rsyslog

Rsyslog – это мощная, безопасная и высокопроизводительная система обработки логов, принимающая данные из различных источников (систем и приложений) и выдающая их в разнообразных форматах. Она представляет собой развитие обычного демона syslog до полнофункциональной системы ведения логов корпоративного уровня. **Rsyslog** работает по модели **«клиент-сервер»**, поэтому ее можно настроить как клиент и/или сервер для централизованного ведения логов других серверов, сетевых устройств и удаленных приложений.

Установка

Для RHEL/CentOS:

```
$ yum install rsyslog
```

Для Ubuntu/Debian:

```
$ apt install rsyslog
```

После установки **rsyslog** нужно запустить службу, активировать автоматический запуск при загрузке и проверить состояние при помощи команды **systemctl.**

```
$ sudo systemctl start rsyslog
$ sudo systemctl enable rsyslog
$ sudo systemctl status rsyslog
```

Связанные файлы

- **/etc/rsyslog.conf** - главный файл конфигурации
- **/etc/rsyslog.d** - директория для хранения файлов конфигурации для различных служб и приложений

Отношения с фаерволами

- Если у вас включена служба **SELinux**, нужно выполнить следующие команды, чтобы разрешить трафик **rsyslog**:

  ```
  $ sudo semanage -a -t syslogd_port_t -p udp 514
  $ sudo semanage -a -t syslogd_port_t -p tcp 514
  ```

- Для **CentOS** (брандмауэр **firewalld**):

  ```
  $ sudo firewall-cmd --permanent --add-port=514/udp
  $ sudo firewall-cmd --permanent --add-port=514/tcp
  $ sudo firewall-cmd --reload
  ```

- Для **Ubuntu** (брандмауэр **ufw**):

```
$ sudo ufw allow 514/udp
$ sudo ufw allow 514/tcp
$ sudo ufw reload
```

Правила

```
источник.уровень_важности   место_записи_лога
```

, где:

- **источник:** тип процесса или приложения, от которого исходит сообщение, значение может быть **auth**, **cron**, **daemon**, **kernel**, **local0..local7**. Использование звездочки (*) означает все источники.
- **уровень_важности:** тип сообщения логов: emerg-0, alert-1, crit-2, err-3 и др. Использование звездочки (*) означает все уровни важности, если ничего не указывать, предполагается отсутствие уровня важности.
  - **0, emerg** – система не работоспособна
  - **1, alert** – система требует немедленного вмешательства
  - **2, crit** – состояние системы критическое
  - **3, err** – сообщение об ошибке
  - **4, warning** – предупреждение о возможной проблеме
  - **5, notice** – нормальное, но важное событие
  - **6, info** – информационное сообщение
  - **7 , debug** – отладочное сообщение
- **место_ записи_лога:** локальный файл или удаленный сервер rsyslog (определенный в формате IP-адрес:порт).

Примеры

1. Для сбора логов удаленных узлов мы будем использовать следующий набор правил с шаблоном **RemoteLogs**. Обратите внимание, что эти правила должны предшествовать правилам обработки локальных сообщений.

```
$template RemoteLogs,"/var/log/%HOSTNAME%/%PROGRAMNAME%.log"
*.* ?RemoteLogs
& ~
```

1) **«$template RemoteLogs,”/var/log/%HOSTNAME%/%PROGRAMNAME%.log”»**. Директива *$template* дает демону rsyslog команду собирать полученные сообщения из источников и записывать их в отдельные логи в директории `/var/logs` в соответствии с именем узла (машины клиента) и источником (программой/приложением), от которых были получены сообщения, что определено соответствующим шаблоном.

2) Вторая строка **«*.* ?RemoteLogs»** означает запись сообщений всех уровней важности от всех источников в соответствии с шаблоном *RemoteLogs*.

3) Последняя строка **«& ~»** задает rsyslog прекратить обработку сообщений после их записи в файл. Если не указать «& ~», сообщения будут записаны в локальные файлы.
Настройка сервера для нашего примера завершена. Теперь нужно сохранить и закрыть файл конфигурации, а также перезапустить демон rsyslog, чтобы изменения вступили в силу:

```
$ sudo systemctl restart rsyslog
```

Настройка сервера

1. Для сбора удаленных логов необходимо включить (раскомментировать) прослушивание на порту 514 для протоколов UDP и/или TCP

```
$ModLoad imudp
$UDPServerRun 514

$ModLoad imtcp
$InputTCPServerRun 514
```

2. Создать необходимое правило на основе примера согласно указанной выше конструкции

Настройка клиента

1. Чтобы демон **rsyslog** работал как клиент и отправлял все локальные логи на **удаленный сервер rsyslog**, добавьте следующее правило перенаправления в конце файла.

   Чтобы использовать **UDP**, поставьте перед IP-адресом **одинарный знак @**. Чтобы использовать **TCP**, поставьте перед ним **два знака @** (@@). Номер порта должен соответствовать номеру порта, прописанному в конфигурации сервера:

```
*. * @@192.168.100.10:514
```

   Приведенное правило будет отправлять сообщения всех уровней важности от всех источников.

2. Для отправки сообщений от конкретного источника, например, **auth**, воспользуйтесь следующим правилом:

```
auth. * @@192.168.100.10:514
```

3. Сохраните и закройте файл, а также перезагрузите службу rsyslog чтобы изменения вступили в силу.

```
$ sudo systemctl restart rsyslog
```

Проверка

Последний этап – проверить, действительно ли rsyslog получает сообщения от клиента и сохраняет их в директории `/var/log` и формате имя_ узла /имя_ программы.log.
Выполните команду ls, чтобы получить список файлов директории логов и проверьте, есть ли там директории под названием ip-172.31.21.58 (или с соответствующим именем узла вашего клиента).

```
$ ls -l /var/log/
```

Если директория существует, проверьте файлы логов в ней следующей командой:

```
$ sudo ls -l /var/log/ip-172-31-21-58/
```

Источники:

- https://itproffi.ru/ustanovka-i-nastrojka-rsyslog-v-linux/
- https://www.dmosk.ru/miniinstruktions.php?mini=rsyslog
- https://www.k-max.name/linux/rsyslog-na-debian-nastrojka-servera/
- https://losst.ru/nastrojka-rsyslog-v-linux