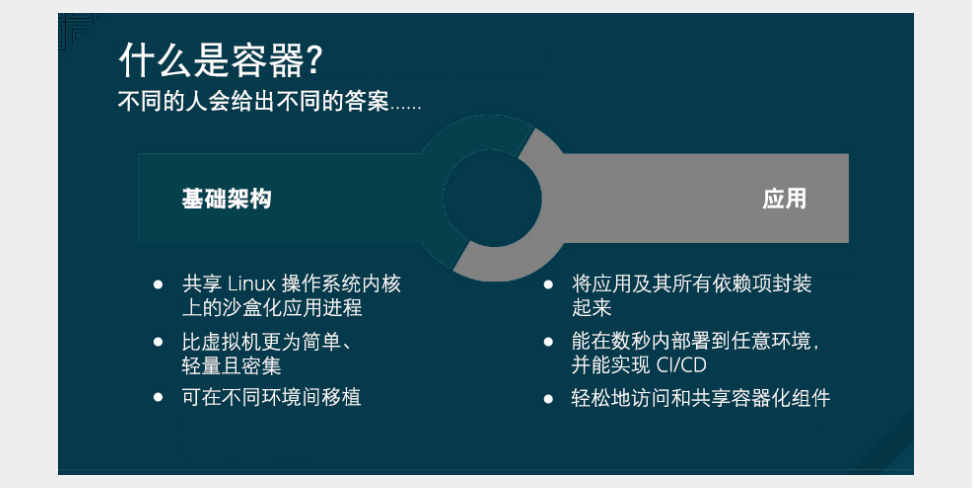
技术详情

容器安全防护的十个层级

简介

容器具有广泛的吸引力，因为用户可使用容器轻松地将应用及其所有依赖项封装到单个镜像中，而且这类镜像无需更改即可用于开发、测试和生产。容器能让您轻松地在多个环境和部署目标（如物理服务器、虚拟机 (VM) 和私有云或公共云）间保持一致性。得益于此，团队可以更加轻松地开发和管理应用，进而创造业务价值。



企业环境对安全性的要求很高，所有在容器中运行重要服务的企业都会问：“容器安全吗？”和“我们能够信任装有我们应用的容器吗？”本文将介绍与容器解决方案堆栈的不同层级及容器生命周期的不同阶段相关的 10 个关键要素。

红帽官方微博　红帽官方微信

cn.redhat.com

确保容器安全：层级和生命周期

确保容器的安全与确保所有运行中进程的安全很类似。在部署和运行容器之前，您要全面考虑解决方案堆栈的各个层级的安全性。您还需要全面考虑应用和容器整个生命周期的安全性。容器安全防护的 10 大关键要素分别为：

1. 容器主机多租户
2. 容器内容
3. 容器注册表
4. 构建容器
5. 部署容器
6. 容器编排
7. 网络隔离
8. 存储
9. 应用编程接口 (API) 管理
10. 联合集群

1. 容器主机操作系统和多租户

借助容器，开发人员可将应用及其依赖项作为一个单元来进行处理，以简化相关的构建和应用工作。容器还可在共享主机上实现多租户应用部署，从而轻松地实现对服务器的充分利用。您可以轻松地在单个主机上部署多个应用，并根据需要启用和关闭单个容器。另外，您也不需要在各个 VM 上分别安装虚拟机监控程序或管理虚拟客户机操作系统，这一点有别于传统虚拟化。容器能让应用进程（而非硬件）实现虚拟化。

要充分利用这种封装和部署技术，运营团队需要为所运行的容器构建适用的环境。运营团队还需要安装能在边界确保容器安全性的操作系统，从而避免主机内核出现容器逃逸情况，并确保容器不会相互影响。

容器就是相互隔离且存在资源约束的 Linux® 进程，它们能让您在共享主机内核上运行沙盒化应用。您的容器安全性方案应与保障 Linux 上所有运行进程安全性时所用的方案保持一致。适时放弃特权非常重要，而且这仍是最为有效的一种方式。但如果能在创建容器时尽可能地少设置一些特权，成效会更好。您应该以普通用户（而非根用户）的身份来运行容器。接着，充分发挥 Linux 多个安全性层级的作用。Linux 命名空间、安全增强型 Linux (SELinux)、Cgroups、功能和安全计算模式 (seccomp) 是五种安全措施，可用于确保红帽® 企业 Linux 上所运行容器的安全性。

- **Linux 命名空间**能为实现容器隔离奠定基础。使用命名空间后，其中所含的进程看上去就像是拥有了自己的全局资源实例。命名空间可以实现抽象化，让您在进入容器后就感觉像是在自己的操作系统上运行一样。
- **SELinux** 可以提供额外的安全保护，以使容器相互隔离开并与主机隔离开。SELinux 允许管理员针对每一个用户、应用、进程和文件实施强制访问控制 (MAC)。SELinux 就像一堵墙，当您设法突破（意外或故意）命名空间所形成的抽象化时，这堵墙会加以阻挡。
- **Cgroups**（控制组）会限制、监管并隔离**进程**集合对**资源的使用**（如 CPU、内存、磁盘 I/O、网络）。它可以确保您的容器不会被同一主机上的其他容器影响。Cgroups 还可用于控制伪设备（一种常见的攻击向量）。
- **Linux 功能**可用于将根锁定到容器中。功能是指可以单独启用或禁用的不同特权单元。借助功能，您可以执行发送原始 IP 数据包或绑定至 1024 以下的端口等操作。运行容器时，您可在不影响绝大多数容器化应用的情况下删除多个功能。
- 最后，**安全计算模式** (seccomp) 配置文件可与容器关联，限制可执行的系统调用。

您还可以将容器部署到专为运行 Linux 容器而进行过优化的轻量级操作系统上，如红帽企业 Linux Atomic Host，以便进一步提高应用和基础架构的安全性。Atomic Host 可最小化主机环境并根据容器对环境进行调优，从而减小攻击面。

近期，红帽企业 Linux 7.1 获得了通用标准认证，其中就包括 Linux 容器框架支持认证。这是表明红帽企业 Linux Atomic Host 能为您提供安全保障的力证之一。

传统虚拟化也支持多租户，但其所用的方式却大不相同。虚拟化有赖于虚拟机监控程序，后者则会使用客户机 VM（每个 VM 都有各自的操作系统） 以及正在运行的应用及其依赖项。借助 VM，虚拟机监控程序可使客户机相互隔离开并与主机隔离开。这样就能减少有权访问虚拟机监控程序的个人和进程，从而减小物理服务器的攻击面。但是，仍要实施安全监控，以检测各种威胁；例如，某个客户机 VM 或许能够利用虚拟机监控程序漏洞来获取访问其他 VM 或主机内核的权限。另外，如果操作系统需要安装补丁，则必须在使用该操作系统的所有客户机 VM 上进行安装。

容器可在客户机 VM 内运行，同时也可能存在需要在客户机 VM 内运行容器的用例。例如，如果您正在容器中部署传统应用，那么您可能会为了将应用提升并转换到云端，而想将容器置于客户机 VM 内。但是，在单个主机上进行容器多租户部署是一种更为轻便、灵活且易于扩展的部署解决方案。这种部署模式尤其适用于云原生应用。

2. 容器内容（使用可信来源）

说到安全性，重要的是保护容器中所含的内容。一段时间以来，应用和基础架构一直都是由各种即用型组件组合而成的。而且，其中的很多组件都是开源数据包，如 Linux 操作系统、Apache Web 服务器、红帽 JBoss® 企业应用平台、PostgreSQL 和 Node.js。现在，这些数据包的容器化版本也已准备就绪，您不必再自行构建。但是，在从外部来源下载任意代码时，您需要知晓这些数据包的原始来源、构建者以及其中是否含有任何恶意代码。请问问自己这几个问题：

- 容器中所含的内容是否会危及我的基础架构？
- 应用层是否存在已知的漏洞？
- 运行时和操作系统层是否处于最新状态？
- 容器将多久更新一次？当容器更新时，我将如何知晓这一情况？

多年来，红帽一直通过红帽企业 Linux 和我们的产品组合来封装和交付可信的 Linux 内容。现在，红帽正通过封装成 Linux 容器的方式来交付同样可信的内容，其中包括红帽企业 Linux 7 和红帽企业 Linux 6 的基本镜像。红帽还通过红帽容器目录针对各种语言运行时、中间件、数据库等提供了大量经过认证的镜像。不管是裸机、 VM 还是云端，只要能运行红帽企业 Linux，就可以运行经过红帽认证的容器。红帽和我们的合作伙伴均支持经过认证的容器。

容器镜像内容是由已知源代码封装而成的。红帽还会提供安全性监管。借助全新的容器健康指数，红帽可以揭示各个容器镜像的“等级”，从而详细指明应该如何管辖、使用和评估容器镜像，以满足生产系统的需求。在对容器进行评级时，所考虑的部分因素是未应用于容器内所有组件的安全勘误的已存在时长和所造成的影响，这样能得出一个安全专家和非专家人士都能理解的容器安全性总体评级。

当红帽发布安全性更新时（如针对 glibc、Drown 或 Dirty Cow 的修复），我们还会重构我们的容器镜像并将其推送到我们的公共注册表。红帽安全公告会向您发出提醒，告知您我们在认证容器镜像中新发现的所有问题，并指引您找到更新后的镜像，以便您转而更新使用该镜像的所有应用。

当然，有时您需要使用红帽并未提供的内容。我们建议您使用漏洞数据库会不断更新的容器扫描工具，以确保您在使用其他来源的容器镜像时始终能获得有关已知漏洞的最新信息。由于已知漏洞列表会不断变更，所以您在首次下载容器镜像时需要检查其所含的内容，并持续跟踪所有已获批和已部署镜像的漏洞状态，就像红帽跟踪红帽容器镜像那样。

红帽在红帽企业 Linux 中提供了一个可插拔式 API，该 API 可支持多种扫描程序，如 OpenSCAP、Black Duck Hub、JFrog Xray 和 Twistlock。红帽 CloudForms 还可与 OpenSCAP 搭配使用，通过扫描来确定容器镜像是否存在安全问题。另外，红帽 OpenShift 还能使用扫描程序来扫描持续整合和持续交付 (CI/CD) 进程。下文会对此进行更为详细的介绍。

3. 容器注册表（安全访问容器镜像）

当然，您的团队会构建内容基于所下载公共容器镜像的容器。您需要按照您在管理其他类型的二进制文件时所用的方式，来管理对已下载的容器镜像和内部构建镜像的访问权限及其应用方式。支持容器镜像存储的私有注册表有很多。我们建议，您选择的私有注册表应能帮助您自动实施与使用注册表中所存储容器镜像相关的策略。

红帽 OpenShift 包含可用于管理容器镜像的私有注册表。OpenShift 注册表提供了基于角色的访问权限控制功能，允许您管理哪些人员可以提取和推送特定的容器镜像。OpenShift 还支持与您可能已在使用的其他私有注册表进行整合，如 JFrog 的 Artifactory 以及 Docker 可信注册表。

由于已知漏洞列表会不断更新，所以您需要持续跟踪已部署的容器镜像和新下载的镜像中的内容。您的注册表应包含相应功能，以帮助您基于容器的相关元数据（包括已知漏洞）来管理内容。例如，您可以使用红帽 CloudForms 的 SmartState 分析功能来标记注册表中的漏洞镜像。标记之后，OpenShift 将会就此阻止该镜像运行。

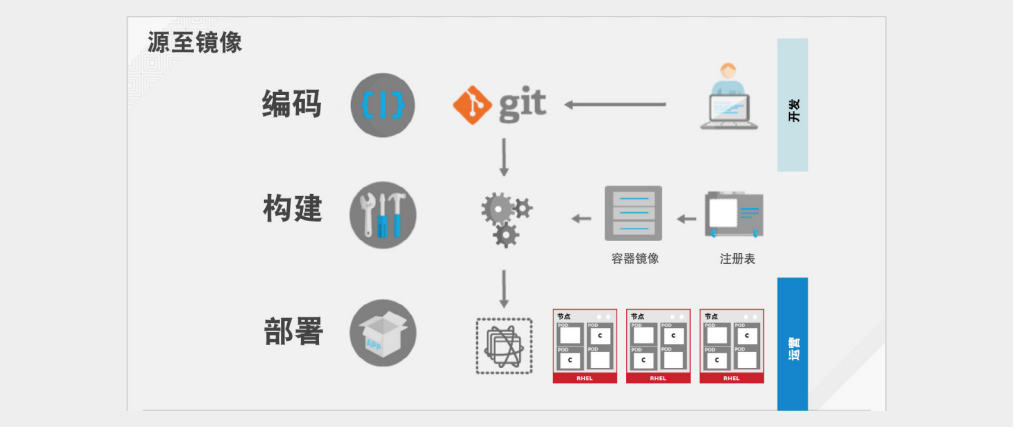
我们将在后续几个部分中重点介绍 OpenShift 支持自动实施容器内容安全策略的其他几种方式。

4. 安全性和构建流程

在容器化环境中，软件构建流程是指生命周期中将应用代码与所需运行时库进行整合的阶段。如何管理这个构建流程，是确保软件堆栈安全性的关键所在。如能遵循“一次构建，随心部署”的理念，即可确保构建流程的产物就是要在生产环境中部署的项目。另外，保持容器的不可变性也十分重要。换句话说，就是不要为正在运行的容器安装补丁，而应重新构建并重新部署这些容器。

红帽 OpenShift 提供了多种功能，可用于进行构建管理并确保安全性：

- “源至镜像”(S2I) 是一个用于组合源代码和基本镜像的开源框架。S2I 能让开发和运营团队轻松地在能够重现的构建环境中开展协作。当开发人员在 S2I 下使用 Git 提交代码时，OpenShift：
  - 可触发（通过代码存储库中的 webhook 或是其他的一些自动持续整合 (CI) 流程）从可用的工件（包括 S2I 基本镜像）和新提交的代码中自动组装新的镜像。
  - 自动部署新构建的镜像，以进行测试。
  - 可将经过测试的镜像置于生产状态，并通过 CI 流程自动部署新的镜像。



红帽 OpenShift 包含可用于实现 CI 的集成式 Jenkins 实例。OpenShift 还包含丰富多样的 RESTful API，可供您用于整合自己的构建工具、CI 工具或私有镜像注册表，如 JFrog 的 Artifactory。

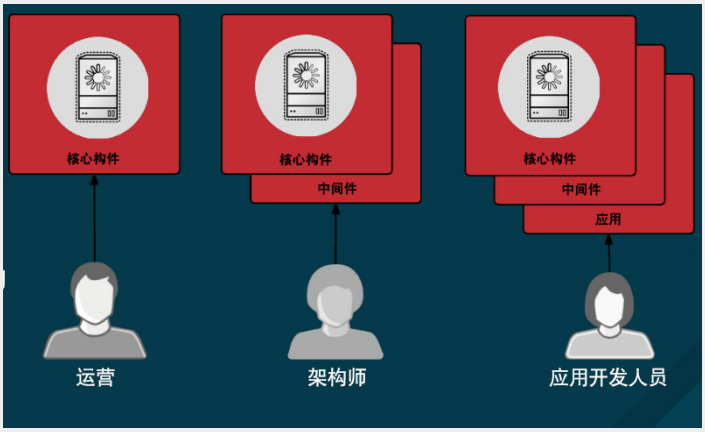
要确保应用的安全性，最好的做法就是将自动化安全测试功能整合到构建或 CI 流程中。例如，整合：

- 静态应用安全测试 (SAST) 和动态应用安全测试 (DAST) 工具，如 HP Fortify 和 IBM AppScan。
- 可用于实时检查已知漏洞的扫描程序，如 Black Duck Hub 和 JFrog Xray。这类工具可为容器中的开源数据包编目、就任何已知漏洞向您发出通知并在先前所扫描数据包中发现新漏洞时向您提供最新信息。

另外，CI 流程还应包含相应策略，以标记通过安全扫描发现存有问题的构件。这样，您的团队便可采取相应措施，尽早处理这些问题。

无论您是身在监管严格的行业，还是单纯地想优化您团队的工作成效，我们都建议您精心设计容器镜像的管理和构建流程，以便充分利用容器的各个层级来实现控制分离，从而使：

- 运营团队能够管理基本镜像。
- 架构师能够管理中间件、运行时、数据库等解决方案。
- 开发人员能专注于应用层开发，而且只需要负责编写代码。



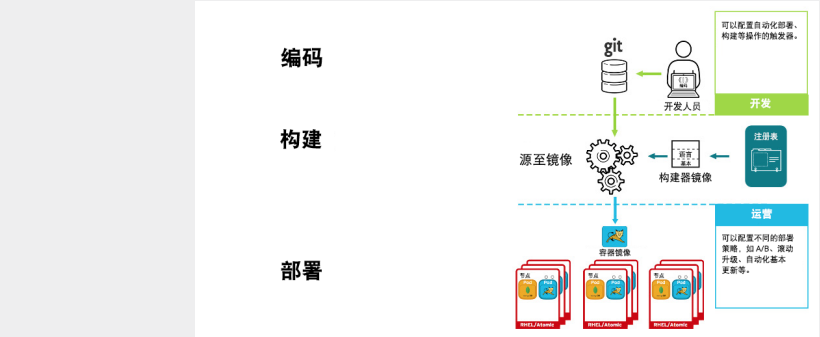
最后，我们建议您对自己的定制容器进行签名，以确保它们不会在构建和部署过程中被篡改。

5. 部署：控制可在集群中部署的内容

对于在构建过程中有操作以失败告终，或是对于在镜像完成部署后才发现漏洞的情况，您会想要实现另一层级的安全性：能够基于策略自动完成部署的工具。

让我们一起来了解一下使用核心层、中间件层和应用层这三个容器镜像层构建而成的应用。发现核心镜像存有问题，然后对该镜像进行重新构建。构建完成后，该镜像会被推送到 OpenShift 注册表。OpenShift 便会检测到该镜像已被更改。对于依赖于该镜像的构件和已定义触发器的构件，OpenShift 将自动重新构建这个应用镜像，并引入已修复的库。

构建完成后，该镜像会被推送到 OpenShift 的内部注册表。OpenShift 会即刻检测出其内部注册表中的镜像发生了变化，并自动为已定义触发器的应用部署更新后的镜像，以确保生产环境中运行的代码与最新的更新后镜像始终保持一致。所有这些功能通过协同合作，确保了 CI/CD 流程和管道的安全性。



高安全性包括可用于从安全角度来管理容器部署的自动化策略。OpenShift 内置安全上下文约束 (SCC)，可用来定义一组条件，容器集（一个容器集合）必须满足这些条件，才能被系统接受。OpenShift 安全上下文约束 (SCC) 有助于实现 Kubernetes 中的容器集安全策略，并允许管理员控制：

- 特权容器的运行。
- 可以请求添加容器的功能。
- 将主机目录用作宗卷。
- 容器的 SELinux 上下文。
- seccomp 配置文件的创建。
- 容器用户 ID。

拥有所需权限的用户可以选择将默认 SCC 策略调整得更为宽容。和 CI 工具一样，如果用户喜欢，他们可以将自己的 CD 工具与容器平台相整合。

通过将 CI/CD 与 OpenShift 相整合，系统完全可以自动重构应用以引入最新的修复和测试，并确保该应用能在环境中的所有位置加以部署。

6. 容器编排：确保容器平台的安全

当然，应用很少会以单个容器的形式来交付。即使是非常简单的应用，通常也会由前端、后端和数据库构成。在容器中部署基于微服务的现代化应用意味着部署多个容器 — 有时在同一个主机上，有时分布在多个主机或节点上（如下图所示）。

在管理大规模的容器部署时，您需要考虑：

- 应将哪个容器部署到哪个主机。
- 要为哪个主机分配更多容量。
- 哪些容器需要相互访问。它们要如何发现彼此？
- 如何控制对于共享资源（如网络和存储）的访问和管理。
- 如何监控容器的健康状况。
- 如何自动扩展应用容量以满足需求。
- 如何在满足安全要求的情况下启用开发人员自助服务。

您可以构建自己的容器管理环境。那么，既然可以使用内置管理和安全功能来部署容器平台，而且这样还能让您的团队将精力集中于构建能够创造业务价值的应用上，为什么还要花时间来构建工具？

红帽 OpenShift 容器平台可通过纳入和扩展开源 Kubernetes 项目来实现容器编排、自动调度，并在由物理机或虚拟机构成的集群上运行应用容器。由 Google 发起的开源项目 Kubernetes 可以使用“主结点”来管理复杂的容器集群编排。OpenShift 也随附有很多功能，其中的红帽 CloudForms 可用于监控私有注册表中容器的健康状况，并防止部署带有最新检测到的漏洞的容器。

由于开发人员和操作员都有大量的功能可以使用，所以基于角色的严格访问权限控制是容器平台的关键要素之一。例如，编排主结点是访问的中央点，应该接受最高等级的安全审查。API 是实现大规模自动化容器管理的关键所在。API 用于为容器集、服务和复制控制器验证和配置数据，对传入的请求执行项目验证，并调用其他主要系统组件上的触发器。

API 访问控制（身份验证和授权）是确保容器平台安全的关键所在。OpenShift **主结点**包含一个内置 **OAuth 服务器**。开发人员和管理员可以获取 **OAuth 访问令牌**，以完成他们的 API 验证。作为管理员，您可以将 OAuth 配置为使用您所选的**身份提供商**来进行验证，例如，轻量级目录访问协议 (LDAP) 目录。

容器平台的另一大重要价值在于，它能为开发人员提供自助服务，从而使得开发团队能够更加轻松快速地交付基于已批准层级构建的应用。为确保各个团队不会在未经授权的情况下访问彼此的环境，平台必须实现多租户安全性。您需要一个能够充分掌控团队情况的自助服务门户，以在确保安全性的情况下促进合作。OpenShift 为 Kubernetes 增添了几个用于维护多租户主结点安全的组件，以确保：

- 针对主结点的所有访问都是通过传输层安全 (TLS) 协议来进行的。
- 针对 API 服务器的访问基于 X.509 证书或基于令牌
- 项目配额用于限制恶意令牌可能会导致的损害程度。
- Etcd 没有直接面向集群开放。

OpenShift 3.5 **增强了集群的管理**，包括改进机密和证书管理。

7. 网络隔离

在容器中部署基于微服务的现代化应用通常意味着要在多个节点上分布式部署多个容器。出于网络防御方面的考量，您需要找到一种方式，以将集群中的应用相互分隔开。

典型的公共云容器服务（如 Google Container Engine (GKE)、Azure 容器服务或 Amazon Web Services (AWS) 容器服务）都是单租户服务。它们能让您在自己发起的 VM 集群上运行自己的容器。为了实现安全容器多租户，您希望通过容器平台获得单个集群并对流量进行划分，以将该集群中的不同用户、团队、应用和环境分隔开来。

借助网络命名空间，各个容器集合（称为“容器集”）都能获得自己所要绑定的 IP 和端口范围，因此能使节点上的容器集网络相互分隔开。在默认情况下，来自不同命名空间（项目）的容器集不能向另一项目的容器集和服务发送数据包或从中接收数据包，但下文注明的例外选项除外。您可以使用这些功能将集群中的开发人员、测试和生产环境分隔开。

但是，IP 地址和端口的这种激增会让网络变得更加复杂。另外，容器都是暂时性的。所以，我们建议您投资能够帮助您应对这种复杂性的工具。如果某个容器平台能使用软件定义的网络 (SDN) 来构建能在整个集群中的各个容器间进行通信的统一集群网络，请首选这种平台。

另外，还可以选择能够利用**路由器**或**防火墙**的方法来控制出口流量的容器平台，以便利用 IP 白名单来进行控制（例如，控制数据库访问）。

除了网络命名空间之外，**SDN** 可通过将主结点（编排）命名空间与 ovs 多租户插件分隔开，来进一步提高安全性。默认情况下，在启用 **ovs 多租户插件**后，一个命名空间中的容器集流量会与其他命名空间（项目）的容器集流量隔离开。为了向 ovs 多租户插件提供例外支持，红帽 OpenShift 容器平台 3.1 引入了“**oadm 容器集网络**”功能，可允许两个项目访问彼此的服务，或允许所有项目访问集群中的所有容器集和服务。这仅限于在整个项目级别运行时，而且所允许的流量始终是双向的。也就是说，如果您能访问项目中的某个服务，您就能访问该项目中的所有服务，您还会被授予对所有项目服务的必要访问权限。由于权限是双向的，所以这只能由集群管理员来配置。

如红帽 OpenShift 容器平台 3.5 的**技术预览**中所述，新的网络策略插件 (ovs-networkpolicy) 旨在改进 ovs 多租户插件用于配置容器集间所允许流量的方式。网络策略允许在单个容器集的级别配置隔离策略。由于网络策略不是双向的，而且仅适用于在项目管理员控制下的容器集入口流量，所以也不需要集群管理员特权。

如果您想部署网络扫描程序，您可以轻松地对它们实现容器化，并将其作为“**超级特权容器**”来运行。

8. 存储

容器对于无状态和有状态应用都大有帮助。保护附加存储是确保有状态服务安全无虞的关键所在。红帽 OpenShift 容器平台可以提供适用于多种**存储**的插件，包括**网络文件系统 (NFS)**、**AWS 弹性块存储 (EBS)**、**GCE 持久磁盘**、**GlusterFS**、**iSCSI**、**RADOS (Ceph)** 和 Cinder。

持久卷 (PV) 能以资源提供商支持的任意方式挂载到主机上。提供商将获得不同的功能，而且每个 PV 的访问模式都会被设置为该特定卷所支持的各种具体模式。例如，NFS 可以支持多个读/写客户端，但是特定的 NFS PV 可能会在服务器上导出为只读模式。每个 PV 都会获得自己的访问模式集，其中描述了相应 PV 的功能。例如，ReadWriteOnce、ReadOnlyMany 和 ReadWriteMany。

对于共享存储（NFS、Ceph、 Gluster 等），有一个实用的技巧，那就是：将共享存储 PV 的组 ID (gid) 作为注释在 PV 资源上进行注册。当容器集声明这个 PV 时，所注释的 gid 将会添加到该容器集的**附加组**中，并为该容器集提供访问共享存储中所含内容的权限。

对于块存储（EBS、GCE 持久磁盘、iSCSI 等），容器平台可以使用 SELinux 功能来保护非特权容器集的已挂载宗卷的根，以使已挂载宗卷归其关联容器所有，并且只有关联容器才能看到。

所有相互通信的容器平台组件都应通过 https 加密传输中的数据。

当然，您还应充分利用所选存储解决方案中提供的各种安全功能。

9. API 管理/端点安全和单点登录 (SSO)

要确保应用安全，需要对应用加以管理并实施 API 身份验证和授权。

Web SSO 功能是现代化应用的一个关键组成部分。容器平台可以随附大量容器化服务，以供开发人员在构建应用时使用，如红帽 SSO (RH-SSO)、受到全面支持的开箱即用型 SAML 2.0 或基于 OpenID Connect 的身份验证、Web 单点登录以及基于上游 Keycloak 项目的联合服务。RH-SSO 7.1 配备适用于红帽 JBoss Fuse 和红帽 JBoss 企业应用平台 (JBoss EAP) 的客户端适配器。RH-SSO 7.1 包含一个全新的 Node.js 客户端适配器，可用于针对 Node.js 应用进行身份验证和 Web 单点登录。RH-SSO 可以与基于 LDAP 的目录服务整合，包括 Microsoft Active Directory 和红帽企业 Linux 身份管理。RH-SSO 还能与社交登录提供商整合，如 Facebook、Google 和 Twitter。

对于由微服务构成的应用而言，API 是关键所在。这些应用拥有多个独立的 API 服务，会导致服务端点激增，所以需要使用额外的工具来加以监管。另外，我们还建议您使用 API 管理工具。红帽 3scale 提供了多种多样的标准 API 身份验证和安全选项，这些选项可以单独或结合使用，以发放凭证和控制访问权限。这些选项包括标准 API 密钥、应用 ID 和密钥对以及 OAuth 2.0。

3Scale 的访问控制功能要优于基本的安全和身份验证功能。您可以利用应用和帐户计划来限制对于特定端点、方法和服务的访问，并为用户组应用访问策略。通过应用计划，您可以为 API 的使用设置速率限值，并控制开发人员小组的流量。您可以按时段为传入的 API 调用设置限值，以便保护您的基础架构并使流量保持平稳。还可以让已达到或超出速率限值的应用自动触发超额警报，并针对超出限值的应用定义相应的行为。

10. 集群联合中的角色和访问管理

2016 年 7 月，Kubernetes 1.3 首次引入 Kubernetes 联合集群。这是 Kubernetes 上游推出的激动人心的新功能之一，目前拥有该功能的最新版本为 Kubernetes 1.6 测试版。在部署和访问跨多个集群且在公共云或企业数据中心内运行的应用服务时，联合非常有用。多个集群可用于实现跨多个可用性区域的应用高可用性，或用于对跨多个云提供商（如 AWS、Google 云和 Azure）的部署或迁移进行常规管理。

现在，在管理联合集群时，您需确保编排工具能够跨不同的部署平台实例提供您所需的安全性。一如既往，身份验证和授权仍是关键所在；另外，能够安全地将数据传输至应用（无论它们在何处运行）并实现跨集群应用多租户管理的能力也非常重要。Kubernetes 正在扩展集群联合，以求涵盖对联合机密、联合命名空间以及入口对象的支持。

联合机密会横跨联合中的所有集群来自动创建和管理机密，确保这些机密保持全局一致并处于最新状态；即使在应用原始更新时一些集群处理离线状态，也要能做到这一点。

联合命名空间与传统的 **Kubernetes 命名空间**类似，并具有相同的功能。在联合控制面板上创建命名空间可确保它们在联合中的所有集群间保持同步。

红帽正与社区开展合作，并会在这些功能趋于成熟后将它们融入 OpenShift 之中。

结论

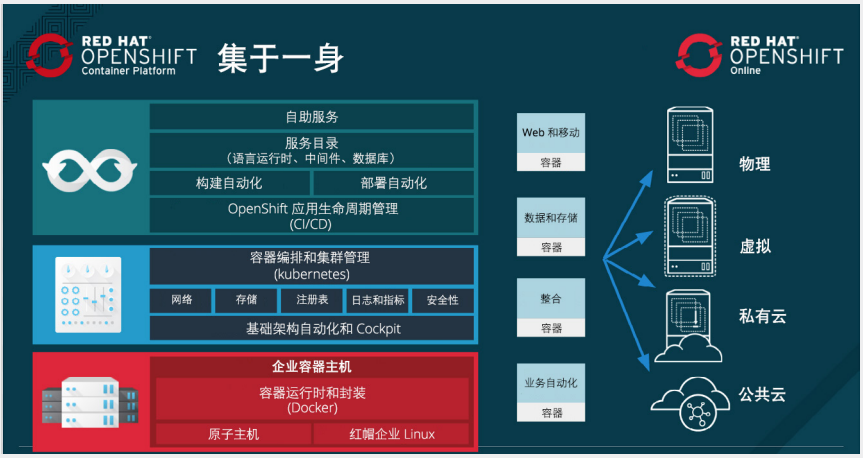
当然，这不仅仅关乎安全。您的容器平台需要提供一种既适用于开发人员、又适用于运营团队的体验。您需要一个基于容器的企业级安全应用平台，这个平台不但要能同时为开发人员和运营人员所用，而且不能影响两个团队各自所需的功能，另外还要能够提升运营效率和基础架构利用率。

OpenShift 3 是以可移植的标准 Linux 容器为核心构建而成的。它提供了多种内置安全功能，其中包括：

- 基于角色的严格访问权限控制，并能与企业身份验证系统相整合。
- 借助 Google Kubernetes，实现涵盖全网的有效容器编排和管理。
- 已与红帽企业 Linux 7 和红帽企业 Linux Atomic Host 整合；已经过优化，可大规模运行容器；已启用 SELinux，可实现严格隔离。
- 与公共和私有注册表相整合。
- 集成式 CI/CD 工具，可确保 DevOps 实践的安全性。
- 全新的容器联网模式。
- 支持远程存储卷。

而且，OpenShift 支持多种编程语言、框架和服务。另外，为联合集群提供支持也已提上我们的日程。

OpenShift 可在 OpenStack、VMware、AWS、GCP、Azure 以及装有红帽企业 Linux 7 的任意平台上运行。红帽还能以公共云服务的形式来提供 OpenShift Dedicated 和 OpenShift 在线版。



过去的 15 余年里，红帽在向企业客户提供安全可信且完全开源的解决方案方面，一直占据着领导地位。现在，我们正借助红帽企业 Linux、红帽 OpenShift 容器平台等解决方案以及基于容器的全套红帽产品组合，来提供同样安全可信的容器。

关于红帽

红帽是世界领先的开源解决方案供应商，依托社区力量为客户提供稳定可靠及高性能的云技术、Linux、中间件、存储和虚拟化产品。红帽还提供屡获殊荣的支持、培训和咨询服务。作为紧密连接全球企业、合作伙伴和开源社区的中心，红帽致力于通过为广大客户提供实用、创新型技术产品，有效释放其宝贵资源以推动业务增长，并为未来 IT 发展奠定坚实基础。

*查看更多**红帽产品组合**信息，请访问 redhat.com/zh*

销售及技术支持

800 810 2100
400 890 2100

红帽软件（北京）有限公司

北京市朝阳区东大桥路 9 号
侨福芳草地大厦 A 座 8 层
邮编：100020
86 10 6533 9300

红帽官方微博　红帽官方微信

cn.redhat.com
#f7530_0517