Engineer

Podman

Motor de contenedores que funciona sin demonio

Qué es

Podman es un motor de contenedores para Linux que construye, descarga y ejecuta las mismas imágenes OCI que Docker, pero sin un servidor central intermediario. Cada comando `podman` hace su trabajo en su propio proceso y termina. Un contenedor iniciado sigue ejecutándose bajo un pequeño proceso supervisor llamado conmon, y ese supervisor es un proceso hijo de lo que lo haya iniciado: una shell, un script o systemd. El proyecto se describe como una herramienta sin demonio, de código abierto y nativa de Linux, e indica que la mayoría de los usuarios pueden crear un alias de `docker` a `podman` y seguir trabajando.

El nombre es una abreviatura de pod manager, gestor de pods. Un pod, concepto tomado de Kubernetes, es un grupo de contenedores que comparten un espacio de nombres de red y pueden comunicarse entre sí mediante localhost. Podman permite crear pods manualmente, y `podman kube play` inicia contenedores y pods a partir de un archivo YAML de Kubernetes, lo que lo convierte en una opción razonable para probar un manifiesto antes de llevarlo a un clúster.

Dos funciones lo distinguen en el uso diario. La primera es el funcionamiento sin privilegios de root. Un usuario normal puede ejecutar contenedores: los espacios de nombres de usuario del kernel asignan a ese usuario el UID 0 dentro del contenedor y su intervalo de identificadores subordinados, obtenido de `/etc/subuid` y `/etc/subgid`, a todos los demás UID. La red de los contenedores sin root pasa por pasta, una pila de red en espacio de usuario, porque un usuario sin privilegios no tiene derecho a reescribir el filtro de paquetes del host.

La segunda es Quadlet. Un archivo Quadlet es una breve descripción declarativa de un contenedor, volumen, red, pod o construcción de una imagen, escrita con la misma sintaxis ini que una unidad de systemd y guardada en

`/etc/containers/systemd/` para el sistema o en `~/.config/containers/systemd/` para un usuario. Al arrancar, y en cada `systemctl daemon-reload`, un generador de systemd convierte `web.container` en un servicio ordinario `web.service`. El antiguo comando `podman generate systemd` sigue funcionando, y el proyecto lo marca como obsoleto en favor de Quadlet.

Utilice el concepto en una decisión

Aplique «¿Qué es Podman?» a una decisión real. Podemos ayudarle a relacionar el concepto con el sistema que está construyendo o comprando. Software empaquetado para comportarse igual en un portátil, un servidor de pruebas y producción, y ampliar su capacidad cuando aumente la actividad.

Explore nuestros servicios Póngase en contacto →

Compruebe los supuestos detrás del término

Utilice la explicación y los trabajos enlazados para formular preguntas sobre su proyecto. Podemos examinar los supuestos con usted antes de elegir una herramienta. 20 aplicaciones Docker en Google Cloud y AWS — 20 aplicaciones en contenedores desplegadas y monitorizadas

20 aplicaciones Docker en Google Cloud y AWS Póngase en contacto →

Por qué importa

La pregunta que plantea Podman es quién controla el árbol de procesos. Docker gestiona los contenedores mediante `dockerd`, con sus propias políticas de reinicio, registros y comprobaciones del estado de los contenedores. En un servidor que ya ejecuta systemd, eso supone un segundo supervisor junto al primero, y ambos no comparten vocabulario.

Con Podman y Quadlet, systemd es el único supervisor. Un contenedor es un servicio como cualquier otro: `systemctl status` lo muestra, journald guarda su salida, `After=` y `Requires=` lo ordenan respecto de la base de datos que necesita, y se le aplican sin cambios `Restart=` y los límites de cgroups que systemd ya conoce. Un operador que sabe leer un archivo de unidad ya sabe leer la capa de contenedores.

También hay una vertiente de seguridad. El demonio de Docker se ejecuta como root salvo que se elija su modo sin root, y la propia guía de seguridad de Docker indica que solo deberían poder controlarlo usuarios de confianza, porque puede darse a un contenedor cualquier directorio del host. Podman no mantiene un demonio con privilegios ejecutándose entre comandos. Una configuración sin root restringe aún más el alcance: un proceso que escapa del contenedor queda como usuario sin privilegios.

Por tanto, elegir entre Docker y Podman es menos una competición de funciones que una cuestión de adecuación. Docker es la opción más cómoda en el portátil de un desarrollador con Docker Desktop, para equipos cuyo flujo de trabajo gira en torno a Compose y para herramientas que esperan un socket de Docker, aunque Podman puede ofrecer una API compatible cuando se le solicita. Podman suele resultar preferible en servidores donde systemd ya es la fuente de verdad y un demonio que reescribe el cortafuegos supone un problema más que una comodidad.

Cómo se aplica en la práctica

Podman se encarga de la capa de contenedores en dos proyectos recientes y, antes de eso, fue la herramienta de diagnóstico para un conjunto de sistemas construido sobre Docker.

En un host, los contenedores se ejecutan como unidades Quadlet bajo systemd, con Podman en lugar de Docker. El motivo es el cortafuegos: la configuración predeterminada de Docker publica los puertos de los contenedores por delante de las reglas que instala el rol de endurecimiento, mientras que estos contenedores usan la red del host sin publicar ningún puerto. Las unidades se ejecutan a nivel del sistema, en lugar de usar el modo sin root. La página recoge el compromiso: una automatización más sencilla frente al aislamiento más estricto que daría el modo sin root, junto con una nota sobre cómo cambiar de enfoque si el riesgo de escape de un contenedor llega a importar más que la sencillez de la automatización. El despliegue está separado del aprovisionamiento. Root prepara una vez la ruta de despliegue. Cada versión posterior se publica con un único script fijo que un usuario sin privilegios puede ejecutar mediante una regla de alcance limitado. El script solo hace dos cosas: construir la imagen y reiniciar el servicio.

La misma organización funciona con muy poco hardware. Un host de un solo núcleo y 512 MB de memoria ejecuta los puntos de acceso TLS públicos de la empresa, la forja de Git y el servicio onion, con la automatización de las copias de seguridad implementada por separado. Su capa de contenedores consiste en unidades Quadlet bajo el mismo supervisor que el resto, en lugar de un demonio con supervisor propio. Durante un día de muestreo registrado, la memoria disponible nunca bajó de unos 310 MB de los 464 MB utilizables.

Antes, en veinte aplicaciones empaquetadas con Docker y distribuidas entre Google Cloud y AWS, el diagnóstico de problemas se hacía mediante Podman, con Kubernetes para la orquestación. Las imágenes eran las mismas; la herramienta que las rodeaba era una elección.

Dónde surgen los problemas

La mayoría de los problemas con Podman nacen de esperar que se comporte como Docker o como una unidad ordinaria de systemd, cuando combina aspectos de ambos.

Editar un archivo Quadlet no cambia nada. El generador se ejecuta al arrancar y con `systemctl daemon-reload`, por lo que un cambio en disco permanece invisible hasta que ocurre una de esas cosas. Después, `systemctl enable` rechaza el servicio generado porque systemd lo considera transitorio. El arranque automático debe declararse en la sección `[Install]` del propio archivo:

```
# /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
```

Después de guardarlo, ejecute `systemctl daemon-reload` y `systemctl start web.service`.

Un servicio sin root se detiene cuando su propietario cierra la sesión. Las unidades de usuario viven dentro del gestor de systemd del propio usuario, que termina con la última sesión. Ejecutar `loginctl enable-linger` para ese usuario mantiene activo el gestor desde el arranque, algo que necesita cualquier servicio sin root que deba sobrevivir a un reinicio.

Un servidor web sin root solicita el puerto 80 y se le deniega. Los puertos inferiores a 1024 requieren root o la capacidad `CAP_NET_BIND_SERVICE`, y esa regla se aplica al usuario con el que se ejecuta Podman. Publique un puerto alto y coloque un proxy delante, o reduzca de forma deliberada el ajuste `ip_unprivileged_port_start` del kernel, sabiendo que se aplica a todos los usuarios del host.

Los archivos de un montaje de enlace pertenecen a un desconocido. En modo sin root, el UID del propio usuario corresponde a root dentro del contenedor y todos los demás UID del contenedor se asignan al intervalo subordinado. Un proceso que se ejecuta con UID 999 dentro escribe archivos que el host ve como propiedad de un identificador como 100998. `podman unshare` abre una shell dentro de la misma asignación, donde `chown` funciona como espera el contenedor, y la opción `U` de un montaje ajusta automáticamente el propietario para que corresponda al esperado por el contenedor.

Lecturas adicionales

- ¿Qué es Podman? — Documentación del propio proyecto, empezando por su definición de la herramienta.
- podman-systemd.unit — Referencia de Quadlet: tipos de archivo, rutas de búsqueda y todas las claves.
- Configuración y uso básicos de Podman en un entorno sin root — Identificadores subordinados, espacios de nombres de usuario y red con pasta.
- podman-unshare — Entrar en el espacio de nombres de usuario sin root para inspeccionar o corregir la propiedad.
- podman-generate-systemd — La forma anterior de generar unidades y su obsolescencia en favor de Quadlet.

Glosario

Elija según sus restricciones reales

Describa la ruta de despliegue, los riesgos operativos y quiénes se encargan de las guardias. Podemos definir el alcance de un cambio con comprobaciones, una vía de recuperación y responsabilidades claras.

Vea cuánto cuesta Póngase en contacto →

Haga práctica la terminología

Pregunte cómo se aplica «¿Qué es Podman?» a su organización. Basta un ejemplo concreto de su sistema actual para iniciar la conversación.

Explore nuestros servicios

welcome@engineer.company +45 50 10 61 84 Copenhague, Dinamarca

Llévese esta página

Tarjeta de contacto

Creado con esmero y cariño por D.

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

Derechos de autor © 2025 – actualidad · Engineer ApS · Todos los derechos reservados.

Impreso desde https://engineer.company/es/glossary/podman/