Logging and Monitoring

CIS 2235 Adv System Administration

Agenda

Logging

- syslog
- systemd journaling
- logrotate

Monitoring

What is logging?

- Most Linux services log….somewhere….somehow
- A log message is a line of text with some properties
  - timestamp
  - type of event
  - severity of event
  - pid and process name
  - message itself
- Could be a process start up message or a failure of a critical service
- Sys admins need to get useful information from these logs
  - Act when necessary!

Log management

Logs should give actionable information.

Log management:

- Collects log messages from a variety of sources
- Provides a structured interface for querying, filtering, monitoring and analyzing log messages
- Manages the retention and expiration of log messages
  - keep messages as long as they are helpful or legally required
  - remove messages no longer needed to conserve resources

syslog

historical log management on Unix is **syslog**

- sorts messages and saves them to appropriate log files, or
- can forward them to another host on the network

syslog only provides log collection, not analysis or monitoring

many applications bypass syslog and write to log files directly

Not much consistency across Unix and Linux distros

syslog logs are text files, and can be read and processed with standard Unix tools: grep, cat, less, awk

systemd

The systemd journal collects messages,

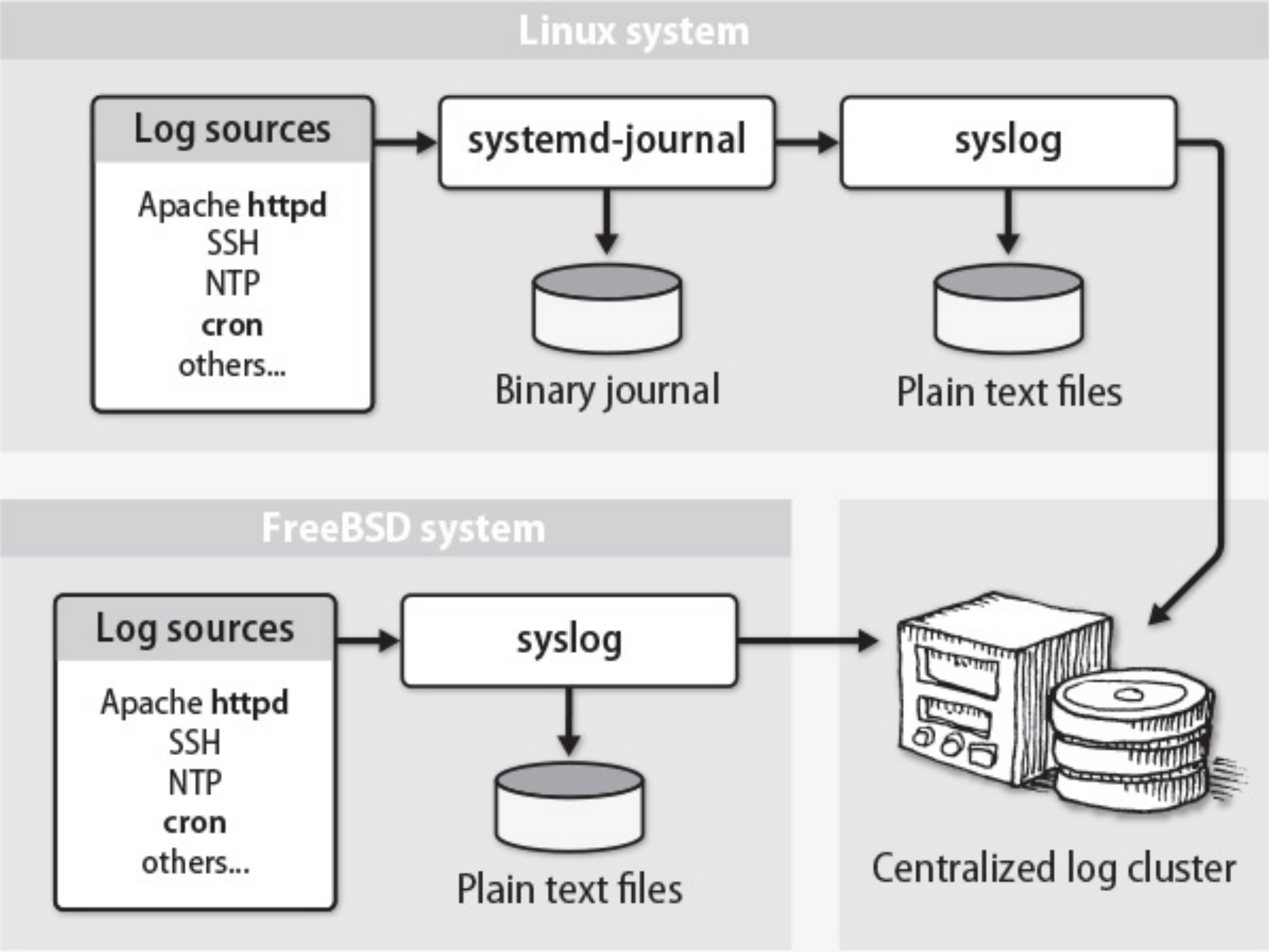
- stores them in indexed, compressed binary format
- has a command line utility for viewing/filtering
- can integrate with syslog

Both of these services can be combined

Use journalctl command to view the (binary) journal

```
journalctl -u ssh
journalctl -f
journalctl —disk-usage
journalctl -n 100 /usr/sbin/sshd
journalctl —help
```

Logging architecture



Log locations

Most apps put log files relative to /var/log

some apps write to other locations

note inconsistent naming

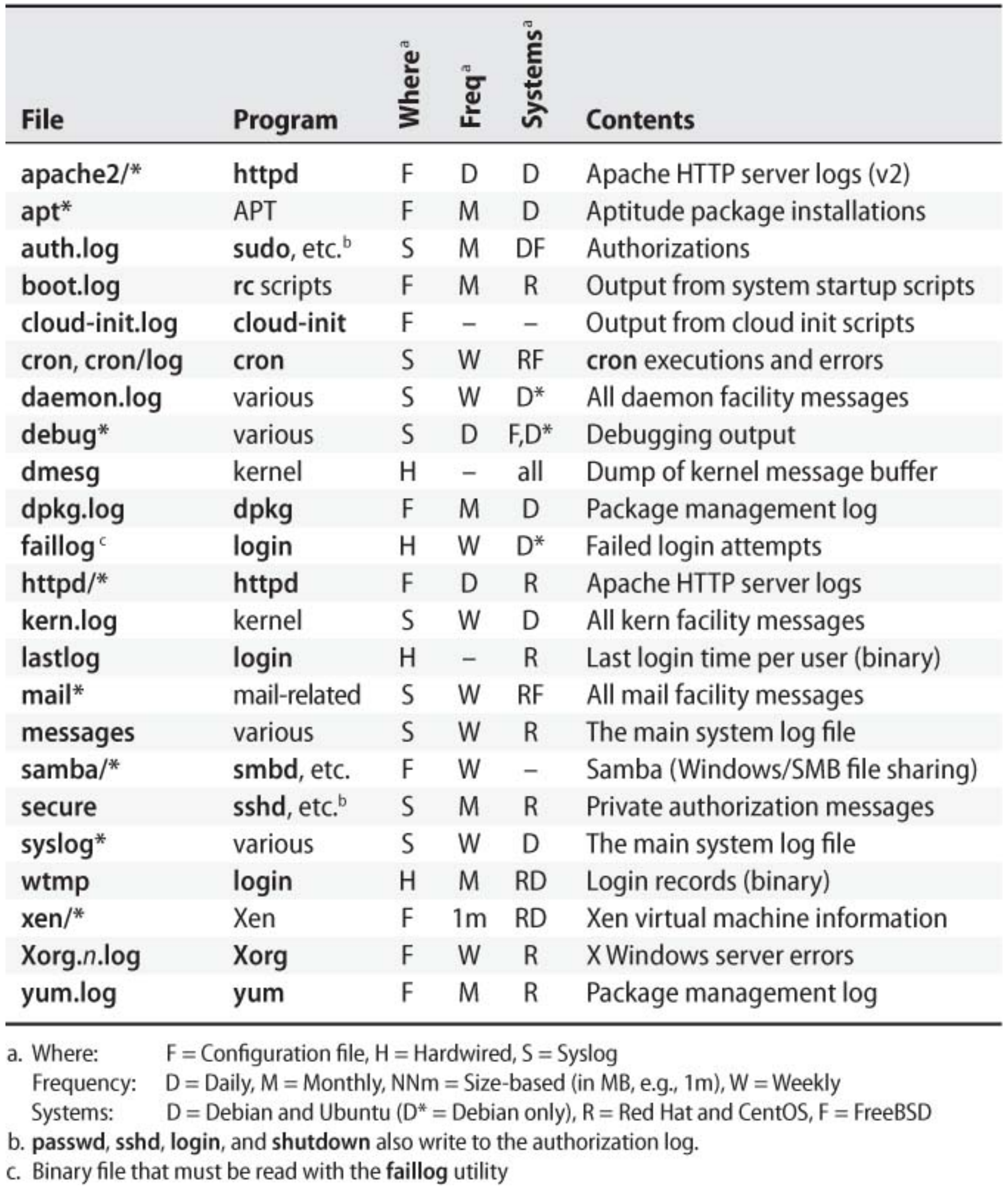
- faillog
- daemon.log
- dmesg

generally owned by root

add admins to the group systemd-journal for access to all files

log files can grow quickly, so monitoring disk space is important

Log locations



File	Program	Where[a]	Freq[a]	Systems[a]	Contents
apache2/*	httpd	F	D	D	Apache HTTP server logs (v2)
apt*	APT	F	M	D	Aptitude package installations
auth.log	sudo, etc.[b]	S	M	DF	Authorizations
boot.log	rc scripts	F	M	R	Output from system startup scripts
cloud-init.log	cloud-init	F	–	–	Output from cloud init scripts
cron, cron/log	cron	S	W	RF	cron executions and errors
daemon.log	various	S	W	D*	All daemon facility messages
debug*	various	S	D	F,D*	Debugging output
dmesg	kernel	H	–	all	Dump of kernel message buffer
dpkg.log	dpkg	F	M	D	Package management log
faillog[c]	login	H	W	D*	Failed login attempts
httpd/*	httpd	F	D	R	Apache HTTP server logs
kern.log	kernel	S	W	D	All kern facility messages
lastlog	login	H	–	R	Last login time per user (binary)
mail*	mail-related	S	W	RF	All mail facility messages
messages	various	S	W	R	The main system log file
samba/*	smbd, etc.	F	W	–	Samba (Windows/SMB file sharing)
secure	sshd, etc.[b]	S	M	R	Private authorization messages
syslog*	various	S	W	D	The main system log file
wtmp	login	H	M	RD	Login records (binary)
xen/*	Xen	F	1m	RD	Xen virtual machine information
Xorg.*n*.log	Xorg	F	W	R	X Windows server errors
yum.log	yum	F	M	R	Package management log

[a] Where: F = Configuration file, H = Hardwired, S = Syslog
Frequency: D = Daily, M = Monthly, NNm = Size-based (in MB, e.g., 1m), W = Weekly
Systems: D = Debian and Ubuntu (D* = Debian only), R = Red Hat and CentOS, F = FreeBSD

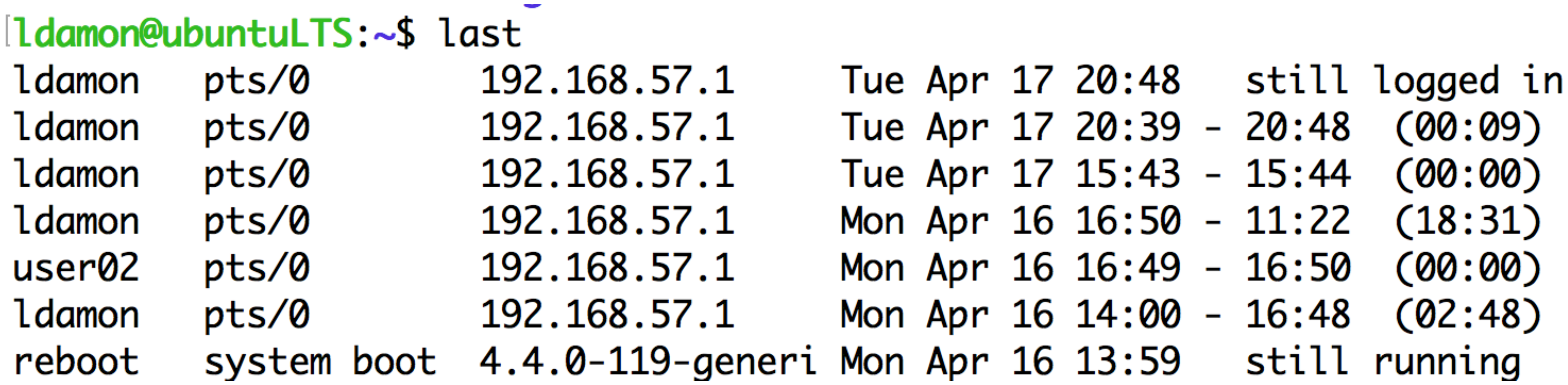
[b] **passwd**, **sshd**, **login**, and **shutdown** also write to the authorization log.

[c] Binary file that must be read with the **faillog** utility

notable logs - wtmp

wtmp: record of user logins and logouts

- also includes system boot time
- binary file, use “last” command to view

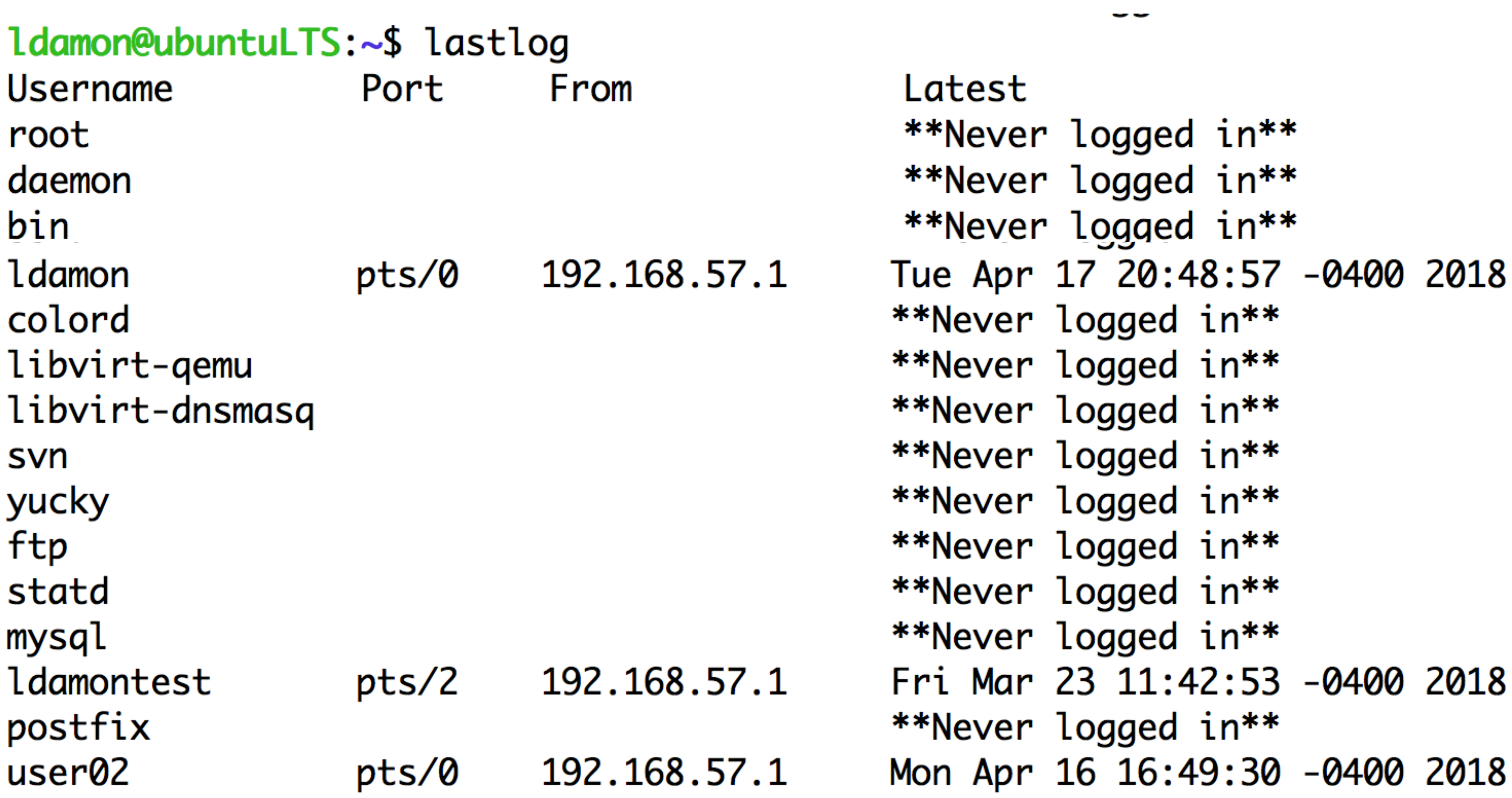


```
[ldamon@ubuntuLTS:~$ last
ldamon    pts/0        192.168.57.1     Tue Apr 17 20:48   still logged in
ldamon    pts/0        192.168.57.1     Tue Apr 17 20:39 - 20:48  (00:09)
ldamon    pts/0        192.168.57.1     Tue Apr 17 15:43 - 15:44  (00:00)
ldamon    pts/0        192.168.57.1     Mon Apr 16 16:50 - 11:22  (18:31)
user02    pts/0        192.168.57.1     Mon Apr 16 16:49 - 16:50  (00:00)
ldamon    pts/0        192.168.57.1     Mon Apr 16 14:00 - 16:48  (02:48)
reboot    system boot  4.4.0-119-generi Mon Apr 16 13:59   still running
```

notable logs - lastlog

lastlog: records last login time of each user

does not grow over time



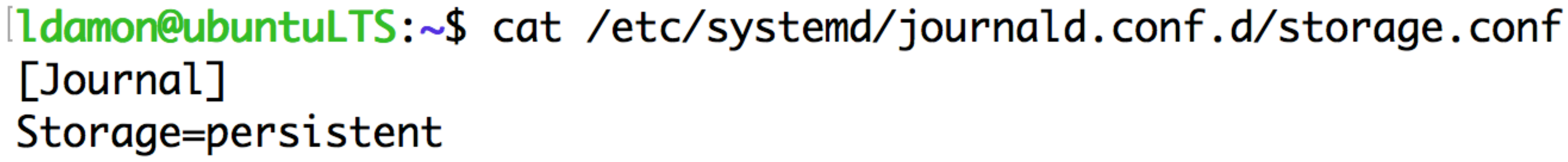
```
ldamon@ubuntuLTS:~$ lastlog
Username         Port     From             Latest
root                                       **Never logged in**
daemon                                     **Never logged in**
bin                                        **Never logged in**
ldamon           pts/0    192.168.57.1    Tue Apr 17 20:48:57 -0400 2018
colord                                    **Never logged in**
libvirt-qemu                              **Never logged in**
libvirt-dnsmasq                           **Never logged in**
svn                                       **Never logged in**
yucky                                     **Never logged in**
ftp                                       **Never logged in**
statd                                     **Never logged in**
mysql                                     **Never logged in**
ldamontest       pts/2    192.168.57.1    Fri Mar 23 11:42:53 -0400 2018
postfix                                   **Never logged in**
user02           pts/0    192.168.57.1    Mon Apr 16 16:49:30 -0400 2018
```

configuring systemd journal

/etc/systemd/journald.conf is the main config file

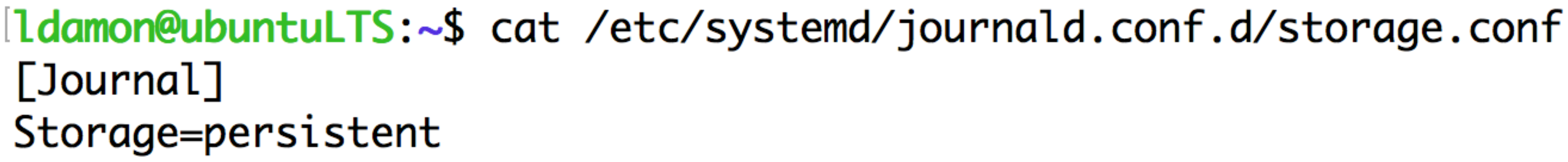
/etc/systemd/journald.conf.d is directory that allows additional conf files

To change config, create this directory and add files with new configs. For example:



```
ldamon@ubuntuLTS:~$ cat /etc/systemd/journald.conf.d/storage.conf
[Journal]
Storage=persistent
```

configuring systemd journal



```
ldamon@ubuntuLTS:~$ cat /etc/systemd/journald.conf.d/storage.conf
[Journal]
Storage=persistent
```

Storage option controls whether the journal is saved to disk

- volatile: in memory only
- auto (default): saves journal in /var/log/journal if (and only if) the directory exists
- persistent: saves journal, creates directory
- none: discard all log data

systemd journal and syslog

On Linux systems, both the systemd journal and syslog are active.

Why?

syslog can get messages from a variety of plugins and forward them to different outputs, based on filters and rules

this ability doesn’t exist with the systemd journal

eventually, systemd journal will probably be enhanced and take over, but not yet

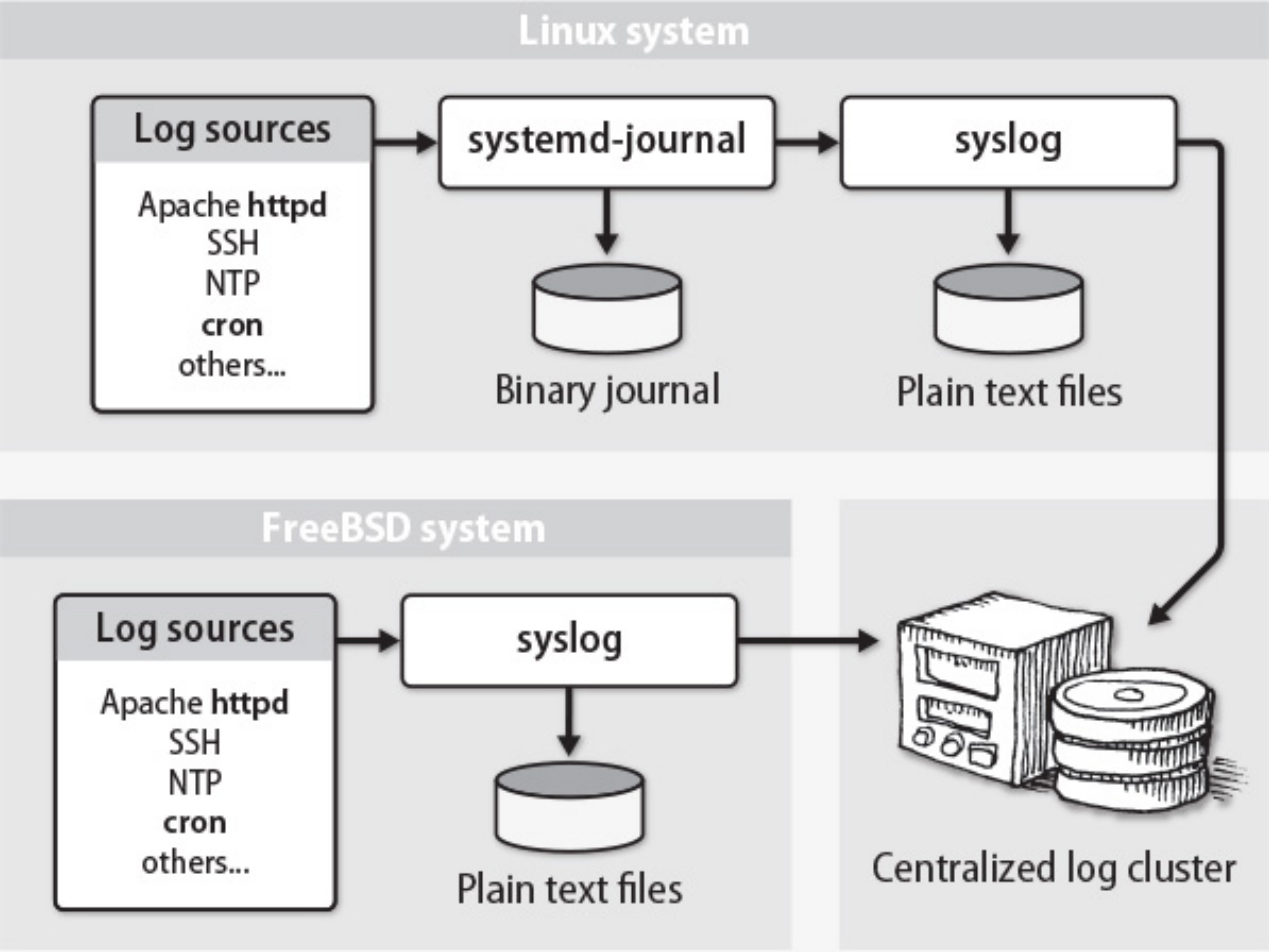
systemd journal and syslog

On Ubuntu, systemd gets the messages initially and forwards them to a syslog socket.

Red Hat and CentOS use a different integration - the Red Hat syslog knows how to read using the journal api

`ForwardToSyslog` option in configuration tells which, "yes" is Ubuntu default — this means systemd will send message to the syslog socket.

Logging architecture



syslog configuration

/etc/rsyslog.conf is the main config file

- it includes any files in /etc/rsyslog.d directory:

```
$IncludeConfig /etc/rsyslog.d/*.conf
```

uses modules to extend behavior

- imfile convert a plain text file to syslog message format
- imtcp/imudp accept network messages over TCP or UDP
- omfile output module that writes messages to a file
- omfwd to forward messages to a remote syslog server
- ommysql to forward messages to a MySQL DB

syslog configuration (cont)

has selectors that route messages appropriately

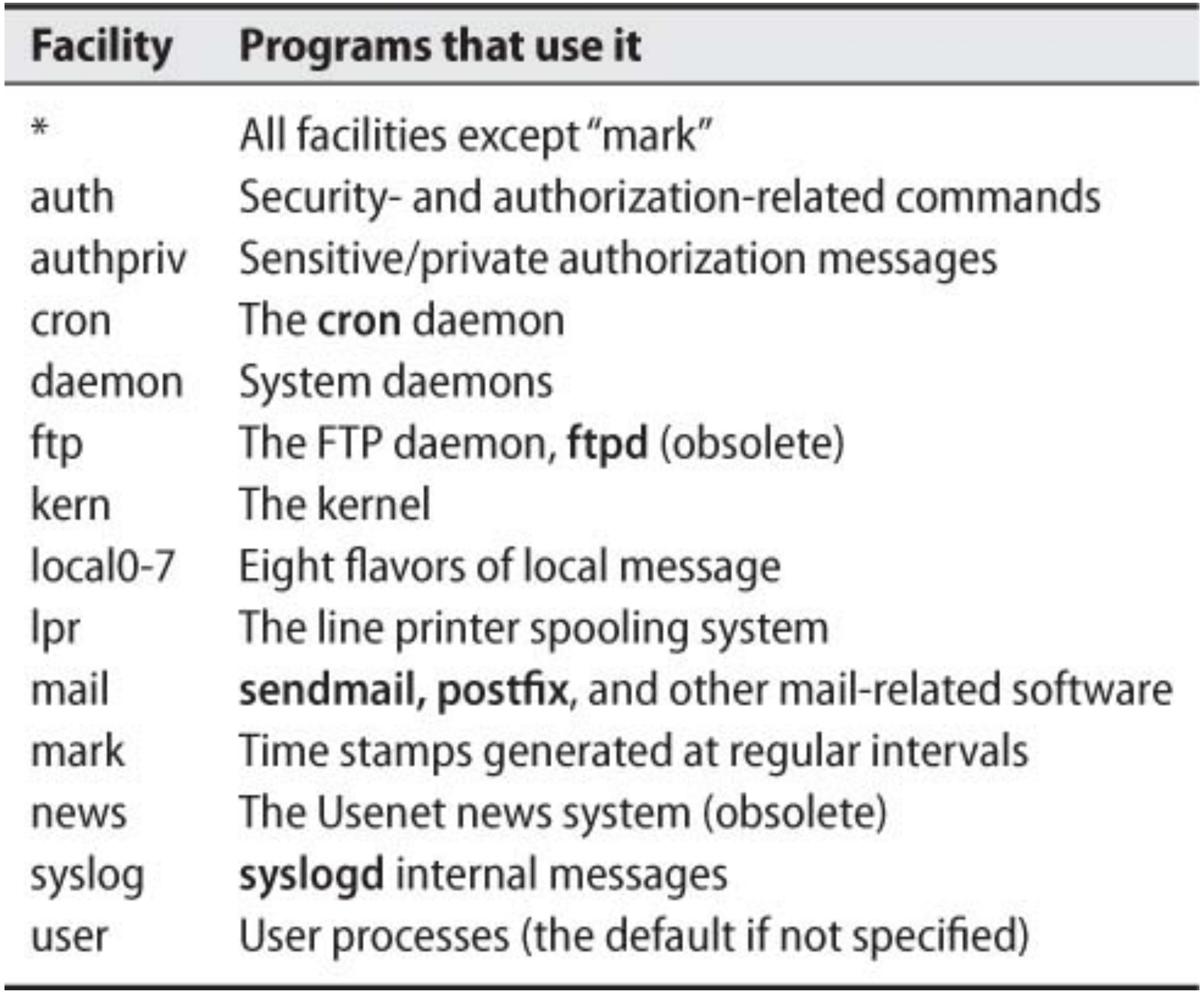
general form: selector   action

```
auth.*   /var/log/auth.log
```

Sends all authentication messages to /var/log/auth.log

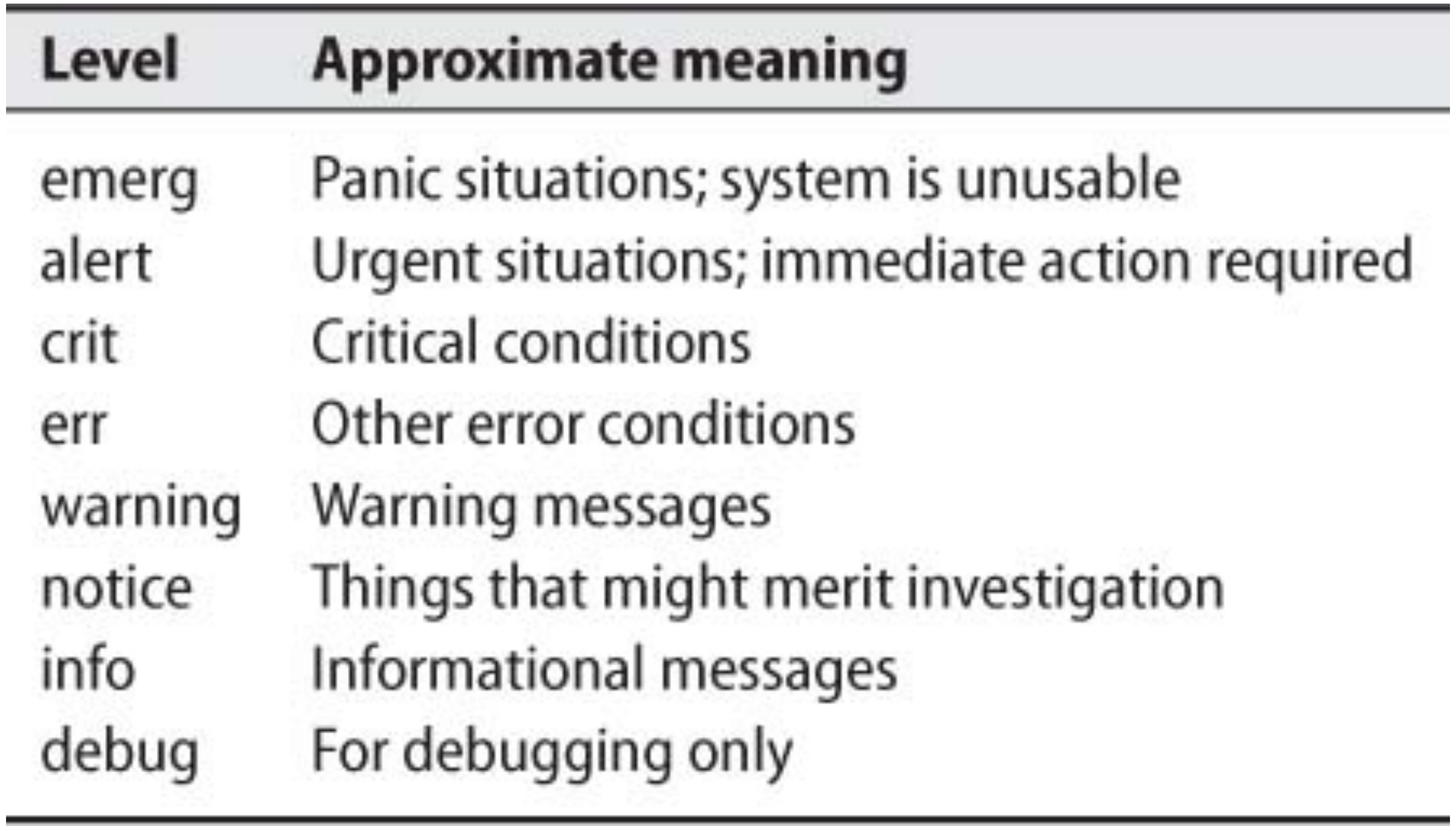
the selector has two fields:
facility.priorityLevel

facilities

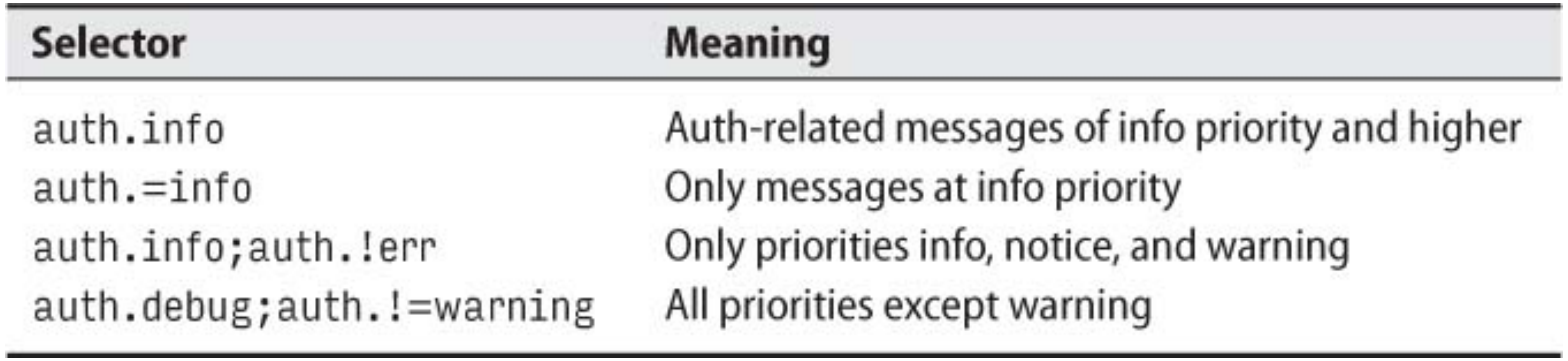


Facility	Programs that use it
*	All facilities except "mark"
auth	Security- and authorization-related commands
authpriv	Sensitive/private authorization messages
cron	The **cron** daemon
daemon	System daemons
ftp	The FTP daemon, **ftpd** (obsolete)
kern	The kernel
local0-7	Eight flavors of local message
lpr	The line printer spooling system
mail	**sendmail, postfix**, and other mail-related software
mark	Time stamps generated at regular intervals
news	The Usenet news system (obsolete)
syslog	**syslogd** internal messages
user	User processes (the default if not specified)

priority level

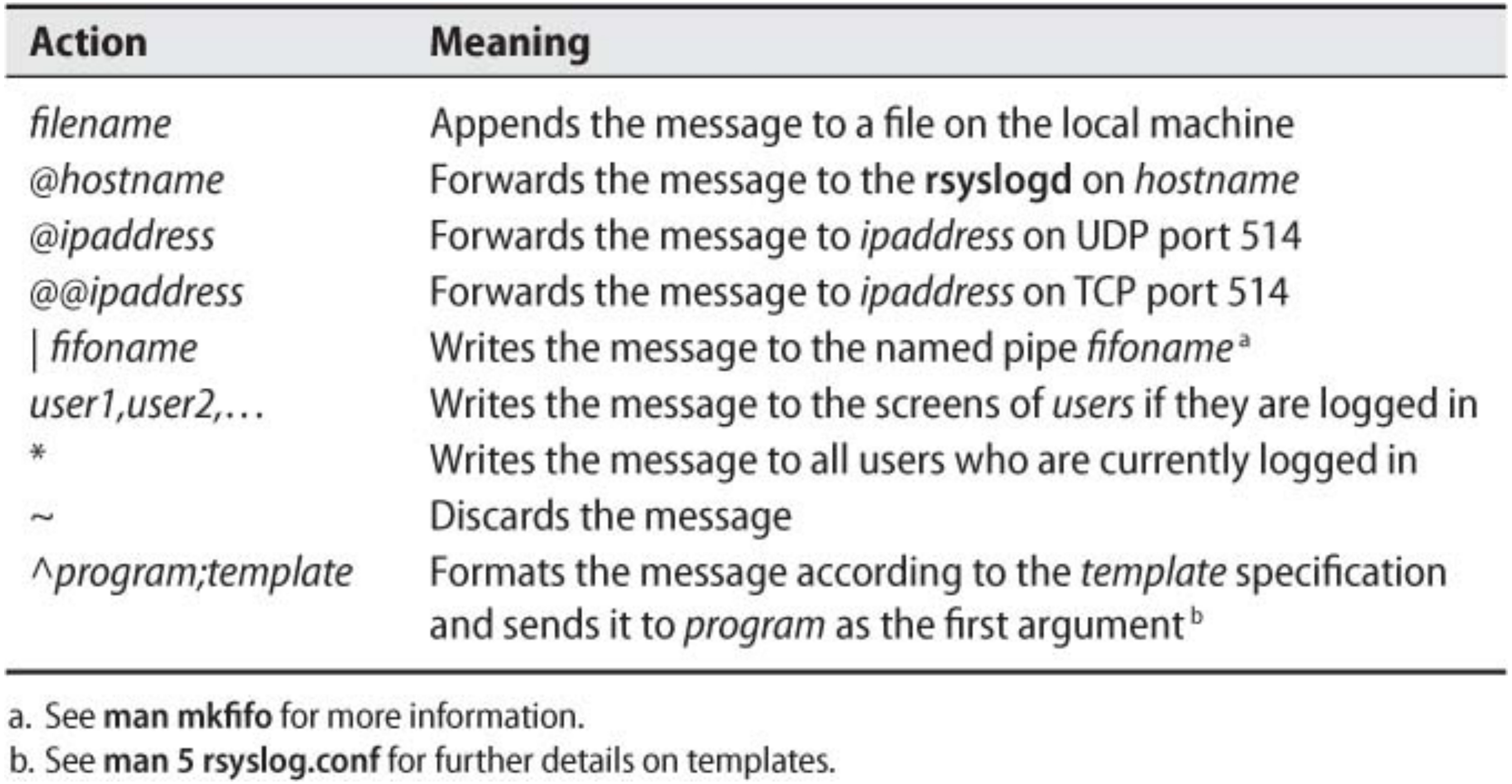


Level	Approximate meaning
emerg	Panic situations; system is unusable
alert	Urgent situations; immediate action required
crit	Critical conditions
err	Other error conditions
warning	Warning messages
notice	Things that might merit investigation
info	Informational messages
debug	For debugging only



Selector	Meaning
`auth.info`	Auth-related messages of info priority and higher
`auth.=info`	Only messages at info priority
`auth.info;auth.!err`	Only priorities info, notice, and warning
`auth.debug;auth.!=warning`	All priorities except warning

actions



Action	Meaning
filename	Appends the message to a file on the local machine
@*hostname*	Forwards the message to the **rsyslogd** on *hostname*
@*ipaddress*	Forwards the message to *ipaddress* on UDP port 514
@@*ipaddress*	Forwards the message to *ipaddress* on TCP port 514
| *fifoname*	Writes the message to the named pipe *fifoname*[a]
user1,user2,…	Writes the message to the screens of *users* if they are logged in
*	Writes the message to all users who are currently logged in
~	Discards the message
^*program;template*	Formats the message according to the *template* specification and sends it to *program* as the first argument[b]

[a] See **man mkfifo** for more information.

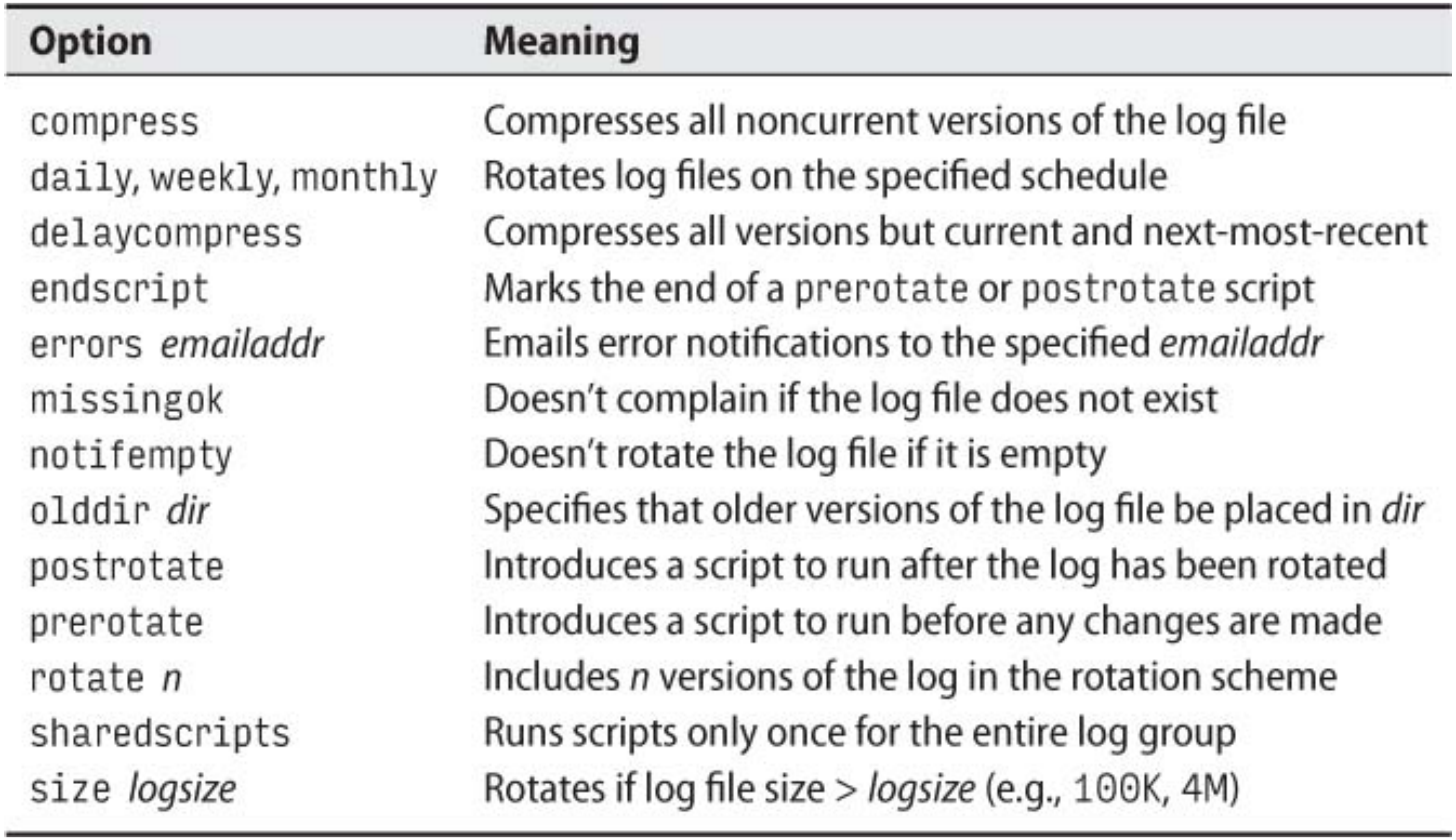
[b] See **man 5 rsyslog.conf** for further details on templates.

logrotate

- Most logs grow over time, and can fill up the disk
  - logrotate utility can help
  - included as standard in most Linux distributions
  - configured via /etc/logrotate.conf
    - also includes conf files in /etc/logrotate.d
- Example:

```
/var/log/samba/log.smbd {
  weekly
  missingok
  rotate 7
  postrotate
     /etc/init.d/smbd reload > /dev/null
  endscript
  compress
  notifempty
}
```

logrotate options



Option	Meaning
compress	Compresses all noncurrent versions of the log file
daily, weekly, monthly	Rotates log files on the specified schedule
delaycompress	Compresses all versions but current and next-most-recent
endscript	Marks the end of a prerotate or postrotate script
errors *emailaddr*	Emails error notifications to the specified *emailaddr*
missingok	Doesn't complain if the log file does not exist
notifempty	Doesn't rotate the log file if it is empty
olddir *dir*	Specifies that older versions of the log file be placed in *dir*
postrotate	Introduces a script to run after the log has been rotated
prerotate	Introduces a script to run before any changes are made
rotate *n*	Includes *n* versions of the log in the rotation scheme
sharedscripts	Runs scripts only once for the entire log group
size *logsize*	Rotates if log file size > *logsize* (e.g., 100K, 4M)

logs everywhere…ELK stack

Managing logs on a single server isn’t too bad
- can scale up to several servers

What if you have 10s or 100s of servers?
- need tools designed to scale

Leader in space is the (R)ELK stack
- Redis - in memory cache
- Elasticsearch - scalable DB and search engine
- Logstash - message parser/handler
- Kibana - visualization tool

Logstash can read in messages, pass them off to Elasticsearch, and then you can use Kibana to do graphs of “interesting” stuff.

Monitoring

If you have lots of logs, you can't read them individually

Want to monitor key data points

- system up/down
- disk space
- cpu usage
- key processes

Can monitor in real time with a tool like Nagios

More monitoring moving to time series —

- What is the normal CPU usage on a DB server?
- We see 10% errors for X, is that high? Or expected?

Steps in monitoring

All monitoring depends on centralized collection of data:

- collect data
- determine actionable data points
- determine the correct response to each data point
  - automated cleanup of disk space
  - displaying info on a dashboard
  - storing for later analysis
  - email notification
  - SMS notification
  - do nothing

Culture

If a system is critical, it must be monitored.

Monitoring requires time/attention from staff as part of regular duties — not just reacting to problems

Data needs to be useful

- false alerts train people to ignore them
- can negatively impact morale

Everyone should respond to alerts — inter-departmental

- the more the merrier

Documentation on responses is essential for each possible alert

Alerts should be fixed, not suppressed

- Don't find out if a primary failed (and was ignored) when the backup fails!

Tips

Avoid burnout

- share the load of off-hour notifications

Only respond 24x7 for critical situations. Defer non-critical to normal work hours

Avoid false positives and non-critical notifications

documentation, documentation, documentation

- if you are responding at 3 am, your brain isn’t going to be awake. Having clear responses is a must

Monitor the monitoring system. Does an unmonitored outage exist?

No servers or services go into production without being added to the monitoring