DETALLE DE LA TECNOLOGÍA

DIEZ CAPAS DE SEGURIDAD DE CONTENEDORES

INTRODUCCIÓN

Los contenedores son muy atractivos porque permiten a los usuarios empaquetar fácilmente una aplicación y todas sus dependencias en una sola imagen que se puede promover desde la etapa de desarrollo hasta las de prueba y producción, sin ningún cambio. Los contenedores aseguran la consistencia entre entornos y varios objetivos de implementación, como servidores físicos, máquinas virtuales (VM) y nubes privadas o públicas. Esto permite que los equipos desarrollen y administren las aplicaciones que generan valor empresarial con más facilidad.

Las empresas requieren una seguridad sólida; por lo tanto, cualquiera que ejecute servicios esenciales en contenedores se preguntará si los contenedores son seguros y si se puede confiar en ellos para las aplicaciones. Este documento describe diez elementos clave de seguridad para distintas capas de la pila de soluciones de contenedores y las diferentes etapas del ciclo de vida del contenedor.

facebook.com/redhatinc
@RedHatIberia
Red Hat EMEA

es.redhat.com

ASEGURAMIENTO DE CONTENEDORES: CAPAS Y CICLO DE VIDA

Asegurar los contenedores es como asegurar cualquier proceso en ejecución. Debe considerar la seguridad en todas las capas de la pila de la solución antes de implementar y ejecutar su contenedor. También es importante considerar la seguridad de todo el ciclo de vida de la aplicación y del contenedor. Los diez elementos clave de la seguridad de contenedores son los siguientes:

1. Arquitectura multiempresa del host de contenedor
2. Contenido del contenedor
3. Registros del contenedor
4. Diseño de contenedores
5. Implementación de contenedores
6. Orquestación de contenedores
7. Aislamiento de redes
8. Almacenamiento
9. Gestión de interfaz de programación de aplicaciones (API)
10. Clústeres federados

1. El sistema operativo y la arquitectura multiempresa del host de contenedor

Los contenedores permiten que los desarrolladores puedan crear y promover más fácilmente una aplicación y sus dependencias como una unidad. Además, le permiten aprovechar al máximo sus servidores porque habilitan las implementaciones de aplicaciones de arquitectura multiempresa en un host compartido. Usted puede implementar múltiples aplicaciones en un solo host con facilidad al aumentar y reducir los contenedores individuales, según sea necesario. Y a diferencia de la virtualización tradicional, no necesita un hipervisor ni gestionar sistemas operativos guest (invitados) en cada máquina virtual. Los contenedores virtualizan los procesos de sus aplicaciones, no el hardware.

Para obtener el máximo provecho de esta tecnología de empaquetado e implementación, el equipo de operaciones necesita el entorno adecuado para ejecutar los contenedores. Las operaciones necesitan un sistema operativo (OS) que pueda proteger los contenedores en los límites: proteger el kernel del host de los escapes del contenedor y proteger los contenedores unos de otros.

Los contenedores son procesos Linux® aislados, con recursos aislados, que le permiten ejecutar aplicaciones separadas en el kernel de un host compartido. Su estrategia de protección de contenedores debe ser la misma que la de cualquier proceso en ejecución en Linux. Descartar los privilegios es importante y todavía, la mejor práctica. Mejor aún es crear contenedores con el menor privilegio posible. Los contenedores se deben ejecutar como usuario, no como raíz. Luego, use los múltiples niveles de seguridad disponibles en Linux. Cinco de las funciones de seguridad disponibles para proteger los contenedores que se ejecutan en Red Hat® Enterprise Linux son los siguientes: espacios de nombres de Linux, Security-Enhanced Linux (SELinux), cgroups, capacidades y modo seguro de computación (seccomp).

- Los espacios de nombres de Linux proporcionan los aspectos básicos del aislamiento de contenedores. Un espacio de nombre hace que parezca que los procesos contenidos en el espacio de nombre tengan su propia instancia de recursos globales. Los espacios de nombre proporcionan la abstracción que da la impresión de que uno ejecuta en su propio sistema operativo cuando se encuentra adentro de un contenedor.
- SELinux proporciona una capa adicional de seguridad para mantener a los contenedores aislados entre sí y del host. Permite a los administradores ejecutar los controles de acceso obligatorios (MAC) para cada usuario, aplicación, proceso y archivo. SELinux es como una pared de ladrillos: impedirá que usted rompa (por accidente o a propósito) la abstracción del espacio de nombre.

- Los cgroups (grupos de control) limitan, justifican y aíslan el uso de recursos (por ejemplo, CPU, memoria, E/S del disco, red) de un grupo de procesos. Utilice los cgroups para asegurarse de que su contenedor no se superponga con otro contenedor en el mismo host. Los cgroups también se pueden usar para controlar los pseudodispositivos, que son un vector de ataque conocido.
- Las capacidades de Linux se pueden usar para inhabilitar la raíz en un contenedor. Las capacidades son unidades de privilegio separadas que se pueden habilitar o deshabilitar de forma independiente. Las capacidades le permiten, por ejemplo, enviar paquetes IP sin formato o enlazar a puertos por debajo de 1024. Al ejecutar contenedores, se pueden dejar de usar múltiples capacidades sin afectar la gran mayoría de las aplicaciones en contenedores.
- Por último, un perfil de modo seguro de computación (seccomp) se puede asociar con un contenedor para limitar las llamadas disponibles del sistema.

La seguridad de sus aplicaciones e infraestructura se puede mejorar aún más al implementar sus contenedores en un sistema operativo ligero y optimizado para ejecutar contenedores de Linux, como Red Hat Enterprise Linux Atomic Host. Atomic Host reduce la superficie de ataque, ya que minimiza el entorno del host y lo ajusta a los contenedores.

Como prueba de las funciones de seguridad disponibles con Red Hat Enterprise Linux y Atomic Host, Red Hat Enterprise Linux 7.1 hace poco recibió la certificación Common Criteria, que incluye la certificación para el soporte de marco en contenedores de Linux.

La virtualización tradicional también habilita la arquitectura multiempresa, pero de una manera totalmente distinta. La virtualización depende de un hipervisor que inicia las máquinas virtuales guest (cada una con su propio sistema operativo) y, además, ejecuta la aplicación y sus dependencias. Con las máquinas virtuales, el hipervisor aísla los guest entre sí y del host. Menos personas y procesos tienen acceso al hipervisor, lo que reduce la superficie de ataque en el servidor físico. De todos modos, la seguridad debe supervisarse en caso de amenazas; por ejemplo, una máquina virtual guest puede usar bugs del hipervisor para obtener acceso a otra máquina virtual o al kernel del host. Y cuando el sistema operativo necesita parches, estos se deben aplicar en todas las máquinas virtuales guest que usen ese sistema operativo.

Los contenedores se deben ejecutar adentro de las máquinas virtuales guest, y puede haber casos prácticos en los que esto es aconsejable. Por ejemplo, si implementará una aplicación tradicional en un contenedor, tal vez desee ubicar el contenedor adentro de la máquina virtual guest, a fin de mejorar y cambiar una aplicación a la nube. Sin embargo, la arquitectura multiempresa del contenedor en un solo host proporciona una solución de implementación más ligera, flexible y fácil de escalar. Este modelo de implementación es especialmente adecuado para las aplicaciones nativas de la nube.

2. Contenido del contenedor (utilice recursos confiables)

En materia de seguridad, lo que está adentro del contenedor es importante. Desde ya hace un tiempo, las aplicaciones e infraestructuras han estado conformadas por componentes fácilmente disponibles. Muchos de estos componentes son paquetes de open source, como el sistema operativo Linux, Apache Web Server, Red Hat JBoss® Enterprise Application Platform, PostgreSQL y Node.js. Las versiones en contenedor de estos paquetes ahora también están ya disponibles, de tal manera que no tiene que diseñarlas usted mismo. Pero, tal como ocurre con cualquier código que descarga de una fuente externa, debe conocer de dónde provienen los paquetes originalmente, quién los diseñó y si hay algún código malicioso en ellos. Pregúntese lo siguiente:

- ¿El contenido de los contenedores comprometerá mi infraestructura?
- ¿Hay alguna vulnerabilidad conocida en la capa de la aplicación?
- ¿Están las capas del tiempo de ejecución y del sistema operativo actualizadas?
- ¿Con qué frecuencia se actualizará el contenedor y cómo sabré cuando se actualiza?

Durante años, Red Hat ha empaquetado y entregado contenido Linux confiable en Red Hat Enterprise Linux y en toda nuestra cartera de productos. Ahora, Red Hat entrega ese mismo contenido empaquetado y confiable como contenedores de Linux. Esto incluye imágenes de base para Red Hat

Enterprise Linux 7 y Red Hat Enterprise Linux 6, y también proporciona un gran número de imágenes certificadas para distintos tiempos de ejecución de lenguajes, middleware, bases de datos y más, a través del catálogo de contenedores de Red Hat. Los contenedores certificados de Red Hat se ejecutan en cualquier lugar en que se ejecute Red Hat Enterprise Linux, desde equipos sin sistema operativo hasta máquinas virtuales y la nube. Los contenedores certificados tienen el soporte de Red Hat y de nuestros partners.

El contenido de la imagen del contenedor se empaqueta desde un código fuente conocido. Red Hat también proporciona supervisión de la seguridad. Con su nuevo índice de estado del contenedor, Red Hat muestra el "grado" de cada imagen de contenedor y detalla cómo las imágenes de contenedores se deben proteger, consumir y evaluar para satisfacer las necesidades de los sistemas de producción. La clasificación de los contenedores se basa en parte en la antigüedad y el impacto de las erratas de seguridad no aplicadas a todos los componentes de un contenedor, lo que proporciona una clasificación agregada de la seguridad de un contenedor que puede ser entendida tanto por expertos en seguridad como por no expertos.

Cuando Red Hat lanza las actualizaciones de seguridad, como correcciones de glibc, Drown o Dirty Cow, también rediseñamos nuestras imágenes de contenedor y las enviamos a nuestro registro público. Las recomendaciones de seguridad de Red Hat lo alertan de cualquier problema detectado recientemente en las imágenes certificadas de contenedor y lo dirigen hacia la imagen actualizada para que usted pueda actualizar cualquier aplicación que use la imagen.

Por supuesto, habrá ocasiones en que necesitará contenido que Red Hat no proporciona. Le recomendamos usar las herramientas de escaneo del contenedor que continuamente usan bases de datos actualizadas de las vulnerabilidades, a fin de asegurarse de tener siempre la información más reciente sobre las vulnerabilidades conocidas cuando use imágenes de contenedor de otras fuentes. Debido a que la lista de vulnerabilidades conocidas cambia constantemente, deberá verificar los contenidos de sus imágenes de contenedor cuando las descargue por primera vez y continuar verificando con el tiempo el estado de vulnerabilidad de todas las imágenes aprobadas e implementadas, tal como Red Hat lo hace con las imágenes de contenedores de Red Hat.

Red Hat proporciona una API que se puede conectar en Red Hat Enterprise Linux para dar soporte a múltiples escáneres, como OpenSCAP, Black Duck Hub, JFrog Xray y Twistlock. Red Hat CloudForms también se puede usar con OPenSCAP para escanear imágenes de contenedor por cuestiones de seguridad. Además, Red Hat OpenShift le proporciona la capacidad de usar escáneres con su proceso de integración continua y entrega continua (CI/CD). Este tema se explica con más detalle a continuación.

3. Registros del contenedor (acceso seguro a las imágenes de contenedor)

Claro está que sus equipos diseñan contenedores que colocan el contenido en una capa por encima de las imágenes públicas de contenedor que usted descarga. El acceso y la promoción de las imágenes de contenedor descargadas y de las imágenes creadas internamente se deben administrar de la misma manera en que se administran otros tipos de binarios. Hay una cantidad de registros privados que son compatibles con el almacenamiento de imágenes de contenedor. Recomendamos seleccionar un registro privado que lo ayude a automatizar las políticas de uso de imágenes de contenedor almacenadas en el registro.

Red Hat OpenShift incluye un registro privado que se puede usar para administrar imágenes de contenedor. El registro de OpenShift proporciona controles de acceso basado en funciones que le permiten administrar quién puede insertar y enviar imágenes de contenedor específicas. OpenShift también es compatible con la integración con otros registros privados que posiblemente ya esté usando, como Artifactory de JFrog y Docker Trusted Registry.

La lista de vulnerabilidades conocidas cambia constantemente, por lo que es necesario que, con el tiempo, supervise los contenidos de las imágenes de contenedor implementadas, así como de las imágenes descargadas recientemente. El registro debe incluir las funciones que lo ayudan a administrar el contenido en función de los metadatos sobre el contenedor, incluidas las vulnerabilidades conocidas. Por ejemplo, puede usar el análisis Red Hat CloudForms SmartState para marcar las imágenes vulnerables de su registro. Una vez marcadas, OpenShift evitará que esas imágenes se ejecuten de forma continua.

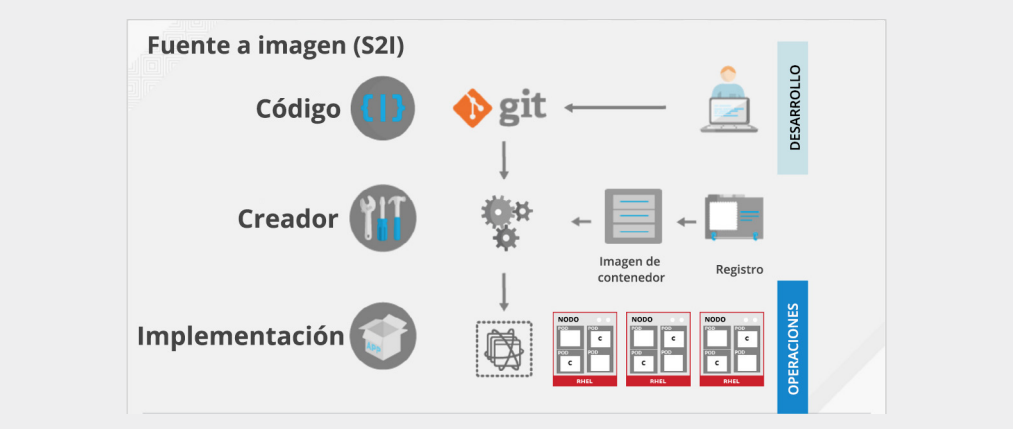
En las próximas secciones, destacaremos algunas de las maneras adicionales en las que OpenShift da soporte a la automatización de las políticas de seguridad en torno al contenido de contenedores.

4. La seguridad y el proceso de diseño

En un entorno de contenedores, el proceso de diseño de software es la etapa en el ciclo de vida en que el código de aplicaciones se integra con las bibliotecas de tiempo de ejecución necesario. Administrar este proceso de diseño es clave para proteger la pila de software. Adherir a la filosofía "diseñar una vez, implementar en todas partes" garantiza que el producto del proceso de diseño sea exactamente igual a lo que se implementa en la etapa de producción. También es importante mantener la inmutabilidad de sus contenedores. En otras palabras, no aplique parches en los contenedores en ejecución; en su lugar, rediséñelos y vuelva a implementarlos.

Red Hat OpenShift proporciona varias capacidades para la gestión y la seguridad del diseño:

- "Fuente a imagen" (S2I) es un marco de open source para combinar código fuente e imágenes de base. S2I permite que sus equipos de desarrollo y operaciones colaboren entre sí en un entorno de diseño reproducible. Cuando un desarrollador asigna código con git, bajo S2I, OpenShift puede hacer lo siguiente:
  - Activar (mediante enlaces web en el repositorio del código o algún otro proceso de integración continua [CI] automatizado) el ensamblaje automático de una nueva imagen desde artefactos disponibles, como la imagen de base S2I y el código confirmado recientemente.
  - Implementar de forma automática la imagen diseñada recientemente para probarla.
  - Promover la imagen probada al estado de producción e implementar automáticamente la nueva imagen a través del proceso de integración continua.



Red Hat OpenShift incluye una instancia integrada de Jenkins para la integración continua. OpenShift también incluye las API RESTful, que se pueden usar para integrar sus propias herramientas de diseño o integración continua o un registro de imagen privado, como Artifactory de JFrog.

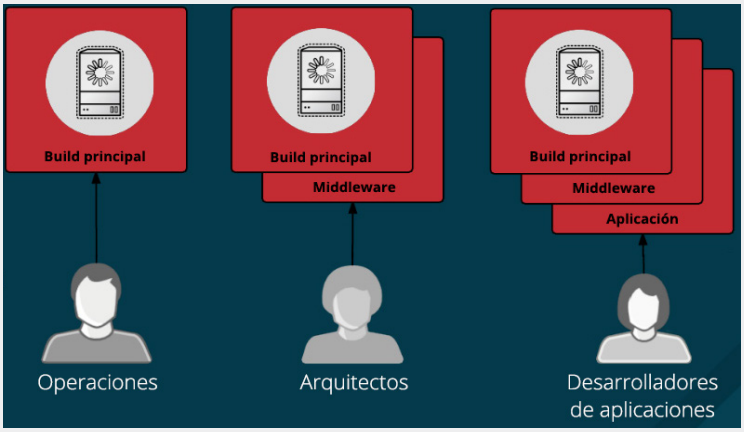
Una práctica recomendada para la seguridad de las aplicaciones es integrar pruebas de seguridad automatizadas en su proceso de diseño o integración continua. Por ejemplo, integre lo siguiente:

- Herramientas para pruebas de seguridad de aplicaciones estáticas (SAST) y pruebas de seguridad de aplicaciones dinámicas (DAST), como HP Fortify y IBM AppScan.
- Escáneres para verificación en tiempo de real de las vulnerabilidades conocidas, como Black Duck Hub y JFrog Xray. Las herramientas como estas catalogan los paquetes de open source en su contenedor, le notifican sobre cualquier vulnerabilidad conocida y lo actualizan cuando se descubren nuevas vulnerabilidades en paquetes escaneados previamente.

Además, su proceso de integración continua debe incluir políticas que marquen los diseños con problemas detectados por los escaneos de seguridad para que su equipo pueda adoptar las medidas adecuadas y abordar esos problemas lo antes posible.

Tanto si trabaja en un sector altamente regulado o si simplemente desea optimizar los esfuerzos de su equipo, le recomendamos diseñar su gestión de imágenes de contenedor y crear procesos para aprovechar las capas del contenedor e implementar la separación del control, de tal manera que:

- El equipo de operaciones gestione las imágenes de base.
- Los arquitectos gestionen el middleware, los tiempos de ejecución, las bases de datos y otras soluciones semejantes.
- Los desarrolladores se centren en las capas de aplicaciones y solo escriban el código.



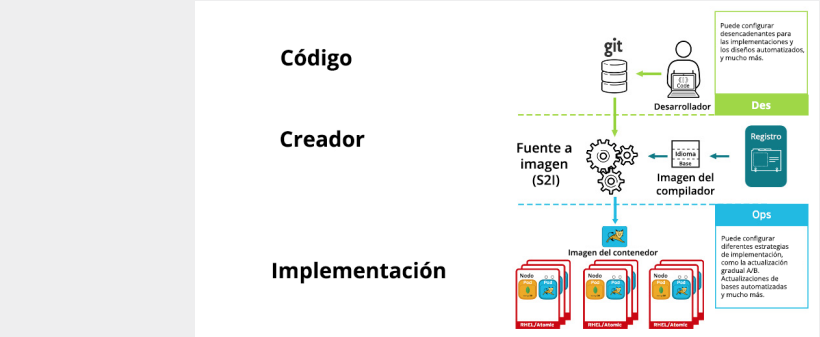
Finalmente, le recomendamos que firme sus contenedores diseñados de forma personalizada, de tal manera de asegurarse de que no sean utilizados de forma indebida entre la etapa de diseño y la de implementación.

5. Implementación: controlar lo que se puede implementar dentro de un clúster

En caso de que algo fracase durante el proceso de diseño, o en las situaciones en las que se detecta una vulnerabilidad después de haber implementado una imagen, usted necesitaría contar con otra capa de seguridad: herramientas para la implementación automatizada y basada en políticas.

Analicemos una aplicación que está diseñada con tres capas de imágenes de contenedor: la capa principal, la capa del middleware y la capa de la aplicación. Se detecta un problema en la imagen principal, y esa imagen se rediseña. Una vez que se completa el diseño, la imagen se envía al registro de OpenShift. OpenShift puede detectar que la imagen ha cambiado. Para los diseños que dependen de esta imagen y que tienen desencadenantes definidos, OpenShift rediseñará de forma automática la imagen de la aplicación incorporando las bibliotecas fijas.

Una vez que se completa el diseño, la imagen se envía al registro interno de OpenShift. OpenShift detecta inmediatamente los cambios de las imágenes en su registro interno y, en el caso de las aplicaciones con desencadenantes definidos, implementa automáticamente la imagen actualizada, asegurándose de que el código en ejecución en la etapa de producción sea siempre idéntico al de la imagen actualizada más recientemente. Todas estas capacidades funcionan juntas para integrar las capacidades de seguridad en el proceso y en el canal de CI/CD.



Una seguridad sólida incluye políticas automatizadas que se pueden usar para administrar la implementación de contenedores desde la perspectiva de la seguridad. OpenShift incluye restricciones del contexto de seguridad (SCC) incorporadas que usted puede utilizar para definir un conjunto de condiciones que un pod (un grupo de contenedores) debe ejecutar a fin de ser aceptado por el sistema. Las SCC de OpenShift, que han sido aportadas a Kubernetes como política de seguridad del pod, permiten que un administrador pueda controlar lo siguiente:

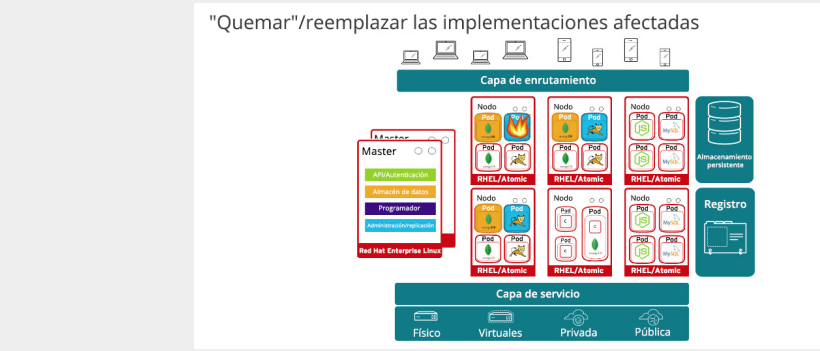
- La ejecución de contenedores con privilegios.
- Las capacidades que un contenedor puede solicitar que se añadan.
- El uso de directorios de host como volúmenes.
- El contexto de SELinux del contenedor.
- La creación de perfiles seccomp.
- La id. de usuario del contenedor.

Los usuarios con los permisos requeridos pueden adaptar las políticas de SCC predeterminadas para ser más permisivos si así lo desean. Tal como con las herramientas de CI, los usuarios pueden integrar sus propias herramientas de CD con su plataforma de contenedores si así lo prefieren.

Al integrar CI/CD con OpenShift y al asegurarse de que todo el proceso de rediseño de la aplicación para incorporar las últimas correcciones y pruebas esté implementado en todas partes adentro del entorno, dicho proceso se puede automatizar totalmente.

6. Orquestación de los contenedores: proteger la plataforma de contenedores

Ciertamente, las aplicaciones pocas veces se entregan en un solo contenedor. Incluso las aplicaciones simples, típicamente, tienen un front end, un back end y una base de datos. Y la implementación de aplicaciones modernas basadas en microservicios en contenedores implica implementar múltiples contenedores —a veces en el mismo host, y otras veces distribuidos en múltiples host o nodos, como se muestra en el diagrama—.



Cuando se administra la implementación de contenedores a escala, se debe tener en cuenta lo siguiente:

- Qué contenedores deben implementarse en qué hosts.
- Qué host tiene más capacidad.
- Qué contenedores necesitan acceso entre sí. ¿Cómo se detectarán unos a otros?
- Cómo controlar el acceso (y la gestión) a los recursos compartidos, como la red y el almacenamiento.
- Cómo supervisar el estado del contenedor.
- Cómo escalar automáticamente la capacidad de la aplicación para satisfacer los requerimientos.
- Cómo habilitar el autoservicio para desarrolladores y a la vez satisfacer los requerimientos de seguridad.

Usted puede diseñar su propio entorno de gestión de contenedores. Pero, ¿por qué perder el tiempo diseñando herramientas cuando puede implementar una plataforma de contenedores con funciones de gestión y seguridad integradas para permitir que su equipo invierta su energía en diseñar las aplicaciones que generan valor empresarial?

Red Hat OpenShift Container Platform proporciona orquestación de contenedores, planificación automatizada y ejecución de contenedores de aplicaciones en clústeres de máquinas físicas o virtuales mediante la inclusión y exclusión del proyecto Kubernetes de open source. Kubernetes, un proyecto de open source iniciado por Google, utiliza "maestros" para gestionar la complejidad de la orquestación en clúster de contenedores. OpenShift también incluye Red Hat CloudForms que, entre otras cosas, se puede usar para supervisar el estado de los contenedores en su registro privado y evitar la implementación de contenedores con vulnerabilidades recientemente detectadas.

Dado el estado de las capacidades para desarrolladores y operadores, un sólido control de acceso basado en funciones es un elemento fundamental de la plataforma de contenedores. Por ejemplo, los maestros de la orquestación son un punto central de acceso y deben recibir el mayor nivel de supervisión de escrutinio de seguridad. Las API son clave para automatizar la gestión de contenedores a escala. Se usan para validar y configurar los datos para pods, servicios y controladores de replicación; realizar validaciones de proyectos en solicitudes entrantes, e invocar desencadenantes en otros componentes del sistema más importantes.

El control de acceso a la API (autenticación y autorización) es fundamental para proteger su plataforma de contenedores. El **maestro** de OpenShift incluye un **servidor OAuth** incorporado. Los desarrolladores y administradores obtienen **tokens de acceso OAuth** para autenticarse a sí mismos en la API. Como administrador, puede configurar OAuth para autenticar con el **proveedor de identidades** de su elección, incluidos los directorios del protocolo ligero de acceso a directorios (LDAP).

Uno de los valores clave de una plataforma de contenedores es la capacidad para habilitar el autoservicio para desarrolladores, a fin de que sus equipos de desarrollo entreguen de forma fácil y ágil las aplicaciones diseñadas en capas aprobadas. La seguridad de la arquitectura multiempresa es fundamental para la plataforma, a fin de asegurarse de que los equipos no accedan a los entornos de los otros sin autorización. Necesita un portal de autoservicio que proporcione el control necesario a los equipos para promover la colaboración y a la vez proporcionar seguridad. OpenShift añade distintos componentes a Kubernetes para mantener un maestro multiempresa seguro, lo que garantiza que:

- Todo el acceso al maestro se encuentre sobre la seguridad de la capa de transporte (TLS).
- El acceso al servidor de la API se base en token o certificado X.509.
- La cuota del proyecto se use para limitar el daño que un token malicioso puede hacer.
- Etcd no esté expuesto directamente al clúster.

OpenShift 3.5 proporciona **gestión de clústeres mejorada**, incluidos secretos mejorados y la gestión de certificados.

7. Aislamiento de redes

Implementar las aplicaciones modernas basadas en microservicios en contenedores implica, frecuentemente, implementar múltiples contenedores distribuidos en nodos múltiples. Si se tiene en cuenta la defensa de la red, es necesaria una manera de aislar las aplicaciones entre sí adentro del clúster.

Un servicio típico de contenedores de la nube pública, como Google Container Engine (GKE), Azure Container Services o Amazon Web Services (AWS) Container Service, es un servicio de tenencia individual. Le permite ejecutar sus contenedores en el clúster de la máquina virtual que usted inició. Para una arquitectura multiempresa de contenedores segura, necesita una plataforma de contenedores que le permita tomar un solo clúster y segmentar el tráfico para aislar, adentro del clúster, a los diferentes usuarios, equipos, aplicaciones y entornos.

Con los espacios de nombres de red, cada grupo de contenedores (conocido como "pod") obtiene su propia IP y su propio intervalo de puerto con los cuales enlazarse; de esta manera, aíslan entre sí a las redes del pod en el nodo. Los pod de distintos espacios de nombre (proyectos) no pueden enviar paquetes o recibirlos desde pods y servicios de un proyecto diferente de forma predeterminada, con opciones excepcionales que se nombran a continuación. Estas funciones se pueden usar para aislar los entornos de desarrollador, de prueba y de producción dentro de un clúster.

Sin embargo, esta proliferación de direcciones de IP y puertos hace que las redes sean más complejas. Además, los contenedores se diseñan para sufrir cambios. Le recomendamos invertir en herramientas que traten con esta complejidad en su lugar. Es preferible una plataforma de contenedores que use una red definida por software (SDN) para proporcionar una red de clústeres unificada que permita la comunicación entre contenedores en todo el clúster.

También es preferible una plataforma de contenedores que proporcione la capacidad de controlar el tráfico de egreso con un router o un método de firewall, de tal manera que pueda usar una lista blanca de IP para controlar, por ejemplo, el acceso a la base de datos.

Además de los espacios de nombre de red, la SDN proporciona seguridad adicional porque ofrece aislamiento entre los espacios de nombre del maestro (orquestación) y el complemento (plug-in) ovs-multitenant. Cuando se habilita el complemento ovs-multitenant, el tráfico de pods en un espacio de nombre se aísla, de forma predeterminada, del tráfico de pods desde otro espacio de nombre (proyecto). Para proporcionar excepciones al complemento ovs-multitenant, se presentó la funcionalidad "oadm de las redes de pod" en Red Hat OpenShift Container Platform 3.1 para permitir que dos proyectos accedan a los servicios del otro o para permitir que todos los proyectos puedan acceder a todos los pod y servicios en el clúster. La limitación de esto es que opera al nivel de todo el proyecto, y el tráfico permitido siempre es bidireccional. Esto significa que, si puede acceder a un servicio en un proyecto, puede acceder a todos los servicios de ese proyecto y también, necesariamente, debe garantizar el acceso a todos los servicios de su proyecto. Debido a que los permisos son bidireccionales, esto solo puede ser configurado por un administrador del clúster.

Presentado como una vista preliminar tecnológica en Red Hat OpenShift Container Platform 3.5, se diseñó un nuevo complemento de la política de redes (política de redes ovs) para mejorar la forma en que el complemento ovs-multitenant se puede usar para configurar el tráfico permitido entre pods. La política de redes permite configurar las políticas de aislamiento a nivel de los pod individuales. Debido a que las políticas en redes no son bidireccionales y aplican solo al tráfico de ingreso de pods bajo el control de un administrador de proyectos, tampoco se requieren privilegios de gestión del clúster.

Si desea implementar escáneres de redes, estos son fáciles de ejecutar en contenedores como "contenedores superprivados".

8. Almacenamiento

Los contenedores son útiles para las aplicaciones sin estado y con estado. Proteger el almacenamiento adjunto es un elemento clave para la seguridad de los servicios con estado. Red Hat OpenShift Container Platform proporciona complementos para las versiones múltiples de almacenamiento, incluidos los sistemas de archivo de red (NFS), AWS Elastic Block Stores (EBS), GCE Persistent Disks, GlusterFS, iSCSI, RADOS (Ceph) y Cinder.

Un **volumen persistente (VP)** se puede montar en un host de cualquier manera que sea compatible con el proveedor de recursos. Los proveedores tendrán distintas capacidades, y los modos de acceso de cada VP están configurados con los modos específicos compatibles con ese volumen en particular. Por ejemplo, NFS es compatible con múltiples clientes de lectura/escritura, pero un VP específico del NFS se podrá exportar al servidor como de solo lectura. Cada VP tiene su propio grupo de modos de acceso que describen las capacidades de ese volumen persistente específico. Como ReadWriteOnce, ReadOnlyMany y ReadWriteMany.

En el caso del almacenamiento compartido (NFS, Ceph, Gluster, etc.), la clave consiste en hacer que el VP del almacenamiento compartido registre su id. de grupo (gid) como una anotación en el recurso del VP. Cuando el VP es solicitado por el pod, el gid registrado se añadirá a los grupos complementarios del pod y le dará acceso al pod a los contenidos del almacenamiento compartido.

Para el almacenamiento en bloques (EBS, GCE Persistent Disks, iSCSI, etc.), las plataformas de contenedores pueden usar las capacidades de SELinux para proteger la raíz del volumen montado para pods sin privilegios, de tal manera que el volumen montado sea propiedad del contenedor con el que está asociado y sea solo visible para este contenedor.

Los datos en tránsito deben estar encriptados mediante https para que todos los componentes de la plataforma de contenedores se comuniquen entre sí.

Y, por supuesto, usted debe aprovechar las funciones de seguridad disponibles en la solución de almacenamiento que eligió.

9. Gestión de API/seguridad del end point e inicio de sesión único (SSO)

Proteger sus aplicaciones incluye administrar la autenticación y autorización de las aplicaciones y API.

Las capacidades de SSO de la web constituyen una parte clave de las aplicaciones modernas. Las plataformas de contenedores pueden incluir una cantidad de servicios en contenedores para que los desarrolladores los utilicen al crear sus aplicaciones, como Red Hat SSO (RH-SSO), un servicio de federación basado en el innovador proyecto Keycloak, de inicio de sesión único de la web, con autenticación basada en la conexión OpenId o SAML 2.0 lista para usar y totalmente compatible. RH-SSO 7.1 presenta adaptadores del cliente para Red Hat JBoss Fuse y Red Hat JBoss Enterprise Application Platform (JBoss EAP). RH-SSO 7.1 incluye un nuevo adaptador del cliente Node.js que permite la autenticación y el inicio de sesión único de la web de las aplicaciones Node.js. RH-SSO se puede integrar con los servicios de directorios basados en LDAP, incluidos Microsoft Active Directory y Red Hat Enterprise Linux Identity Management. RH-SSO también se integra con proveedores de inicio de sesión social, como Facebook, Google y Twitter.

Las API son clave para las aplicaciones compuestas por microservicios. Estas aplicaciones tienen múltiples servicios de API independientes que conducen a la proliferación de endpoints de servicios que requieren herramientas adicionales para fines de control. Además, recomendamos usar una herramienta de gestión de API: 3Scale de Red Hat. Esta herramienta le proporciona una variedad de opciones para la autenticación y seguridad de las API, que se pueden usar por separado o combinadas, para emitir credenciales y controlar el acceso. Estas opciones incluyen claves API estándar, id. de aplicaciones y par de claves, y OAuth 2.0.

El control de acceso de 3Scale presenta seguridad y autenticación que superan los requisitos básicos. Los planes de cuentas y aplicaciones le permiten restringir el acceso a endpoints, métodos y servicios específicos y aplicar políticas de acceso a los grupos de usuarios. Los planes de aplicaciones le permiten establecer límites de velocidad al uso de las API y controlar el flujo del tráfico para los

grupos de desarrolladores. Los límites establecidos por períodos para las API entrantes promueven la protección de la infraestructura y mantienen el flujo constante del tráfico. El excedente activado automáticamente alerta sobre las aplicaciones que alcanzan o exceden los límites de velocidad y definen el comportamiento de las aplicaciones que exceden los límites.

10. Gestión de funciones y acceso en una federación de clústeres

En julio de 2016, Kubernetes 1.3 presentó los clústeres federados de Kubernetes por primera vez. Esta es una de las nuevas y fascinantes funciones que han evolucionado en las innovadoras Kubernetes, actualmente en beta en Kubernetes 1.6. La federación es útil para implementar y acceder a los servicios de aplicaciones que abarcan múltiples clústeres y se ejecutan en la nube pública o centros de datos comerciales. Los múltiples clústeres pueden ser útiles para habilitar la alta disponibilidad de las aplicaciones en todas las zonas de disponibilidad múltiples o para habilitar la gestión común de las implementaciones o migraciones en proveedores de nube múltiples, como AWS, Google Cloud y Azure.

Cuando administre clústeres federados, necesitará asegurarse de que sus herramientas de orquestación proporcionen la seguridad requerida en todas las diversas instancias de la plataforma de implementación. Como siempre, la autenticación y la autorización son clave, como así también la capacidad para transmitir datos de forma segura a sus aplicaciones, donde sea que se ejecuten, y para administrar la arquitectura multiempresa de las aplicaciones en todos los clústeres. Kubernetes amplía la federación de clústeres para incluir soporte para secretos federados, espacios de nombre federados y objetos de ingreso.

Los secretos federados crean y administran automáticamente los secretos en todos los clústeres de una federación, y garantizan que se mantengan actualizados y uniformes a nivel global, incluso si algunos de los clústeres no están conectados al momento de la aplicación de las actualizaciones originales.

Los espacios de nombre federados son similares a los espacios de nombre de Kubernetes tradicionales y proporcionan la misma funcionalidad. Crear espacios de nombre en el plano de control de la federación asegura que estén sincronizados en todos los clústeres de la federación.

Red Hat trabaja con la comunidad y añadirá estas capacidades a OpenShift a medida que se desarrollen.

CONCLUSIÓN

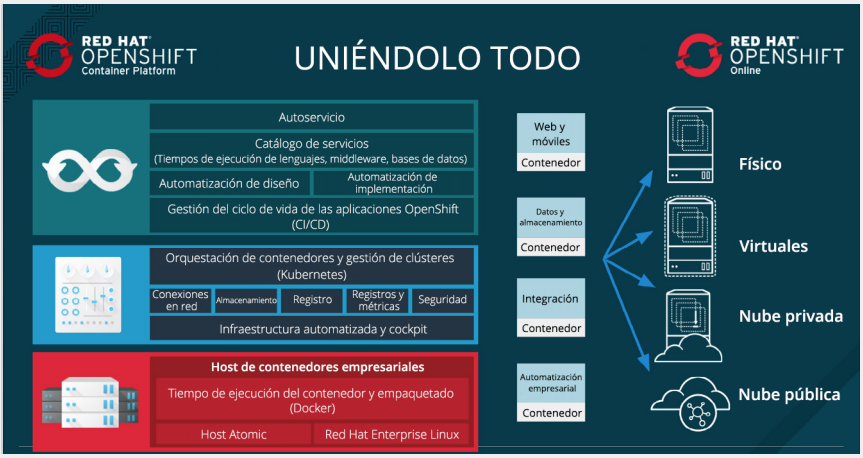
Ciertamente, no solo se trata de la seguridad. Su plataforma de contenedores necesita proporcionar una experiencia que funcione para sus desarrolladores y su equipo de operaciones. Necesita una plataforma de aplicaciones basada en contenedores y de grado empresarial que habilite tanto a desarrolladores como operadores, sin comprometer las funciones requeridas por cada equipo, y que también mejore la eficiencia operativa y el uso de la infraestructura.

OpenShift 3 está diseñado en un núcleo de contenedores de Linux estándar y portátiles. OpenShift proporciona funciones de seguridad incorporadas que incluyen lo siguiente:

- Sólidos controles de acceso basados en funciones, con integraciones con los sistemas de autenticación de la empresa.
- Orquestación de contenedores potente y a escala de la web, y gestión con Google Kubernetes.
- Red Hat Enterprise Linux 7 y Red Hat Enterprise Linux Atomic Host integrados, optimizados para ejecutar contenedores a escala con SELinux habilitado para lograr un aislamiento sólido.
- Integración con registros públicos y privados.
- Herramientas de CI/CD integradas para prácticas DevOps seguras.
- Un nuevo modelo para conexiones en red de contenedores.
- Soporte para volúmenes de almacenamiento remoto.

OpenShift también proporciona el grupo más grande de marcos, servicios y lenguajes de programación compatibles. Y el soporte para clústeres federados ya se encuentra en camino.

OpenShift está disponible para ejecutar en OpenStack, VMware, AWS, GCP, Azure y en cualquier plataforma que proporcione Red Hat Enterprise Linux 7. Red Hat también proporciona OpenShift Dedicated y OpenShift Online como servicios de nube pública.



Como proveedor líder de soluciones completas de open source, confiables y seguras a clientes empresariales durante más de 15 años, Red Hat ahora brinda el mismo nivel de confianza y seguridad para contenedores mediante soluciones como Red Hat Enterprise Linux, Red Hat OpenShift Container Platform y toda nuestra cartera de productos Red Hat habilitados para contenedores.

ACERCA DE RED HAT, INC.

Red Hat es el proveedor líder mundial de soluciones open source empresarial, con un enfoque impulsado por la comunidad para la obtención de tecnologías cloud, Linux, middleware, almacenamiento y virtualización de alta fiabilidad y rendimiento. Red Hat también ofrece servicios de soporte, formación y consultoría. Como eje central de una red global de empresas, partners y comunidades open source, Red Hat ayuda a crear tecnologías competentes e innovadoras que liberan recursos para el crecimiento y preparación de los consumidores para el futuro de las TI. Conozca más en http://es.redhat.com.

ARGENTINA
Ingeniero Butty 240, 14º piso
Ciudad de Buenos Aires
Argentina
+54 11 4329 7300

CHILE
Avda. Apoquindo Nº 2827
oficina 701, Piso 7
Los Condes, Santiago, Chile
+562 2597 7000

COLOMBIA
Red Hat Colombia S.A.S
Cra 9 No. 115-06 Piso 19 Of 1906
Edificio Tierra Firme Bogota,
Colombia
+571 5088631
+52 55 8851 6400

MÉXICO
Calle Río Lerma 232
Cuauhtémoc
06500 Ciudad de México
Mexico
+52 55 8851 6400

ESPAÑA
Torre de Cristal
Paseo de la Castellana 259C
Piso 17 Norte
28046 Madrid
+34 914148800

facebook.com/redhatinc
@RedHatIberia
Red Hat EMEA

es.redhat.com
#f7530_0517