Journald

Els registres del sistema (received from the kernel, from user processes, from standard input and error of system services) són recopilats per un demoni anomenat systemd-journald (per tant, haurà d'estar encés i habilitat per a què funcioni: *systemctl start systemd-journald && systemctl enable systemd-journald*

Basic configuration

The journal stores log data in /run/log/journal/ by default (in binary form). Because the /run/ directory is volatile by nature, log data is lost at reboot. To make the log data persistent, the directory /var/log/journal/ with correct ownership and permissions must exist, where the systemd-journald service can store its data. systemd will create the directory for you—and switch to persistent logging—if you edit /etc/systemd/journald.conf file leaving following line like this (the contrary of "persistent" is "volatile"; it is possible to use "no" as a value, too):

Storage=persistent

Journal doesn't keep data through a time-limit (for example last week only), instead it occupies 10% of available partition space, and when it gets it, it rotates the logs like a list(deleting the last one). To find out the disk that currently uses you can execute *journalctl --disk-usage* command ; if you want to change ths maximum possible usage, you can edit /etc/systemd/journald.conf file to leave following lines like this:

SystemMaxUse=50G

You can forward the journal to a terminal device to inform you about system messages on a preferred terminal screen, for example /dev/tty12 . Change the following options in /tc/systemd/journald.conf to:

#ForwardToSyslog=no
#ForwardToKMsg=no
ForwardToConsole=yes
TTYPath=/dev/tty12

By default, Journal users without root privileges can only see log files generated by them. In Fedora, adding an user to "systemd-journal" group gives him access to all logs.

Parámetros de journalctl

Si se pone más de un parámetro a la vez es equivalente a un AND (a no ser que entre ellos se escriba un "+", interpretándose entonces un OR):

-f : Similar a tail -f

-r : Muestra los mensajes al revés (el más moderno primero)

-e : Muestra todos los mensajes y se mueve hasta el último (por defecto se queda en el primero)

-o {verbose|json|export|short-iso|cat} : Muestra salida en modo verboso (viendo todos los campos de cada registro), en formato JSON, en formato binario (especialmente pensado para exportaciones a través de la red), en formato similar al normal pero añadiendo el año a la fecha mostrada (cosa que por defecto no se hace) o mostrando solo el texto del mensaje propiamente dicho.

--utc : Muestra la fecha de cada mensaje en formato Universal en vez de Local (que es como Systemd las muestra por defecto)

--no-pager : No usa paginador para mostrar los mensajes sino que los "vomita" todos de golpe (útil para entubar la salida a otro comando)

-n nº : Muestra solo los últimos nº mensajes. Si nº no se escribe, por defecto es 10

/ruta/ejecutable : Filtra solo los mensajes relacionados con el ejecutable idicado.

_PID nº : Filtra solo los mensajes relacionados con el PID idicado

-u nombreUnit : Filtra solo los mensajes relacionados con el "unit" indicado

-k : Filtra solo los mensajes relacionados con el kernel (similar a dmesg)

_UID nº : Filtra solo los mensajes relacionados con el UID indicado

_HOSTNAME nombre : Filtra solo los mensajes relacionados con el host indicado

-t unaTag : Filtra solo los mensajes etiquetados con el valor indicado (ver más abajo systemd-cat)

-p nombrePri : Filtra solo los mensajes con una prioridad igual o superior a la indicada (también se puede poner -p pri1..pri2). Las prioridades son (en vez del nombre se podría indicar su número): debug (7), info (6), notice (5), warning (4), err (3), crit (2), alert (1), emerg (0)

--since=xxxx : Filtra solo los mensajes ocurridos desde la fecha y hora indicadas
El formato de la fecha ha de ser "YYYY-MM-DD hh:mm:ss"; si se omite el tiempo se asume 00:00:00; si se omiten los segundos se asume :00; si se omite la fecha se asume hoy. También se pueden indicar las expresiones "yesterday", "today" y "tomorrow", que se refieren a la medianoche del día anterior, actual o siguiente. También se puede escribir "now", que se refiere a la hora actual (útil con -f). También se pueden indicar tiempos relativos respecto la hora actual con - (antes) o + (después), así: "-1week", "-1month", "-20day" (ver man systemd.time)

--until=xxx : Filtra solo los mensajes ocurridos hasta la fecha y hora indicadas

-b N : Filtra solo los mensajes dhere N is the boot you want information from. N=0 (o no ponerlo) significa el inicio actual; -1 el anterior, y así). Para ver todos los inicios posibles hacer journalctl –list-boots The second field is the boot ID. These first two IDs can be used when referring to a specific boot. Next there is the day, date, time and timezone, when the first entry entered the journal. These are then followed by the same fields, representing the last entry of the journal. Observando el second field entonces se puede escribir un filtro como: journalctl _BOOT_ID=uuid

-F nombreCampo : Muestra solo los valores del campo indicado (con -o verbose se pueden consultar cuáles son o, consultar https://www.freedesktop.org/software/systemd/man/systemd.journal-fields.html)

nombreCampo=valor : Muestra solo los registros que tengan en el campo indicado el valor indicado

--vacuum-size=1G : Borra del disco las entradas necesarias (empezando por las más antiguas) para que el registro ocupe un total de 1G de espacio

--vacuum-time=1years : Borra del disco las entradas más antiguas que el tiempo indicado

--verify : Comprueba la integridad interna del registro

De momento no hay la posibilidad de indicar filtros excluyentes del tipo "todo excepto esto" (https://github.com/systemd/systemd/issues/2720)

Para enviar un mensaje manualment al registro existe el comando específico systemd-cat, que funciona así: *echo "Backup Failed" | systemd-cat -t "terminalid-home_backup" -p "crit"* (la fecha se añade sola) O también así: *systemd-cat -t "nombreSimuladoAplicacion" -p "crit" echo "Backup Failed"* (lo mismo que el caso anterior pero además registra stderr). Una alternativa a systemd-cat es el venerable comando *logger -s -t -p " "*

Existen programas complementarios que lanzan alertas (ejecución de un determinado programa, envío de mails o similares, etc) cuando se detecta un determinado mensaje en el journal. Ejemplos son:
* https://github.com/jjk-jacky/journal-triggerd (su página web es https://jjacky.com/journal-triggerd)
* https://github.com/The-Compiler/journalwatch (solo sirve para enviar mails)