journalctl Query the journal

START HERE

`journalctl -u <unit>`
Logs of one service

`journalctl --since "1 hour ago"`
Recent entries

`journalctl -f`
Follow new entries

`journalctl -g <regex>`
Search messages

`journalctl -b`
Since this boot

`journalctl -e`
Jump to the end

`journalctl -p err`
Errors and worse

`journalctl --disk-usage`
How big is the journal?

Journal all entries → `-b -u -p` → **Filters** boot, unit, priority → `--since -g` → **Time and text** narrowed set → `-o -n -r` → **Output** pager or stdout

CONCEPTS

The journal is a binary, indexed log. Filters narrow it before output, so combine them freely: different fields are ANDed.

Reading the system journal needs root or membership of the systemd-journal, adm or wheel group; otherwise you only see your own entries.

The journal is only kept across reboots if /var/log/journal exists (Storage=persistent or auto in journald.conf). Otherwise -b -1 has nothing.

SELECT ENTRIES

`journalctl -u <unit>`
Logs of a unit
Repeat -u to see several units interleaved

`journalctl -b`
Current boot only

`journalctl -b -1`
Previous boot
Needs a persistent journal; find IDs with --list-boots

`journalctl -k`
Kernel messages (dmesg)

`journalctl -t <name>`
Entries by syslog identifier

`journalctl -T <name>`
Hide a syslog identifier

`journalctl --user-unit <unit>`
Logs of a user unit

`journalctl -I -u <unit>`
Latest invocation of a unit
Shows only the most recent run, handy for oneshot services and timers

`journalctl --list-invocations -u <unit>`
List runs of a unit

`journalctl _PID=<pid>`
Match a process ID

`journalctl _UID=<uid>`
Match a user ID

`journalctl <path>`
Entries of an executable

`journalctl _COMM=<name> + _PID=<pid>`
OR two match groups with +

`journalctl --user`
The user journal

`journalctl -m`
Merge all available journals

SELECT ENTRIES (cont.)

`journalctl -D <dir>`
Read journal files from a directory

`journalctl --namespace=<ns>`
Read a journal namespace

TIME AND PRIORITY

`journalctl -p err`
Errors and worse
A level includes everything more severe

`journalctl -p warning..err`
Priority range

`journalctl --since "<date>"`
From a point in time
Accepts 2026-10-09 08:00, yesterday, now, or relative like -2h

`journalctl --until "<date>"`
Up to a point in time

`journalctl -S -1h`
Last hour

`journalctl -S today`
Since midnight

`journalctl --utc`
Show times in UTC

`journalctl --facility=<name>`
Match a syslog facility

Priority levels

Name	Number
emerg	0
alert	1
crit	2
err	3
warning	4
notice	5
info	6
debug	7

SEARCH AND FOLLOW

`journalctl -f`
Follow new entries

`journalctl -f -u <unit>`
Follow one unit

`journalctl -n <num>`
Last N entries

`journalctl -n +<num>`
First N entries
The plus form counts from the start of the selection

`journalctl -g <regex>`
Search messages by pattern
Case-insensitive unless the pattern has uppercase; force with --case-sensitive

SEARCH AND FOLLOW (cont.)

`journalctl --case-sensitive -g <regex>`
Force case-sensitive search

`journalctl -e`
Open pager at the end

`journalctl -r`
Newest entries first

`journalctl --no-pager`
Print without a pager

`journalctl -x`
Add catalog explanations

`journalctl -q`
Hide info and privilege warnings

OUTPUT

`journalctl -o json-pretty`
Entries as readable JSON

`journalctl -o json`
One JSON object per line
Pipe to jq for scripting

`journalctl -o cat`
Messages only

`journalctl -o short-iso`
ISO timestamps

`journalctl -o short-precise`
Microsecond timestamps

`journalctl -o verbose`
Every field of each entry

`journalctl -o with-unit`
Prefix lines with unit name

`journalctl --output-fields=<list>`
Limit fields in verbose/json

`journalctl -a`
Show long and binary fields

`journalctl --show-cursor`
Print the cursor at the end

`journalctl --after-cursor=<cursor>`
Resume after a cursor

`journalctl --cursor-file=<file>`
Resume and save position

Output modes

Mode	Use
short	Default, syslog-like
cat	Message text only
verbose	All fields
json-pretty	Debugging fields
json	Scripts and jq

FIELDS AND BOOTS

`journalctl --list-boots`
List recorded boots

`journalctl -b <id>`
A specific boot by ID

FIELDS AND BOOTS (cont.)

`journalctl -N`
List all field names

`journalctl -F <field>`
List values of a field
Example: -F _SYSTEMD_UNIT lists units that logged

`journalctl <FIELD>=<value>`
Match any field exactly

DISK AND MAINTENANCE

`journalctl --disk-usage`
Total journal size

`sudo journalctl --vacuum-time=<time>`
Delete archives older than a time
Example: 2weeks. Only archived files are removed

`sudo journalctl --vacuum-size=<size>`
Shrink archives to a size
Example: 500M

`sudo journalctl --vacuum-files=<num>`
Keep only N archive files

`sudo journalctl --rotate`
Rotate files now
Combine with a vacuum option to make recent data eligible

`sudo journalctl --flush`
Move /run data into /var

`journalctl --verify`
Check journal file integrity

Cap the journal permanently

```
# /etc/systemd/journald.conf.d/size.conf
[Journal]
SystemMaxUse=500M
MaxRetentionSec=1month
# then: sudo systemctl restart systemd-journald
```

RECIPES

Why did it fail?

```
systemctl status <unit>
journalctl -u <unit> -b -p warning -e
journalctl -I -u <unit> -x
```

Last boot's errors

```
journalctl --list-boots
journalctl -b -1 -p err --no-pager
```

Export to a file

```
journalctl -u <unit> --since yesterday -o short-iso --no-pager > unit.log
```

Extract fields with jq

```
journalctl -u <unit> -n 50 -o json --no-pager \
  | jq -r '[.__REALTIME_TIMESTAMP, .MESSAGE] | @tsv'
```