Jak używać journalctl do przeglądania i manipulowania dziennikami systemowymi

Wprowadzenie

Niektóre z najbardziej przekonujących zalet systemd są związane z rejestrowaniem procesów i systemu. Podczas korzystania z innych narzędzi dzienniki są zwykle rozproszone w całym systemie, obsługiwane przez różne demony i procesy, i mogą być dość trudne do interpretacji, gdy obejmują wiele aplikacji. Systemd próbuje rozwiązać te problemy, zapewniając scentralizowane rozwiązanie do zarządzania do rejestrowania wszystkich procesów jądra i przestrzeni użytkownika. System, który zbiera te dzienniki i zarządza nimi, jest znany jako dziennik.

Dziennik jest implementowany za pomocą demona journald, który obsługuje wszystkie komunikaty generowane przez jądro, initrd, usługi itp. Omówimy sposób korzystania z narzędzia journalctl, które może służyć do uzyskiwania dostępu do danych przechowywanych w nim i manipulowania nimi.

Główny pomysł

Jednym z impulsów stojących za systemd jest centralizacja zarządzania dziennikami, niezależnie od tego, skąd pochodzą wiadomości. Ponieważ większość procesu rozruchu i zarządzania usługami jest obsługiwana przez proces systemd, sensowne jest ustandaryzowanie sposobu gromadzenia dzienników i uzyskiwania do nich dostępu. Demon journald zbiera dane ze wszystkich dostępnych źródeł i przechowuje je w formacie binarnym dla łatwej i dynamicznej manipulacji.

Daje nam to wiele istotnych zalet. Poprzez interakcję z danymi za pomocą jednego narzędzia administratorzy mogą dynamicznie wyświetlać dane dziennika zgodnie ze swoimi potrzebami. Może to być tak proste, jak przeglądanie danych rozruchowych sprzed trzech rozruchów lub łączenie wpisów dziennika kolejno z dwóch powiązanych usług w celu debugowania problemu z komunikacją.

Przechowywanie danych dziennika w formacie binarnym oznacza również, że dane mogą być wyświetlane w dowolnych formatach wyjściowych, w zależności od potrzeb. Na przykład, do codziennego zarządzania logami możesz być przyzwyczajony do przeglądania logów w standardowym formacie syslog, ale jeśli zdecydujesz się później wykreślić przerwy w usłudze, możesz wyprowadzić każdy wpis jako obiekt JSON, aby uczynić go użytecznym w postaci graficznej. Ponieważ dane nie są zapisywane na dysk w postaci zwykłego tekstu, nie jest wymagana konwersja, gdy potrzebny jest inny format na żądanie.

systemd może być używany zarówno z istniejącą implementacją syslog, lub może zastąpić funkcje syslog, w zależności od potrzeb. Chociaż dziennik systemd zaspokoi większość potrzeb związanych z rejestrowaniem zdarzeń przez administratora, może również uzupełniać istniejące mechanizmy rejestrowania. Na przykład możesz mieć scentralizowany serwer syslog, którego używasz do kompilacji danych z wielu serwerów, ale możesz także chcieć przeplatać dzienniki z wielu usług w jednym systemie z dziennikiem systemd. Możesz to zrobić, łącząc te technologie.

1. Ustawianie czasu systemowego

Jedną z zalet korzystania z dziennika binarnego do rejestrowania jest możliwość przeglądania rekordów dziennika w UTC lub w czasie lokalnym do woli. Domyślnie systemd wyświetla wyniki w czasie lokalnym.

Z tego powodu, zanim zaczniemy korzystać z dziennika, upewnimy się, że strefa czasowa jest poprawnie skonfigurowana. systemd Apartament faktycznie jest wyposażony w narzędzie o nazwie timedatectl, które mogą pomóc w tym.

Najpierw sprawdź, jakie strefy czasowe są dostępne opcją z list-timezones:

timedatectl list-timezones

Spowoduje to wyświetlenie stref czasowych dostępnych w systemie. Gdy znajdziesz tą, która pasuje do lokalizacji twojego serwera, możesz go ustawić, korzystając z opcji set-timezone:

sudo timedatectl set-timezone zone

Aby mieć pewność, że Twój komputer używa teraz właściwego czasu, użyj timedatectl polecenia samodzielnie lub z status opcją. Wyświetlacz będzie taki sam:

timedatectl status

Local time: Thu 2020-02-05 14:08:06 UTC

Universal time: Thu 2020-02-05 19:08:06 UTC

RTC time: Thu 2020-02-05 19:08:06

Time zone: Etc/UTC (UTC, -0000)

NTP service: active

NTP synchronized: yes

RTC in local TZ: no

DST active: n/a

Pierwszy wiersz powinien wyświetlać prawidłowy czas.

Podstawowe przeglądanie dziennika

Aby wyświetlić dzienniki journald zebrane przez demona, użyj polecenia journalctl.

W przypadku korzystania z niego osobno każda pozycja dziennika w systemie będzie wyświetlana na stronie w celu przeglądania. Najstarsze wpisy będą na górze:

journalctl

-- Logs begin at Tue 2020-02-03 21:48:52 UTC, end at Tue 2020-02-03 22:29:38 UTC. --

Feb 03 21:48:52 localhost.localdomain systemd-journal[243]: Runtime journal is using 6.2M (max allowed 49.

Feb 03 21:48:52 localhost.localdomain systemd-journal[243]: Runtime journal is using 6.2M (max allowed 49.

Feb 03 21:48:52 localhost.localdomain systemd-journald[139]: Received SIGTERM from PID 1 (systemd).

Feb 03 21:48:52 localhost.localdomain kernel: audit: type=1404 audit(1423000132.274:2): enforcing=1 old_en

Feb 03 21:48:52 localhost.localdomain kernel: SELinux: 2048 avtab hash slots, 104131 rules.

Feb 03 21:48:52 localhost.localdomain kernel: SELinux: 2048 avtab hash slots, 104131 rules.

Feb 03 21:48:52 localhost.localdomain kernel: input: ImExPS/2 Generic Explorer Mouse as /devices/platform/

Feb 03 21:48:52 localhost.localdomain kernel: SELinux: 8 users, 102 roles, 4976 types, 294 bools, 1 sens,

Feb 03 21:48:52 localhost.localdomain kernel: SELinux: 83 classes, 104131 rules

. . .

Prawdopodobnie będziesz mieć strony danych do przewijania, które mogą mieć dziesiątki lub setki tysięcy linii, jeśli systemd były w twoim systemie przez dłuższy czas. Pokazuje to, ile danych jest dostępnych w bazie danych dziennika.

Format będzie znany osobom przyzwyczajonym do standardowego syslog rejestrowania.
Jednak w rzeczywistości gromadzi to dane z większej liczby źródeł niż syslog jest w stanie wykonać tradycyjne implementacje. Obejmuje dzienniki z wczesnego procesu rozruchu, jądra, initrd i standardowy błąd aplikacji i obecne.

Możesz zauważyć, że wszystkie wyświetlane znaczniki czasu są czasem lokalnym. Jest to dostępne dla każdego wpisu dziennika, gdy mamy już poprawnie ustawiony czas lokalny w naszym systemie. Wszystkie dzienniki są wyświetlane przy użyciu tych nowych informacji.

Jeśli chcesz wyświetlić znaczniki czasu w UTC, możesz użyć flagi --utc:

journalctl --utc

2. Filtrowanie dzienników według czasu

Chociaż dostęp do tak dużego zbioru danych jest zdecydowanie przydatny, tak duża ilość informacji może być trudna lub niemożliwa do kontrolowania i przetwarzania. Z tego powodu jedną z najważniejszych cech journalctl są opcje filtrowania.

Wyświetlanie dzienników z bieżącego rozruchu

Najbardziej podstawowym z nich, z którego możesz korzystać codziennie, jest flaga -b. Spowoduje to wyświetlenie wszystkich pozycji kroniki zebranych od ostatniego restartu.

journalctl -b

Pomoże Ci to zidentyfikować i zarządzać informacjami, które są istotne dla twojego aktualnego środowiska.

W przypadkach, gdy nie korzystasz z tej funkcji i wyświetlasz więcej niż jeden dzień rozruchu, zobaczysz, że journalctl wstawiono linię, która wygląda następująco, gdy system ulegnie awarii:

. . .

-- Reboot --

. . .

Można to wykorzystać, aby pomóc logicznie podzielić informacje na sesje rozruchowe.

Past Boots

Chociaż zwykle będziesz chciał wyświetlać informacje z bieżącego rozruchu, z pewnością są chwile, w których wcześniejsze rozruchy również byłyby pomocne. W dzienniku można zapisać informacje z wielu poprzednich uruchomień, dzięki czemu journalctl można łatwo wyświetlać te informacje.

Niektóre dystrybucje domyślnie umożliwiają zapisywanie poprzednich informacji o rozruchu, podczas gdy inne wyłączają tę funkcję. Aby włączyć trwałe informacje o rozruchu, możesz utworzyć katalog do przechowywania dziennika, wpisując:

sudo mkdir -p /var/log/journal

Lub możesz edytować plik konfiguracji dziennika:

sudo nano /etc/systemd/journald.conf

W sekcji [Journal] ustaw Storage=opcję „trwałe”, aby włączyć trwałe rejestrowanie:

/etc/systemd/journald.conf

. . .

[Journal]

Storage=persistent

Gdy na serwerze jest włączone zapisywanie poprzednich rozruchów, journalctl udostępnia kilka poleceń, które pomogą w pracy z rozruchami jako jednostką podziału. Aby zobaczyć rozruchy, o których wie journald użyj opcji --list-boots z journalctl:

journalctl --list-boots

-2 caf0524a1d394ce0bdbcff75b94444fe Tue 2020-02-03 21:48:52 UTC-- Mon2020-02-03 22:17:00 UTC

-1 13883d180dc0420db0abcb5fa26d6198 Tue 2020-02-03 22:17:03 UTC-- Mon2020-02-03 22:19:08 UTC

0 bed718b17a73415fade0e4e7f4bea609 Tue 2020-02-03 22:19:12 UTC-- Mon2020-02-03 23:01:01 UTC

Spowoduje to wyświetlenie wiersza dla każdego rozruchu. Pierwsza kolumna to wartość dla rozruchu, za pomocą której można łatwo odwołać się do rozruchu journalctl. Jeśli potrzebujesz bezwzględnego odwołania, identyfikator rozruchu znajduje się w drugiej kolumnie. Czas, o którym mówi sesja rozruchu, można określić na podstawie dwóch specyfikacji czasu podanych pod koniec.

Aby wyświetlić informacje z tych rozruchów, możesz użyć informacji z pierwszej lub drugiej kolumny.

Na przykład, aby zobaczyć dziennik z poprzedniego rozruchu, użyj -1względnego wskaźnika z -b flagą:

journalctl -b -1

Możesz także użyć identyfikatora rozruchu, aby oddzwonić do danych z rozruchu:

journalctl -b caf0524a1d394ce0bdbcff75b94444fe – tu wpisz odczytany identyfikator

3. Czas okna

Podczas gdy wyświetlanie wpisów w dzienniku po rozruchu jest niezwykle przydatne, często możesz chcieć poprosić o okna czasowe, które nie są dobrze dopasowane do rozruchu systemu. Może to być szczególnie przydatne, gdy mamy do czynienia z długo działającymi serwerami o znacznym czasie sprawności.

Możesz filtrować według dowolnych limitów czasowych za pomocą opcji --sincei --until, które ograniczają pozycje wyświetlane odpowiednio do po lub przed danym czasem.

Wartości czasu mogą występować w różnych formatach. W przypadku bezwzględnych wartości czasu należy użyć następującego formatu:

YYYY-MM-DD HH:MM:SS

Na przykład możemy zobaczyć wszystkie wpisy od 10 stycznia 2020 o 17:15, wpisując:

journalctl --since "2020-04-27 17:13:15"

Jeśli składniki powyższego formatu zostaną pominięte, zostaną zastosowane niektóre wartości domyślne. Na przykład, jeśli data zostanie pominięta, założona zostanie bieżąca data.
Jeśli brakuje komponentu czasu, „00:00:00” (północ) zostanie zastąpione.
Pole sekund można również pominąć i ustawić domyślnie na „00”:

journalctl --since "2020-04-27" --until "2020-04-27 17:13"

Dziennik rozumie również niektóre wartości względne i nazwane skróty. Na przykład możesz użyć słów „wczoraj”, „dziś”, „jutro” lub „teraz”. Czasy względne utworzysz, dodając „-” lub „+” do wartości numerowanej lub używając słów takich jak „temu” w konstrukcji zdania.

Aby uzyskać dane z wczoraj, możesz wpisać:

journalctl --since yesterday

Jeśli otrzymałeś raporty o zakłóceniach usługi rozpoczynających się o 9:00 i trwających do godziny temu, możesz wpisać:

journalctl --since 09:00 --until "1 hour ago"

Jak widać, stosunkowo łatwo jest zdefiniować elastyczne okna czasowe do filtrowania wpisów, które chcesz zobaczyć.

Filtrowanie według zainteresowań wiadomości

Nauczyliśmy się powyżej kilku sposobów filtrowania danych dziennika przy użyciu ograniczeń czasowych. W tej sekcji omówimy sposób filtrowania w oparciu o interesującą Cię usługę lub komponent. Dziennik Systemd zawiera różne sposoby na zrobienie tego.

4. Według jednostki

Być może najbardziej przydatnym sposobem filtrowania jest jednostka, którą jesteś zainteresowany. Możemy użyć opcji -u filtrowania w ten sposób.

Na przykład, aby zobaczyć wszystkie dzienniki z jednostki Nginx w naszym systemie, możemy wpisać:

journalctl -u nginx.service

Zazwyczaj prawdopodobnie chcesz również filtrować według czasu, aby wyświetlić linie, które Cię interesują. Na przykład, aby sprawdzić, jak usługa działa dzisiaj, możesz wpisać:

journalctl -u nginx.service --since today

Ten rodzaj skupienia staje się niezwykle pomocny, gdy korzystasz ze zdolności dziennika do przeplatania rekordów z różnych jednostek. Na przykład, jeśli proces Nginx jest podłączony do jednostki PHP-FPM w celu przetworzenia zawartości dynamicznej, można scalić wpisy z obu w kolejności chronologicznej, określając obie jednostki:

journalctl -u nginx.service -u php-fpm.service --since today

Może to znacznie ułatwić wykrywanie interakcji między różnymi programami i systemami debugowania zamiast poszczególnych procesów.

Według ID procesu, użytkownika lub grupy

Niektóre usługi sprawują różnorodne procesy potomne. Jeśli sprawdziłeś dokładny PID procesu, który Cię interesuje, możesz go również przefiltrować.

W tym celu możemy filtrować, określając pole PID. Na przykład, jeśli interesujący nas PID to 8088, możemy wpisać:

journalctl PID=8088

W innym przypadku możesz chcieć pokazać wszystkie wpisy zarejestrowane od określonego użytkownika lub grupy. Można to zrobić za pomocą filtrów _UID lub _GID. Na przykład, jeśli serwer WWW działa pod użytkownikiem www-data, możesz znaleźć identyfikator użytkownika, wpisując:

id -u www-data

33

Następnie można użyć zwróconego identyfikatora do filtrowania wyników dziennika:

journalctl UID=33 --since today

Dziennik systemd ma wiele pól, które mogą być używane do filtrowania. Niektóre z nich są przekazywane z rejestrowanego procesu, a niektóre są stosowane przy journald użyciu informacji zebranych z systemu w momencie rejestrowania.

Wiodący znak podkreślenia wskazuje, że pole PID jest tego ostatniego typu. Dziennik automatycznie rejestruje i indeksuje PID procesu, który się loguje w celu późniejszego filtrowania. Możesz dowiedzieć się o wszystkich dostępnych polach dziennika, wpisując:

man systemd.journal-fields

Omówimy niektóre z nich w tym przewodniku. Na razie jednak omówimy jeszcze jedną przydatną opcję związaną z filtrowaniem według tych pól. Opcja -F może być stosowany w celu pokazania wszystkich dostępnych wartości dla danej dziedzinie dziennika.

Na przykład, aby zobaczyć, dla których identyfikatorów grup dziennik systemd ma wpisy, możesz wpisać:

journalctl -F _GID

102

133

81

84

100

0

1

Spowoduje to wyświetlenie wszystkich wartości zapisanych w dzienniku dla pola identyfikatora grupy. Może to pomóc w konstruowaniu filtrów.

5. Według ścieżki komponentu

Możemy również filtrować, podając lokalizację ścieżki.

Jeśli ścieżka prowadzi do pliku wykonywalnego, journalctl wyświetli wszystkie wpisy dotyczące danego pliku wykonywalnego. Na przykład, aby znaleźć te wpisy, które dotyczą pliku wykonywalnego bash, możesz wpisać:

journalctl /usr/bin/bash

Zwykle, jeśli jednostka jest dostępna dla pliku wykonywalnego, metoda ta jest czystsza i zapewnia lepsze informacje (wpisy z powiązanych procesów potomnych itp.). Czasami jednak nie jest to możliwe.

6. Wyświetlanie komunikatów jądra

Komunikaty jądra, zwykle znajdujące się w danych wyjściowych dmesg, można również pobrać z dziennika.

Aby wyświetlić tylko te wiadomości, możemy dodać flagi -k lub --dmesg do naszego polecenia:

journalctl -k

Domyślnie wyświetli to komunikaty jądra z bieżącego rozruchu. Możesz określić alternatywny rozruch przy użyciu normalnych flag wyboru rozruchu omówionych wcześniej. Na przykład, aby uzyskać wiadomości sprzed pięciu rozruchów, możesz wpisać:

journalctl -k -b -5

7. Według priorytetu

Jednym z filtrów, którym często interesują się administratorzy systemu, jest priorytet wiadomości. Chociaż często przydatne jest rejestrowanie informacji na bardzo szczegółowym poziomie, podczas gdy faktycznie niszczą dostępne informacje, dzienniki o niskim priorytecie mogą być mylące.

journalctl za pomocą opcji -p można wyświetlać tylko wiadomości o określonym priorytecie lub wyższym. Pozwala to odfiltrować wiadomości o niższym priorytecie.

Na przykład, aby wyświetlić tylko wpisy zarejestrowane na poziomie błędu lub wyższym, możesz wpisać:

journalctl -p err -b

Spowoduje to wyświetlenie wszystkich wiadomości oznaczonych jako błąd, krytyczny, alert lub awaria. Dziennik implementuje standardowe poziomy komunikatów syslog. Możesz użyć nazwy priorytetu lub odpowiadającej jej wartości liczbowej. W kolejności od najwyższego do najniższego priorytetu są to:

0: emerg	0: emerg
1: alert	1: alert
2: kryt	2: crit
3: err	3: err
4: ostrzeżenie	4: warning
5: zawiadomienie	5: notice
6: informacje	6: info
7: debugowanie	7: debug

Powyższe liczby lub nazwy mogą być używane zamiennie z opcją -p. Wybranie priorytetu spowoduje wyświetlenie wiadomości oznaczonych na określonym poziomie i tych powyżej.

Modyfikowanie wyświetlania dziennika

Powyżej zademonstrowaliśmy wybór wpisów poprzez filtrowanie. Istnieją jednak inne sposoby modyfikowania danych wyjściowych. Możemy dostosować journalctl do różnych potrzeb.

8. Obetnij lub rozwiń wyjście

Możemy dostosować sposób wyświetlania danych journalctl, nakazując im zmniejszenie lub rozszerzenie wyniku.

Domyślnie journalctl wyświetla cały wpis na ekranie, umożliwiając przejście do pozycji z prawej strony ekranu. Dostęp do tych informacji można uzyskać, naciskając klawisz strzałki w prawo.

Jeśli wolisz wyciąć dane wyjściowe, wstawiając wielokropek, w którym informacje zostały usunięte, możesz użyć opcji --no-full:

journalctl --no-full

. . .

Feb 04 20:54:13 journalme sshd[937]: Failed password for root from 83.234.207.60...h2

Feb 04 20:54:13 journalme sshd[937]: Connection closed by 83.234.207.60 [preauth]

Feb 04 20:54:13 journalme sshd[937]: PAM 2 more authentication failures; logname...ot

Możesz także iść w przeciwnym kierunku i nakazać journalctlwyświetlanie wszystkich jego informacji, niezależnie od tego, czy zawiera znaki niedrukowalne. Możemy to zrobić za pomocą flagi -a:

journalctl -a

9. Wyjście do standardowego wyjścia

Domyślnie journalctl wyświetla dane wyjściowe w ekranie dla łatwiejszego odczytania. Jeśli jednak planujesz przetwarzanie danych za pomocą narzędzi do manipulacji tekstem, prawdopodobnie chcesz mieć możliwość wyjścia na standardowe wyjście.

Możesz to zrobić z opcją --no-pager:

journalctl --no-pager

Można to natychmiast przesłać do narzędzia przetwarzającego lub przekierować do pliku na dysku, w zależności od potrzeb.

Formaty wyjściowe

W przypadku przetwarzania pozycji dziennika, jak wspomniano powyżej, najprawdopodobniej łatwiej będzie parsować dane, jeśli są one w formacie używalnym. Na szczęście dziennik może być wyświetlany w różnych formatach w zależności od potrzeb. Możesz to zrobić za pomocą opcji -o ze specyfikatorem formatu.

Na przykład możesz wyprowadzać zapisy księgowe w JSON, wpisując:

journalctl -b -o json

{ "__CURSOR" : "s=13a21661cf4948289c63075db6c25c00;i=116f1;b=81b58db8fd9046ab9f847ddb82a2fa2d;m=19f0daa;t=50e33c33587ae;x=e307daadb4858635", "__REALTIME_TIMESTAMP" : "1422990364739502", "__MONOTONIC_TIMESTAMP" : "27200938", "_BOOT_ID" : "81b58db8fd9046ab9f847ddb82a2fa2d", "PRIORITY" : "6", "_UID" : "0", "_GID" : "0", "_CAP_EFFECTIVE" : "3fffffffff", "_MACHINE_ID" : "752737531a9d1a9c1e3cb52a4ab967ee", "_HOSTNAME" : "desktop", "SYSLOG_FACILITY" : "3", "CODE_FILE" : "src/core/unit.c", "CODE_LINE" : "1402", "CODE_FUNCTION" : "unit_status_log_starting_stopping_reloading", "SYSLOG_IDENTIFIER" : "systemd", "MESSAGE_ID" : "7d4958e842da4a758f6c1cdc7b36dcc5", "_TRANSPORT" : "journal", "_PID" : "1", "_COMM" : "systemd", "_EXE" : "/usr/lib/systemd/systemd", "_CMDLINE" : "/usr/lib/systemd/systemd", "_SYSTEMD_CGROUP" : "/", "UNIT" : "nginx.service", "MESSAGE" : "Starting A high performance web server and a reverse proxy server...", "_SOURCE_REALTIME_TIMESTAMP" : "1422990364737973" }

. . .

Jest to przydatne do analizowania za pomocą narzędzi. Można użyć tego formatu json-pretty, aby uzyskać lepszą obsługę struktury danych przed przekazaniem jej konsumentowi JSON:

journalctl -b -o json-pretty

{

"__CURSOR" : "s=13a21661cf4948289c63075db6c25c00;i=116f1;b=81b58db8fd9046ab9f847ddb82a2fa2d;m=19f0daa;t=50e33c33587ae;x=e307daadb4858635",

"__REALTIME_TIMESTAMP" : "1422990364739502",

"__MONOTONIC_TIMESTAMP" : "27200938",

"_BOOT_ID" : "81b58db8fd9046ab9f847ddb82a2fa2d",

"PRIORITY" : "6",

"_UID" : "0",

"_GID" : "0",

"_CAP_EFFECTIVE" : "3fffffffff",
"_MACHINE_ID" : "752737531a9d1a9c1e3cb52a4ab967ee",
"_HOSTNAME" : "desktop",
"SYSLOG_FACILITY" : "3",
"CODE_FILE" : "src/core/unit.c",
"CODE_LINE" : "1402",
"CODE_FUNCTION" : "unit_status_log_starting_stopping_reloading",
"SYSLOG_IDENTIFIER" : "systemd",
"MESSAGE_ID" : "7d4958e842da4a758f6c1cdc7b36dcc5",
"_TRANSPORT" : "journal",
"_PID" : "1",
"_COMM" : "systemd",
"_EXE" : "/usr/lib/systemd/systemd",
"_CMDLINE" : "/usr/lib/systemd/systemd",
"_SYSTEMD_CGROUP" : "/",
"UNIT" : "nginx.service",
"MESSAGE" : "Starting A high performance web server and a reverse proxy server...",
"_SOURCE_REALTIME_TIMESTAMP" : "1422990364737973"
}

. . .

Do wyświetlania można użyć następujących formatów:

cat: Wyświetla tylko samo pole komunikatu.

eksport: format binarny odpowiedni do przesyłania lub tworzenia kopii zapasowych.

json: Standardowy JSON z jednym wpisem w wierszu.

json-pretty: JSON sformatowany dla lepszej czytelności dla człowieka

json-sse: sformatowane dane wyjściowe JSON są opakowane, aby dodać zdarzenie wysłane przez serwer

short: Domyślny syslogstyl wyjściowy

short-iso: Domyślny format rozszerzony w celu wyświetlania znaczników czasowych zegara ściennego ISO 8601.

short-monotonic: Domyślny format z monotonicznymi znacznikami czasu.

short-precision: Domyślny format z mikrosekundową precyzją

verbose: Pokazuje wszystkie pola dziennika dostępne dla pozycji, w tym te zwykle ukryte wewnętrznie.

Te opcje pozwalają wyświetlać zapisy księgowe w formacie, który najlepiej odpowiada Twoim bieżącym potrzebom.

Aktywny monitoring procesu

Komenda journalctl imituje jak wielu administratorów wykorzystywać tail do monitorowania aktywnego lub niedawnej działalności. Ta funkcja jest wbudowana journalctl, umożliwiając dostęp do tych funkcji bez konieczności łączenia się z innym narzędziem.

10. Wyświetlanie ostatnich dzienników

Aby wyświetlić określoną liczbę rekordów, możesz użyć opcji -n, która działa dokładnie tak jak tail -n.

Domyślnie wyświetla 10 ostatnich wpisów:

journalctl -n

Możesz określić liczbę wpisów, które chcesz zobaczyć, za pomocą liczby po -n:

journalctl -n 20

Następujące dzienniki

Aby aktywnie śledzić zapisywane dzienniki, możesz użyć flagi -f. Ponownie działa to tak, jak można się spodziewać, jeśli masz doświadczenie w korzystaniu z tail -f:

journalctl -f

Konserwacja dziennika

Być może zastanawiasz się nad kosztem przechowywania wszystkich danych, które widzieliśmy do tej pory. Możesz być zainteresowany w czyszczeniu niektórych starszych dzienników i zwalnianiu miejsca.

11. Znajdowanie bieżącego użycia dysku

Można dowiedzieć się jak dużo miejsca, dzienniki obecnie zajmują na dysku za pomocą flagi --disk-usage:

journalctl --disk-usage

Journals take up 72.0M on disk.

12. Usuwanie starych dzienników

Jeśli chcesz zmniejszyć dziennik, możesz to zrobić na dwa różne sposoby (dostępne w systemd wersji 218 i nowszych).

Jeśli skorzystasz z opcji --vacuum-size, możesz zmniejszyć dziennik, wskazując rozmiar. Spowoduje to usunięcie starych pozycji, dopóki łączne miejsce w dzienniku zajmowane na dysku nie osiągnie żądanego rozmiaru:

sudo journalctl --vacuum-size=1G

Innym sposobem na zmniejszenie dziennika jest zapewnienie czasu granicznego z opcją --vacuum-time. Wszelkie wpisy, poza tym czasem zostaną usunięte. Pozwala to zachować wpisy utworzone po określonym czasie.

Na przykład, aby zatrzymać wpisy z ostatniego roku, możesz wpisać:

sudo journalctl --vacuum-time=1years

13. Ograniczanie rozszerzenia dziennika

Możesz skonfigurować serwer tak, aby ograniczał ilość miejsca, które dziennik może zająć. Można to zrobić, edytując plik /etc/systemd/journald.conf.

Aby ograniczyć wzrost dziennika, można użyć następujących elementów:

SystemMaxUse=: Określa maksymalną przestrzeń dyskową, którą dziennik może wykorzystać w pamięci trwałej.

SystemKeepFree=: Określa ilość miejsca, które dziennik powinien pozostawić wolne podczas dodawania pozycji dziennika do pamięci trwałej.

SystemMaxFileSize=: Kontroluje, jak duże pojedyncze pliki dziennika mogą rosnąć w trwałym magazynie przed ich obróceniem.

RuntimeMaxUse=: Określa maksymalne miejsce na dysku, które można wykorzystać w pamięci ulotnej (w /runsystemie plików).

RuntimeKeepFree=: Określa ilość miejsca, które ma zostać przeznaczone na inne zastosowania podczas zapisywania danych w pamięci ulotnej (w systemie plików /run).

RuntimeMaxFileSize=: Określa ilość miejsca, które pojedynczy plik dziennika może zająć w pamięci ulotnej (w systemie plików /run) przed obróceniem.

Ustawiając te wartości, możesz kontrolować, w jaki sposób journald zużywa i oszczędza miejsce na serwerze. Pamiętaj, że SystemMaxFileSizei RuntimeMaxFileSize będzie ukierunkowane na zarchiwizowane pliki, aby osiągnąć określone limity. Należy o tym pamiętać podczas interpretowania liczby plików po operacji odkurzania.

14. Wniosek

Jak widać, dziennik systemd jest niezwykle przydatny do gromadzenia danych o systemie i aplikacjach oraz zarządzania nimi. Większość elastyczności wynika z automatycznie rejestrowanych obszernych metadanych i scentralizowanego charakteru dziennika. Polecenia journalctl ułatwia skorzystanie z zaawansowanych funkcji i wykonanie obszernej analizy i debugowanie różnych komponentów aplikacji.