Log Storage

Log Rotation

Log rotation is closing a log file and opening a new log file when the first file is considered to be complete. Log rotation is typically performed according to a schedule (e.g., hourly, daily, weekly) or when a log file reaches a certain size. The primary benefits of log rotation are preserving log entries and keeping the size of log files manageable. When a log file is rotated, the preserved log file can be compressed to save space. Also, during log rotation, scripts are often run that act on the archived log. For example, a script might analyze the old log to identify malicious activity, or might perform filtering that causes only log entries meeting certain characteristics to be preserved. Many log generators offer log rotation capabilities; many log files can also be rotated through simple scripts or third-party utilities, which in some cases offer features not provided by the log generators.

Log Archival

Log archival is retaining logs for an extended period of time, typically on removable media, a storage area network (SAN), or a specialized log archival appliance or server. Logs often need to be preserved to meet legal or regulatory requirements. Section 4.2 provides additional information on log archival. There are two types of log archival: retention and preservation. Log retention is archiving logs on a regular basis as part of standard operational activities. Log preservation is keeping logs that normally would be discarded, because they contain records of activity of particular interest. Log preservation is typically performed in support of incident handling or investigations.

LOG INDEXING AND ROTATION FOR OPTIMIZED ARCHIVAL

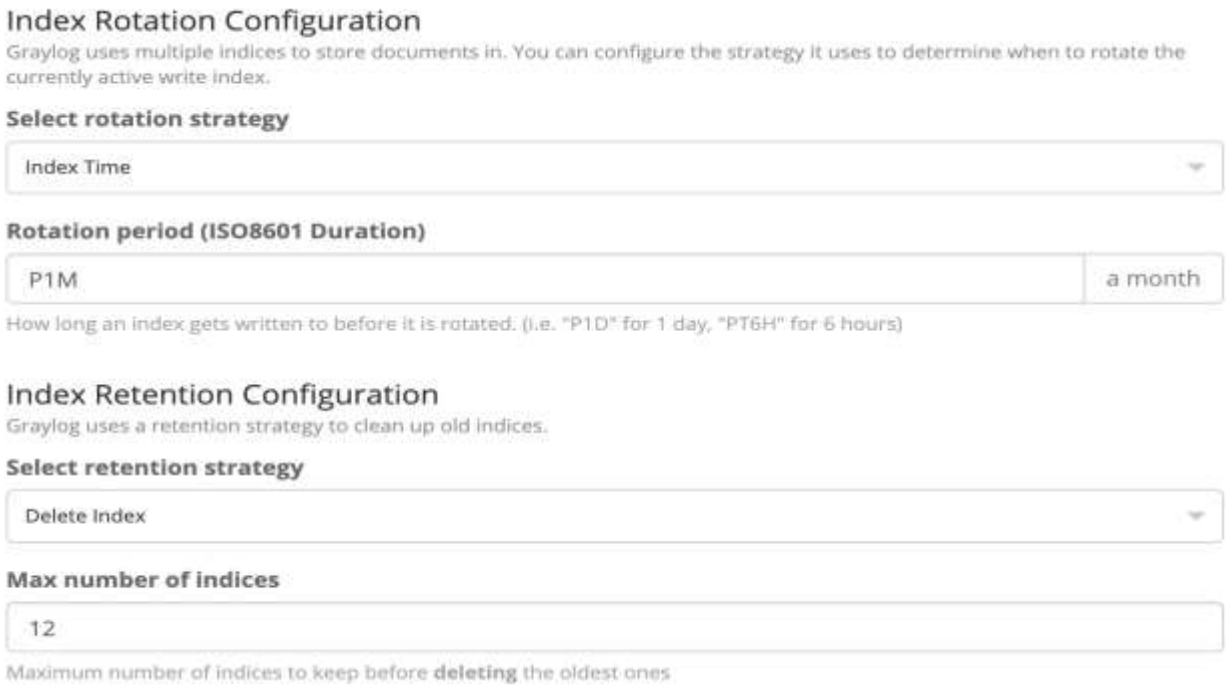
You have Gigabytes or Terabytes of logs coming in on a daily basis, but now what do you do with them? Should I keep 10 days, 30 days or 1 year? How do I rotate around my logs and configure them in Graylog? Let's talk about the best practices around log retention and how to configure them in Graylog.

Log rotation can be done for various reasons ranging from meeting a compliance goal, keeping the size of the index down for faster searches or to get rid of data after a set amount of time. Graylog enables you to rotate the indexes based on a few methods. Message count, will rotate the index after a number of messages have been written into the index. Index size rotates the index after the size defined has been reached and Index time rotates the index after the specified time. All have their uses, with the most common being index time. With index time, you can meet the three months online, but

setting the index to rotate every week, and keeping 12 of them online, or every day and setting the max indices to 90.

INDEX ROTATION

To get into your index rotation strategy you need to go to: System -> Indices and select Edit next to the index you would like to modify. In this example below, we have a 1-month rotation and are keeping 12 indexes for a full year of data.



After you set your rotation strategy, you will also need to select your retention configuration. Let’s have a quick overview of each choice and why you might want to pick that option.

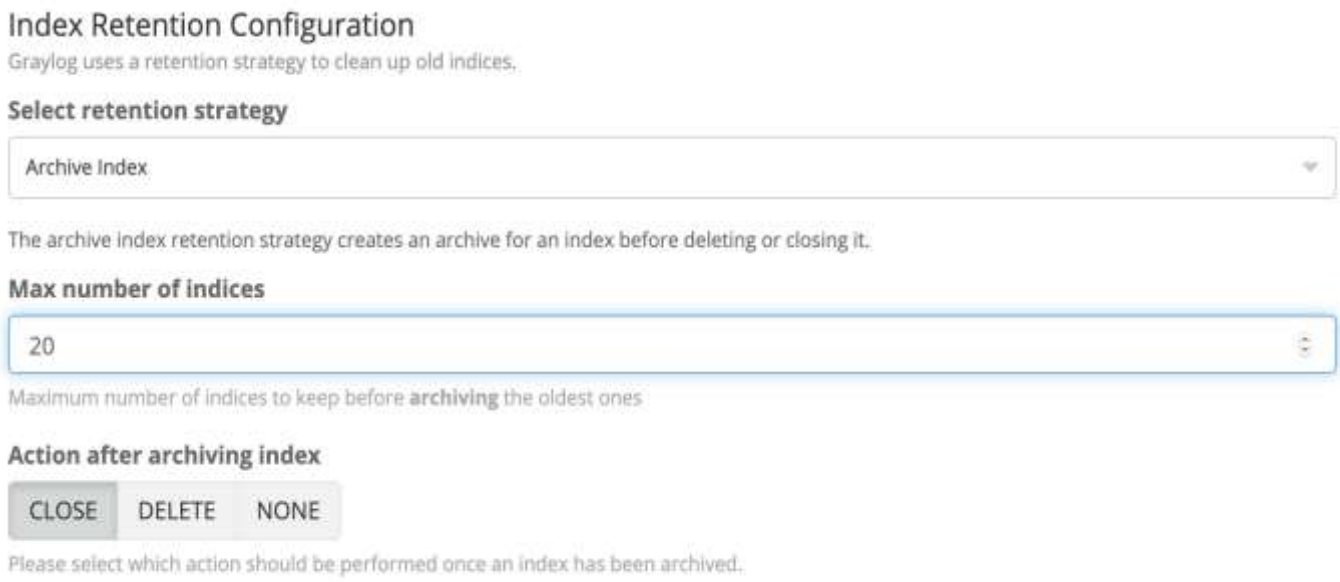
● Delete: When you delete the indices you are having the minimal resource consumption by Elasticsearch and removes the index from disk thus saving disk space. This would be a good setting for operational data, which after a couple of weeks has no value (System Metrics, Flow Data, etc.)

● Close: Closing an index, blocks Elasticsearch from writing more data into it, but keeps it online and maintains the index’s metadata so you can still search it.

● Do Nothing: No resource saving on elasticsearch, and will keep the index open and on disk until manual removal.

ARCHIVING

If you would like to archive your data you can use the Enterprise version of Graylog to set up a backend storage location, allowing older indexes to be moved and compressed for long term storage. After you have a mount location on your server and configured that as the storage location of your archives, you can then set up the “Archive Index” retention strategy and give the number for when it will start moving them to the archive. In this example, the 21st index will be moved to the archive location, and the index closed. Many will choose to delete the index after archiving, as the archive feature allows for restoration if needed.



CONCLUSION

Understanding your log retention and rotation strategy is essential in any deployment of Graylog. With correct rotation strategies, your logs will be collected and maintained as expected, and allow for a fast and useful log aggregation tool.

Log archival: Storing log data to address future requirements.

Log data is vital information that contains records about events that have happened in a network. Log data is essential to monitor the network and understand the network activities, user actions, and their motives.

As every device in the network generates logs, the amount of data collected is huge, and managing and storing all this data becomes a challenge. Log archiving is a process that helps administrators use available storage efficiently.

Why archiving log data is important.

- Meet regulatory standards

  Most compliance regulations compel enterprises to retain log data for at least a year to facilitate forensic analysis. For instance, section 802 of SOX requires organizations to archive their data for at least seven years.

- Identify patterns and trends

  When log data spanning a longer period of time is archived, it can be loaded back into an analytical solution to identify network activity trends and patterns. These trends and patterns support the design and implementation of preventive security strategies.

- Optimize log data storage

  Archiving log data by employing compression techniques, as well as storing archived logs in a location that doesn't need to be optimized for quick access, are two good ways to save storage space and reduce costs. Furthermore, since the data can be decompressed and loaded into active databases any time without any data loss, it can still be used for forensic analysis or any other operation easily.

Managing Nagios logs

Nagios natively supports log rotation, a functionality managed using `log_rotation` main configuration option. This is the configuration option description taken from official nagios documentation

Format:*log_rotation_method=[n/h/d/w/m]*
Example*:log_rotation_method=d*

This is the rotation method that you would like Nagios to use for your log file. Values are as follows:

n = None (don't rotate the log - this is the default)
h = Hourly (rotate the log at the top of each hour)
d = Daily (rotate the log at midnight each day)

w = Weekly (rotate the log at midnight on Saturday)
m = Monthly (rotate the log at midnight on the last day of the month)

Many times people become confused by the Nagios log management capabilities and believe that, besides rotating, Nagios will erase older files. this is not real neither in Nagios nor in Centreon systems and older logs remain in our disk for months or even years.

This simple script can be very helpful in order to address the previous fact. It manages log files in two combinable ways: Compressing and or deleting files older than x days. It takes three arguments:

- Directory where nagios logs are stored
- Age, in days, for files that will be compressed
- Age, in days, for files that will be deleted

For instance, and given that it is named as manage_naglogs, this example would delete files older than 30 days and would compress files older than 7 days:

```
manage_naglogs /var/log/nagios 7 30
```

And here comes the script:

```
#!/bin/bash


if [ $3 -gt 0 ]
then
        find $1/nagios-*.gz -mtime +$3 -exec rm {} \;
        find $1/nagios-*.log -mtime +$3 -exec rm {} \;
fi
if [ $2 -gt 0 ]
```

```
then

        find $1/nagios-*.log -mtime +$2 -exec gzip {} \;

fi
```

In order to run it periodically, I recommend adding the needed commands to cron. In systems like Debian where /etc/cron.daily stores scripts run every day, and assuming you have saved the previous script in /usr/local/nagios/bin, create an script like this, save it in /etc/cron.daily and set proper file permissions for being run for cron daemon (chmod 755 will do the job):

```
#!/bin/bash

/usr/local/nagios/bin/manage_naglogs /var/nagios/logs 7 30
```

In systems where only crontab is available, next entry will do the job. It will run our script once every day at 3:00am:

```
00 3 * * * root /usr/local/nagios/bin/manage_naglogs /var/nagios/log 7 30
```

Finally for Centreon: Keep, at least, one nagios rotated log file untouched (ie, neither compressed nor deleted). Have in mind that centreon run every day (usually at 1:00am) an script for parsing Nagios log files in order to create availability reports. To achieve it, use values higher than 1 for the second and third script arguments.

LOG ROTATION USING LINUX COMMAND-LINE UTILITY LOGROTATE

Let us consider a scenario when all the service and custom logs of your ubuntu server gets backed up on S3 on a daily basis after compressing them and the allocated space is freed without using any third party software.

We can achieve this task by the use of logrotate utility provided by the ubuntu server.

Log rotation means the task of archiving any application's or system's current log, starting a fresh log, and deleting older logs.Log files left unrotated can lead to the filling of disk space which can be an alarming situation for the server.You can backup the archived logs on to your S3 bucket. Log rotation helps the system to ensure that the useful space is getting utilized in an appropriate manner.

Here is a demo which will help your log files to get rotated in an effective manner and also ensure safe backup of these archived logs on to your S3 bucket.

Step 1 :- **Installing logrotate utility**

Log rotation is a utility of ubuntu and might be preinstalled in your ubuntu server (/etc/logrotate.d).If it is not installed then you can install it manually by the following commands.

sudo apt-get update

sudo apt-get install logrotate

Step 2 :- **Configuring Logrotate**

There are two ways to configure logrotate.

1. By making changes in default global configuration file /etc/logrotate.conf for all services :- The main configuration file is nicely commented and you can trim this file to your needs.You can also make specified blocks for your application in the file.The attributes of logrotate are described below.By default,, logrotate reads from this configuration file if you haven't made any specific configuration in the directory /etc/logrotate.d .The configuration in logrotate.conf applies to all the service logs.If you want to set different options on the basis of the applications, then go for option 2.

2. By making individual configuration file in directory /etc/logrotate.d/ for each service and application .

This will illustrate how you can configure the logrotate for efficient use by going through option 2.This is the optimized way to manage your logs depending upon the type of application and its use.

In the directoy /etc/logrotate.d/, you may find many configuration files already present there.Generally you will find configuration files of those applications in this directory which are installed using package manager.

Lets take an example of a configuration file in /etc/logrotate.d/apache2 . If there is no file in this directory, you can create one.But if you have apache2 installed in your system using package manager (apt-get), you must have this file.Below is my apache configuration file for logrotation :-

```
/var/log/apache2/*.log {
weekly
missingok
rotate 4
compress
delaycompress
notifempty
create 640 root adm
sharedscripts
postrotate
  aws s3 sync /var/log/apache2/ s3://<Your-bucket-Name>/<Server-Name>/<Apache2-
logs>/ --region <your-bucket-region>
endscript
}
```

Note :- In the above file you can also create specify more than one log directory.For example, /var/log/apache2/*.log /var/<app-directory>/*.log and then the block can be started for the configuration file.The configuration will be applicable for both the log directories.

The attributes are described below :-

- **1. Rotate Interval :-** This command tells logrotate that how often the logs will be rotated.It basically specifies the time interval after which the logrotation is done.

The options can be :-

1. daily
2. weekly
3. monthly
4. yearly

If you want to set some other option, for example say hourly then you have to make some adjustments.If you want to rotate logs on per hour basis, then you need to create a directory /etc/cron.hourly and make a new file in it which will contain the following line :-

/usr/sbin/logrotate /etc/logrotate.d/<name of configuration file for which log rotation is done hourly>

If the rotation interval is not specified then the logs are rotated meeting to some other condition like size exceeding 100k, 100M, 100G.

- **2. Size :-** The size attribute tells logrotate to perform the operation when the size of the file becomes greater than the specified size.
    1. size 100k :- The logs are rotated when the size becomes 100 kilobytes
    2. size 100M :- The logs are rotated when the size becomes 100 Megabytes
    3. size 100G :- The logs are rotated when the size becomes 100 Gigabytes
- **3. Rotate Count :-** The rotate count specifies how many archived log files will be kept around before the logrotate starts deleting the older files.
    1. Rotate 4

This will specify that, at a time 4 archived files will be kept by logrotate and when there are four archived files already present then the oldest one named "<log-file-name>.4 " will be deleted to make space for the newly created archive.

- **4. Missingok :-** This attribute tells that if the log file is missing then logrotate will not throw any error and will jump to next one.

- **5. Compress :-** This command will archive log files in compressed (gzip format).It is generally a good idea to compress the files to save the useful space since log files are only text files so a level of compression can be

really helpful in proper utilization of the storage.If you don't want to compress the log files, you can simply write nocompress instead of compress in the configuration file.

- **6. Delaycompress :-** With delaycompress option active, the compression of the previous log files is postponed for the next rotation cycle.This option will work in combination with compress option and is really helpful when the application keeps writing logs to the previous log files.

- **7. Notifempty :-** This option tells logrotate that do not rotate the logs if empty.

- **8. create 640 root adm :-** This option specifies the permission of newly created log files.It specifies that the newly created log files will be having permission of read/write for owner root and only read for group adm.

- **9. postrotate :-** The postrotate script is run by the logrotate every time it rotates log in specified configuration block.The postrotate command specifies that the next line will be the starting line of script and the script will be ended when the endscript command is encountered.Here you can specifiy something like restarting apache server to switch to new log file.I am writing a command to sync the directory of my logs to have a backup on S3 bucket.You can also include your custom logic like making a date folder on S3 bucket to create dated backups of the logs.

- **10. sharedscripts :-** This option tells that the postrotate script will be run only once for multiple configuration files having same log directory.For example, apache2 has both error logs as well as access logs but the directory for both is same (/var/log/apache2/*.log) . In this case, if the sharedscripts tag is on then it will run postrotate script only once otherwise postrotate script will be run everytime the log rotation is done.

Similarly, configuration files for other services and applications can be made to manage their logs.In each file you can specify a new folder of your S3 bucket to manage logs according to the application name.

To start the logrotate service manually or to check the configuration for proper functioning, you can use the command :-

logrotate -v /etc/logrotate.d/apache2

where option -v means verbose so that you can view all the progress done by logrotate utility.

Logrotate utility provides an easy and efficient way to manage service logs as well as application logs.