ports. The following command installs as a normal user, changing the listen and trap ports:

   ```
   $ dsconfig \
    set-connection-handler-prop \
   ```

```
 --handler-name SNMP \
 --set listen-port:11161 \
 --set trap-port:11162 \
 --hostname localhost \
 --port 4444 \
 --bindDN uid=admin \
 --bindPassword password \
 --usePkcs12TrustStore /path/to/opendj/config/keystore \
 --trustStorePassword:file
/path/to/opendj/config/keystore.pin \
 --no-prompt
```

5. Restart the SNMP connection handler to take the changes into account:

```
$ dsconfig \
 set-connection-handler-prop \
 --handler-name SNMP \
 --set enabled:false \
 --hostname localhost \
 --port 4444 \
 --bindDN uid=admin \
 --bindPassword password \
 --usePkcs12TrustStore /path/to/opendj/config/keystore \
 --trustStorePassword:file
/path/to/opendj/config/keystore.pin \
 --no-prompt

$ dsconfig \
 set-connection-handler-prop \
 --handler-name SNMP \
 --set enabled:true \
 --hostname localhost \
 --port 4444 \
 --bindDN uid=admin \
 --bindPassword password \
 --usePkcs12TrustStore /path/to/opendj/config/keystore \
 --trustStorePassword:file
/path/to/opendj/config/keystore.pin \
 --no-prompt
```

6. Check that connection handler works as expected.

   The following command reads the response on the SNMP listen port:

```
$ snmpwalk -v 2c -c OpenDJ@OpenDJ localhost:11161

iso.3.6.1.2.1.66.1.1.1.1 = STRING: "ForgeRock Directory
Services version"
iso.3.6.1.2.1.66.1.1.2.1 = STRING: "/path/to/opendj" ...
```

JMX-Based Monitoring

A number of tools support Java Management Extensions (JMX), including the **`jconsole`** command bundled with the Java platform, and VisualVM. JMX is not configured by default.

Configure JMX

1. Set server Java arguments appropriately to avoid regular full garbage collection (GC) events.

   JMX is based on Java Remote Method Invocation (RMI), which uses references to objects. By default, the JMX client and server perform a full GC periodically to clean up stale references. As a result, the default settings cause JMX to cause a full GC every hour.

   To prevent hourly full GCs when using JMX, add the `-XX:+DisableExplicitGC` option to the list of `start-ds.java-args` arguments. You can do this by editing the `config/java.properties` file and restarting the server.

   Avoid using this argument when importing LDIF online using the **`import-ldif`** command. The import process uses GC to work around memory management issues.

2. Configure the server to activate JMX access.

   The following example uses the reserved port number, `1689`:

   ```
   $ dsconfig \
    create-connection-handler \
    --handler-name JMX \
    --type jmx \
    --set enabled:true \
    --set listen-port:1689 \
    --hostname localhost \
   ```

```
 --port 4444 \
 --bindDN uid=admin \
 --bindPassword password \
 --usePkcs12TrustStore /path/to/opendj/config/keystore \
 --trustStorePassword:file
/path/to/opendj/config/keystore.pin \
 --no-prompt
```

The change takes effect immediately.

Connect Over JMX

1. Add appropriate privileges to access JMX monitoring information.

   By default, no users have privileges to access the JMX connection. The following commands create a user with JMX privileges, who can authenticate over an insecure connection:

   ▼ *Show commands*

   ```
   # Create a password policy to allow the user to
   authenticate insecurely:
   $ dsconfig \
    create-password-policy \
    --policy-name "Allow insecure authentication" \
    --type password-policy \
    --set default-password-storage-scheme:PBKDF2-HMAC-SHA256
   \
    --set password-attribute:userPassword \
    --hostname localhost \
    --port 4444 \
    --bindDN uid=admin \
    --bindPassword password \
    --usePkcs12TrustStore /path/to/opendj/config/keystore \
    --trustStorePassword:file
   /path/to/opendj/config/keystore.pin \
    --no-prompt

   # Create a backend for the JMX monitor user entry:
   $ dsconfig \
   create-backend \
    --backend-name jmxMonitorUser \
    --type ldif \
    --set enabled:true \
   ```

```
 --set base-dn:"uid=JMX Monitor" \
 --set ldif-file:db/jmxMonitorUser/jmxMonitorUser.ldif \
 --set is-private-backend:true \
 --hostname localhost \
 --port 4444 \
 --bindDN uid=admin \
 --bindPassword password \
 --usePkcs12TrustStore /path/to/opendj/config/keystore \
 --trustStorePassword:file
/path/to/opendj/config/keystore.pin \
 --no-prompt

# Prepare the JMX monitor user entry.
# Notice the privileges and password policy settings:
$ cat > /tmp/jmxMonitorUser.ldif << EOF
dn: uid=JMX Monitor
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
cn: JMX Monitor
sn: User
uid: JMX Monitor
userPassword: password
ds-privilege-name: monitor-read
ds-privilege-name: jmx-notify
ds-privilege-name: jmx-read
ds-privilege-name: jmx-write
ds-pwp-password-policy-dn: cn=Allow insecure
authentication,cn=Password Policies,cn=config
EOF

# Import the JMX monitor user:
$ import-ldif \
 --backendID jmxMonitorUser \
 --includeBranch "uid=JMX Monitor" \
 --ldifFile /tmp/jmxMonitorUser.ldif \
 --hostname localhost \
 --port 4444 \
 --bindDN uid=admin \
 --bindPassword password \
 --usePkcs12TrustStore /path/to/opendj/config/keystore \
 --trustStorePassword:file
/path/to/opendj/config/keystore.pin
```

2. Connect using the service URI, username, and password:

   Service URI

   Full URI to the service including the hostname or IP address and port number for JMX where the DS server listens for connections.

   For example, if the server hostname is `localhost`, and the DS server listens for JMX connections on port `1689`, then the service URI is:

   ```
   service:jmx:rmi:///jndi/rmi://localhost:1689/org.opends.
   server.protocols.jmx.client-unknown
   ```

   Username

   The full DN of the user with privileges to connect over JMX, such as `uid=JMX Monitor`.

   Password

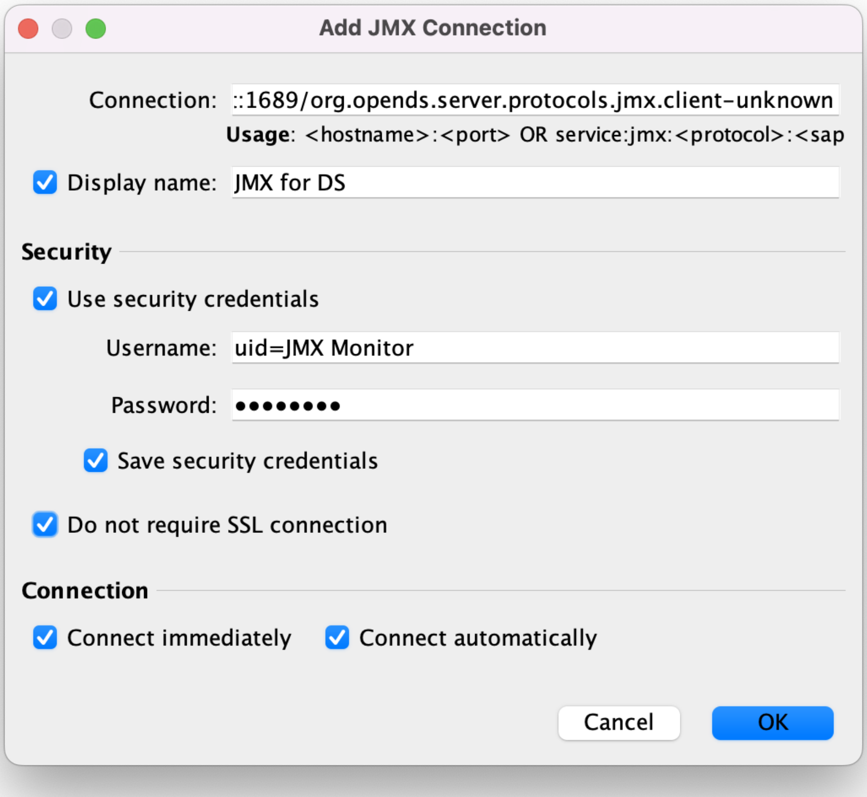
   The bind password for the user.

3. Connect remotely.

   The following steps show how you connect using VisualVM:

   a. Start VisualVM.

   b. Select **File** > **Add JMX Connection...** to configure the connection:



Add JMX Connection

Connection: ::1689/org.opends.server.protocols.jmx.client-unknown

Usage: <hostname>:<port> OR service:jmx:<protocol>:<sap

☑ Display name: JMX for DS

Security

☑ Use security credentials

Username: uid=JMX Monitor

Password: ●●●●●●●●●

☑ Save security credentials

☑ Do not require SSL connection

Connection

☑ Connect immediately ☑ Connect automatically

> c. Select the connection in the left menu to view JMX monitoring information.
>
> For additional details, see Monitoring and Management Using JMX Technology.

Status and Tasks

The **`status`** command functions in offline mode, but provides more information with the server is running. The command describes the server’s capabilities, including the ports and disks it uses, and the backends it serves. With the `--script-friendly` option, the command returns JSON output. The command requires administrative credentials to read a running server’s configuration:

```
$ status \
 --bindDn uid=admin \
 --bindPassword password \
 --hostname localhost \
 --port 4444 \
 --usePkcs12TrustStore /path/to/opendj/config/keystore \
 --trustStorePassword:file /path/to/opendj/config/keystore.pin \
 --script-friendly
```

The **`manage-tasks`** command lets you manage tasks scheduled on a server, such as regular backup. The command connects to the administration port of a local or remote server:

```
$ manage-tasks \
 --hostname localhost \
 --port 4444 \
 --bindDN uid=admin \
 --bindPassword password \
 --usePkcs12TrustStore /path/to/opendj/config/keystore \
 --trustStorePassword:file /path/to/opendj/config/keystore.pin \
 --no-prompt
```

Push to Graphite

The Graphite application stores numeric time-series data of the sort produced by monitoring metrics, and allows you to render graphs of that data.

Your applications, in this case DS servers, push data into Graphite. You do this by configuring the Graphite Monitor Reporter Plugin with the host and port number of the Graphite service, and with a prefix for your server, such as its FQDN. By default, the plugin pushes all metrics it produces to the Graphite service. You can opt to limit this by setting the `excluded-metric-pattern` or `included-metric-pattern` properties.

The following example configures the plugin to push metrics to Graphite at `graphite.example.com:2004` every 10 seconds (default):

```
$ dsconfig \
 create-plugin \
 --hostname localhost \
 --port 4444 \
 --bindDN uid=admin \
 --bindPassword password \
 --plugin-name Graphite \
 --type graphite-monitor-reporter \
 --set enabled:true \
 --set graphite-server:graphite.example.com:2004 \
 --set metric-name-prefix:ds.example.com \
 --usePkcs12TrustStore /path/to/opendj/config/keystore \
 --trustStorePassword:file /path/to/opendj/config/keystore.pin \
 --no-prompt
```

To view metrics stored in Graphite, you can use the Graphite render API or Grafana, for example. See the Graphite and Grafana documentation for details.

Alerts

DS servers can send alerts for significant server events.

The following example enables JMX alert notifications:

```
$ dsconfig \
 set-alert-handler-prop \
 --hostname localhost \
 --port 4444 \
 --bindDN uid=admin \
 --bindPassword password \
 --handler-name "JMX Alert Handler" \
 --set enabled:true \
 --usePkcs12TrustStore /path/to/opendj/config/keystore \
 --trustStorePassword:file /path/to/opendj/config/keystore.pin \
 --no-prompt
```

The following example sets up an SMTP server, and configures email alerts:

```
$ dsconfig \
 create-mail-server \
 --hostname localhost \
 --port 4444 \
 --bindDN uid=admin \
 --bindPassword password \
 --server-name "SMTP server" \
 --set enabled:true \
 --set auth-username:mail.user \
 --set auth-password:password \
 --set smtp-server:smtp.example.com:587 \
 --set trust-manager-provider:"JVM Trust Manager" \
 --set use-start-tls:true \
 --usePkcs12TrustStore /path/to/opendj/config/keystore \
 --trustStorePassword:file /path/to/opendj/config/keystore.pin \
 --no-prompt

$ dsconfig \
 create-alert-handler \
 --hostname localhost \
 --port 4444 \
 --bindDN uid=admin \
 --bindPassword password \
 --handler-name "SMTP Alert Handler" \
 --type smtp \
 --set enabled:true \
 --set message-subject:"DS Alert, Type: %%alert-type%%, ID:
%%alert-id%%" \
 --set message-body:"%%alert-message%%" \
 --set recipient-address:kvaughan@example.com \
 --set sender-address:ds@example.com \
 --usePkcs12TrustStore /path/to/opendj/config/keystore \
 --trustStorePassword:file /path/to/opendj/config/keystore.pin \
 --no-prompt
```

Alert Types

DS servers use the following alert types. For alert types that indicate server problems, check `logs/errors` for details:

org.opends.server.AccessControlDisabled

The access control handler has been disabled.

`org.opends.server.AccessControlEnabled`
The access control handler has been enabled.

`org.opends.server.authentiation.dseecompat.ACIParseFailed`
The dseecompat access control subsystem failed to correctly parse one or more ACI rules when the server first started.

`org.opends.server.CannotCopySchemaFiles`
A problem has occurred while attempting to create copies of the existing schema configuration files before making a schema update, and the schema configuration has been left in a potentially inconsistent state.

`org.opends.server.CannotRenameCurrentTaskFile`
The server is unable to rename the current tasks backing file in the process of trying to write an updated version.

`org.opends.server.CannotRenameNewTaskFile`
The server is unable to rename the new tasks backing file into place.

`org.opends.server.CannotScheduleRecurringIteration`
The server is unable to schedule an iteration of a recurring task.

`org.opends.server.CannotWriteConfig`
The server is unable to write its updated configuration for some reason and therefore the server may not exhibit the new configuration if it is restarted.

`org.opends.server.CannotWriteNewSchemaFiles`
A problem has occurred while attempting to write new versions of the server schema configuration files, and the schema configuration has been left in a potentially inconsistent state.

`org.opends.server.CannotWriteTaskFile`
The server is unable to write an updated tasks backing file for some reason.

`org.opends.server.DirectoryServerShutdown`
The server has begun the process of shutting down.

`org.opends.server.DirectoryServerStarted`
The server has completed its startup process.

`org.opends.server.DiskFull`
Free disk space has reached the full threshold.

Default is 6% of the size of the file system.

`org.opends.server.DiskSpaceLow`
Free disk space has reached the low threshold.

Default is 10% of the size of the file system.

org.opends.server.EnteringLockdownMode
The server is entering lockdown mode, wherein only root users are allowed to perform operations and only over the loopback address.

org.opends.server.LDAPHandlerDisabledByConsecutiveFailures
Consecutive failures have occurred in the LDAP connection handler and have caused it to become disabled.

org.opends.server.LDAPHandlerUncaughtError
Uncaught errors in the LDAP connection handler have caused it to become disabled.

org.opends.server.LDIFBackendCannotWriteUpdate
An LDIF backend was unable to store an updated copy of the LDIF file after processing a write operation.

org.opends.server.LDIFConnectionHandlerIOError
The LDIF connection handler encountered an I/O error that prevented it from completing its processing.

org.opends.server.LDIFConnectionHandlerParseError
The LDIF connection handler encountered an unrecoverable error while attempting to parse an LDIF file.

org.opends.server.LeavingLockdownMode
The server is leaving lockdown mode.

org.opends.server.ManualConfigEditHandled
The server detects that its configuration has been manually edited with the server online, and those changes were overwritten by another change made through the server. The manually edited configuration will be copied to another location.

org.opends.server.ManualConfigEditLost
The server detects that its configuration has been manually edited with the server online, and those changes were overwritten by another change made through the server. The manually edited configuration could not be preserved due to an unexpected error.

org.opends.server.replication.UnresolvedConflict
Multimaster replication cannot resolve a conflict automatically.

org.opends.server.UncaughtException
A server thread has encountered an uncaught exception that caused that thread to terminate abnormally. The impact that this problem has on the server depends on which thread was impacted and the nature of the exception.

org.opends.server.UniqueAttributeSynchronizationConflict
A unique attribute conflict has been detected during synchronization processing.

`org.opends.server.UniqueAttributeSynchronizationError`

An error occurred while attempting to perform unique attribute conflict detection during synchronization processing.

Metric Types Reference

The following monitoring metrics are available in each interface:

Type	Description
Counter	Cumulative metric for a numerical value that only increases while the server is running. Counts that reflect volatile data, such as the number of requests, are reset to 0 when the server starts up.
Gauge	Metric for a numerical value that can increase or decrease.

<table>
<tr><th>Type</th><th>Description</th></tr>
<tr><td>Summary</td><td>

Metric that samples observations, providing a count of observations, sum total of observed amounts, average rate of events, and moving average rates across sliding time windows.

Common REST and LDAP views show summaries as JSON objects. JSON summaries have the following fields:[1]

```
{
  "count": number,      // Number of events since the server started
  "total": number,      // Sum of quantities measured for each event
                        // since the server started
  // The following are related to the "count":
  "mean_rate": number,  // Average event rate per second
                        // since the server started
  "m1_rate": number,    // One-minute average event rate per second
                        // (exponentially decaying)
  "m5_rate": number,    // Five-minute average event rate per second
                        // (exponentially decaying)
  "m15_rate": number,   // Fifteen-minute average event rate per second
                        // (exponentially decaying)
}
```

The `"total"` depends on the type of events measured. For example, if the `"count"` is the number of requests, then the `"total"` is the total etime in milliseconds to process all the requests. If the `"count"` is the number of times the server read bytes of data, then the `"total"` is the total number of bytes read.

The Prometheus view does not provide time-based statistics, as rates can be calculated from the time-series data. Instead, the Prometheus view includes summary metrics whose names have the following suffixes or labels:

- `_count` : number of events since the server started
- `_total` : sum of quantities measured for each event since the server started
- `{quantile="0.5"}` : 50% at or below this value since the server started

</td></tr>
</table>

Type	Description
	• `{quantile="0.75"}` : 75% at or below this value since the server started • `{quantile="0.95"}` : 95% at or below this value since the server started • `{quantile="0.98"}` : 98% at or below this value since the server started • `{quantile="0.99"}` : 99% at or below this value since the server started • `{quantile="0.999"}` : 99.9% at or below this value since the server started

Type	Description
Timer	Metric combining a summary with other statistics. Common REST and LDAP views show summaries as JSON objects. JSON summaries have the following fields[1]

```
{
  "count": number,     // Number of events since the
server started
  "total": number,     // Total duration for all
events
                       // since the server started,
in ms
                       // (for requests, sum of the
etimes
                       // since the server started,
in ms)
  // The following are related to the "count":
  "mean_rate": number, // Average event rate per
second
                       // since the server started
  "m1_rate": number,   // One-minute average event
rate per second
                       // (exponentially decaying)
  "m5_rate": number,   // Five-minute average event
rate per second
                       // (exponentially decaying)
  "m15_rate": number,  // Fifteen-minute average
event rate per second
                       // (exponentially decaying)
  // The following are related to the "total":
  "mean": number,      // Average duration over all
events
                       // since the server started,
in ms
  "min": number,       // Minimum duration recorded
                       // since the server started,
in ms
  "max": number,       // Maximum duration recorded
                       // since the server started,
in ms
  "stddev": number,    // Standard deviation of
durations
                       // since the server started,
in ms
  "p50": number,       // 50% durations at or below
```

<table>
<tr><th>Type</th><th>Description</th></tr>
<tr><td></td><td>
<pre>
this value
                              // (median) since the server
started, in ms
  "p75": number,   // 75% durations at or below
this value
                              // since the server started,
in ms
  "p95": number,   // 95% durations at or below
this value
                              // since the server started,
in ms
  "p98": number,   // 98% durations at or below
this value
                              // since the server started,
in ms
  "p99": number,   // 99% durations at or below
this value
                              // since the server started,
in ms
  "p999": number,   // 99.9% durations at or below
this value
                              // since the server started,
in ms
  "p9999": number,   // 99.99% durations at or
below this value
                              // since the server started,
in ms
  "p99999": number   // 99.999% durations at or
below this value
                              // since the server started,
in ms
}
</pre>
The Prometheus view does not provide time-based statistics. Rates can be calculated from the time-series data.
</td></tr>
</table>

[1] Monitoring metrics reflect sample observations made while the server is running. The values are not saved when the server shuts down. As a result, metrics of this type reflect data recorded since the server started.

Metrics that show etime measurements in milliseconds (ms) continue to show values in ms even if the server is configured to log etimes in nanoseconds.

The calculation of moving averages is intended to be the same as that of the **`uptime`** and **`top`** commands, where the moving average plotted over time is smoothed by weighting that decreases exponentially. For an explanation of the mechanism, see the Wikipedia section, Exponential moving average.

LDAP Metrics Reference

LDAP metrics are exposed as LDAP attributes on entries under `cn=monitor` . Metrics entry object class names start with `ds-monitor` . Metrics attribute names start with `ds-mon` . For details, see the About This Reference.

For examples of common monitoring requests, see LDAP-Based Monitoring.

> NOTE
>
> Some `ds-mon-jvm-*` metrics depend on the JVM version and configuration. In particular, GC-related metrics depend on the garbage collector that the server uses. The GC metric names are *unstable*, and can change even in a minor JVM release.

Name	Syntax	Description
`ds-mon-abandoned-requests`	Counter metric	Total number of abandoned operations since startup
`ds-mon-active-connections-count`	Integer	Number of active client connections
`ds-mon-active-persistent-searches`	Integer	Number of active persistent searches
`ds-mon-admin-hostport`	Host port	The administrative host and port
`ds-mon-alive`	Boolean	Indicates whether the server is alive
`ds-mon-alive-errors`	Directory String	Lists server errors preventing the server from operating correctly that require administrative action
`ds-mon-backend-degraded-index-count`	Integer	Number of degraded indexes in the backend
`ds-mon-backend-degraded-index`	Directory String	Backend degraded index
`ds-mon-backend-entry-count`	Integer	Number of entries contained in the backend
`ds-mon-backend-filter-use-indexed`	Integer	Number of indexed searches performed against the backend

Name	Syntax	Description
`ds-mon-backend-filter-use-start-time`	Generalized Time	Time when recording started for statistical information about the simple search filters processed against the backend
`ds-mon-backend-filter-use-unindexed`	Integer	Number of unindexed searches performed against the backend
`ds-mon-backend-filter-use`	Json	Information about the simple search filter processed against the backend
`ds-mon-backend-is-private`	Boolean	Whether the base DNs of this backend should be considered public or private
`ds-mon-backend-proxy-base-dn`	DN	Base DNs routed to remote LDAP servers by the proxy backend
`ds-mon-backend-proxy-shard`	Summary metric	Remote LDAP servers that the proxy backend forwards requests to
`ds-mon-backend-ttl-entries-deleted`	Summary metric	Summary for entries purged by time-to-live
`ds-mon-backend-ttl-is-running`	Boolean	Indicates whether time-to-live is in the process of purging expired entries
`ds-mon-backend-ttl-last-run-time`	Generalized Time	Last date and time when time-to-live finished purging expired entries
`ds-mon-backend-ttl-queue-size`	Integer	Number of entries queued for purging by the time-to-live service
`ds-mon-backend-ttl-thread-count`	Integer	Number of active time-to-live threads
`ds-mon-backend-writability-mode`	Directory String	Current backend behavior when processing write operations, can either be "disabled", "enabled" or "internal-only"
`ds-mon-base-dn-entry-count`	Integer	Number of subordinate entries of the base DN, including the base DN
`ds-mon-base-dn`	DN	Base DN handled by a backend
`ds-mon-build-number`	Integer	Build number of the Directory Server

Name	Syntax	Description
`ds-mon-build-time`	Generalized Time	Build date and time of the Directory Server
`ds-mon-bytes-read`	Summary metric	Network bytes read summary
`ds-mon-bytes-written`	Summary metric	Network bytes written summary
`ds-mon-cache-entry-count`	Integer	Current number of entries held in this cache
`ds-mon-cache-max-entry-count`	Integer	Maximum number of entries allowed in this cache
`ds-mon-cache-max-size-bytes`	Size in bytes	Memory limit for this cache
`ds-mon-cache-misses`	Summary metric	Number of attempts to retrieve an entry that was not held in this cache
`ds-mon-cache-total-tries`	Summary metric	Number of attempts to retrieve an entry from this cache
`ds-mon-certificate-expires-at`	Generalized Time	Certificate expiration date and time
`ds-mon-certificate-issuer-dn`	DN	Certificate issuer DN
`ds-mon-certificate-serial-number`	Integer	Certificate serial number
`ds-mon-certificate-subject-dn`	DN	Certificate subject DN
`ds-mon-changelog-id`	Directory String	Changelog identifier
`ds-mon-changelog-hostport`	Host port	The host and port of the changelog server
`ds-mon-changelog-purge-delay`	Duration in milli-seconds	The purge delay of the changelog

Name	Syntax	Description
`ds-mon-compact-version`	Directory String	Compact version of the Directory Server
`ds-mon-config-dn`	DN	DN of the configuration entry
`ds-mon-connected-to-server-hostport`	Host port	Host and replication port of the server that this server is connected to
`ds-mon-connected-to-server-id`	Integer	Identifier of the server that this server is connected to
`ds-mon-connection`	Json	Client connection summary information
`ds-mon-connections`	Summary metric	Connection summary
`ds-mon-current-connections`	Integer	Number of client connections currently established with the Directory Server
`ds-mon-current-delay`	Duration in milli-seconds	Current local delay in receiving replicated operations
`ds-mon-current-receive-window`	Integer	Current replication window size for receiving messages, indicating the number of replication messages a remote server can send before waiting on acknowledgement from this server. This does not depend on the TCP window size
`ds-mon-current-send-window`	Integer	Current replication window size for sending messages, indicating the number of replication messages this server can send before waiting on acknowledgement from the receiving server. This does not depend on the TCP window size
`ds-mon-current-time`	Generalized Time	Current date and time
`ds-mon-db-cache-evict-internal-nodes-count`	Integer	Number of internal nodes evicted from the database cache

Name	Syntax	Description
`ds-mon-db-cache-evict-leaf-nodes-count`	Integer	Number of leaf nodes (data records) evicted from the database cache
`ds-mon-db-cache-leaf-nodes`	Boolean	Whether leaf nodes are cached
`ds-mon-db-cache-misses-internal-nodes`	Integer	Number of internal nodes requested by btree operations that were not in the database cache
`ds-mon-db-cache-misses-leaf-nodes`	Integer	Number of leaf nodes (data records) requested by btree operations that were not in the database cache
`ds-mon-db-cache-size-active`	Size in bytes	Size of the database cache
`ds-mon-db-cache-size-total`	Size in bytes	Maximum size of the database cache
`ds-mon-db-cache-total-tries-internal-nodes`	Integer	Number of internal nodes requested by btree operations
`ds-mon-db-cache-total-tries-leaf-nodes`	Integer	Number of leaf nodes (data records) requested by btree operations
`ds-mon-db-checkpoint-count`	Integer	Number of checkpoints run so far
`ds-mon-db-log-cleaner-file-deletion-count`	Integer	Number of cleaner file deletions
`ds-mon-db-log-files-open`	Integer	Number of files currently open in the database file cache
`ds-mon-db-log-files-opened`	Integer	Number of times a log file has been opened
`ds-mon-db-log-size-active`	Size in bytes	Estimate of the amount in bytes of live data in all data files (i.e., the size of the DB, ignoring garbage)

Name	Syntax	Description
`ds-mon-db-log-size-total`	Size in bytes	Size used by all data files on disk
`ds-mon-db-log-utilization-max`	Integer	Current maximum (upper bound) log utilization as a percentage
`ds-mon-db-log-utilization-min`	Integer	Current minimum (lower bound) log utilization as a percentage
`ds-mon-db-version`	Directory String	Database version used by the backend
`ds-mon-disk-dir`	Filesystem path	A monitored directory containing data that may change over time
`ds-mon-disk-free`	Size in bytes	Amount of free disk space
`ds-mon-disk-full-threshold`	Size in bytes	Effective full disk space threshold
`ds-mon-disk-low-threshold`	Size in bytes	Effective low disk space threshold
`ds-mon-disk-root`	Filesystem path	Monitored disk root
`ds-mon-disk-state`	Directory String	Current disk state, can be either "normal", "low" or "full"
`ds-mon-domain-generation-id`	Integer	Replication domain generation identifier
`ds-mon-domain-name`	DN	Replication domain name
`ds-mon-entries-awaiting-updates-count`	Duration in milli-seconds	Number of entries for which an update operation has been received but not replayed yet by this replica
`ds-mon-fix-ids`	Directory String	IDs of issues that have been fixed in this Directory Server build
`ds-mon-full-version`	Directory String	Full version of the Directory Server
`ds-mon-group-id`	Directory String	Unique identifier of the group in which the directory server belongs

Name	Syntax	Description
`ds-mon-healthy`	Boolean	Indicates whether the server is able to handle requests
`ds-mon-healthy-errors`	Directory String	Lists transient server errors preventing the server from temporarily handling requests
`ds-mon-install-path`	Filesystem path	Directory Server root installation path
`ds-mon-instance-path`	Filesystem path	Directory Server instance path
`ds-mon-je-environment-nbytes-evicted-critical`	Size in bytes	Number of bytes evicted by the DB worker threads For details, see Cache Internal Nodes.
`ds-mon-jvm-architecture`	Directory String	Java virtual machine architecture (e.g. 32-bit, 64-bit)
`ds-mon-jvm-arguments`	Directory String	Input arguments passed to the Java virtual machine
`ds-mon-jvm-available-cpus`	Integer	Number of processors available to the Java virtual machine
`ds-mon-jvm-class-path`	Filesystem path	Path used to find directories and JAR archives containing Java class files
`ds-mon-jvm-classes-loaded`	Integer	Number of classes loaded since the Java virtual machine started
`ds-mon-jvm-classes-unloaded`	Integer	Number of classes unloaded since the Java virtual machine started
`ds-mon-jvm-java-home`	Filesystem path	Installation directory for Java runtime environment (JRE)
`ds-mon-jvm-java-vendor`	Directory String	Java runtime environment (JRE) vendor
`ds-mon-jvm-java-version`	Directory String	Java runtime environment (JRE) version

Name	Syntax	Description
`ds-mon-jvm-memory-heap-init`	Size in bytes	Amount of heap memory that the Java virtual machine initially requested from the operating system
`ds-mon-jvm-memory-heap-max`	Size in bytes	Maximum amount of heap memory that the Java virtual machine will attempt to use
`ds-mon-jvm-memory-heap-reserved`	Size in bytes	Amount of heap memory that is committed for the Java virtual machine to use
`ds-mon-jvm-memory-heap-used`	Size in bytes	Amount of heap memory used by the Java virtual machine
`ds-mon-jvm-memory-init`	Size in bytes	Amount of memory that the Java virtual machine initially requested from the operating system
`ds-mon-jvm-memory-max`	Size in bytes	Maximum amount of memory that the Java virtual machine will attempt to use
`ds-mon-jvm-memory-non-heap-init`	Size in bytes	Amount of non-heap memory that the Java virtual machine initially requested from the operating system
`ds-mon-jvm-memory-non-heap-max`	Size in bytes	Maximum amount of non-heap memory that the Java virtual machine will attempt to use
`ds-mon-jvm-memory-non-heap-reserved`	Size in bytes	Amount of non-heap memory that is committed for the Java virtual machine to use
`ds-mon-jvm-memory-non-heap-used`	Size in bytes	Amount of non-heap memory used by the Java virtual machine
`ds-mon-jvm-memory-reserved`	Size in bytes	Amount of memory that is committed for the Java virtual machine to use
`ds-mon-jvm-memory-used`	Size in bytes	Amount of memory used by the Java virtual machine
`ds-mon-jvm-supported-tls-ciphers`	Directory String	Transport Layer Security (TLS) cipher suites supported by this Directory Server

Name	Syntax	Description
`ds-mon-jvm-supported-tls-protocols`	Directory String	Transport Layer Security (TLS) protocols supported by this Directory Server
`ds-mon-jvm-threads-blocked-count`	Integer	Number of threads in the BLOCKED state
`ds-mon-jvm-threads-count`	Integer	Number of live threads including both daemon and non-daemon threads
`ds-mon-jvm-threads-daemon-count`	Integer	Number of live daemon threads
`ds-mon-jvm-threads-deadlock-count`	Integer	Number of deadlocked threads
`ds-mon-jvm-threads-deadlocks`	Directory String	Diagnostic stack traces for deadlocked threads
`ds-mon-jvm-threads-new-count`	Integer	Number of threads in the NEW state
`ds-mon-jvm-threads-runnable-count`	Integer	Number of threads in the RUNNABLE state
`ds-mon-jvm-threads-terminated-count`	Integer	Number of threads in the TERMINATED state
`ds-mon-jvm-threads-timed-waiting-count`	Integer	Number of threads in the TIMED_WAITING state
`ds-mon-jvm-threads-waiting-count`	Integer	Number of threads in the WAITING state
`ds-mon-jvm-vendor`	Directory String	Java virtual machine vendor
`ds-mon-jvm-version`	Directory String	Java virtual machine version
`ds-mon-last-seen`	Generalized Time	Time that this server was last seen
`ds-mon-ldap-hostport`	Host port	The host and port to connect using LDAP (no support for start TLS)

Name	Syntax	Description
`ds-mon-ldap-starttls-hostport`	Host port	The host and port to connect using LDAP (with support for start TLS)
`ds-mon-ldaps-hostport`	Host port	The host and port to connect using LDAPS
`ds-mon-listen-address`	Directory String	Host and port
`ds-mon-lost-connections`	Duration in milli-seconds	Number of times the replica lost its connection to the replication server
`ds-mon-major-version`	Integer	Major version number of the Directory Server
`ds-mon-max-connections`	Integer	Maximum number of simultaneous client connections that have been established with the Directory Server
`ds-mon-max-receive-window`	Integer	Maximum replication window size for receiving messages, indicating the number of replication messages a remote server can send before waiting on acknowledgement from this server. This does not depend on the TCP window size
`ds-mon-max-send-window`	Integer	Maximum replication window size for sending messages, indicating the number of replication messages this server can send before waiting on acknowledgement from the receiving server. This does not depend on the TCP window size
`ds-mon-minor-version`	Integer	Minor version number of the Directory Server
`ds-mon-newest-change-number`	Integer	Newest change number present in the change number index database
`ds-mon-newest-csn-timestamp`	Generalized Time	Timestamp of the newest CSN present in the replica database

Name	Syntax	Description
`ds-mon-newest-csn`	CSN (Change Sequence Number)	Newest CSN present in the replica database
`ds-mon-oldest-change-number`	Integer	Oldest change number present in the change number index database
`ds-mon-oldest-csn-timestamp`	Generalized Time	Timestamp of the oldest CSN present in the replica database
`ds-mon-oldest-csn`	CSN (Change Sequence Number)	Oldest CSN present in the replica database
`ds-mon-os-architecture`	Directory String	Operating system architecture
`ds-mon-os-name`	Directory String	Operating system name
`ds-mon-os-version`	Directory String	Operating system version
`ds-mon-point-version`	Integer	Point version number of the Directory Server
`ds-mon-process-id`	UUID	Process ID of the running directory server
`ds-mon-product-name`	Directory String	Full name of the Directory Server
`ds-mon-protocol`	Directory String	Network protocol
`ds-mon-receive-delay`	Duration in milli-seconds	Current local delay in receiving replicated operations
`ds-mon-replay-delay`	Duration in milli-seconds	Current local delay in replaying replicated operations

Name	Syntax	Description
`ds-mon-replayed-updates-conflicts-resolved`	Counter metric	Number of updates replayed on this replica for which replication naming conflicts have been resolved
`ds-mon-replayed-updates-conflicts-unresolved`	Counter metric	Number of updates replayed on this replica for which replication naming conflicts have not been resolved
`ds-mon-replayed-internal-updates`	Counter metric	Number of updates replayed on this replica which modify the internal state but not user data
`ds-mon-replayed-updates`	Timer metric	Timer for updates that have been replayed on this replica
`ds-mon-replica-hostport`	Host port	Host and port of a replica server
`ds-mon-replication-domain`	DN	The replication domain
`ds-mon-replication-protocol-version`	Integer	The protocol version used for replication
`ds-mon-requests-abandon`	Timer metric	Abandon request timer
`ds-mon-requests-add`	Timer metric	Add request timer
`ds-mon-requests-bind`	Timer metric	Bind request timer
`ds-mon-requests-compare`	Timer metric	Compare request timer
`ds-mon-requests-delete`	Timer metric	Delete request timer
`ds-mon-requests-extended`	Timer metric	Extended request timer

Name	Syntax	Description
`ds-mon-requests-failure-client-invalid-request`	Timer metric	Timer for requests that failed because there was a problem while attempting to perform the associated operation (associated LDAP result codes: 1, 2, 12, 15, 16, 17, 18, 19, 20, 21, 23, 34, 35, 36, 37, 38, 39; associated HTTP status codes: client error (4xx) except 401 and 403)
`ds-mon-requests-failure-client-redirect`	Timer metric	Timer for requests that could not complete because further action is required (associated HTTP status codes: redirection (3xx))
`ds-mon-requests-failure-client-referral`	Timer metric	Timer for requests that failed because the server did not hold the request targeted entry (but was able to provide alternative servers that may) (associated LDAP result code: 10)
`ds-mon-requests-failure-client-resource-limit`	Timer metric	Timer for requests that failed because they were trying to exceed the resource limits allocated to the associated clients (associated LDAP result codes: time, size and admin limit exceeded (respectively 4, 5 and 11)
`ds-mon-requests-failure-client-security`	Timer metric	Timer for requests that failed for security reasons (associated LDAP result codes: 8, 9, 13, 25, 26, 27; associated HTTP status codes: unauthorized (401) and forbidden (403))
`ds-mon-requests-failure-server`	Timer metric	Timer for apparently valid requests that failed because the server was not able to process them (associated LDAP result codes: busy (51), unavailable (52), unwilling to perform (53) and other (80); associated HTTP status codes: server error (5xx))
`ds-mon-requests-failure-uncategorized`	Timer metric	Timer for requests that failed due to uncategorized reasons

Name	Syntax	Description
`ds-mon-requests-get`	Timer metric	GET request timer
`ds-mon-requests-in-queue`	Integer	Number of requests in the work queue that have not yet been picked up for processing
`ds-mon-requests-modify-dn`	Timer metric	Modify DN request timer
`ds-mon-requests-modify`	Timer metric	Modify request timer
`ds-mon-requests-patch`	Timer metric	PATCH request timer
`ds-mon-requests-post`	Timer metric	POST request timer
`ds-mon-requests-put`	Timer metric	PUT request timer
`ds-mon-requests-search-base`	Timer metric	Base object search request timer
`ds-mon-requests-search-one`	Timer metric	One level search request timer
`ds-mon-requests-search-sub`	Timer metric	Subtree search request timer
`ds-mon-requests-submitted`	Summary metric	Summary for operations that have been successfully submitted to the work queue
`ds-mon-requests-unbind`	Timer metric	Unbind request timer
`ds-mon-requests-uncategorized`	Timer metric	Uncategorized request timer
`ds-mon-revision`	Directory String	Revision ID in the source repository from which the Directory Server is build
`ds-mon-sent-updates`	Counter metric	Number of replication updates sent by this replica

Name	Syntax	Description
`ds-mon-server-hostport`	Host port	Host and port of a server
`ds-mon-server-id`	Integer	Server identifier
`ds-mon-server-is-local`	Boolean	Indicates whether this is the topology server that has handled the monitoring request
`ds-mon-server-state`	CSN (Change Sequence Number)	Replication server state
`ds-mon-short-name`	Directory String	Short name of the Directory Server
`ds-mon-ssl-encryption`	Boolean	Whether SSL encryption is used when exchanging messages with this server
`ds-mon-start-time`	Generalized Time	Start date and time for the Directory Server
`ds-mon-status-last-changed`	Generalized Time	Last date and time the replication status of the local replica changed
`ds-mon-status`	Directory String	Replication status of the local replica, can either be "Invalid", "Not connected", "Normal", "Degraded", "Full update", "Bad generation id"
`ds-mon-system-name`	Directory String	Fully qualified domain name of the system where the Directory Server is running
`ds-mon-total-connections`	Integer	Total number of client connections that have been established with the Directory Server since it started
`ds-mon-updates-already-in-progress`	Counter metric	Number of duplicate updates: updates received by this replica which cannot be applied because they are already in progress. Can happen when a directory server fails over to another replication server

Name	Syntax	Description
ds-mon-updates-inbound-queue	Integer	Number of remote updates received from the replication server but not replayed yet on this replica
ds-mon-updates-outbound-queue	Integer	Number of local updates that are waiting to be sent to the replication server once they complete
ds-mon-updates-totals-per-replay-thread	Json	JSON array of the number of updates replayed per replay thread
ds-mon-vendor-name	Directory String	Vendor name of the Directory Server
ds-mon-version-qualifier	Directory String	Version qualifier of the Directory Server
ds-mon-working-directory	Filesystem path	Current working directory of the user running the Directory Server

Prometheus Metrics Reference

The following list puts Prometheus labels in braces. For example, the labels in `ds_backend_db_cache_misses_internal_nodes{backend,type}` are `backend` and `type`.

For examples of common monitoring requests, see HTTP-Based Monitoring.

NOTE

Some `ds_jvm_*` metrics depend on the JVM version and configuration. In particular, GC-related metrics depend on the garbage collector that the server uses. The GC metric names are *unstable*, and can change even in a minor JVM release.

Name	Type	Description
ds_all_entry_caches_cache_entry_count	Gauge	Current number of entries held in this cache
ds_all_entry_caches_cache_misses_count	Summary	Number of attempts to retrieve an entry that was not held in this cache

Name	Type	Description
`ds_all_entry_caches_cache_misses_total`	Summary	Number of attempts to retrieve an entry that was not held in this cache
`ds_all_entry_caches_cache_total_tries_count`	Summary	Number of attempts to retrieve an entry from this cache
`ds_all_entry_caches_cache_total_tries_total`	Summary	Number of attempts to retrieve an entry from this cache
`ds_backend_db_cache_evict_internal_nodes_count{backend,type}`	Gauge	Number of internal nodes evicted from the database cache
`ds_backend_db_cache_evict_leaf_nodes_count{backend,type}`	Gauge	Number of leaf nodes (data records) evicted from the database cache
`ds_backend_db_cache_leaf_nodes{backend,type}`	Gauge	Whether leaf nodes are cached
`ds_backend_db_cache_misses_internal_nodes{backend,type}`	Gauge	Number of internal nodes requested by btree operations that were not in the database cache
`ds_backend_db_cache_misses_leaf_nodes{backend,type}`	Gauge	Number of leaf nodes (data records) requested by btree operations that were not in the database cache
`ds_backend_db_cache_size_active_bytes{backend,type}`	Gauge	Size of the database cache
`ds_backend_db_cache_size_total_bytes{backend,type}`	Gauge	Maximum size of the database cache
`ds_backend_db_cache_total_tries_internal_nodes{backend,type}`	Gauge	Number of internal nodes requested by btree operations

Name	Type	Description
ds_backend_db_cache_total_tries_leaf_nodes{backend,type}	Gauge	Number of leaf nodes (data records) requested by btree operations
ds_backend_db_checkpoint_count{backend,type}	Gauge	Number of checkpoints run so far
ds_backend_db_log_cleaner_file_deletion_count{backend,type}	Gauge	Number of cleaner file deletions
ds_backend_db_log_files_open{backend,type}	Gauge	Number of files currently open in the database file cache
ds_backend_db_log_files_opened{backend,type}	Gauge	Number of times a log file has been opened
ds_backend_db_log_size_active_bytes{backend,type}	Gauge	Estimate of the amount in bytes of live data in all data files (i.e., the size of the DB, ignoring garbage)
ds_backend_db_log_size_total_bytes{backend,type}	Gauge	Size used by all data files on disk
ds_backend_db_log_utilization_max{backend,type}	Gauge	Current maximum (upper bound) log utilization as a percentage
ds_backend_db_log_utilization_min{backend,type}	Gauge	Current minimum (lower bound) log utilization as a percentage
ds_backend_degraded_index_count{backend,type}	Gauge	Number of degraded indexes in the backend
ds_backend_entry_count{backend,base_dn,dc,type}	Gauge	Number of subordinate entries of the base DN, including the base DN

Name	Type	Description
`ds_backend_entry_count{backend,base_dn,type}`	Gauge	Number of subordinate entries of the base DN, including the base DN
`ds_backend_filter_use_indexed{backend,type}`	Gauge	Number of indexed searches performed against the backend
`ds_backend_filter_use_start_time_seconds{backend,type}`	Gauge	Time when recording started for statistical information about the simple search filters processed against the backend
`ds_backend_filter_use_unindexed{backend,type}`	Gauge	Number of unindexed searches performed against the backend
`ds_backend_is_private{backend,type}`	Gauge	Whether the base DNs of this backend should be considered public or private
`ds_backend_ttl_entries_deleted_count{backend,type}`	Summary	Summary for entries purged by time-to-live
`ds_backend_ttl_entries_deleted_total{backend,type}`	Summary	Summary for entries purged by time-to-live
`ds_backend_ttl_is_running{backend,type}`	Gauge	Indicates whether time-to-live is in the process of purging expired entries
`ds_backend_ttl_last_run_time_seconds{backend,type}`	Gauge	Last date and time when time-to-live finished purging expired entries
`ds_backend_ttl_queue_size{backend,type}`	Gauge	Number of entries queued for purging by the time-to-live service
`ds_backend_ttl_thread_count{backend,type}`	Gauge	Number of active time-to-live threads
`ds_certificates_certificate_expires_at_seconds{alias,key_manager}`	Gauge	Certificate expiration date and time

Name	Type	Description
`ds_connection_handlers_http_active_connections_count{http_handler}`	Gauge	Number of active client connections
`ds_connection_handlers_http_bytes_read_count{http_handler}`	Summary	Network bytes read summary
`ds_connection_handlers_http_bytes_read_total{http_handler}`	Summary	Network bytes read summary
`ds_connection_handlers_http_bytes_written_count{http_handler}`	Summary	Network bytes written summary
`ds_connection_handlers_http_bytes_written_total{http_handler}`	Summary	Network bytes written summary
`ds_connection_handlers_http_requests_count{http_handler,type}`	Summary	Delete request timer
`ds_connection_handlers_http_requests_count{http_handler,type}`	Summary	GET request timer
`ds_connection_handlers_http_requests_count{http_handler,type}`	Summary	PATCH request timer
`ds_connection_handlers_http_requests_count{http_handler,type}`	Summary	POST request timer
`ds_connection_handlers_http_requests_count{http_handler,type}`	Summary	PUT request timer
`ds_connection_handlers_http_requests_count{http_handler,type}`	Summary	Uncategorized request timer

Name	Type	Description
`ds_connection_handlers_http_requests_failure_count{http_handler,type}`	Summary	Timer for apparently valid requests that failed because the server was not able to process them (associated LDAP result codes: busy (51), unavailable (52), unwilling to perform (53) and other (80); associated HTTP status codes: server error (5xx))
`ds_connection_handlers_http_requests_failure_count{http_handler,type}`	Summary	Timer for requests that could not complete because further action is required (associated HTTP status codes: redirection (3xx))
`ds_connection_handlers_http_requests_failure_count{http_handler,type}`	Summary	Timer for requests that failed because there was a problem while attempting to perform the associated operation (associated LDAP result codes: 1, 2, 12, 15, 16, 17, 18, 19, 20, 21, 23, 34, 35, 36, 37, 38, 39; associated HTTP status codes: client error (4xx) except 401 and 403)
`ds_connection_handlers_http_requests_failure_count{http_handler,type}`	Summary	Timer for requests that failed due to uncategorized reasons
`ds_connection_handlers_http_requests_failure_count{http_handler,type}`	Summary	Timer for requests that failed for security reasons (associated LDAP result codes: 8, 9, 13, 25, 26, 27; associated HTTP status codes: unauthorized (401) and forbidden (403))
`ds_connection_handlers_http_requests_failure_seconds_total{http_handler,type}`	Summary	Timer for apparently valid requests that failed because the server was not able to process them (associated LDAP result codes: busy (51), unavailable (52), unwilling to perform (53) and other (80); associated HTTP status codes: server error (5xx))
`ds_connection_handlers_http_requests_failure_seconds_total{http_handler,type}`	Summary	Timer for requests that could not complete because further action is required (associated HTTP status codes: redirection (3xx))

Name	Type	Description
ds_connection_handlers_http_requests_failure_seconds_total{http_handler,type}	Summary	Timer for requests that failed because there was a problem while attempting to perform the associated operation (associated LDAP result codes: 1, 2, 12, 15, 16, 17, 18, 19, 20, 21, 23, 34, 35, 36, 37, 38, 39; associated HTTP status codes: client error (4xx) except 401 and 403)
ds_connection_handlers_http_requests_failure_seconds_total{http_handler,type}	Summary	Timer for requests that failed due to uncategorized reasons
ds_connection_handlers_http_requests_failure_seconds_total{http_handler,type}	Summary	Timer for requests that failed for security reasons (associated LDAP result codes: 8, 9, 13, 25, 26, 27; associated HTTP status codes: unauthorized (401) and forbidden (403))
ds_connection_handlers_http_requests_failure_seconds{http_handler,type,quantile}	Summary	Timer for apparently valid requests that failed because the server was not able to process them (associated LDAP result codes: busy (51), unavailable (52), unwilling to perform (53) and other (80); associated HTTP status codes: server error (5xx))
ds_connection_handlers_http_requests_failure_seconds{http_handler,type,quantile}	Summary	Timer for requests that could not complete because further action is required (associated HTTP status codes: redirection (3xx))
ds_connection_handlers_http_requests_failure_seconds{http_handler,type,quantile}	Summary	Timer for requests that failed because there was a problem while attempting to perform the associated operation (associated LDAP result codes: 1, 2, 12, 15, 16, 17, 18, 19, 20, 21, 23, 34, 35, 36, 37, 38, 39; associated HTTP status codes: client error (4xx) except 401 and 403)
ds_connection_handlers_http_requests_failure_seconds{http_handler,type,quantile}	Summary	Timer for requests that failed due to uncategorized reasons

Name	Type	Description
`ds_connection_handlers_http_requests_failure_seconds{http_handler,type,quantile}`	Summary	Timer for requests that failed for security reasons (associated LDAP result codes: 8, 9, 13, 25, 26, 27; associated HTTP status codes: unauthorized (401) and forbidden (403))
`ds_connection_handlers_http_requests_seconds_total{http_handler,type}`	Summary	Delete request timer
`ds_connection_handlers_http_requests_seconds_total{http_handler,type}`	Summary	GET request timer
`ds_connection_handlers_http_requests_seconds_total{http_handler,type}`	Summary	PATCH request timer
`ds_connection_handlers_http_requests_seconds_total{http_handler,type}`	Summary	POST request timer
`ds_connection_handlers_http_requests_seconds_total{http_handler,type}`	Summary	PUT request timer
`ds_connection_handlers_http_requests_seconds_total{http_handler,type}`	Summary	Uncategorized request timer
`ds_connection_handlers_http_requests_seconds{http_handler,type,quantile}`	Summary	Delete request timer
`ds_connection_handlers_http_requests_seconds{http_handler,type,quantile}`	Summary	GET request timer

Name	Type	Description
`ds_connection_handlers_http_requests_seconds{http_handler,type,quantile}`	Summary	PATCH request timer
`ds_connection_handlers_http_requests_seconds{http_handler,type,quantile}`	Summary	POST request timer
`ds_connection_handlers_http_requests_seconds{http_handler,type,quantile}`	Summary	PUT request timer
`ds_connection_handlers_http_requests_seconds{http_handler,type,quantile}`	Summary	Uncategorized request timer
`ds_connection_handlers_ldap_abandoned_requests{ldap_handler}`	Counter	Total number of abandoned operations since startup
`ds_connection_handlers_ldap_active_connections_count{ldap_handler}`	Gauge	Number of active client connections
`ds_connection_handlers_ldap_active_persistent_searches{ldap_handler}`	Gauge	Number of active persistent searches
`ds_connection_handlers_ldap_bytes_read_count{ldap_handler}`	Summary	Network bytes read summary
`ds_connection_handlers_ldap_bytes_read_total{ldap_handler}`	Summary	Network bytes read summary
`ds_connection_handlers_ldap_bytes_written_count{ldap_handler}`	Summary	Network bytes written summary

Name	Type	Description
`ds_connection_handlers_ldap_bytes_written_total{ldap_handler}`	Summary	Network bytes written summary
`ds_connection_handlers_ldap_connections_count{ldap_handler}`	Summary	Connection summary
`ds_connection_handlers_ldap_connections_total{ldap_handler}`	Summary	Connection summary
`ds_connection_handlers_ldap_requests_count{ldap_handler,scope,type}`	Summary	Base object search request timer
`ds_connection_handlers_ldap_requests_count{ldap_handler,scope,type}`	Summary	One level search request timer
`ds_connection_handlers_ldap_requests_count{ldap_handler,scope,type}`	Summary	Subtree search request timer
`ds_connection_handlers_ldap_requests_count{ldap_handler,type}`	Summary	Abandon request timer
`ds_connection_handlers_ldap_requests_count{ldap_handler,type}`	Summary	Add request timer
`ds_connection_handlers_ldap_requests_count{ldap_handler,type}`	Summary	Bind request timer
`ds_connection_handlers_ldap_requests_count{ldap_handler,type}`	Summary	Compare request timer

Name	Type	Description
`ds_connection_handlers_ldap_requests_count{ldap_handler,type}`	Summary	Delete request timer
`ds_connection_handlers_ldap_requests_count{ldap_handler,type}`	Summary	Extended request timer
`ds_connection_handlers_ldap_requests_count{ldap_handler,type}`	Summary	Modify DN request timer
`ds_connection_handlers_ldap_requests_count{ldap_handler,type}`	Summary	Modify request timer
`ds_connection_handlers_ldap_requests_count{ldap_handler,type}`	Summary	Unbind request timer
`ds_connection_handlers_ldap_requests_count{ldap_handler,type}`	Summary	Uncategorized request timer
`ds_connection_handlers_ldap_requests_failure_count{ldap_handler,type}`	Summary	Timer for apparently valid requests that failed because the server was not able to process them (associated LDAP result codes: busy (51), unavailable (52), unwilling to perform (53) and other (80); associated HTTP status codes: server error (5xx))
`ds_connection_handlers_ldap_requests_failure_count{ldap_handler,type}`	Summary	Timer for requests that failed because the server did not hold the request targeted entry (but was able to provide alternative servers that may) (associated LDAP result code: 10)
`ds_connection_handlers_ldap_requests_failure_count{ldap_handler,type}`	Summary	Timer for requests that failed because there was a problem while attempting to perform the associated operation (associated LDAP result codes: 1, 2, 12, 15, 16, 17, 18, 19, 20, 21, 23, 34, 35, 36, 37, 38, 39; associated HTTP status codes: client error (4xx) except 401 and 403)

Name	Type	Description
`ds_connection_handlers_ldap_requests_failure_count{ldap_handler,type}`	Summary	Timer for requests that failed because they were trying to exceed the resource limits allocated to the associated clients (associated LDAP result codes: time, size and admin limit exceeded (respectively 4, 5 and 11)
`ds_connection_handlers_ldap_requests_failure_count{ldap_handler,type}`	Summary	Timer for requests that failed due to uncategorized reasons
`ds_connection_handlers_ldap_requests_failure_count{ldap_handler,type}`	Summary	Timer for requests that failed for security reasons (associated LDAP result codes: 8, 9, 13, 25, 26, 27; associated HTTP status codes: unauthorized (401) and forbidden (403))
`ds_connection_handlers_ldap_requests_failure_seconds_total{ldap_handler,type}`	Summary	Timer for apparently valid requests that failed because the server was not able to process them (associated LDAP result codes: busy (51), unavailable (52), unwilling to perform (53) and other (80); associated HTTP status codes: server error (5xx))
`ds_connection_handlers_ldap_requests_failure_seconds_total{ldap_handler,type}`	Summary	Timer for requests that failed because the server did not hold the request targeted entry (but was able to provide alternative servers that may) (associated LDAP result code: 10)
`ds_connection_handlers_ldap_requests_failure_seconds_total{ldap_handler,type}`	Summary	Timer for requests that failed because there was a problem while attempting to perform the associated operation (associated LDAP result codes: 1, 2, 12, 15, 16, 17, 18, 19, 20, 21, 23, 34, 35, 36, 37, 38, 39; associated HTTP status codes: client error (4xx) except 401 and 403)

Name	Type	Description
`ds_connection_handlers_ldap_requests_failure_seconds_total{ldap_handler,type}`	Summary	Timer for requests that failed because they were trying to exceed the resource limits allocated to the associated clients (associated LDAP result codes: time, size and admin limit exceeded (respectively 4, 5 and 11)
`ds_connection_handlers_ldap_requests_failure_seconds_total{ldap_handler,type}`	Summary	Timer for requests that failed due to uncategorized reasons
`ds_connection_handlers_ldap_requests_failure_seconds_total{ldap_handler,type}`	Summary	Timer for requests that failed for security reasons (associated LDAP result codes: 8, 9, 13, 25, 26, 27; associated HTTP status codes: unauthorized (401) and forbidden (403))
`ds_connection_handlers_ldap_requests_failure_seconds{ldap_handler,type,quantile}`	Summary	Timer for apparently valid requests that failed because the server was not able to process them (associated LDAP result codes: busy (51), unavailable (52), unwilling to perform (53) and other (80); associated HTTP status codes: server error (5xx))
`ds_connection_handlers_ldap_requests_failure_seconds{ldap_handler,type,quantile}`	Summary	Timer for requests that failed because the server did not hold the request targeted entry (but was able to provide alternative servers that may) (associated LDAP result code: 10)
`ds_connection_handlers_ldap_requests_failure_seconds{ldap_handler,type,quantile}`	Summary	Timer for requests that failed because there was a problem while attempting to perform the associated operation (associated LDAP result codes: 1, 2, 12, 15, 16, 17, 18, 19, 20, 21, 23, 34, 35, 36, 37, 38, 39; associated HTTP status codes: client error (4xx) except 401 and 403)

Name	Type	Description
`ds_connection_handlers_ldap_requests_failure_seconds{ldap_handler,type,quantile}`	Summary	Timer for requests that failed because they were trying to exceed the resource limits allocated to the associated clients (associated LDAP result codes: time, size and admin limit exceeded (respectively 4, 5 and 11)
`ds_connection_handlers_ldap_requests_failure_seconds{ldap_handler,type,quantile}`	Summary	Timer for requests that failed due to uncategorized reasons
`ds_connection_handlers_ldap_requests_failure_seconds{ldap_handler,type,quantile}`	Summary	Timer for requests that failed for security reasons (associated LDAP result codes: 8, 9, 13, 25, 26, 27; associated HTTP status codes: unauthorized (401) and forbidden (403))
`ds_connection_handlers_ldap_requests_seconds_total{ldap_handler,scope,type}`	Summary	Base object search request timer
`ds_connection_handlers_ldap_requests_seconds_total{ldap_handler,scope,type}`	Summary	One level search request timer
`ds_connection_handlers_ldap_requests_seconds_total{ldap_handler,scope,type}`	Summary	Subtree search request timer
`ds_connection_handlers_ldap_requests_seconds_total{ldap_handler,type}`	Summary	Abandon request timer
`ds_connection_handlers_ldap_requests_seconds_total{ldap_handler,type}`	Summary	Add request timer

Name	Type	Description
`ds_connection_handlers_ldap_requests_seconds_total{ldap_handler,type}`	Summary	Bind request timer
`ds_connection_handlers_ldap_requests_seconds_total{ldap_handler,type}`	Summary	Compare request timer
`ds_connection_handlers_ldap_requests_seconds_total{ldap_handler,type}`	Summary	Delete request timer
`ds_connection_handlers_ldap_requests_seconds_total{ldap_handler,type}`	Summary	Extended request timer
`ds_connection_handlers_ldap_requests_seconds_total{ldap_handler,type}`	Summary	Modify DN request timer
`ds_connection_handlers_ldap_requests_seconds_total{ldap_handler,type}`	Summary	Modify request timer
`ds_connection_handlers_ldap_requests_seconds_total{ldap_handler,type}`	Summary	Unbind request timer
`ds_connection_handlers_ldap_requests_seconds_total{ldap_handler,type}`	Summary	Uncategorized request timer
`ds_connection_handlers_ldap_requests_seconds{ldap_handler,scope,type,quantile}`	Summary	Base object search request timer

Name	Type	Description
`ds_connection_handlers_ldap_requests_seconds{ldap_handler,scope,type,quantile}`	Summary	One level search request timer
`ds_connection_handlers_ldap_requests_seconds{ldap_handler,scope,type,quantile}`	Summary	Subtree search request timer
`ds_connection_handlers_ldap_requests_seconds{ldap_handler,type,quantile}`	Summary	Abandon request timer
`ds_connection_handlers_ldap_requests_seconds{ldap_handler,type,quantile}`	Summary	Add request timer
`ds_connection_handlers_ldap_requests_seconds{ldap_handler,type,quantile}`	Summary	Bind request timer
`ds_connection_handlers_ldap_requests_seconds{ldap_handler,type,quantile}`	Summary	Compare request timer
`ds_connection_handlers_ldap_requests_seconds{ldap_handler,type,quantile}`	Summary	Delete request timer
`ds_connection_handlers_ldap_requests_seconds{ldap_handler,type,quantile}`	Summary	Extended request timer
`ds_connection_handlers_ldap_requests_seconds{ldap_handler,type,quantile}`	Summary	Modify DN request timer

Name	Type	Description
`ds_connection_handlers_ldap_requests_seconds{ldap_handler,type,quantile}`	Summary	Modify request timer
`ds_connection_handlers_ldap_requests_seconds{ldap_handler,type,quantile}`	Summary	Unbind request timer
`ds_connection_handlers_ldap_requests_seconds{ldap_handler,type,quantile}`	Summary	Uncategorized request timer
`ds_current_connections`	Gauge	Number of client connections currently established with the Directory Server
`ds_current_time_seconds`	Gauge	Current date and time
`ds_disk_free_space_bytes{disk}`	Gauge	Amount of free disk space
`ds_disk_free_space_full_threshold_bytes{disk}`	Gauge	Effective full disk space threshold
`ds_disk_free_space_low_threshold_bytes{disk}`	Gauge	Effective low disk space threshold
`ds_entry_cache_entry_count{cache}`	Gauge	Current number of entries held in this cache
`ds_entry_cache_max_entry_count{cache}`	Gauge	Maximum number of entries allowed in this cache
`ds_entry_cache_max_size_bytes{cache}`	Gauge	Memory limit for this cache
`ds_entry_cache_misses_count{cache}`	Summary	Number of attempts to retrieve an entry that was not held in this cache
`ds_entry_cache_misses_total{cache}`	Summary	Number of attempts to retrieve an entry that was not held in this cache

Name	Type	Description
`ds_entry_cache_total_tries_count{cache}`	Summary	Number of attempts to retrieve an entry from this cache
`ds_entry_cache_total_tries_total{cache}`	Summary	Number of attempts to retrieve an entry from this cache
`ds_health_status_alive`	Gauge	Indicates whether the server is alive
`ds_health_status_healthy`	Gauge	Indicates whether the server is able to handle requests
`ds_jvm_available_cpus`	Gauge	Number of processors available to the Java virtual machine
`ds_jvm_classes_loaded`	Gauge	Number of classes loaded since the Java virtual machine started
`ds_jvm_classes_unloaded`	Gauge	Number of classes unloaded since the Java virtual machine started
`ds_jvm_memory_heap_init_bytes`	Gauge	Amount of heap memory that the Java virtual machine initially requested from the operating system
`ds_jvm_memory_heap_max_bytes`	Gauge	Maximum amount of heap memory that the Java virtual machine will attempt to use
`ds_jvm_memory_heap_reserved_bytes`	Gauge	Amount of heap memory that is committed for the Java virtual machine to use
`ds_jvm_memory_heap_used_bytes`	Gauge	Amount of heap memory used by the Java virtual machine
`ds_jvm_memory_init_bytes`	Gauge	Amount of memory that the Java virtual machine initially requested from the operating system
`ds_jvm_memory_max_bytes`	Gauge	Maximum amount of memory that the Java virtual machine will attempt to use
`ds_jvm_memory_non_heap_init_bytes`	Gauge	Amount of non-heap memory that the Java virtual machine initially requested from the operating system

Name	Type	Description
`ds_jvm_memory_non_heap_max_bytes`	Gauge	Maximum amount of non-heap memory that the Java virtual machine will attempt to use
`ds_jvm_memory_non_heap_reserved_bytes`	Gauge	Amount of non-heap memory that is committed for the Java virtual machine to use
`ds_jvm_memory_non_heap_used_bytes`	Gauge	Amount of non-heap memory used by the Java virtual machine
`ds_jvm_memory_reserved_bytes`	Gauge	Amount of memory that is committed for the Java virtual machine to use
`ds_jvm_memory_used_bytes`	Gauge	Amount of memory used by the Java virtual machine
`ds_jvm_threads_blocked_count`	Gauge	Number of threads in the BLOCKED state
`ds_jvm_threads_count`	Gauge	Number of live threads including both daemon and non-daemon threads
`ds_jvm_threads_daemon_count`	Gauge	Number of live daemon threads
`ds_jvm_threads_deadlock_count`	Gauge	Number of deadlocked threads
`ds_jvm_threads_new_count`	Gauge	Number of threads in the NEW state
`ds_jvm_threads_runnable_count`	Gauge	Number of threads in the RUNNABLE state
`ds_jvm_threads_terminated_count`	Gauge	Number of threads in the TERMINATED state
`ds_jvm_threads_timed_waiting_count`	Gauge	Number of threads in the TIMED_WAITING state
`ds_jvm_threads_waiting_count`	Gauge	Number of threads in the WAITING state

Name	Type	Description
`ds_max_connections`	Gauge	Maximum number of simultaneous client connections that have been established with the Directory Server
`ds_replication_changelog_connected_changelogs_current_receive_window{changelog_id,domain_name,dc}`	Gauge	Current replication window size for receiving messages, indicating the number of replication messages a remote server can send before waiting on acknowledgement from this server. This does not depend on the TCP window size
`ds_replication_changelog_connected_changelogs_current_receive_window{changelog_id,domain_name}`	Gauge	Current replication window size for receiving messages, indicating the number of replication messages a remote server can send before waiting on acknowledgement from this server. This does not depend on the TCP window size
`ds_replication_changelog_connected_changelogs_current_send_window{changelog_id,domain_name,dc}`	Gauge	Current replication window size for sending messages, indicating the number of replication messages this server can send before waiting on acknowledgement from the receiving server. This does not depend on the TCP window size
`ds_replication_changelog_connected_changelogs_current_send_window{changelog_id,domain_name}`	Gauge	Current replication window size for sending messages, indicating the number of replication messages this server can send before waiting on acknowledgement from the receiving server. This does not depend on the TCP window size
`ds_replication_changelog_connected_changelogs_domain_generation_id{changelog_id,domain_name,dc}`	Gauge	Replication domain generation identifier

Name	Type	Description
ds_replication_changelog_connected_changelogs_domain_generation_id{changelog_id,domain_name}	Gauge	Replication domain generation identifier
ds_replication_changelog_connected_changelogs_max_receive_window{changelog_id,domain_name,dc}	Gauge	Maximum replication window size for receiving messages, indicating the number of replication messages a remote server can send before waiting on acknowledgement from this server. This does not depend on the TCP window size
ds_replication_changelog_connected_changelogs_max_receive_window{changelog_id,domain_name}	Gauge	Maximum replication window size for receiving messages, indicating the number of replication messages a remote server can send before waiting on acknowledgement from this server. This does not depend on the TCP window size
ds_replication_changelog_connected_changelogs_max_send_window{changelog_id,domain_name,dc}	Gauge	Maximum replication window size for sending messages, indicating the number of replication messages this server can send before waiting on acknowledgement from the receiving server. This does not depend on the TCP window size
ds_replication_changelog_connected_changelogs_max_send_window{changelog_id,domain_name}	Gauge	Maximum replication window size for sending messages, indicating the number of replication messages this server can send before waiting on acknowledgement from the receiving server. This does not depend on the TCP window size
ds_replication_changelog_connected_changelogs_ssl_encryption{changelog_id,domain_name,dc}	Gauge	Whether SSL encryption is used when exchanging messages with this server

Name	Type	Description
`ds_replication_changelog_connected_changelogs_ssl_encryption{changelog_id,domain_name}`	Gauge	Whether SSL encryption is used when exchanging messages with this server
`ds_replication_changelog_connected_replicas_current_receive_window{domain_name,dc,server_id}`	Gauge	Current replication window size for receiving messages, indicating the number of replication messages a remote server can send before waiting on acknowledgement from this server. This does not depend on the TCP window size
`ds_replication_changelog_connected_replicas_current_receive_window{domain_name,server_id}`	Gauge	Current replication window size for receiving messages, indicating the number of replication messages a remote server can send before waiting on acknowledgement from this server. This does not depend on the TCP window size
`ds_replication_changelog_connected_replicas_current_send_window{domain_name,dc,server_id}`	Gauge	Current replication window size for sending messages, indicating the number of replication messages this server can send before waiting on acknowledgement from the receiving server. This does not depend on the TCP window size
`ds_replication_changelog_connected_replicas_current_send_window{domain_name,server_id}`	Gauge	Current replication window size for sending messages, indicating the number of replication messages this server can send before waiting on acknowledgement from the receiving server. This does not depend on the TCP window size
`ds_replication_changelog_connected_replicas_domain_generation_id{domain_name,dc,server_id}`	Gauge	Replication domain generation identifier

Name	Type	Description
`ds_replication_changelog_connected_replicas_domain_generation_id{domain_name,server_id}`	Gauge	Replication domain generation identifier
`ds_replication_changelog_connected_replicas_max_receive_window{domain_name,dc,server_id}`	Gauge	Maximum replication window size for receiving messages, indicating the number of replication messages a remote server can send before waiting on acknowledgement from this server. This does not depend on the TCP window size
`ds_replication_changelog_connected_replicas_max_receive_window{domain_name,server_id}`	Gauge	Maximum replication window size for receiving messages, indicating the number of replication messages a remote server can send before waiting on acknowledgement from this server. This does not depend on the TCP window size
`ds_replication_changelog_connected_replicas_max_send_window{domain_name,dc,server_id}`	Gauge	Maximum replication window size for sending messages, indicating the number of replication messages this server can send before waiting on acknowledgement from the receiving server. This does not depend on the TCP window size
`ds_replication_changelog_connected_replicas_max_send_window{domain_name,server_id}`	Gauge	Maximum replication window size for sending messages, indicating the number of replication messages this server can send before waiting on acknowledgement from the receiving server. This does not depend on the TCP window size
`ds_replication_changelog_connected_replicas_ssl_encryption{domain_name,dc,server_id}`	Gauge	Whether SSL encryption is used when exchanging messages with this server

Name	Type	Description
`ds_replication_changelog_connected_replicas_ssl_encryption{domain_name,server_id}`	Gauge	Whether SSL encryption is used when exchanging messages with this server
`ds_replication_changelog_domain_generation_id{domain_name,dc}`	Gauge	Replication domain generation identifier
`ds_replication_changelog_domain_generation_id{domain_name}`	Gauge	Replication domain generation identifier
`ds_replication_changelog_missing_changes{domain_name,dc}`	Gauge	Missing changes for replication
`ds_replication_changelog_missing_changes{domain_name}`	Gauge	Missing changes for replication
`ds_replication_changelog_newest_change_number`	Gauge	Newest change number present in the change number index database
`ds_replication_changelog_oldest_change_number`	Gauge	Oldest change number present in the change number index database
`ds_replication_changelog_replica_dbs_newest_csn_timestamp_seconds{domain_name,dc,server_id}`	Gauge	Timestamp of the newest CSN present in the replica database
`ds_replication_changelog_replica_dbs_oldest_csn_timestamp_seconds{domain_name,dc,server_id}`	Gauge	Timestamp of the oldest CSN present in the replica database

Name	Type	Description
`ds_replication_replica_current_receive_window`	Gauge	Current replication window size for receiving messages, indicating the number of replication messages a remote server can send before waiting on acknowledgement from this server. This does not depend on the TCP window size
`ds_replication_replica_current_send_window`	Gauge	Current replication window size for sending messages, indicating the number of replication messages this server can send before waiting on acknowledgement from the receiving server. This does not depend on the TCP window size
`ds_replication_replica_domain_generation_id`	Gauge	Replication domain generation identifier
`ds_replication_replica_entries_awaiting_updates_count`	Gauge	Number of entries for which an update operation has been received but not replayed yet by this replica
`ds_replication_replica_lost_connections`	Gauge	Number of times the replica lost its connection to the replication server
`ds_replication_replica_max_receive_window`	Gauge	Maximum replication window size for receiving messages, indicating the number of replication messages a remote server can send before waiting on acknowledgement from this server. This does not depend on the TCP window size
`ds_replication_replica_max_send_window`	Gauge	Maximum replication window size for sending messages, indicating the number of replication messages this server can send before waiting on acknowledgement from the receiving server. This does not depend on the TCP window size

Name	Type	Description
`ds_replication_replica_remote_replicas_current_delay_seconds{domain_name,dc,remote_server_id,server_id}`	Gauge	Current local delay in receiving replicated operations
`ds_replication_replica_remote_replicas_receive_delay_seconds{domain_name,dc,remote_server_id,server_id}`	Gauge	Current local delay in receiving replicated operations
`ds_replication_replica_remote_replicas_replay_delay_seconds{domain_name,dc,remote_server_id,server_id}`	Gauge	Current local delay in replaying replicated operations
`ds_replication_replica_remote_replicas_replayed_updates_count{domain_name,dc,remote_server_id,server_id}`	Summary	Timer for updates that have been replayed on this replica
`ds_replication_replica_remote_replicas_replayed_updates_seconds_total{domain_name,dc,remote_server_id,server_id}`	Summary	Timer for updates that have been replayed on this replica
`ds_replication_replica_remote_replicas_replayed_updates_seconds{domain_name,dc,remote_server_id,server_id,quantile}`	Summary	Timer for updates that have been replayed on this replica
`ds_replication_replica_replayed_internal_updates{domain_name,server_id}`	Counter	Number of updates replayed on this replica which modify the internal state but not user data

Name	Type	Description
`ds_replication_replica_replayed_updates_conflicts_resolved`	Counter	Number of updates replayed on this replica for which replication naming conflicts have been resolved
`ds_replication_replica_replayed_updates_conflicts_unresolved`	Counter	Number of updates replayed on this replica for which replication naming conflicts have not been resolved
`ds_replication_replica_replayed_updates_count`	Summary	Timer for updates that have been replayed on this replica
`ds_replication_replica_replayed_updates_seconds_total`	Summary	Timer for updates that have been replayed on this replica
`ds_replication_replica_replayed_updates_seconds{quantile}`	Summary	Timer for updates that have been replayed on this replica
`ds_replication_replica_sent_updates`	Counter	Number of replication updates sent by this replica
`ds_replication_replica_ssl_encryption`	Gauge	Whether SSL encryption is used when exchanging messages with this server
`ds_replication_replica_status_last_changed_seconds`	Gauge	Last date and time the replication status of the local replica changed
`ds_replication_replica_updates_already_in_progress{domain_name,server_id}`	Counter	Number of duplicate updates: updates received by this replica which cannot be applied because they are already in progress. Can happen when a directory server fails over to another replication server
`ds_replication_replica_updates_inbound_queue`	Gauge	Number of remote updates received from the replication server but not replayed yet on this replica
`ds_replication_replica_updates_outbound_queue`	Gauge	Number of local updates that are waiting to be sent to the replication server once they complete

Name	Type	Description
`ds_start_time_seconds`	Gauge	Start date and time for the Directory Server
`ds_topology_servers_server_is_local{server_id}`	Gauge	Indicates whether this is the topology server that has handled the monitoring request
`ds_total_connections`	Gauge	Total number of client connections that have been established with the Directory Server since it started
`ds_work_queue_requests_in_queue`	Gauge	Number of requests in the work queue that have not yet been picked up for processing
`ds_work_queue_requests_submitted_count`	Summary	Summary for operations that have been successfully submitted to the work queue
`ds_work_queue_requests_submitted_total`	Summary	Summary for operations that have been successfully submitted to the work queue

Was this helpful? 👍 👎