Journald

Els registres del sistema (els quals poden representar des de simples missatges informatius fins missatges crítics d'errors, i poden ser rebuts directament per part del kernel o bé per part de processos d'usuari o de dimonis, entre d'altres orígens) en sistemes gestionats per Systemd són recopilats per un demoni central anomenat "systemd-journald". Per tant, haurà d'estar encés i habilitat per a què funcioni aquesta recopilació de registres: *sudo systemctl start systemd-journald && sudo systemctl enable systemd-journald*

Configuració bàsica

El dimoni "systemd-journald" per defecte emmagatzema les dades de registre a la carpeta "**/run/log/journal/*MACHINE-ID***" (en forma binària). Com que el directori "/run" és volàtil per naturalesa (es troba a "tmpfs", que és, de fet, RAM), aquestes dades es perden en reiniciar-se.

> NOTA: El valor "MACHINE-ID" és el que s'emmagatzema dins del fitxer "**/etc/machine-id**". Aquest valor, que serveix per identificar internament una màquina, es genera aleatòriament en el moment de la instal·lació del sistema i no canvia més.

Per fer que les dades del registre siguin persistents, és requisit imprescindible que existeixi la carpeta "**/var/log/journal/*MACHINE-ID***" perquè és l'únic lloc on el servei "systemd-journald" pot emmagatzemar les dades. Generalment, "systemd-journald" crearà aquest directori automàticament si no existeix, així que l'únic que caldrà fer per canviar al registre persistent és editar el fitxer "**/etc/systemd/journald.conf**" deixant la línia següent (i reiniciant el servei):

> *Storage=persistent*

> NOTA: Si no es pot crear "/var/log/journal/*MACHINE-ID*" (ja que el sistema de fitxers on està muntat és de només lectura, per exemple), el valor "persistent" farà que automàticament es desin les dades del registre a "/run/log/journal/*MACHINE-ID*" (basat en RAM). D'altra banda, si volguéssim desar les dades de registre només a "/run/log/journal/*MACHINE-ID*" (és a dir, sense tocar mai el disc), hauríem d'utilitzar el valor "volatile". A més, hi ha el valor "auto", que és igual que "persistent" si existeix la carpeta "/var/log/journal/*MACHINE-ID*" i igual a "volatile" si no existeix (fent, per tant, l'existència d'aquesta carpeta el "commutador" d'aquest comportament tan diferent). Finalment, també és possible utilitzar el valor "none"

Per esbrinar la quantitat de disc que el registre utilitza, es pot executar l'ordre *journalctl --disk-usage*. Cal tenir en compte que la sortida de l'ordre anterior diu *"Archived and active journals ..."* Això significa que el registre està format en realitat per diversos fitxers (tots ells amb l'extensió ".journal") dins de la carpeta "/var/log/journal/MACHINE-ID". Només un d'ells és el fitxer "actiu" (on realment s'emmagatzemen les dades) i els altres són els fitxers "arxivats" (on no s'escriuran més registres perquè només dades passades ja rotades i comprimides).

> NOTA: La directiva *SystemMaxFileSize* (o *RuntimeMaxFileSize*) de l'arxiu "journald.conf" defineixen el tamany màxim (en K,M,G,T...) que tindrà un fitxer ".journal". Un cop s'arribi a aquest tamany, el fitxer automàticament es comprimirà, es rotarà i s'arxivarà. Per defecte aquesta directiva val 1/8 del tamany indicat a la directiva *SystemMaxUse* (veure més avall), i això vol dir, a la pràctica, que es podran tenir fins a 7 arxius "*.journal" arxivats més l'actiu. L'arxiu arxivat més antic s'eliminarà a cada rotació. En qualsevol cas, el número màxim de fitxers ".journal" que es volen mantenir es pot configurar, independentment de *SystemMaxUse*, amb la directiva *SystemMaxFiles* (o *RuntimeMaxFiles*); l'arxiu arxivat més antic que superi el número indicat en aquesta directiva s'eliminarà a cada rotació.
> NOTA: La directiva *MaxFileSec=nº* ofereix una forma alternativa de rotació de logs que, en comptes de tenir en compte el tamany màxim dels arxius arxivats (com passava amb *SystemMaxFileSize*) té en compte el temps màxim que ha de passar per arxivar un arxiu ".journal"
> NOTA: Quan *systemd-journald* deixa d'escriure en un fitxer del registre (és a dir, quan "arxiva" un fitxer), aquest fitxer passarà a canviar-se el nom a "nomOriginal@sufixeRandom.journal"
> NOTA: Si *systemd-journald* s'aturés de manera no neta o si es constatés que els fitxers "*.journal" estan danyats, es canviaria la seva extensió a ".journal ~" i *systemd-journald* començaria a escriure en un fitxer nou.
> NOTA: Ja hem dit que els fitxers ".journal" són de tipus binari. Això implica que serà necessari utilitzar eines que siguin capaces d'entendre aquest format binari (documentat a https://systemd.io/JOURNAL_FILE_FORMAT) per tal de llegir (o escriure-hi!) les entrades pertinents (és a dir, no ens serviran les comandes textuals típiques com *cat*, *cut*, *grep*, etc). En aquest sentit, si un desenvolupador volgués interactuar amb el Journal pot utilitzar l'API que el servei Journald proporciona, basat en el protocol documentat a https://systemd.io/JOURNAL_NATIVE_PROTOCOL. En tot cas, nosaltres com administradors només ens interessarem en llegir les entrades publicades allà, per la qual cosa farem servir la comanda especialitzada *journalctl*, proporcionada pel propi Systemd.

Per defecte, tots els fitxers que formen el registre (és a dir, la suma dels actius i arxivats) poden ocupar un 10% de l'espai de partició màxim disponible. Per canviar l'espai total d'emmagatzematge possible utilitzat pels fitxers de diari, es pot editar el fitxer "/etc/systemd/journald.conf" per deixar la línia següent:

SystemMaxUse=50G

NOTA: Existeix una directiva semblant però que afecta a l'emmagatzematge del Journal en RAM (és a dir, a "/run/log/journal") anomenada *RuntimeMaxUse* En tot cas, es poden fer servir els sufixes "K","M","G","T",...
NOTA: Alternativament, es pot indicar no un tamany total màxim sinó un temps màxim, passat el qual s'esborraran els arxius arxivats més antics. Això es pot fer indicant la directiva *MaxRetentionSec=nº* (on el número per defecte representa segons però es pot convertir en minuts, hores, dies, setmanes, mesos o anys si s'afegeix el sufixe "m","h","d","w","M" o "y", respectivament). Per defecte val 0, que vol dir que no s'aplica

Per defecte, al fitxer "/etc/systemd/journald.conf" hi ha la línia *SplitMode=uid* . Això significa que les dades que arriben a *systemd-journald* s'emmagatzemaran o bé al fitxer "**system.journal**" o bé al/s fitxer/s "**user-*UID*.journal**". El primer emmagatzema les dades generades pels diferents usuaris del sistema (és a dir, els que tenen un UID menor de 1000) incloent l'usuari "root"; el/ segon/s emmagatzemen les dades generades per l'usuari "estàndard" respectiu. Com que el primer només el poden llegir l'usuari "root" (o els usuaris del grup "systemd-journal") i el/s segon/s el pot/poden llegir, a més dels anteriors, l'usuari que l'/els ha generat., un usuari "estàndard" podrà veure només els missatges de registre generats pels programes executats per ell mateix mentre que l'usuari administrador els podrà veure tots.

NOTA: Tots els fitxers del journal són, per defecte, propietat i llegibles pel grup "systemd-journal". ¿Per què els fitxers de registre específics de cada usuari no són propietat de l'usuari en qüestió? Això es fa per evitar que l'usuari els pugui escriure directament. A canvi s'utilitzen ACLs per garantir que l'usuari només en tingui accés de lectura. De fet, es pot concedir a usuaris i grups addicionals accés als fitxers de registre mitjançant aquest mètode; per exemple, les distribucions i els administradors poden optar per concedir accés de lectura, a més de al grup "systemd-journal", a tots els membres dels grups de sistemes "wheel" i "adm" amb una ordre ACL com la següent: *setfacl -Rnm g:wheel:rx,d:g:wheel:rx,g:adm:rx,d:g:adm:rx /var/log/journal* (cal tenir en compte que aquesta ordre actualitzarà les ACL tant per als fitxers del registre existents com per als futurs fitxers de registre creats al directori "/var/log/journal")
NOTA: L'altre valor de SplitMode és "none". En aquest cas, els fitxers del registre no es divideixen per usuari sinó que tots els missatges s'emmagatzemen al registre del sistema únic. En aquest mode, els usuaris no privilegiats no tenen accés a les seves pròpies dades de registre
NOTA: La divisió de fitxers del registre per usuari només està disponible si l'emmagatzematge és persistent: si les dades s'emmagatzemen en un suport volàtil només s'utilitzarà un sol fitxer.

D'altra banda, es poden reenviar les dades del registre a un terminal en temps real, com per exemple "/dev/tty12"; només cal indicar les opcions següents a "/etc/systemd/journald.conf":

ForwardToConsole=yes
TTYPath=/dev/tty12
MaxLevelConsole=err #El seu significat és similar al del paràmetre *-p* de la comanda *journalctl*

NOTA: De forma similar a *MaxLevelConsole,* també existeix la directiva *MaxLevelStore* per definir quina és la prioritat màxima dels missatges que es voldran guardar en disc

Una altra forma de personalitzar la configuració del servei Journald és crear un nom de fitxer "*elquesigui*.conf" dins de la carpeta "**/etc/systemd/journald.conf.d**" amb el contingut següent (per exemple):

[Journal]
Storage=persistent
SystemMaxUse=10G
...

El fitxer de configuració principal té la precedència més baixa; això significa que les entrades col·locades en un fitxer dins del directori "journald.conf.d" anul·len les entrades homònimes col·locades en el fitxer de configuració principal. D'altra banda, si hi haguessin diversos fitxers dins de la carpeta "journald.conf.d", s'ordenaran pel seu nom de fitxer en ordre lexicogràfic, de manera que si varis fitxers especifiquen la mateixa opció, l'entrada al fitxer amb el darrer nom té prioritat. És per això que es recomana anomenar els fitxers ubicats en aquesta carpeta començant amb un número de dos dígits per simplificar l'ordenació dels fitxers.

Paràmetres de *journalctl*

-f	Similar a *tail -f*
-r	Mostra els missatges al revés (el més modern primer)
-e	Mostra tots els missatges i es mou fins el darrer (per defecte es queda en el primer)
-n [nº]	Mostra només els darrers nº missatges. Si nº no s'escriu, per defecte és 10
-x	Afegeix informació extra respecte les línies mostrades (enllaços a documentació, context dels missatges, possibles solucions a errors, etc)
_PID=nº	Mostra només els missatges relacionats amb el PID indicat NOTA: A més d'aquest camp _PID, hi ha molts més camps pels quals es pot filtrar la informació guardada al registre. La llista d'aquests camps es pot obtenir executant *journalctl -N* (o mirant la página *man systemd.journal-fields*). De fet, les següents files d'aquesta taula en mostren alguns exemples més.
_EXE=/ruta/ejecutable	Mostra només els missatges relacionats amb l'executable indicat. *En el cas de ser un script, però, és millor utilitzar el camp _COMM. *En qualsevol cas, es pot usar el camp _CMDLINE per filtrar no només per la ruta de l'executable sinó també pels seus paràmetres. *Si s'escriu com a paràmetre de *journalctl* directament la ruta de l'executable (o dispositiu) es filtrarà pels camps _EXE, _COMM o _KERNEL_DEVICE indistintament (aquest darrer camp serveix per filtrar els missatges relacionats amb un determinat dispositiu "/dev/xxx").
_UID=nº	Mostra només els missatges relacionats amb el UID indicat
_HOSTNAME=nomMaquina	Mostra només els missatges relacionats amb el "host" indicat. És interessant en el cas de tenir centralitzats els missatges de vàries màquines en un sol registre (veure més avall)
-F _HOSTNAME	Mostra tots els valors guardats en el registre del camp indicat
Escriure més d'un camp a la vegada és equivalent a un AND, a no ser que entre ells s'escrigui un "+"; interpretant-se llavors un OR. Per exemple: *journalctl _UID=1000 _PID=1234 + _UID=1001 _PID=1234* mostraria els missatges generats pel programa amb PID=1234 sempre que l'hagi executat o bé l'usuari 1000 o bé l'usuari 1001. Cal dir que si es repeteix varis cops el mateix camp, també es considerarà un OR.	
-g exprReg	Mostra només els missatges el valor del camp MESSAGE dels quals coincideixi amb l'expressió regular indicada. Per conèixer la sintaxi admesa es pot consultar *man pcre2pattern* Si l'expressió regular s'escriu tota en minúscules, la recerca serà "case-insensitive"; en cas contrari, serà "case-sensitive" (aquest comportament es pot modificar amb l'ús del paràmetre extra *--case-sensitive={yes|no}*
-u nomUnit	Mostra només els missatges generats (o relacionats) amb la "unit" indicat. Es poden fer servir comodins.
-t unaTag	Mostra només els missatges etiquetats amb el valor indicat, el qual es correspon al del camp SYSLOG_IDENTIFIER (normalment sol ser el nom de l'executable)
-p nomPrioritat	Les prioritats són (en lloc del seu nom es podria indicar el número associat): "debug" (7), "info" (6), "notice" (5), "warning" (4), "err" (3), "crit" (2), "alert" (1), "emerg" (0). Aquest paràmetre mostra només els missatges amb una prioritat igual o inferior a la indicada (és a dir, mostra missatges igual o més importants). També es pot escriure *-p pri1..pri2* para establir un rang concret de prioritats a mostrar.
-S xxxx	Mostra només els missatges ocorreguts des de la data i hora indicats NOTA:El format de la data ha de ser "YYYY-MM-DD hh:mm:ss"; si s'omet el temps s'assumeix 00:00:00; si s'ometen els segons s'assumeix :00; si s'omet la data s'assumeix avui. També es poden indicar les expressions "yesterday", "today" i "tomorrow", que es refereixen a la mitjanit del dia anterior, actual o següent, respectivament. També es pot escriure "now", que es refereix a la hora actual (útil amb *-f*). També es poden indicar temps relatius respecte l'hora actual amb – (abans) o + (després), així: "-15s", "-15m", "-15h", "-15d", "-15w", "-15M", "-15y"...(veure *man systemd.time* per més possibilitats)

-U xxx	Mostra només els missatges ocorreguts fins la data i hora indicats
-o { short | verbose | json | json-pretty | cat | export | short-iso}	Altera el format de la sortida mostrada per *journalctl*. NOTA: Per ordre, els valors indicats signifiquen el següent: *Mode por defecte (mostrant els camps data i hora, _HOSTNAME, _COMM, _PID i MESSAGE) *Mode verbós (mostra tots els camps de cada registre) *Mode (casi) verbós amb format JSON *Mode (casi) verbós amb format JSON però pensat per ser vist per humans *Mode que mostra només el valor del camp MESSAGE dels registres i res més *Format binari (pensat per a exportacions a través de la xarxa) *Format similar al per defecte però afegeix l'any a la data mostrada (per defecte no ho fa).Hi ha més variants de format de tipus "short-xxx" que modifiquen la presentació de la data i hora de forma diferent
--no-pager	No usa paginador per mostrar els missatges sinó que els "vomita" tots de cop (útil per encanonar la sortida a un altre comanda). Com efecte colateral, parteix les línies més llargues que el terminal per a què es puguin veure sense haver-lo de desplaçar horitzontalment..
--utc	Mostra la data de cada missatge en format Universal en lloc de Local (que és com Systemd les mostra per defecte)
-b [nº]	Filtra només els missatges pertanyents a l'ús de la màquina després de l'arranc nº indicat (on nº=0 -o no posar-lo- representa l'arranc actual, nº=-1 l'anterior, i així). Òbviament, hi haurà més d'un inici a escollir només si *systemd-journald* guarda els registres de forma permanent al disc i no pas a RAM. NOTA: Per veure tots els inicis possibles cal fer *journalctl –list-boots* El segon camp mostrat per aquesta comanda és el "boot ID", el qual serveix per referir-se a un arranc concret, i es podria usar, per exemple, per filtra d'aquesta manera: *journalctl _BOOT_ID=valorBootID*. A continuació del "boot ID" apareix la data i hora de la primera entrada guardada per cadascun d'aquests arrancs. Finalment, apareix la data i hora de la darrera entrada.
--system o ***--user***	El primer paràmetre mostra els missatges del kernel i dels serveis del sistema. El segon paràmetre mostra els missatges dels processos executats sota el context de l'usuari actual. Si no s'hi indica cap paràmetre, es mostraran tots els missatges que l'usuari actual pugui veure
--vacuum-size = nº{ K|M|G|T }	Esborra del disc les entrades necessàries (començant per les més antigues) per a què el registre ocupi en disc només el tamany indicat. No opera amb les entrades actives, només les arxivades.
--vacuum-time = nº{ s|m|h|d|w|M|y }	Esborra del disc les entrades més antigues que el temps indicat. Si s'invoca juntament amb *--vacuum-size* s'aplicarà la restricció que esborri més. Si un fitxer ".journal" inclou entrades més antigues però també més noves, no serà esborrat.
--flush	Demana a *systemd-journald* que esborri les dades emmagatzemades a la memòria RAM (específicament, a "/run/log/journal") i (si l'emmagatzematge persistent està habilitat), les gravi a "/var/log/journal". Hi ha un servei anomenat *systemd-journal-flush* que normalment s'invoca durant l'arranc del sistema (després d'encendre's el servei *systemd-journald*) i que és responsable d'executar aquesta ordre automàticament a cada inici.
--disk-usage	Tal com ja s'ha dit, mostra el tamany ocupat en disc pels arxius .journal (actius i arxivats)
--rotate	Força una rotació dels fitxers ".journal"
--verify	Comprova la integritat interna del registre
-D /ruta/carpeta/journal	Útil quan es vol inspeccionar des d'un sistema Live el journal (previ muntatge de la partició on resideix) d'un sistema no arrencat/arrencable
--file=arxiu.journal	Només mostra els missatges emmagatzemats físicament a l'arxiu ".journal" indicat (admet comodins)

NOTA: De moment no hi ha la possibilitat d'indicar filtres excloents del tipus "tot excepte això" (https://github.com/systemd/systemd/issues/2720)

NOTA: En el cas d'usar contenidors *systemd-nspawn*, existeix l'opció d'indicar si desitjem que els missatges de registre generats per aquests contenidors vagin a parar al journal del propi contenidor (creant-se llavors un enllaç a ells a la màquina amfitriona) o bé al journal de la màquina amfitriona (creant-se llavors un enllaç a ells dins del contenidor en qüestió); això es pot aconseguir afegint a la comanda *systemd-nspawn -b -M nomContenidor ...* (usada per iniciar el contenidor) el paràmetre *--link-journal* indicant-li el valor *"guest"* o *"host"*, respectivament. A partir de llavors, per veure els missatges de registre d'un determinat contenidor en la màquina amfitriona s'haurà d'executar indistintament la comanda *journalctl -m _HOSTNAME=nomContenedor ...*

Per enviar un missatge manualment al registre (acció molt útil en scripts, per exemple) existeix la comanda específica *systemd-cat*, que funciona així: *echo "Backup Failed" | systemd-cat -t "nomSimulatAplicacio" -p "crit"* (la data s'afegeix sola).O també així: *systemd-cat -t "nomSimulatAplicacio" -p "crit" echo "Backup Failed"* (faria el mateix que l'anterior però ara a més registra stderr). Una alternativa a *systemd-cat* és la venerable comanda *logger -s -t "nom" -p " crit"*

Transports

Per saber la via utilitzada pels diferents programes per enviar els seus respectius missatges al dimoni *systemd-journald* es pot consultar el valor del camp _TRANSPORT. Valors possibles són: "journal" (on els programes fan ús directament del mecanisme nadiu ofert per *systemd-journald* per rebre missatges), "syslog" (on els programes fan ús del mecanisme tradicional pre-Systemd però al que *systemd-journald* atén també), "kernel" (mecanisme exclusiu reservat per missatges provinents del kernel... existeix fins i tot un paràmetre exclusiu de *journalctl* per veure només aquest tipus de missatge, *-k*), "stdout" (via usada per *systemd-journald* per recopilar els missatges enviats pels diferents serveis del sistema a stdout) o "audit" (missatges provinents específicament del dimoni Audit), entre d'altres. Respecte això, és interessant llegir el següent quadre:

systemd-journald is a system service that collects and stores logging data. It creates and maintains structured, indexed journals based on logging information that is received from the kernel, from user processes via the libc syslog(3) call, from STDOUT/STDERR of system services or via its native API. As far as protocols are concerned, *systemd-journald* ...

- … is the listener on a stream socket named "**/run/systemd/journal/stdout**" systemd connects the raw standard outputs and errors of services (that have defaulted to or that explicitly have *StandardOutput=journal/StandardError=journal*) to this socket. It thus receives the protocol of variable length free-format records terminated with linefeeds.
- … is the listener on datagram sockets named "**/run/systemd/journal/dev-log**", which is symbolically linked from "/dev/log". This receives the protocol that the *syslog()* function in the GNU C library, linked into applications, speaks.
- … tries to be a client of another service listening on a datagram socket named "**/run/systemd/journal/syslog**". This also receives the protocol that the *syslog()* function in the GNU C library speaks (although *systemd-journald* actually uses another library and another function to speak it).
- … is a reader from a character device named "**/dev/kmsg**". This receives the protocol that the Linux kernel speaks, which is a protocol of variable length, largely free-format, records terminated with linefeeds.
- … is the listener on a datagram socket named "**/run/systemd/journal/socket**". This is analogous to the GNU C library case in that applications link to a library that speaks a certain protocol to this socket; except that the function is *sd_journal_sendv()*, it is in a systemd C library that applications link to, and the protocol is a systemd-only protocol comprising an array of key=value pairs, and optionally a readable file descriptor in each datagram.

Accés remot de logs

Si volem veure els missatges de registre d'una màquina remota es pot fer via HTTP gràcies a un "miniservidor web" específic anomenat *systemd-journal-gatewayd*, activable mitjançant socket (i que ve inclòs dins del paquet "**systemd-journal-remote**", el qual caldrà instal.lar prèviament). Si es mira la línia *ListenStream* del seu arxiu de configuració es pot comprovar que per defecte escolta a totes les interfícies pel port 19531 (procura doncs que aquest port estigui obert per l'eventual tallafocs que hi hagi al sistema!)

NOTA: Aquest servidor també pot oferir els registres via HTTP**S** si s'indica, a l'executable present a la línia *ExecStart=* del seu arxiu .service, un determinat certificat i clau privada (amb els paràmetres *--cert=* i *--key=*, respectivament).

Un cop funcionant el servidor "gatewayd" (*sudo systemctl start systemd-journal-gatewayd*), per veure els logs en una altra màquina via HTTP podem fer servir qualsevol client com ara un navegador (indicant el protocol "http://" perquè amb http**s**:// no funcionarà!!) o *curl*. En aquest darrer cas, es faria així:

curl -s -H "Accept:application/json" http://ip.Del.Servidor:19531/entries

on el valor de la capçalera de client Accept indica el format de les dades a rebre (altres formats vàlids són "text/plain" (valor per defecte, de fet) o "application/vnd.fdo.journal" (indica format verbós).

NOTA: També es pot afegir una altra capçalera per indicar un rang d'entrades concretes a obtenir, així: *"Range:entries=cursor[[:nºskip]:nºentrades]"*

Després de la ruta "/entries" es poden indicar diferents paràmetres en forma de "querystring" (és a dir, així */entries?param1&param2&param3...*), com ara:

boot : Similar al paràmetre *-b=0* de *journalctl*
follow : Similar al paràmetre *-f* de *journalctl*
nomCamp=valor : Per filtrar per un valor concret d'un camp propi del Journal (_UID, _HOSTNAME, etc)

D'altra banda, en substitució de la ruta "/entries" també es poden escriure aquestes altres rutes (teniu més informació sobre més rutes possibles a *man systemd-journal-gatewayd)*

/browse : Especialment pensat per utilitzar un navegador: permet visualitzar els registres de forma interactiva via web
/fields/nomCamp : Mostra els diferents valors trobats pel camp propi del Journal (_UID, _HOSTNAME, etc) indicat

NOTA: Trobareu més opcions a *man systemd-journal-gatewayd*

Enregistrament remot de logs

L'enregistrament en una determinada màquina central dels logs generats a màquines remotes és un procediment similar a l'explicat a l'apartat anterior: les màquines-generadores hauran de tenir funcionant cadascuna el servei *systemd-journal-gatewayd* i la màquina-magatzem central haurà d'obtenir d'elles els seus missatges demanant-los "activament" (ja sigui de forma periòdica o puntual) amb una comanda recopiladora específica (diferent, aquí sí, de la comanda *curl* utlitzada a l'apartat anterior).

Per a què aquest procediment funcioni caldrà tenir instal.lat prèviament el paquet "systemd-journal-remote" a totes les màquines, no solament les de tipus "generadores" sinó també a la màquina recopiladora central. A partir d'aquí, els passos per configurar la infraestructura necessària són els següents:

1.-A les màquines "generadores" de logs només cal fer una cosa, tal com hem dit: iniciar i/o habilitar el servei *systemd-journal-gatewayd.service* (o, alternativament, el socket *systemd-journal-gatewayd.socket*). En aquest sentit, doncs, aquest pas no és diferent de l'indicat a l'apartat anterior.

2.-Al sistema "recolector" cal tenir creada prèviament la carpeta "**/var/log/journal/remote**". A partir d'aquí, cada cop que es vulguin obtenir d'una determinada màquina remota els logs apareguts des de la darrera vegada que es va preguntar (i guardar-los en un fitxer local anomenat "/var/log/journal/remote/remote-ip.maq.gener.journal"), caldrà executar la següent comanda: *sudo /lib/systemd/systemd-journal-remote --url http://ip.maq.gener:19531* Es pot establir com una tasca programada, per exemple.

Per veure al servidor els registres obtinguts de màquines remots es pot executar la comanda *journalctl -m* , la qual mescla tots els registres ja siguin locals o remots, provinguin d'on provinguin; si es vol consultar només els d'una màquina en concret caldrà afegir a més el filtre *_HOSTNAME=nomMaquina*)

Hem de tenir en compte, però que la comanda anterior només funcionarà si l'executem com a "root" (amb *sudo*) ja que per defecte la carpeta "/var/log/journal/remote" i el seu contingut només tindran permisos de lectura per aquest usuari. Si volem poder veure els logs de màquines remotes sense haver de ser administrador del sistema, una opció seria (suposant que el nostre usuari pertany, això sí, al grup "systemd-journal"), executar la següent comanda: *sudo chown -R root:systemd-journal /var/log/journal/remote*

> NOTA: Recordeu que per afegir un usuari a un determinat grup (com ara "systemd-journal") simplement caldria executar *sudo usermod -a -G systemd-journal nomUsuari* El canvi s'aplicarà quan aquest usuari inicïi sessió de nou

També es pot (tal com ja s'indica a la taula groga anterior que llista els paràmetres més importants de la comanda *journalctl*) consultar el fitxer "*.journal" concret on es guarden els logs d'una determinada màquina desitjada (amb la comanda *journalctl --file /var/log/journal/remote/remote-ipMaqGen.journal*) o tots els fitxers "*.journal" que hi ha a una carpeta (amb *journalctl -D /var/log/journal/remote*)

EXERCICIS:

1.-a) Arrenca una màquina virtual on tinguis permisos d'administrador. Configura-hi el dimoni *systemd-journald* (i reinicia'l!) per a què emmagatzemi en disc tots els missatges de registre ocupant com a màxim 50M. Executa *journalctl --disk-usage* (o també *du -h /var/log/journal/xxxxx*) per comprovar que, efectivament, aquest màxim no se supera (o per molt poc, veure la nota següent).

> NOTA: És possible que el tamany del registre sigui una mica superior als 50M. Això és perquè existeix una directiva anomenada *SystemKeepFree* que contradiu a la que has especificat a l'exercici (*SystemMaxUse*) i on guanya la menys restrictiva; per defecte *SystemKeepFree* obliga a tenir lliure un 15% de l'espai de la partició on està ubicat el journal.

aII) Executa *ls -lh /var/log/journal/xxxxx* per veure els arxius "*.journal" que guarda el dimoni *systemd-journald* (bàsicament un anomenat "system.journal" i un altre anomenat "user-*uid*.journal" amb els seus arxius "arxivats" respectius) i confirmar que la sortida de la comanda *journalctl --disk-usage* anterior quadra amb la suma dels tamanys d'aquests fitxers.

b) Configura ara el dimoni *systemd-journald* (i reinicia'l!) per a què realitzi una rotació cada 10M. Pista: per saber com fer-ho llegeix la tercera "Nota" de la primera pàgina de teoria.

bII) Força una rotació manual de logs (encara que els fitxers de log no hagin arribat al tamany màxim indicat a l'apartat anterior) executant *journalctl --rotate* .¿Quins fitxers veus ara si tornes a executar *ls -lh /var/log/journal/xxxxx*? ¿Segueix quadrant amb el que ensenya ara la comanda *journalctl –disk-usage* ?

> NOTA: Segurament veuràs que Journald genera fitxers "arxivats" de 8MB (si són logs d'usuaris) o 16MB (si són logs de sistema) cada cop que es realitza una rotació forçada. Aquest tamany és degut a què a la configuració només s'indica un *MaxFileSize* (és a dir, un màxim) però no cap valor mínim. Si es realitza una rotació abans d'arribar al màxim, sigui quina sigui la quantitat real de bytes a arxivar, el tamany mínim predefinit per cada fitxer a arxivar és aquest valor de 8 o 16MB, i no es pot canviar.

bIII) Si tornes a executar un altre cop *journalctl --rotate*, ¿què veus ara amb les comandes *ls -lh /var/log/journal/xxxxx* i *journalctl --disk-usage*? Repeteix la rotació de logs fins arribar al màxim de 50M, ¿se supera en algun moment? Per què?

c) Executa *journalctl --vacuum-size=3M* i torna a comprovar l'espai ocupat pel registre amb les comandes *ls -lh /var/log/journal/xxxxx* i *journalctl --disk-usage*. ¿Què ha passat? ¿Per què creus que se supera el màxim marcat? D'altra banda, ¿per a què serviria la comanda *journalctl --vacuum-time=1d*?

d) Configura el dimoni *systemd-journald* (i reinicia'l!) per tal de què els missatges de log de tipus "info" (o més crítics) els mostri també al terminal virtual */dev/pts/1*. Comprova-ho (això ho pots fer, per exemple, aturant un servei com ara el Cups o qualsevol altre...aquest fet emetrà un missatge que hauries de veure al terminal */dev/pts/1*).

2.-a) Executa la comanda *journalctl* de la forma adient per mostrar per pantalla només els logs corresponents a la unit "polkit".

b) Executa la comanda *journalctl* de la forma adient per mostrar per pantalla només els logs corresponents al programa "sudo" que siguin de tipus "warning" (o pitjors)

c) Executa la comanda *journalctl* de la forma adient per mostrar per pantalla només els últims 5 logs corresponents a l'usuari amb l'UID 1000 (o el qui tingui el teu usuari en aquest moment)

d) Executa la comanda *journalctl -f* . Obre a continuació un terminal virtual qualsevol i executa-hi la comanda *timedatectl*. Torna a la pantalla on veies els logs i digues quines línies noves han aparegut i per què.

e) Executa la comanda *journalctl* de la forma adient per mostrar per pantalla només els logs corresponents a tot el que ha passat durant la darrera mitja hora

f) Executa la comanda *journalctl* per tal de mostrar en format JSON les dades de tots els missatges que continguin la paraula "auth".

g) Després de llegir la pàgina del manual de la comanda *dmesg* per esbrinar què fa (especialment interessants són els seus paràmetres: *-n*, *-T*, *-d* i *-w*), respon: ¿per què creus que les comandes *journalctl -k | wc -l* i *dmesg | wc -l* mostren el mateix número? Comprova també, a la pàgina del manual de la comanda *journalctl*, si executar *journalctl -k* seria equivalent a executar *journalctl -b _TRANSPORT=kernel* (o no).

> NOTA: El "kernel ring buffer" és una memòria temporal ("buffer") circular (és a dir, que si s'omple s'anirà eliminant el contingut més antic a mesura que entra el més nou) ubicat en RAM (en forma de dispositiu "/dev/kmsg") on es guarden els missatges generats per kernel des del principi de l'arranc (abans que es reconegui qualsevol dispositiu de disc ni que es posi en marxa cap servei de logging). D'altra banda, a "/proc/kmsg" hi ha una còpia actualitzada en temps real del contingut del ring buffer en format text, que és la que consulten programes com *journalctl*

gII) ¿Per què creus, en canvi, que les comandes *journalctl -k | wc -l* i *journalctl* --system | *wc -l* NO mostren el mateix número? Pista: recorda el significat del paràmetre --system. A partir d'aquí, ¿per què creus que la suma dels números mostrats per *journalctl --system | wc -l* i *journalctl* --user | *wc -l* és igual al número mostrat per *journalctl | wc -l* ?

h) Executa la comanda *journalctl* així, sense cap paràmetre. Seguidament, pulsa la tecla "/" i busca al llarg dels missatges de registre la paraula "session". Pots saltar d'una ocurrència a l'altra pulsant la tecla "N" (cap enrrera) o "n" (cap endavant); per sortir pots pulsar "q".

> NOTA: En comptes de "/" també podries fer servir "?", que fa la recerca des del final en comptes de des del principi (llavors, el significat de les tecles "n" i "N" s'inverteix). En general, es poden fer servir totes les tecles pròpies del visualitzador *less*, que és el que Journalctl fa servir per defecte (hi ha més explicades aquí: https://www.atareao.es/como/menos-es-mas-con-less)

i) Si s'estableix el valor de la variable d'entorn SYSTEMD_COLORS a "true" (per defecte no està definida), en el terminal on s'executi *journalctl*, se li estarà indicant que els codis de colors els mantingui tot i que la sortida no es vegi a pantalla (com passa, per exemple, si s'encanona a una altra comanda). Sabent això, digues què es mostra a pantalla en executar al terminal la següent línia de comandes: *SYSTEMD_COLORS=true journalctl -b -p4 | less -R* ¿Què veus si tornes a executar el mateix però sense escriure el paràmetre *-R* de la comanda *less*?

> NOTA: El fet que *journalctl* mostri per defecte els registres en color ve donat pel valor per defecte "yes" la directiva *LogColor*= present a l'arxiu "/etc/systemd/system.conf"

iII) Sabent que la llista de totes les variables d'entorn relacionades amb Systemd es pot consultar a l'apartat "ENVIRONMENT" de la pàgina *man systemd*, esbrina per a què serveixen les variables *SYSTEMD_PAGER* i *SYSTEMD_LESS* . I també què passaria si la variable *SYSTEMD_LOG_LEVEL* valgués "debug".

j) Envia manualment amb *systemd-cat* un missatge al registre que tingui l'etiqueta "VIRUS" amb prioritat crítica i que com a texte mostri "Atenció, la pantalla explotarà". Comprova amb *journalctl -e* que ha arribat.

3.-Assegura't que la teva màquina virtual tingui la seva tarja de xarxa en mode "adaptador pont". Posa en marxa una altra màquina virtual, també amb la seva tarja de xarxa en mode "adaptador pont". Si no tens una altra màquina, pots crear-ne un clon de la primera (també et pots posar d'acord amb algun company per fer aquest exercici en parelles o bé, si ets "root" a la màquina amfitriona, usar la màquina amfitriona com a segona màquina).

a) Segueix les instruccions indicades a l'apartat "Accés remot de logs" per tal veure en el navegador d'una màquina els logs de l'altra.

b) Segueix les instruccions indicades a l'apartat "Enregistrament remot de logs" per tal implementar la recollida de logs remots Comprova-ho executant la comanda *journalctl* a la màquina "recollidora" fent servir el paràmetre *-m* juntament amb el filtre *_HOSTNAME=nomMaqRemota*