

# **ASTRA LINUX**

## **GUIDE**

**LOGORATE**

## НАСТРОЙКА LOGROTATE

В Linux, большинство сервисов и программ, которые работают в фоне, таких как Apache, Nginx, Postfix и других записывают информацию о своем состоянии, результатах работы и ошибках в лог файлы. Стандартное расположение логов или как их еще называют - журналов - в папке /var/log.

С помощью анализа логов вы можете понять что работает не так, почему произошла ошибка и как решить возникшую проблему. Но тот кроется одна проблема. Размер логов постоянно растет и они занимают все больше и больше места на диске, поэтому необходимо вовремя чистить логи и удалять устаревшие записи, чтобы они не мешали нормально работать. Это можно делать вручную время от времени или настроить скрипты Cron, но есть еще более простой вариант - утилиты logrotate. В этой статье будет рассмотрена настройка logrotate и ее использование.

## КАК РАБОТАЕТ LOGROTATE?

Утилита Logrotate предназначена для автоматизации обработки журналов. Она может выполнять с ними необходимые действия в зависимости от определенных условий и правил соответствия. Например, можно сжимать журналы в архив или отправлять на другой сервер когда они достигают определенного размера, возраста, или других параметров.

Проверку условий можно настроить ежедневно, еженедельно или ежемесячно. Это позволяет создать схему ротации логов, удобную именно для вас и вашего сервера. Также ротация логов может быть полезна на домашнем компьютере, но здесь она не так важна как на серверах, где только в логи Apache могут записываться до сотен тысяч строк ежедневно.

## НАСТРОЙКА LOGROTATE

Logrotate - это популярная утилита, поэтому в большинстве дистрибутивов она поставляется по умолчанию. Вы можете убедиться, что программа установлена в вашем дистрибутиве, попытавшись ее установить. Например, в CentOS:

```
sudo apt install logrotate
```

Теперь, даже если утилита не была установлена, вы ее установите. Все основные настройки программы находятся в файле /etc/logrotate.conf, дополнительные настройки, касаемо правил и других возможностей могут быть размещены в папке /etc/logrotate.d/. Вы можете размещать все настройки logroatae прямо в основном конфигурационном файле, будет более правильно, если настройки для каждого отдельного сервиса будут находиться в отдельном файле, в папке /etc/logrotate.d/.

Чтобы конфигурационные файлы из этой папки загружались программой, необходимо добавить в основной конфигурационный файл такую строчку:

```
vi /etc/logrotate.conf
```

```
include /etc/logrotate.d
```

Просто убедитесь что она там уже есть. Сначала давайте рассмотрим основные директивы, которые мы будем применять во время настройки. Здесь директивы выглядят не совсем обычно, сама директива и определяет что и когда нужно делать, а уже если нужно, ей передаются дополнительные параметры. Чтобы указать как часто нужно выполнять проверку совпадению условий используются такие директивы:

- hourly - каждый час;
- daily - каждый день;
- weekly - каждую неделю;
- monthly - каждый месяц;
- yearly - каждый год.

Основные директивы управления и обработки логов:

- rotate - указывает сколько старых логов нужно хранить, в параметрах передается количество;
- create - указывает, что необходимо создать пустой лог файл после перемещения старого;
- dateext - добавляет дату ротации перед заголовком старого лога;
- compress - указывает, что лог необходимо сжимать;
- delaycompress - не сжимать последний и предпоследний журнал;
- extension - сохранять оригинальный лог файл после ротации, если у него указанное расширение;
- mail - отправлять Email после завершения ротации;
- maxage - выполнять ротацию журналов, если они старше, чем указано;
- missingok - не выдавать ошибки, если лог файла не существует;
- olddir - перемещать старые логи в отдельную папку;
- postrotate/endscript - выполнить произвольные команды после ротации;
- start - номер, с которого будет начата нумерация старых логов;
- size - размер лога, когда он будет перемещен;

Это те основные директивы, которые мы будем использовать. В главном конфигурационном файле находится глобальная конфигурация, директивы, которые будут распространяться на все логи если не было отменено их действие. Каждый лог, который подлежит ротации описывается таким образом:

```
адрес_файла_лога {  
директивы  
}
```

Теперь давайте создадим файл rsyslog.conf в папке /etc/logrotate.d/ и поместим в него настройки для ротации этого лога:

```
/var/log/messages {  
daily  
rotate 3  
size 10M  
compress  
delaycompress  
}
```

Эти настройки означают, что ротация журналов будет выполняться ежедневно, и мы будем хранить три последних журнала, более старые копии будут автоматически удаляться. Минимальный размер для ротации - 10 мегабайт, ротация не будет выполнена, если лог не занимает более 10 мегабайт. Будет использоваться сжатие, для всех журналов кроме последнего и предпоследнего. Точно по такому же принципу вы можете настроить ротацию логов для любого из журналов. Нужно создать такую секцию для каждого из логов, которыми вы хотите управлять.

Теперь осталось протестировать как работает наша конфигурация. Для этого запустим утилиту logrotate с опцией -d. Она выведет все, что планируется сделать, но не будет изменять файлы на диске. У нас есть файл /var/log/messages, размером 40 Мегабайт, посмотрим что будет делать утилиты:

```
logrotate -d /etc/logrotate.d/rsyslog.conf
```

Как видите, программа обнаруживает файл лога и разделяет его на несколько частей. Вы можете убедиться, что logrotate будет запускаться как положено проверив расписание cron:

```
ls /etc/cron.daily/
```

Настройка Logrotate завершена, а вам осталось всего лишь расписать как будет выполняться ротация логов для каждого из журналов, которые занимают много места.

## CRON

Cron (Command Run ON) — система для автоматического запуска программ и скриптов на сервере в определённое время.

Управлять cron нужно с помощью команды 'crontab'.

Команда crontab -l покажет список текущих заданий, crontab -e автоматически запустит текстовый редактор и загрузит в него файл конфигурации cron. После выхода из редактора, новая конфигурация cron будет установлена (вступит в силу) автоматически.

Конфигурационный файл содержит последовательность командных строк и расписание их вызова. Пустые строки и строки, начинающиеся с символа '#' игнорируются. Остальные строки являются установками переменных окружения и командами cron.

Запись crontab- файла в общем случае состоит из семи полей:

минуты часы день месяц день\_недели имя\_пользователя команда

Допустимые значения временных параметров:

Параметр Допустимый интервал

минуты 0-59

часы 0-23

день месяца 1-31

месяц 1-12

день недели 0-7 (0-Вс,1-Пн,2-Вт,3-Ср,4-Чт,5-Пт,6-Сб,7-Вс)

Поле может быть задано явно или шаблоном:

\* — любая цифра;

целое число;

целые числа через запятую — задание дискретного множества значений, например 1,2,5;

два целых числа, разделенные дефисом, соответствующие диапазону значений, например 3-6.

Пример готовой строки сценария cron:

```
# Выполнять задание в 18 часов 7 минут 13 мая если это пятница  
7 18 13 5 5 /home/www/myscript.pl
```

```
# Выполнять задание раз в час в 0 минут
```

```
0 */1 * * * /home/www/myscript.pl
```

```
# Выполнять задание каждые семь часов в 0 минут
```

```
0 */7 * * * /home/www/myscript.pl
```

```
# Выполнять задание по воскресеньям в 10 час 30 минут
```

```
30 10 * * 0 /home/www/myscript.pl
```

Выполнение PHP-файлов

Для выполнения php скриптов из cron необходим бинарный интерпретатора PHP.

Реальный путь на сервере к корневой директории сайта и путь к PHP уточните в службе поддержки хостинга.

```
# Пример запуска скрипта в 5 утра каждый день:
```

```
0 5 * * * /usr/bin/php -q /home/www/script.php
```

Обратите внимание, при запуске из cron опция -q указывает PHP не выдавать HTTP-заголовки.