# 日志轮换工具Logrotate

## 一、Logrotate 概述

**定义**：Logrotate 是Linux系统中用于自动化管理日志文件的工具，通过轮换（rotation）、压缩（compression）、删除旧日志和归档，防止日志文件过大占用磁盘空间。它广泛用于系统日志（如`/var/log/syslog`）、Web服务器日志（如Nginx、Apache）以及应用程序日志。

**核心功能**：

- 按时间（每日、每周、每月）或大小触发日志轮换。
- 压缩旧日志（通常为`.gz`格式）。
- 删除过期日志，控制磁盘使用。
- 执行轮换前/后脚本（如重启服务）。
- 设置新日志文件的权限和所有者。

**运行机制**：

- 通常由`cron`调度（如`/etc/cron.daily/logrotate`）自动运行。
- 配置文件：
  - 主配置文件：`/etc/logrotate.conf`（全局设置）。
  - 子配置文件：`/etc/logrotate.d/`（特定服务配置，如`/etc/logrotate.d/nginx`）。
- 状态文件：`/var/lib/logrotate/status`，记录上次轮换时间。

---

## 二、Logrotate 常用用法

以下是Logrotate的常见配置和操作，适合大多数日志管理需求。

### 1. 基本配置结构

Logrotate的配置文件使用简单的键值对和指令，格式如下：

```
/path/to/logfile {
    directive1
    directive2
    ...
}
```

- `/path/to/logfile`：指定要轮换的日志文件（支持通配符，如`*.log`）。

- `directive`：轮换规则，如轮换频率、压缩、权限等。

**常用指令**：

- **轮换频率**：
  - `daily`：每天轮换。
  - `weekly`：每周轮换。
  - `monthly`：每月轮换。
  - `size <大小>`：当日志达到指定大小（如`100M`、`1G`）时轮换。
- **轮换数量**：
  - `rotate <N>`：保留N个旧日志文件。
- **压缩**：
  - `compress`：压缩旧日志（默认使用gzip）。
  - `nocompress`：不压缩。
- **文件处理**：
  - `missingok`：忽略缺失的日志文件，不报错。
  - `notifempty`：空日志不轮换。
  - `create <模式> <用户> <组>`：轮换后创建新日志，指定权限、所有者和组。
  - `nocreate`：不创建新日志。
- **脚本执行**：
  - `prerotate`：轮换前执行脚本。
  - `postrotate`：轮换后执行脚本。

### 2. 简单配置示例

假设你要管理Web服务器的日志`/var/log/nginx/access.log`，每天轮换，保留7天日志，压缩旧日志，并设置新日志权限为`0640`，所有者为`www-data`，组为`adm`。

```
/var/log/nginx/access.log {
 daily
 rotate 7
 compress
 missingok
 notifempty
 create 0640 www-data adm
 postrotate
 /usr/sbin/nginx -s reload
 endscript
}
```

**解释：**

- `daily`：每天轮换日志。
- `rotate 7`：保留最近7天的日志（access.log.1.gz 到 access.log.7.gz）。
- `compress`：旧日志压缩为 `.gz` 格式。
- `missingok`：如果日志文件不存在，不报错。
- `notifempty`：空日志不轮换。
- `create 0640 www-data adm`：新日志权限为 `rw-r-----`，所有者为 `www-data`，组为 `adm`。
- `postrotate`：轮换后重载Nginx服务，确保新日志生效。

### 3. 运行Logrotate

- **自动运行**：Logrotate通常通过 `cron` 调度运行，默认位于：
  - `/etc/cron.daily/logrotate`（每天运行）。
  - 配置文件路径：`/etc/logrotate.conf` 和 `/etc/logrotate.d/*`。
- **手动运行**：

```
logrotate /etc/logrotate.conf # 正常运行
logrotate -f /etc/logrotate.conf # 强制轮换（忽略状态文件）
logrotate -d /etc/logrotate.conf # 调试模式（模拟运行，不实际执行）
```

- **查看状态**：

```
cat /var/lib/logrotate/status # 查看上次轮换时间
```

### 4. 常见场景与配置

**场景1：管理系统日志**
管理 `/var/log/syslog`，每周轮换，保留4周日志，压缩旧日志：

```
/var/log/syslog {
 weekly
 rotate 4
 compress
 missingok
 notifempty
 create 0640 root adm
}
```

**场景2：管理应用程序日志**
管理自定义应用日志 `/var/log/myapp/app.log`，按大小（100MB）轮换，保留10个日志：

```
/var/log/myapp/app.log {
 size 100M
 rotate 10
 compress
 missingok
 create 0640 myuser mygroup
 postrotate
 /bin/kill -HUP `cat /var/run/myapp.pid`
 endscript
}
```

**与权限管理结合**：

- Logrotate以 `root` 用户运行，新日志的权限由 `create` 指令控制。
- 确保日志文件的所有者和组与服务进程匹配（如Nginx的 `www-data`），避免权限问题。
- 示例：检查日志权限：

```
ls -l /var/log/nginx/access.log
# 输出：-rw-r----- 1 www-data adm 12345 May 28 08:44 access.log
```

**5. 验证与调试**

- **验证配置**：

```
logrotate -d /etc/logrotate.d/nginx # 模拟运行，检查配置错误
```

- **手动触发轮换**：

```
logrotate -f /etc/logrotate.d/nginx  # 强制轮换Nginx日志
ls /var/log/nginx/  # 检查生成的文件（如access.log.1.gz）
```

- **查看日志**：

```
zcat /var/log/nginx/access.log.1.gz  # 查看压缩日志内容
```

---

## 三、Logrotate 进阶用法

以下是Logrotate的高级功能，适用于复杂场景，如多日志管理、自定义压缩、邮件通知等。

### 1. 高级配置指令

- **压缩选项**：
  - `compresscmd <命令>`：自定义压缩工具（如`bzip2`、`xz`）。

    ```
    compresscmd /usr/bin/bzip2
    compressext .bz2
    ```

  - `delaycompress`：延迟压缩到下次轮换（适合服务持续写入日志）。

    ```
    delaycompress
    ```

    - 效果：生成`access.log.1`（未压缩），下次轮换压缩为`access.log.2.gz`。
- **邮件通知**：
  - `mail <email>`：将轮换的日志发送到指定邮箱。
  - `mailfirst` / `maillast`：发送第一个或最后一个轮换日志。
  - 示例：

    ```
    mail admin@example.com
    maillast
    ```

- **轮换时间控制**：

- `dateext`：在轮换文件名中添加日期（如`access.log-20250528.gz`）。
- `dateformat <格式>`：自定义日期格式。

```
dateext
dateformat -%Y%m%d
```

- **共享脚本**：
  - `sharedscripts`：多个日志文件共享`prerotate`/`postrotate`脚本，减少重复执行。

```
sharedscripts
postrotate
    /usr/sbin/nginx -s reload
endscript
```

- **最大文件年龄**：
  - `maxage <天数>`：删除超过指定天数的旧日志。

```
maxage 30
```

- **轮换后复制**：
  - `copytruncate`：复制日志内容后截断原文件（适合无法重启的服务）。

```
copytruncate
```

**2. 复杂配置示例**

**场景：管理多日志文件的高并发服务**
管理Nginx的`access.log`和`error.log`，按大小（50MB）轮换，保留14天日志，使用bzip2压缩，延迟压缩，添加日期后缀，并发送邮件通知。

```
/var/log/nginx/*.log {
 size 50M
 rotate 14
 compress
 compresscmd /usr/bin/bzip2
 compressext .bz2
 delaycompress
 dateext
 dateformat -%Y%m%d
 missingok
 notifempty
 create 0640 www-data adm
 sharedscripts
 postrotate
 /usr/sbin/nginx -s reload
 endscript
 mail admin@example.com
 maillast
}
```

**解释：**

- `size 50M`：日志超过50MB触发轮换。
- `compresscmd /usr/bin/bzip2`：使用bzip2压缩。
- `delaycompress`：延迟压缩，生成未压缩的`access.log.1`，下次轮换压缩为`access.log.2.bz2`。
- `dateext`：文件名带日期（如`access.log-20250528.bz2`）。
- `sharedscripts`：对所有日志文件只执行一次`postrotate`脚本。
- `mail admin@example.com`：发送最后一个轮换日志到邮箱。

### 3. 自定义Cron调度

- 默认Logrotate通过`/etc/cron.daily/logrotate`每天运行，但可自定义调度：
  - **每小时运行**：编辑`/etc/crontab`：

    ```
    0 * * * * root /usr/sbin/logrotate /etc/logrotate.conf
    ```

  - **特定时间运行**：

```
30 2 * * * root /usr/sbin/logrotate /etc/logrotate.conf # 每天2:30运行
```

### 4. 日志轮换与权限管理

- **动态权限设置**：
  - 使用`create`指令确保新日志文件具有正确权限：

    ```
    create 0640 www-data adm
    ```

  - 如果服务以非root用户运行（如Nginx的`www-data`），确保Logrotate生成的文件可被服务写入：

    ```
    chown www-data:adm /var/log/nginx/access.log
    chmod 640 /var/log/nginx/access.log
    ```

- **ACL支持**：
  - 如果需要为特定用户添加额外权限，使用`setfacl`：

    ```
    setfacl -m u:monitor:rx /var/log/nginx/access.log
    ```

- **SUID/SGID/Sticky Bit**：
  - 通常不建议对日志文件设置特殊权限，但可用于共享目录：

    ```
    chmod 1777 /var/log/myapp # 设置Sticky Bit，防止非所有者删除
    ```

### 5. 错误排查与监控

- **检查配置错误**：

  ```
  logrotate -d /etc/logrotate.d/nginx # 调试模式，检查语法
  ```

- **监控日志轮换**：

  ```
  tail -f /var/log/syslog | grep logrotate # 查看Logrotate运行日志
  ```

- **磁盘空间检查**：

```
df -h /var/log # 检查日志目录磁盘使用
du -sh /var/log/nginx/* # 查看日志文件大小
```

### 6. 进阶场景示例

**场景：管理数据库日志**
管理MySQL慢查询日志 `/var/log/mysql/slow.log`，按大小（200MB）轮换，保留5个日志，延迟压缩，使用xz压缩，并重启MySQL服务。

```
/var/log/mysql/slow.log {
 size 200M
 rotate 5
 compress
 compresscmd /usr/bin/xz
 compressext .xz
 delaycompress
 missingok
 notifempty
 create 0640 mysql mysql
 postrotate
 /usr/bin/systemctl reload mysql
 endscript
}
```

**场景：多服务器日志同步**
在分布式系统中，将轮换后的日志复制到远程服务器：

```
/var/log/myapp/*.log {
 daily
 rotate 7
 compress
 missingok
 create 0640 myuser mygroup
 postrotate
 /usr/bin/rsync -avz /var/log/myapp/*.gz remote:/backup/logs/
 endscript
}
```

## 四、实际操作案例

1. **配置Nginx日志轮换：**
   - 创建配置文件：

     ```
     sudo nano /etc/logrotate.d/nginx
     ```

     写入上述Nginx配置（artifact_id: e8e616e0-d894-4936-a3f5-391682ee794c）。
   - 测试：

     ```
     logrotate -d /etc/logrotate.d/nginx
     ```
   - 手动轮换：

     ```
     logrotate -f /etc/logrotate.d/nginx
     ls -l /var/log/nginx/  # 检查access.log.1.gz
     ```
   - 验证权限：

     ```
     ls -l /var/log/nginx/access.log
     # 应显示：-rw-r----- 1 www-data adm ...
     ```
2. **清理过期日志：**
   - 修改配置，添加`maxage`：

     ```
     nano /etc/logrotate.d/myapp
     ```

     添加：`maxage 30`。
   - 强制运行：

     ```
     logrotate -f /etc/logrotate.d/myapp
     ```
3. **调试配置错误：**
   - 模拟运行：

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

- 检查错误日志：

```
tail /var/log/syslog
```

---

## 五、注意事项与最佳实践

1. **权限管理**：
   - 确保`create`指令设置的权限与服务用户匹配，避免服务无法写入日志。
   - 使用`chown`和`chmod`手动修复权限：

```
chown www-data:adm /var/log/nginx/*.log
chmod 640 /var/log/nginx/*.log
```

   - 对敏感日志（如`/var/log/auth.log`）设置严格权限（如`600`）。
2. **性能优化**：
   - 使用`delaycompress`避免高并发服务日志压缩时的性能问题。
   - 限制`rotate`数量和`maxage`，防止磁盘空间耗尽。
3. **错误预防**：
   - 定期检查`/var/lib/logrotate/status`，确保轮换正常运行。
   - 使用`missingok`和`notifempty`减少不必要的错误。
4. **备份与监控**：
   - 将压缩日志备份到远程服务器（如上述rsync示例）。
   - 使用监控工具（如Zabbix、Nagios）检查日志目录大小：

```
du -sh /var/log/*
```

5. **与文件系统交互**：

- 确保日志目录在支持的Linux文件系统（如ext4、XFS）上，避免权限或性能问题。
- 检查挂载选项（如`noatime`）以减少日志写入开销：

```
mount -o remount,noatime /var/log
```

---

## 六、补充说明

- **与其他工具对比**：
  - **Cronolog**：适合实时日志分割，但配置复杂，功能不如Logrotate灵活。
  - **Systemd-journald**：管理系统日志，但更适合短周期日志，长期归档仍需Logrotate。
  - **自定义脚本**：可通过shell脚本实现日志轮换，但维护成本高。
- **扩展阅读**：
  - 官方手册：`man logrotate`
  - 检查默认配置：`cat /etc/logrotate.conf`
- **云环境**：
  - 在云服务（如AWS、阿里云）中，日志可能由云服务管理（如AWS CloudWatch），但Logrotate仍可用于本地日志处理。