Engineer

Podman

рушій контейнерів, якому не потрібен демон

Що це таке

Podman — рушій контейнерів для Linux, який збирає, завантажує й запускає ті самі OCI-образи, що й Docker, але без центрального сервера посередині. Кожна команда `podman` виконує роботу у власному процесі й завершується. Запущений контейнер працює далі під невеликим процесом-наглядачем conmon, а той є дочірнім процесом того, хто його запустив: оболонки, скрипту чи systemd. Проєкт сам називає свій інструмент бездемонним, відкритим і рідним для Linux і пише, що більшість користувачів може просто зробити псевдонім `docker` для `podman` і працювати далі.

Назва походить від pod manager. Под — поняття з Kubernetes: група контейнерів зі спільним мережевим простором імен, тож вони бачать одне одного через localhost. Podman уміє створювати поди напряму, а `podman kube play` запускає контейнери й поди з YAML-файлу Kubernetes. Тому це зручне місце, щоб перевірити маніфест, перш ніж він потрапить до кластера.

У щоденній роботі його вирізняють дві речі. Перша — безкореневий режим. Звичайний користувач може запускати контейнери, бо простори імен користувачів у ядрі роблять його UID 0 усередині контейнера, а решту UID відображають у підлеглий діапазон із `/etc/subuid` і `/etc/subgid`. Мережа для безкореневих контейнерів іде через pasta, мережевий стек у просторі користувача, адже непривілейований користувач не має права переписувати фільтр пакетів хоста.

Друга — Quadlet. Файл Quadlet — короткий декларативний опис контейнера, тому, мережі, пода чи збирання образу, написаний у тому самому ini-синтаксисі, що й модуль systemd, і збережений у `/etc/containers/systemd/` для всієї системи або в `~/.config/containers/systemd/` для одного користувача. Під час завантаження та після кожного `systemctl`

`daemon-reload` генератор systemd перетворює `web.container` на звичайний `web.service`. Старіша команда `podman generate systemd` досі працює, проте проєкт оголосив її застарілою на користь Quadlet.

Застосуйте поняття до рішення

Застосуйте «Що таке Podman?» до конкретного рішення. Ми можемо пов'язати поняття із системою, яку ви створюєте або купуєте. Запакована так, що поводиться однаково на ноутбуці, на тестовому сервері й у продакшні, і масштабується, коли настає напружений тиждень.

Переглянути наші послуги Зв'язатися з нами →

Перевірте припущення за терміном

Використайте пояснення та пов'язані роботи, щоб сформулювати запитання для проєкту. Ми можемо перевірити припущення разом до вибору інструмента. Вела 20 застосунків Docker у Google Cloud і AWS — 20 контейнеризованих застосунків розгорнуто й поставлено під моніторинг

Вела 20 застосунків Docker у Google Cloud і AWS Зв'язатися з нами →

Чому це важливо

Podman змушує відповісти на питання: кому належить дерево процесів? Docker керує контейнерами через `dockerd`, із власними правилами перезапуску, журналюванням і перевірками стану контейнерів. На сервері, де вже працює systemd, це другий наглядач поруч із першим, і мови в них різні.

З Podman і Quadlet наглядач лише один, systemd. Контейнер стає такою самою службою, як усі: `systemctl status` показує його стан, journald зберігає вивід, `After=` і `Requires=` впорядковують його відносно бази даних, від якої він залежить, а `Restart=` і ліміти cgroup, які systemd і так розуміє, діють без змін. Хто вміє читати файл модуля, той уже вміє читати й контейнерну площину.

Є й безпековий бік. Демон Docker працює від root, якщо не обрано його безкореневий режим, і власний посібник Docker з безпеки радить давати керування ним лише довіреним користувачам, бо контейнерові можна відкрити будь-який каталог хоста. Podman не тримає привілейованого демона між командами. Безкореневе налаштування звужує ризик ще більше: процес, що втік із контейнера, опиняється в системі як непривілейований користувач.

Отже, Docker vs Podman — це радше питання доречності, ніж змагання можливостей. Docker зручніший на ноутбуці розробника з Docker Desktop, для команд, чия робота побудована довкола Compose, і для інструментів, що

чекають на сокет Docker, хоча Podman за потреби теж віддає сумісний API. Podman зазвичай виграє на серверах, де systemd уже є джерелом істини і де демон, що переписує брандмауер, — тягар, а не зручність.

Як це виглядає на практиці

Podman несе контейнерну площину у двох нещодавніх проєктах, а раніше був інструментом налагодження для парку застосунків на Docker.

На одному хості контейнери працюють як модулі Quadlet під systemd, а Podman стоїть на місці Docker. Причина — брандмауер. Docker за замовчуванням публікує порти контейнерів вище за правила, які встановлює роль посилення захисту, а ці контейнери користуються мережею хоста й не публікують жодного порту. Модулі працюють на системному рівні, а не в безкореневому режимі, і сторінка записує це як компроміс: простіша автоматизація замість суворішої ізоляції, яку дав би безкореневий режим, плюс примітка, куди рухатися, якщо втеча з контейнера колись стане важливішою. Розгортання відокремлене від підготовки хоста. Root один раз готує шлях розгортання, а далі непривілейований користувач розгортає новий реліз, запускаючи один фіксований скрипт через вузьке правило: зібрати образ, перезапустити службу, і більше нічого.

Така сама схема працює на дуже скромному обладнанні. Хост з одним ядром і 512 МБ пам’яті обслуговує публічні кінцеві точки TLS компанії, git-форджі onion-сервіс, а автоматизацію резервного копіювання реалізовано окремо. Його контейнерний шар — це юніти Quadlet під тим самим супервізором, що й усе інше, а не демон із власним супервізором. За добу, протягом якої фіксували вибірки, доступна пам’ять жодного разу не опускалася нижче приблизно 310 МБ із 464 МБ придатних до використання.

Ще раніше, коли двадцять застосунків, упакованих у Docker, працювали в Google Cloud і AWS, усунення несправностей ішло через Podman, а оркестрацію налагоджували через Kubernetes. Образи були ті самі, а інструмент довкола них був справою вибору.

Де виникають проблеми

Здебільшого клопіт із Podman з’являється, коли від нього чекають поведінки Docker або звичайного модуля systemd, а він трохи і те, і те.

Виправлений файл Quadlet ніби ні на що не впливає. Генератор працює під час завантаження і після `systemctl daemon-reload`, тож зміна на диску лишається непомітною, доки не станеться одне з двох. Далі `systemctl enable` відмовляється вмикати згенеровану службу, бо systemd вважає її тимчасовою. Автозапуск задають у власній секції `[Install]` цього файлу:

```
# /etc/containers/systemd/web.container
[Unit]
Description=Web front end

[Container]
Image=docker.io/library/nginx:1.27
Network=host

[Service]
Restart=always

[Install]
WantedBy=multi-user.target
```

Після збереження виконують `systemctl daemon-reload` і `systemctl start web.service`.

Безкоренева служба зупиняється, щойно власник виходить із системи. Модулі користувача живуть у його власному менеджері systemd, який завершується разом з останнім сеансом. Команда `loginctl enable-linger` для цього користувача тримає менеджер живим від самого завантаження, і без неї безкоренева служба не переживе перезавантаження.

Безкореневий вебсервер просить порт 80 і отримує відмову. Порти нижче 1024 вимагають root або можливості `CAP_NET_BIND_SERVICE`, і це правило стосується й користувача, від якого працює Podman. Публікуйте високий порт і ставте перед ним проксі або свідомо знижуйте параметр ядра `ip_unprivileged_port_start`, пам’ятаючи, що він діє для всіх користувачів хоста.

Файли в bind mount належать незнайомцеві. У безкореневому режимі власний UID користувача стає root усередині контейнера, а решта UID контейнера потрапляє в підлеглий діапазон. Процес, що працює всередині як UID 999, пише файли, які хост бачить як власність чогось на кшталт

100998. `podman unshare` відкриває оболонку в тому самому відображенні, де `chown` поводиться так, як очікує контейнер, а параметр `U` у монтуванні автоматично змінює власника відповідно до контейнера.

Що почитати далі

- What is Podman? — власна документація проєкту, що починається з його визначення інструмента.
- podman-systemd.unit — довідник Quadlet: типи файлів, шляхи пошуку й кожен ключ.
- Basic Setup and Use of Podman in a Rootless environment — підлеглі ідентифікатори, простори імен користувачів і мережа через pasta.
- podman-unshare — як увійти в безкореневий простір імен, щоб перевірити або виправити власника.
- podman-generate-systemd — старіший спосіб створювати модулі та його заміна на Quadlet.

Словник

Обирайте за фактичними умовами

Опишіть шлях розгортання, експлуатаційні ризики та чергову команду. Ми можемо визначити зміну з перевірками, шляхом відновлення та чіткою відповідальністю.

Дивитися, скільки це коштує Зв'язатися з нами →

Зробіть термінологію практичною

Запитайте, як «Що таке Podman?» стосується вашої організації. Конкретного прикладу з поточної системи достатньо для початку розмови.

Переглянути наші послуги

welcome@engineer.company +45 50 10 61 84 Копенгаген, Данія

Візьміть цю сторінку із собою

Візитівка

Дбайливо зроблено з любов'ю — D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Авторське право © 2025 – дотепер · Engineer ApS · Усі права захищено.

Надруковано з https://engineer.company/uk/glossary/podman/