logrotate: configuration and operation

Rotation policy, retention, compression, state and systemd scheduling

What logrotate does

logrotate rotates and manages log files according to declarative policies. It can rename or copy logs, create replacement files, compress old logs, remove expired files and run scripts around rotation. It is not itself a logging daemon: applications and syslog services continue writing logs independently.

Most distributions have a main /etc/logrotate.conf and service-specific rules under /etc/logrotate.d/. Execution is typically scheduled by a systemd timer or cron job. A rule's frequency determines when rotation is eligible; it does not mean logrotate continuously watches the file.

```
cat /etc/logrotate.conf
ls /etc/logrotate.d
systemctl list-timers --all | grep logrotate
```

A complete rotation rule

The example rotates a service log daily, retaining 14 old rotations, compressing older files and skipping empty or absent logs. create establishes a new file after rotation with the specified mode and owner. These settings must match the service's permissions and reopen behaviour.

A rotation policy can combine time and size conditions. size is evaluated independently of time rules in a way that depends on directive ordering; minsize and maxsize provide more explicit combinations. Read the installed manual before mixing them.

```
/var/log/myapp/*.log {
    daily
    rotate 14
    missingok
    notifempty
    compress
    delaycompress
    create 0640 syslog adm
}
```

Rename versus copytruncate

The normal rename-and-create approach moves the old pathname and creates a fresh log. A process that keeps an open file descriptor may continue writing to the renamed file until it reopens its log. A postrotate script can signal a service to reopen the file.

copytruncate copies the existing file and then truncates it in place, useful for programs that cannot reopen logs. It has an unavoidable race between copying and truncation during which some log lines may be lost. Prefer proper reopen support when available.

```
postrotate
    systemctl kill -s HUP rsyslog.service 2>/dev/null || true
endscript
```

Testing and state

logrotate keeps state (often /var/lib/logrotate/status) to decide whether a file is due for rotation. Debug mode -d shows what would happen without changing files or state. Verbose mode -v provides operational detail. Forced mode -f rotates regardless of schedule and can change files, so reserve it for controlled troubleshooting.

Test a single service rule when possible rather than forcing the entire global configuration. Be aware that running logrotate manually with the production state file may affect the next scheduled run.

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

Troubleshooting and retention

If a file never rotates, check whether the rule matches the actual pathname, whether the service writes to a different log, whether the file is empty, and whether the timer runs. Permission errors, shared directory ownership and unsafe parent directories can also prevent rotation; the su directive can be required for certain ownership arrangements.

Retention is determined by rotate count, age and disk growth. Compression reduces storage but does not eliminate the need for capacity alerts. journalctl-managed binary journals use journald's own vacuum and retention controls, not ordinary text-file logrotate rules.

```
journalctl -u logrotate.service --since "7 days ago"
systemctl status logrotate.timer
man logrotate
```

Reference scope: commands and defaults vary by distribution and software version. Consult the locally installed manual pages for exact syntax and behaviour.