Linux Log Commands Every Admin Should Know

Core Log Analysis Commands

1. `journalctl` – Modern systemd log management

The powerhouse for systemd-based distributions:

```bash
journalctl                          # View all system logs
journalctl -u nginx                 # Logs for specific service
journalctl -f                       # Follow logs in real-time
journalctl -k                       # Kernel messages only
journalctl --since "1 hour ago"     # Time-based filtering
journalctl --until "2024-01-15"     # Logs until specific date
journalctl -p err                   # Filter by priority (err, warning, info)
journalctl -n 50                    # Show last 50 entries
journalctl --disk-usage             # Check log disk usage
```

2. `dmesg` – Hardware and kernel diagnostics

Essential for hardware troubleshooting:

```bash
dmesg                               # All kernel ring buffer messages
dmesg | grep -i error               # Find error messages
dmesg | grep -i "usb\|network"      # Hardware-specific issues
dmesg -T                            # Human-readable timestamps
dmesg -w                            # Watch for new messages
```

3. `tail` & `head` – Quick file viewing

Fast access to log file contents:

```bash
tail -f /var/log/syslog             # Follow log in real-time
tail -n 100 /var/log/messages       # Last 100 lines
head -n 50 /var/log/auth.log        # First 50 lines
tail -f /var/log/nginx/error.log    # Monitor web server errors
```

4. `less` & `more` – Interactive log browsing

Navigate large log files efficiently:

bash

```
less /var/log/syslog          # Scrollable view with search
less +F /var/log/messages      # Follow mode (like tail -f)
# Inside less: press '/' to search, 'q' to quit, 'G' to go to end
```

5. `grep` – Pattern searching powerhouse

Find exactly what you need:

bash

```
grep -i "failed\|error" /var/log/auth.log      # Case-insensitive search
grep -r "connection refused" /var/log/          # Recursive search
grep -A 5 -B 5 "critical" /var/log/syslog      # Show 5 lines before/after
grep -v "INFO" /var/log/app.log               # Exclude INFO messages
grep -E "(error|warning|critical)" /var/log/*  # Multiple patterns
```

Essential Log Locations

System-wide logs:

- `/var/log/syslog` – General system messages (Debian/Ubuntu)
- `/var/log/messages` – General system messages (RHEL/CentOS)
- `/var/log/auth.log` – Authentication attempts
- `/var/log/secure` – Security-related messages (RHEL/CentOS)
- `/var/log/kern.log` – Kernel messages

Service-specific logs:

- `/var/log/apache2/` or `/var/log/httpd/` – Web server logs
- `/var/log/nginx/` – Nginx web server logs
- `/var/log/mysql/` – MySQL database logs
- `/var/log/cron` – Scheduled task logs

Advanced Log Analysis Techniques

Power combinations:

```bash
# Failed SSH attempts with IP addresses
journalctl -u sshd | grep "Failed password" | awk '{print $11}' | sort | uniq -c

# Monitor multiple logs simultaneously
multitail /var/log/syslog /var/log/auth.log

# Real-time error monitoring across all logs
find /var/log -name "*.log" -exec tail -f {} + | grep -i error

# Log rotation and compression analysis
logrotate -d /etc/logrotate.conf   # Debug log rotation
```

Filtering and formatting:

```bash
# Show only today's entries
journalctl --since today

# JSON output for parsing
journalctl -o json | jq '.MESSAGE'

# Boot-specific logs
journalctl -b              # Current boot
journalctl -b -1           # Previous boot
```

Pro Tips for Efficient Log Analysis

🔍 Search Strategy:

- Start broad, then narrow down with specific filters
- Use time ranges to limit scope: `--since "2024-01-15 14:00"`
- Combine multiple tools: `journalctl -u apache2 | grep -E "(error|warning)"`

⚡ Performance Tips:

- Use `journalctl --vacuum-size=100M` to clean up old logs
- Set up log rotation to prevent disk space issues
- Use `rsyslog` or `syslog-ng` for centralized logging

🚨 Common Troubleshooting Patterns:

```bash
# System boot issues
journalctl -b | grep -i "failed\|error"

# Network connectivity problems
journalctl -u NetworkManager | tail -50

# Service startup failures
systemctl status service_name
journalctl -u service_name --since "10 minutes ago"

# Disk space and I/O issues
dmesg | grep -i "i/o error\|disk\|filesystem"
```

Bonus Tools for Advanced Users

- `multitail` – Monitor multiple files simultaneously
- `lnav` – Advanced log file navigator with syntax highlighting
- `goaccess` – Real-time web log analyzer
- `rsyslog` – Centralized logging solution
- `logrotate` – Automatic log rotation and compression

Master these commands and you'll solve system issues 10x faster!

By SHAHJAHAN