DETALHES DA TECNOLOGIA

DEZ CAMADAS PARA SEGURANÇA DE CONTAINERS

INTRODUÇÃO

Os containers têm muitos atrativos, pois eles permitem que os usuários empacotem com facilidade uma aplicação e todas as suas dependências em uma única imagem. Essa imagem pode ser utilizada durante as fases de desenvolvimento, teste e produção, sem alterações. Com os containers, é mais fácil garantir a consistência nos ambientes e em vários destinos de implantação, como servidores físicos, máquinas virtuais (VMs) e clouds públicas ou privadas. Isso ajuda as equipes a desenvolver e gerenciar com mais facilidade as aplicações que agregam valor ao negócio.

As empresas precisam de segurança reforçada. E qualquer pessoa que executa serviços essenciais em containers se questiona se os containers são seguros e se é possível confiar suas aplicações neles. Este artigo descreve os dez elementos principais de segurança para camadas diferentes do stack de solução de container e os diversos estágios do ciclo de vida do container.

PROTEÇÃO DE CONTAINERS: CAMADAS E CICLO DE VIDA

A proteção de containers é muito semelhante à de qualquer outro processo em execução. Antes de implantar e executar o container, é preciso pensar na segurança em todas as camadas do stack da solução, bem como em todo o ciclo de vida da aplicação e do container. Os dez elementos principais para a segurança de containers são:

1. Multilocação do host de containers
2. Conteúdo dos containers
3. Registros dos containers
4. Compilação de containers
5. Implantação de containers
6. Orquestração de containers
7. Isolamento de rede
8. Armazenamento
9. Gerenciamento das interfaces de programação de aplicações (API)
10. Clusters federados

1. O sistema operacional host do container e a multilocação

Com os containers, os desenvolvedores têm mais facilidade para criar e promover uma aplicação e suas dependências como uma unidade. Os containers também facilitam o uso máximo dos servidores habilitando as implantações de aplicações de multilocação em um host compartilhado. É possível implantar várias aplicações em um único host inicializando e encerrando containers individuais conforme necessário. E, diferentemente da virtualização tradicional, não é necessário um hipervisor nem o gerenciamento dos sistemas operacionais guest em cada máquina virtual. Os containers virtualizam os processos da aplicação e não o hardware.

Para aproveitar todos os benefícios desta tecnologia de empacotamento e implantação, a equipe de operações precisa do ambiente ideal para executar os containers. Ou seja, um sistema operacional que proteja os containers nos seus limites: protegendo o kernel do host a partir de container escapes, além de protegê-los uns dos outros.

Os containers são processos Linux® com isolamento e confinamento de recursos que permitem a execução de aplicações em sandbox em um kernel de host compartilhado. A abordagem adotada para a proteção dos containers deve ser igual à de qualquer outro processo em execução no Linux. Eliminar privilégios é importante e ainda é a melhor prática. O ideal é criar containers com a menor quantidade possível de privilégios. Os containers devem ser executados como usuário e não como root. E então, usar os vários níveis de segurança disponíveis no Linux. Há cinco recursos de segurança disponíveis para proteger os containers executados no Red Hat® Enterprise Linux. São eles: namespaces do Linux, Security-Enhanced Linux (SELinux), grupos de controle (Cgroups), recursos e modo de computação segura (seccomp).

- Namespaces do Linux fornecem a base do isolamento do container. Um namespace indica aos processos que eles têm suas próprias instâncias de recursos globais. Os namespaces criam uma abstração que dá a entender que a execução está acontecendo em seu próprio sistema operacional, quando, na verdade, ela ocorre no container.
- SELinux fornece uma camada adicional de segurança para manter os containers isolados uns dos outros e do host. O SELinux permite que os administradores imponham controles de acesso obrigatórios (MAC) a todos os usuários, aplicações, processos e arquivos. O SELinux funciona como uma parede de tijolos que impede o rompimento (acidentalmente ou de propósito) da abstração do namespace.

- Cgroups (grupos de controle) limitam, representam e isolam o uso de recursos (por exemplo, CPU, memória, E/S de disco, rede) de uma coleção de processos. Use os Cgroups para garantir que o container não seja suplantado por outro no mesmo host. Outra atribuição dos Cgroups é controlar dispositivos falsos, um vetor de ataque popular.
- Recursos do Linux podem ser usados para bloquear o acesso root em um container. Os recursos são unidades de privilégio distintas que podem ser ativadas ou desativas de forma independente. Com eles, é possível executar ações como envio de pacotes IP brutos ou vinculação a portas abaixo de 1024. Durante a execução dos containers, é possível eliminar vários recursos sem impactar a grande maioria das aplicações em containers.
- Por fim, um perfil de modo de computação segura (seccomp) pode ser associado a um container para restringir as chamadas de sistema disponíveis.

É possível aumentar ainda mais a segurança das aplicações e da infraestrutura implantando os containers em um sistema operacional leve e otimizado para executar containers Linux, como o Red Hat Enterprise Linux Atomic Host. O Atomic Host reduz a superfície de ataque ao minimizar o ambiente host e ajustá-lo aos containers.

Como comprovação dos recursos de segurança disponíveis com o Red Hat Enterprise Linux e o Atomic Host, recentemente o Red Hat Enterprise Linux 7.1 recebeu a certificação Common Criteria, incluindo a certificação do Linux Container Framework Support.

A virtualização tradicional também possibilita a multilocação, mas de uma maneira bem diferente. A virtualização depende de um hipervisor que inicializa as máquinas virtuais guest, cada uma com seu próprio sistema operacional, bem como a aplicação em execução e suas dependências. Com as máquinas virtuais, o hipervisor isola as máquinas guest umas das outras e do host. Assim, menos usuários e processos têm acesso ao hipervisor, o que reduz a superfície de ataque no servidor físico. No entanto, a segurança ainda deve ser monitorada quanto a ameaças. Por exemplo, uma máquina virtual guest pode usar bugs do hipervisor para obter acesso a outra máquina virtual ou ao kernel do host. E, quando o sistema operacional precisar da aplicação de um patch, isso deverá ser feito em todas as máquinas virtuais guest que usam o mesmo sistema.

Os containers podem ser executados nas máquinas virtuais guest e, em alguns caso, isso será desejável. Por exemplo, se estiver implantando uma aplicação tradicional em um container, talvez seja necessário colocar o container em uma máquina virtual guest para fazer o "lift and shift" da aplicação para a cloud. No entanto, a multilocação de containers em um único host fornece uma solução de implantação mais leve, flexível e fácil de escalar. Esse modelo de implantação é adequado principalmente para aplicações nativas da cloud.

2. Conteúdo dos containers (use fontes confiáveis)

Quando o assunto é segurança, não ignore o conteúdo do container. Atualmente, as aplicações e as infraestruturas são formadas por componentes com disponibilidade imediata. Muitos deles são pacotes open source, como ocorre com o sistema operacional Linux, o Apache Web Server, o Red Hat JBoss® Enterprise Application Platform, o PostgreSQL e o Node.js. Atualmente, as versões em container desses pacotes também têm disponibilidade imediata. Portanto, não é necessário criá-las. No entanto, como acontece com qualquer código vindo de uma fonte externa, é preciso conhecer a procedência dos pacotes, quem os criou e se há códigos maliciosos neles. Pergunte a si mesmo:

- O conteúdo dos containers pode comprometer a minha infraestrutura?
- Existem vulnerabilidades conhecidas na camada da aplicação?
- As camadas do sistema operacional e ambiente de execução estão atualizados?
- Com que frequência o container será atualizado? Como saberei quando ele está atualizado?

Há muitos anos a Red Hat empacota e fornece conteúdo Linux confiável no Red Hat Enterprise Linux e em todo o nosso portfólio de soluções. Agora, a Red Hat está fornecendo o mesmo conteúdo confiável empacotado como containers Linux. Isso inclui: imagens de base do Red Hat Enterprise Linux 7 e Red Hat Enterprise Linux 6, uma grande quantidade de imagens certificadas para vários ambientes de execução de linguagem, middleware, bancos de dados e muito mais por meio do Red Hat Container Catalog. Os containers certificados da Red Hat são executados em qualquer ambiente que o Red Hat Enterprise Linux seja executado, inclusive bare-metal, máquinas virtuais e a cloud. Esses containers contam com o suporte da Red Hat e de nossos parceiros.

O conteúdo da imagem do container é empacotado a partir do código-fonte conhecido. A Red Hat também fornece monitoramento de segurança. Com o novo Container Health Index, a Red Hat classifica o "grau" de cada imagem de container detalhando como as imagens do container devem ser curadas, consumidas e avaliadas para atender às necessidades dos sistemas de produção. Parte da classificação dos containers toma como base a maturidade e o impacto da errata de segurança não aplicada para todos os componentes do container. Isso proporciona uma classificação agregada do grau de segurança do container, que pode ser compreendida por especialistas em segurança e por leigos.

Quando a Red Hat lança atualizações de segurança, como correções na biblioteca Glibc, no Drown ou no Dirty Cow, as imagens de container também são recriadas e enviadas para o nosso registro público. Por meio do Red Hat Security Advisories, a Red Hat envia alertas de segurança sobre qualquer problema detectado nas imagens de container certificadas e o direciona para a versão que foi atualizada. Assim, é possível atualizar todas as aplicações que usam essa imagem.

Obviamente, haverá casos em que você precisará de um conteúdo que a Red Hat não fornece. Recomendamos a adoção de ferramentas de verificação de containers que usam bancos de dados de vulnerabilidade que são atualizados com frequência. Isso garante que você tenha sempre as informações mais recentes sobre as vulnerabilidades conhecidas ao usar imagens de container de outras fontes. A lista de vulnerabilidades conhecidas está em constante evolução. Por isso, quando fizer o primeiro download das imagens de container, verifique o conteúdo delas e acompanhe sempre o status de vulnerabilidade de todas as imagens aprovadas e implantadas. É isso que a Red Hat faz com suas próprias imagens de container.

A Red Hat fornece uma API plugável no Red Hat Enterprise Linux para dar suporte a vários verificadores, como OpenSCAP, Black Duck Hub, JFrog Xray e Twistlock. O Red Hat CloudForms também pode ser usado com o OpenSCAP para verificar problemas de segurança nas imagens de container. Além disso, o Red Hat OpenShift Container Platform permite que você use verificadores com o seu processo de integração e entrega contínuas (CI/CD). Isso é abordado de forma mais detalhada a seguir.

3. Registros de containers (acesso seguro às imagens de container)

Naturalmente, suas equipes estão criando containers que acrescentam conteúdo às imagens de container públicas das quais você faz download. É preciso gerenciar o acesso e divulgação das imagens de container baixadas, bem como das imagens criadas internamente, da mesma forma que gerencia outros tipos de binários. Há vários registros privados que oferecem suporte ao armazenamento de imagens de container. Recomendamos a seleção de um registro privado que ajude você a automatizar as políticas de uso das imagens de container armazenadas no registro.

O Red Hat OpenShift Container Platform inclui um registro privado que pode ser usado para gerenciar suas imagens de container. O registro do OpenShift fornece controles de acesso baseados em funções que permitem que você determine quem pode extrair e implantar imagens de container específicas. O OpenShift também oferece suporte à integração com outros registros privados que talvez já esteja em uso na sua empresa, como o JFrog's Artifactory e o Docker Trusted Registry.

A lista de vulnerabilidades conhecidas está em constante evolução. Por isso, acompanhe sempre o conteúdo das imagens de container implantadas, bem como as imagens recém-baixadas. O seu registro deve incluir recursos que ajudem você a gerenciar o conteúdo com base nos metadados sobre o container, incluindo as vulnerabilidades conhecidas. Por exemplo, você pode usar a análise SmartState do Red Hat CloudForms para sinalizar imagens vulneráveis no registro. Assim que uma imagem é sinalizada, o OpenShift impede que ela seja executada.

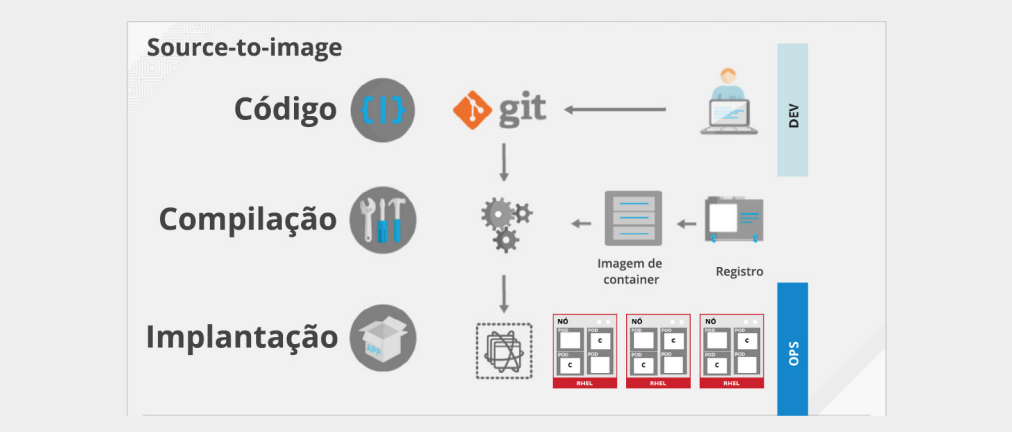
Nas próximas seções, destacaremos outras maneiras como o OpenShift oferece suporte às políticas de segurança de automação para conteúdo de containers.

4. Segurança e processo de compilação

Em um ambiente de container, o processo de compilação do software é o estágio do ciclo de vida em que o código da aplicação é integrado às bibliotecas do ambiente de execução necessárias. Gerenciar esse processo de compilação é fundamental para proteger o stack de software. Ao aderir a filosofia de "compilar uma vez, implantar em qualquer ambiente", você tem a garantia de que o resultado desse processo de compilação será exatamente o que foi implantado na produção. Também é importante manter a imutabilidade dos seus containers, isto é, não aplicar patches em containers em execução. Nesse caso, os containers devem ser recompilados e reimplantados.

O Red Hat OpenShift Container Platform fornece diversos recursos para o gerenciamento e a segurança da compilação:

- O Source-to-image (S2I) é uma estrutura open source para combinar código-fonte e imagens base. Com o S2I, suas equipes de desenvolvimento e operações têm facilidade de colaborar em um ambiente de compilação reproduzível. Quando um desenvolvedor atualiza o código com o git, no S2I, o OpenShift pode:
  - Acionar (via webhooks no repositório do código ou em algum outro processo de integração contínua automatizada) a montagem automática de uma nova imagem a partir de artefatos disponíveis, incluindo a imagem de base do S2I e o código recém-atualizado.
  - Implantar automaticamente a imagem recém-criada para teste.
  - Promover a imagem testada ao status de produção e implantar a nova imagem automaticamente por meio do processo de integração contínua (CI).



O Red Hat OpenShift Container Platform inclui uma instância integrada do Jenkins para a CI. O OpenShift também inclui APIs RESTful que você pode usar para integrar sua própria compilação ou ferramentas de integração contínua (CI) ou ainda o registro de imagens privadas, como o JFrog's Artifactory.

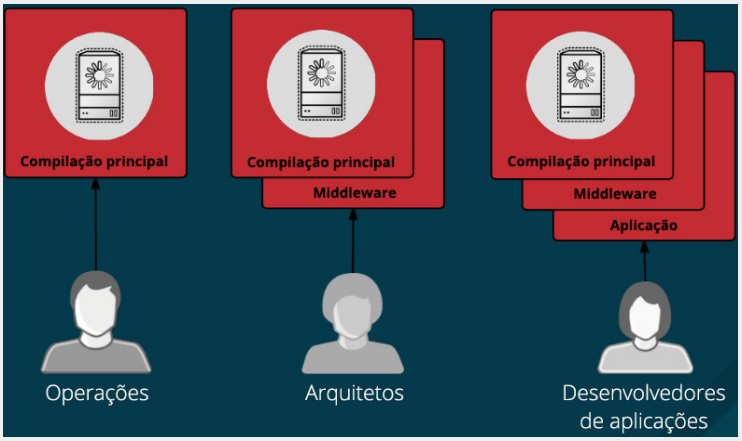
Uma prática recomendada para a segurança da aplicação é integrar o teste de segurança automatizado à compilação ou ao processo de CI. Por exemplo, é possível integrar:

- Ferramentas de teste estático de segurança de aplicações (SAST) e de teste dinâmico de segurança de aplicações (DAST) como HP Fortify e IBM AppScan.
- Verificadores em tempo real de vulnerabilidades conhecidas como o Black Duck Hub e o JFrog Xray. Ferramentas como essas catalogam os pacotes open source no seu container e notificam você sobre qualquer vulnerabilidade conhecida. Elas também informam novas vulnerabilidades descobertas em pacotes verificados anteriormente.

Além disso, o processo de integração contínua (CI) deve incluir políticas que sinalizam compilações com problemas detectados por verificações de segurança. Dessa forma, sua equipe pode tomar as medidas adequadas para resolver esses problemas o quanto antes.

Não importa se você trabalha em um setor altamente regulamentado ou se apenas deseja otimizar os esforços da sua equipe. Por isso, recomendamos que você planeje o gerenciamento da sua imagem de container e seu processo de compilação para se beneficiar das camadas do container e implementar a separação do controle. Dessa forma:

- A equipe de operações gerencia as imagens base.
- Os arquitetos gerenciam o middleware, os ambientes de execução, os bancos de dados e outras soluções semelhantes.
- Os desenvolvedores se concentram nas camadas da aplicação e apenas escrevem códigos.



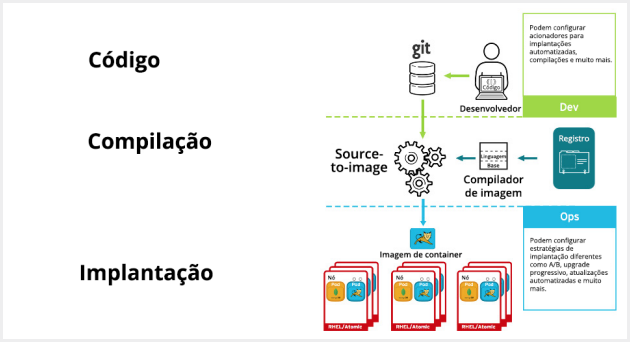
Por fim, recomendamos que você assine seus containers personalizados para ter certeza de que eles não serão adulterados entre a compilação e a implantação.

5. Implantação: controle o que pode ser implantado em um cluster

Caso ocorra algum problema durante o processo de compilação ou se uma vulnerabilidade for descoberta depois da implantação de uma imagem, você precisará de outra camada de segurança: ferramentas para implantação automatizada e baseadas em políticas.

Vejamos uma aplicação criada com três camadas de imagens de container: principal, middleware e camada de aplicação. É descoberto um problema na imagem principal, e essa imagem é recriada. Depois de concluída a compilação, a imagem é enviada para o registro do OpenShift. O OpenShift consegue detectar que a imagem foi alterada. No caso das compilações que dependem dessa imagem e que têm acionadores definidos, o OpenShift recriará automaticamente a imagem da aplicação incorporando as bibliotecas corrigidas.

Depois de concluída a compilação, a imagem é enviada para o registro interno do OpenShift. O OpenShift detecta de imediato as alterações nas imagens no registro interno e, no caso das aplicações com acionadores definidos, implanta automaticamente a imagem atualizada. Isso garante que o código executado em produção seja sempre igual ao da imagem atualizada mais recentemente. Todos esses mecanismos funcionam em conjunto para integrar os recursos de segurança ao processo e pipeline de integração e entrega contínuas (CI/CD).



A segurança reforçada inclui políticas automatizadas que podem ser usadas para gerenciar a implantação do container do ponto de vista da segurança. O OpenShift vem com restrições de contexto de segurança (Security Context Constraints, SCCs) integradas que podem ser usadas para definir um conjunto de condições que um pod (coleção de containers) deve executar para ser aceito no sistema. As restrições de contexto de segurança (SCCs) do OpenShift, que contribuem para o Kubernetes como uma política de segurança do pod, permitem ao administrador controlar:

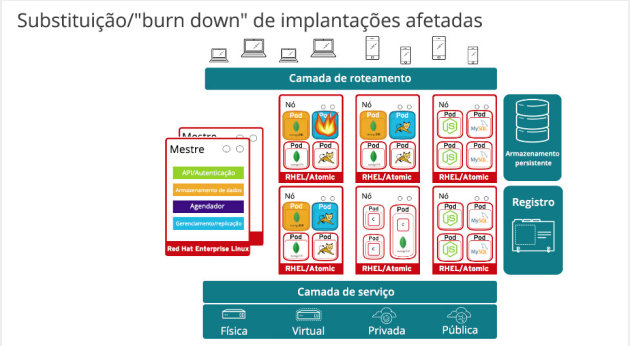
- A execução dos containers com privilégios.
- Os recursos que um container pode solicitar para ser adicionado.
- O uso dos diretórios de host como volumes.
- O contexto do SELinux do container.
- A criação dos perfis de seccomp.
- A ID de usuário do container.

Os usuários que têm as permissões exigidas podem ajustar as políticas de restrição de contexto de segurança (SCC) para que elas sejam mais permissivas, se assim for desejado. Assim como ocorre com as ferramentas de integração contínua (CI), se preferirem, os usuários podem integrar suas próprias ferramentas de entrega contínua (CD) à plataforma do container.

Ao integrar a CI/CD ao OpenShift, é possível automatizar todo o processo de recompilação da aplicação para incorporar as últimas correções e testes e garantir que ele seja implantado em todo o ambiente.

6. Orquestração de containers: proteção da plataforma de containers

Naturalmente, é muito difícil que as aplicações sejam entregues em um único container. Mesmo as aplicações mais simples costumam ter um fron-end, um back-end e um banco de dados. E implantar aplicações modernas baseadas em microsserviços em containers significa implantar vários containers. A implantação pode ser feita no mesmo host ou distribuída entre vários hosts ou nós como mostra o diagrama a seguir.



Ao gerenciar a implantação de um container em escala, considere o seguinte:

- Quais containers devem ser implantados em quais hosts?
- Qual host tem mais capacidade?
- Quais containers precisam acessar outros? Como eles se descobrirão?
- Como controlar o acesso aos recursos compartilhados (rede e armazenamento, por exemplo) e como gerenciá-los?
- Como monitorar a integridade dos containers?
- Como escalar a capacidade da aplicação automaticamente para atender à demanda?
- Como habilitar o autosserviço do desenvolvedor e atender aos requisitos de segurança?

É possível criar seu próprio ambiente de gerenciamento de containers. No entanto, para que investir tempo criando ferramentas se é possível implantar uma plataforma de containers que oferece recursos integrados de gerenciamento e segurança? Com essa plataforma, sua equipe pode se concentrar na criação de aplicações que agreguem valor ao negócio.

O Red Hat OpenShift Container Platform oferece orquestração de containers, automação de agendamento e execução de containers de aplicação em clusters de máquinas físicas ou virtuais por meio da inclusão e extensão do projeto Kubernetes open source. O Kubernetes, um projeto open source iniciado pelo Google, usa "mestres" para gerenciar a complexidade da orquestração do cluster do container. O OpenShift também vem com o Red Hat CloudForms que, entre outras tarefas, pode ser usado para monitorar a integridade dos containers no seu registro privado e impedir a implantação de containers com vulnerabilidades recém-detectadas.

Graças aos inúmeros recursos para as esquipes de desenvolvimento e operações, o forte controle de acesso baseado em funções é um elemento crítico da plataforma de containers. Por exemplo, os mestres de orquestração são um ponto central de acesso e devem receber o mais alto nível de análise de segurança. As APIs são fundamentais para automatizar o gerenciamento de containers em escala. Elas são usadas para validar e configurar os dados para os pods, serviços e controladores de replicação, para validar os projetos nas solicitações recebidas e chamar acionadores nos outros principais componentes do sistema.

O controle de acesso à API (autenticação e autorização) é crítico para proteger a plataforma de containers. O **mestre** do OpenShift inclui um **servidor OAuth** integrado. Os desenvolvedores e administradores obtêm **tokens de acesso ao OAuth** para se autenticar na API. Como administrador, você pode configurar o OAuth para se autenticar usando o **provedor de identidade** de sua escolha, incluindo os diretórios Lightweight Directory Access Protocol (LDAP).

Um dos principais valores de uma plataforma de containers é a capacidade de habilitar o autosserviço do desenvolvedor. Isso permite que as equipes de desenvolvimento sejam mais ágeis para fornecer aplicações criadas em camadas aprovadas. A segurança de multilocação é obrigatória para a própria plataforma garantir que as equipes não acessem os ambientes umas das outras sem autorização. Você precisa de um portal de autosserviço que ofereça controle suficiente às equipes para promover a colaboração sem deixar de proporcionar segurança. O OpenShift acrescenta vários componentes aos Kubernetes para manter um mestre de multilocação segura, garantindo que:

- Todo o acesso ao mestre seja feito por uma segurança de camada de transporte (TLS).
- O acesso ao servidor de API seja baseado em token ou em certificados X.509.
- A cota de projeto seja usada para limitar os danos causados por um token não autorizado.
- O Etcd não seja exposto diretamente ao cluster.

O OpenShift 3.5 oferece **gerenciamento aperfeiçoado de cluster**, incluindo chaves secretas aprimoradas e gerenciamento de certificado.

7. Isolamento de sistema de rede

Em geral, implantar aplicações modernas baseadas em microsserviços em containers significa implantar vários containers distribuídos em diversos nós. É necessário isolar as aplicações umas das outras em um cluster, sem ignorar a defesa de rede.

Serviços comuns de container de cloud pública, como o Google Container Engine (GKE), o Azure Container Services ou a Amazon Web Services (AWS), são de locatário único. Com eles, é possível executar os containers no cluster da máquina virtual iniciada. Para uma multilocação segura de container, você precisa de uma plataforma que adote um único cluster e segmente o tráfego para isolar usuários, equipes, aplicações e ambientes diferentes naquele cluster.

Com os namespaces de rede, cada coleção de containers (conhecida como "pod") obtém seu próprio endereço IP e intervalo de porta ao qual se vincular, isolando as redes de pod umas das outras no nó. Por padrão, os pods de diferentes namespaces (projetos) não podem enviar ou receber pacotes dos pods e serviços de um projeto diferente, com exceção das opções observadas a seguir. É possível usar esses recursos para isolar os ambientes de desenvolvedor, teste e produção em um cluster.

No entanto, essa proliferação de endereços IP e portas causa mais complicações ao sistema de rede. Além disso, os containers são projetados para serem independentes. Recomendamos que você invista em ferramentas que resolvam essa complexidade. O ideal é adotar uma plataforma de containers que use um sistema de rede definido por software (SDN) para oferecer uma rede de clusters unificados que habilita a comunicação entre os containers pelo cluster.

Além disso, é melhor optar por uma plataforma de containers que permita o controle do tráfego de saída usando um método de roteador ou firewall. Dessa forma, é possível usar a lista de permissões do IP para controlar o acesso ao banco de dados, por exemplo.

Além dos namespaces de rede, o SDN fornece uma segurança adicional ao oferecer isolamento entre os namespaces mestre (orquestração) com o plug-in ovs-multitenant. Quando o plug-in ovs-multitenant é ativado, por padrão, o tráfego do pod em um namespace é isolado do tráfego de pod de outro namespace (projeto). Para fornecer exceções ao plug-in ovs-multitenant, a funcionalidade "oadm pod-network" foi incluída no Red Hat OpenShift Container Platform 3.1. O objetivo é permitir que dois projetos acessem os serviços uns dos outros ou permitir que todos os projetos acessem todos os pods e serviços no cluster. A limitação dessa funcionalidade é que ela opera no nível do projeto como um todo e o tráfego permitido é sempre bidirecional. Em outras palavras, se é possível acessar um dos serviços em um projeto, todos os outros também estão disponíveis nesse mesmo projeto. Sendo assim, você automaticamente concedeu acesso a todos os serviços do projeto. Como as permissões são bidirecionais, isso só pode ser configurado pelo administrador do cluster.

Anunciado como uma apresentação prévia da tecnologia no Red Hat OpenShift Container Platform 3.5, um novo plug-in de política de rede (ovs-networkpolicy) foi desenvolvido para melhorar a forma como o plug-in ovs-multitenant pode ser usado para configurar o tráfego permitido entre os pods. A política de rede permite a configuração das políticas de isolamento no nível dos pods individuais. Como as políticas de rede não são bidirecionais e se aplicam apenas ao tráfego de entrada dos pods em um controle do administrador do projeto, os privilégios de administrador do cluster também não são necessários.

Se você quiser implantar verificadores de rede, será fácil colocá-los em containers. Além disso, eles podem ser executados como "containers com superprivilégios".

8. Armazenamento

Os containers são úteis para as aplicações com ou sem monitoração de estado. A proteção do armazenamento anexado é um elemento importante dos serviços de segurança com monitoração de estado. O Red Hat OpenShift Container Platform fornece plug-ins para diversas opções de armazenamento, incluindo os sistemas de arquivo de rede (NFS), o AWS Elastic Block Stores (EBS), os discos persistentes do GCE, o GlusterFS, o iSCSI, o RADOS (Ceph) e o Cinder.

Um **volume persistente (PV)** pode ser montado em um host de qualquer forma compatível com o suporte do provedor de recursos. Os provedores terão recursos diferentes, e os modos de acesso de cada volume persistente serão definidos para os modos específicos que recebem suporte de um volume específico. Por exemplo, o NFS pode dar suporte a vários clientes de leitura/gravação, mas o volume persistente de um NFS específico pode ser exportado no servidor como somente leitura. Cada volume persistente tem seu próprio conjunto de modos de acesso que descreve os próprios recursos específicos. Alguns exemplos são ReadWriteOnce, ReadOnlyMany e ReadWriteMany.

Para o armazenamento compartilhado (NFS, Ceph, Gluster etc.), o truque é fazer com que o volume persistente de armazenamento registre sua ID de grupo (gid) como uma anotação no recurso do volume persistente. Quando o volume persistente é chamado pelo pod, o gid anotado é adicionado aos grupos complementares do pod e dá ao pod acesso ao conteúdo do armazenamento compartilhado.

Para o armazenamento em blocos (EBS, discos persistentes do GCE, iSCSI etc.), as plataformas de containers podem usar os recursos do SELinux para proteger a raiz do volume montado para pods sem privilégio. Assim, o volume montado fica sendo de propriedade do container ao qual está associado e só pode ser visualizado por esse container.

Os dados em trânsito devem ser criptografados via https para todos os componentes da plataforma de containers comunicarem entre si.

E, obviamente, você pode aproveitar os recursos de segurança disponíveis na solução de armazenamento escolhida.

9. Gerenciamento de API/segurança do endpoint e logon único (SSO)

A segurança das aplicações inclui o gerenciamento delas, bem como a autenticação e a autorização da API.

Os recursos de SSO da web é parte essencial das aplicações modernas. As plataformas de containers podem vir com vários serviços em container para uso dos desenvolvedores na criação de aplicações, como o Red Hat SSO (RH-SSO), uma autenticação SAML 2.0 com suporte total e pronta para uso ou com base no OpenID Connect, logon único na web e serviço de federação com base no projeto upstream Keycloak. O RH-SSO 7.1 conta com adaptadores de cliente para o Red Hat Fuse e para o JBoss Enterprise Application Platform (JBoss EAP). O RH-SSO 7.1 inclui um novo adaptador de cliente do Node.js, que habilita a autenticação e o logon único na web para aplicações Node.js. O RH-SSO pode ser integrado aos serviços de diretório com base em LDAP, incluindo o Microsoft Active Directory e o Red Hat Enterprise Linux Identity Management. O RH-SSO também se integra a provedores com login por mídia social, como Facebook, Google e Twitter.

As APIs são fundamentais para aplicações compostas de microsserviços. Essas aplicações têm vários serviços de API independentes. Isso gera a proliferação de endpoints de serviço que exigem ferramentas adicionais para governança. Além disso, recomendamos o uso de uma ferramenta de gerenciamento de API. O 3Scale by Red Hat oferece uma variedade de opções padrão para autenticação e segurança de API, que podem ser usadas sozinhas ou combinadas para emitir credenciais e controlar o acesso. Essas opções incluem chaves de API padrão, ID de aplicação e par de chaves e o OAuth 2.0.

Os recursos de controle de aceso do 3Scale vão além da segurança e autenticação básicas. Os planos de aplicação e conta permitem que você restrinja o acesso a endpoints, métodos e serviços específicos além de aplicar política de acesso a grupos de usuários. Os planos de aplicação permitem a definição de limites de taxa para o uso da API e fluxo de controle de tráfego para grupos de desenvolvedores. Defina limites por período para chamadas de API recebidas, a fim de proteger sua infraestrutura

e manter o tráfego fluindo sem problemas. Acione automaticamente alertas de sobrecarga para aplicações que atinjam ou excedam limites de taxa e defina o comportamento para aplicações além do limite.

10. Funções e gerenciamento de acesso em uma federação de clusters

Em julho de 2016, o Kubernetes 1.3 apresentou o Kubernetes Federated Clusters pela primeira vez. Esse é um dos novos recursos interessantes que estão sendo desenvolvidos no upstream do Kubernetes, atualmente na versão beta do Kubernetes 1.6. A federação é útil para implantar e acessar serviços de aplicação que abrangem vários clusters executados na cloud pública ou em datacenters corporativos. Vários clusters podem ser úteis para habilitar a alta disponibilidade da aplicação em várias zonas ou para habilitar o gerenciamento comum das implantações ou migrações em vários provedores de cloud, como a AWS, o Google Cloud e o Azure.

Ao gerenciar clusters federados, é necessário ter certeza de que as ferramentas de orquestração fornecem a segurança necessária nas diferentes instâncias da plataforma de implementação. Como sempre, a autenticação e a autorização são essenciais, assim como a capacidade de transmitir dados com segurança para suas aplicações, onde quer que elas sejam executadas, e gerenciar a multilocação da aplicação em clusters. O Kubernetes está ampliando a federação de clusters para incluir suporte às chaves secretas federadas, aos namespaces federados e aos objetos de entrada.

As chaves secretas federadas criam e gerenciam automaticamente chaves secretas em todos os clusters de uma federação. Isso garante que elas sejam mantidas consistentes e atualizadas globalmente, mesmo que alguns clusters estejam offline quando as atualizações originais forem aplicadas.

Os namespaces federados são semelhantes aos tradicionais do Kubernetes e fornecem a mesma funcionalidade. Criar namespaces no plano de controle da federação garante que eles sejam sincronizados em todos os clusters.

A Red Hat trabalha em colaboração com a comunidade upstream. Sendo assim, à medida que esses recursos amadurecem, eles são adicionados ao OpenShift.

CONCLUSÃO

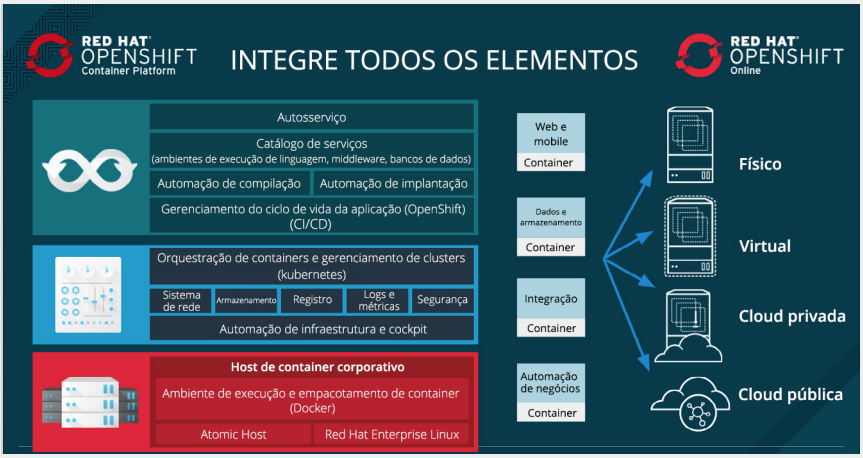
É claro que não estamos falando apenas de segurança. Sua plataforma de containers precisa proporcionar uma experiência válida para as equipes de desenvolvimento e de operações. Além disso, é necessária uma plataforma de aplicações corporativa segura e baseada em containers que facilite o trabalho dessas equipes, sem comprometer as funções necessárias de cada uma. E essa plataforma também precisa aumentar a eficiência operacional e o uso da infraestrutura.

O OpenShift 3 é criado em uma base de containers Linux padrão e portáteis. O OpenShift oferece recursos de segurança integrados que incluem:

- Controles de acesso reforçado baseados em funções com integrações para os sistemas de autenticação corporativos.
- Gerenciamento e orquestração avançados de containers em escala web com o Google Kubernetes.
- Red Hat Enterprise Linux 7 e Red Hat Enterprise Linux Atomic Host integrados e otimizados para executar containers em escala com o SELinux habilitado para isolamento sólido.
- Integração com os registros públicos e privados.
- Ferramentas de integração e entrega contínuas (CI/CD) para práticas seguras de DevOps.
- Um novo modelo de sistema de rede de container.
- Suporte para volumes de armazenamento remoto.

O OpenShift também fornece a maior coleção de linguagens de programação, estruturas e serviços para os quais é oferecido suporte. E o suporte para os clusters federados já está no roadmap.

O OpenShift está disponível para ser executado no OpenStack, VMware, AWS, GCP, Azure e em qualquer plataforma que execute o Red Hat Enterprise Linux 7. A Red Hat também fornece o OpenShift Dedicated e o OpenShift Online como serviços de cloud pública.



Ao assumir a liderança no fornecimento de soluções confiáveis, seguras e totalmente open source para clientes corporativos há mais de 15 anos, a Red Hat traz esse mesmo nível de confiança e segurança para os containers. Para isso, ela usa soluções como o Red Hat Enterprise Linux, o Red Hat OpenShift Container Platform e todo seu portfólio de soluções habilitadas para containers.

SOBRE A RED HAT

A Red Hat é a líder mundial no fornecimento de soluções de software open source, utilizando uma abordagem de parceria com as comunidades para oferecer tecnologias confiáveis e de alto desempenho de cloud, Linux, middleware, armazenamento e virtualização. A Red Hat conta com premiados serviços de suporte, treinamento e consultoria. Como um hub de conectividade em uma rede global de empresas, parceiros e comunidades open source, a Red Hat ajuda a criar tecnologias relevantes e inovadoras que permitem a ampliação recursos disponíveis e preparam os clientes para o futuro da TI.

Saiba mais em
http://www.redhat.com/pt-br

AMÉRICA LATINA
+54 11 4329 7300
latammktg@redhat.com

BRASIL
+55 11 3629 6000
marketing-br@redhat.com

facebook.com/redhatinc
@redhat
linkedin.com/company/red-hat

br.redhat.com
#f7530_0517