AsiaInfo | 亚信安全

关于Linux容器的安全建议

什么是Linux容器?

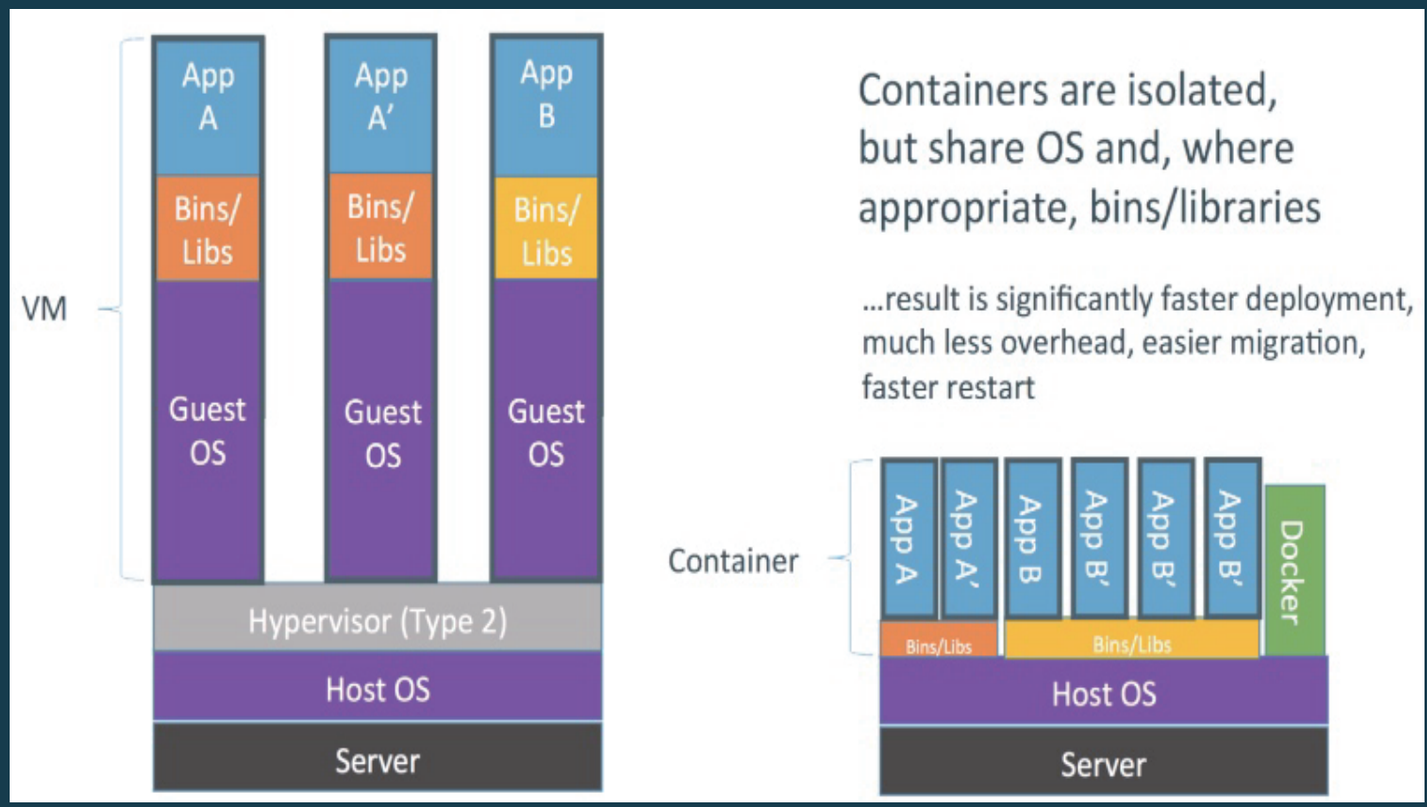
Linux 容器技术能够让您对应用及其整个运行时环境（包括全部所需文件）一起进行打包或隔离。从而让您可以在不同环境（如开发、测试和生产等环境）之间轻松迁移应用，同时还可保留应用的全部功能。

什么是Linux容器安全?

容器安全保护容器的完整性，保护容器管道和应用，保护容器部署环境和基础架构，整合企业安全工具，遵循或增强现有的安全策略。

AsiaInfo | 亚信安全

如何加强Linux容器的安全?



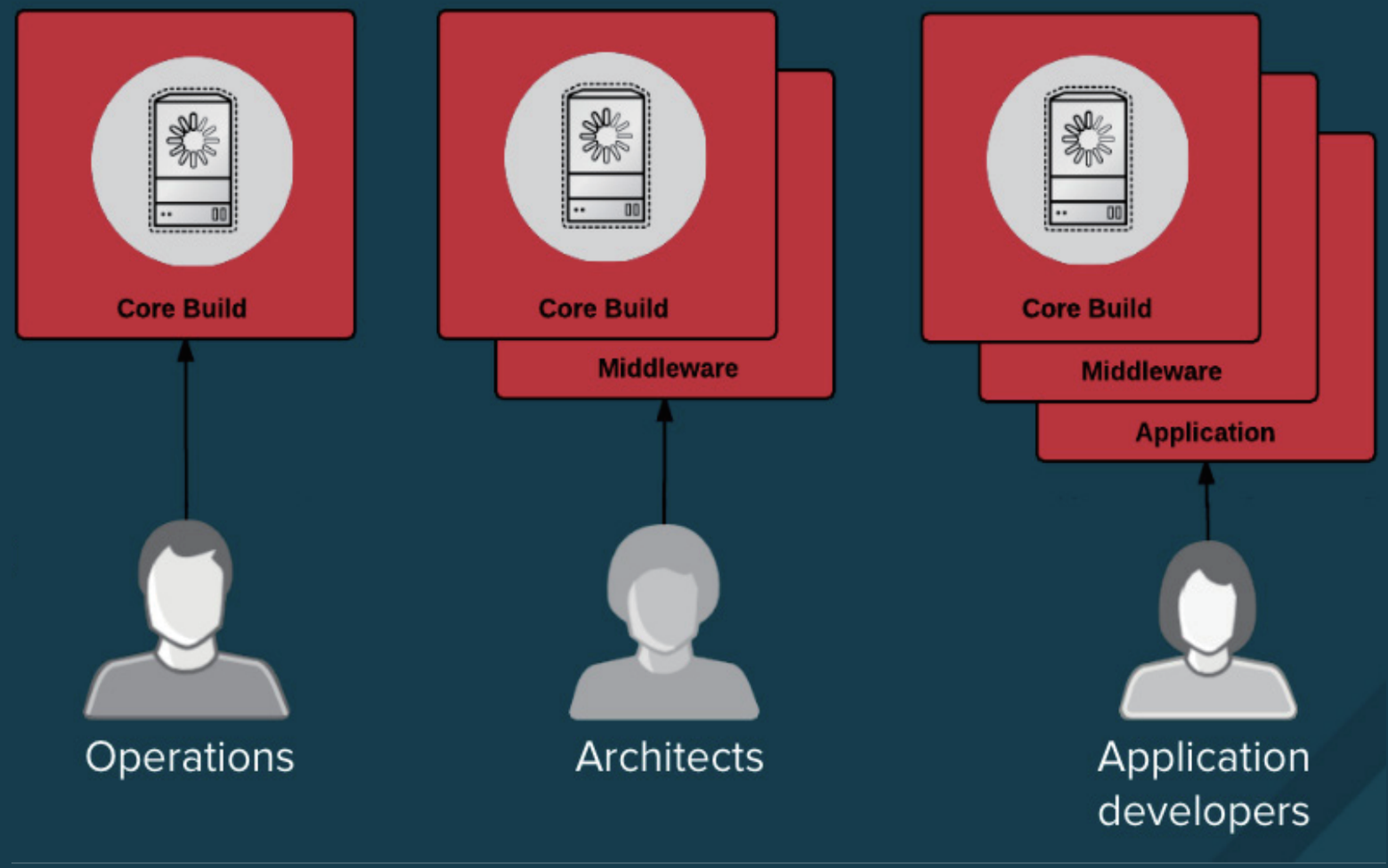
1. 容器操作系统与多租户

容器是隔离和约束资源的 Linux进程，使您能够在共享宿主内核中运行沙盒应用程序。故保护容器的方法应该与确保 Linux上任何正在运行的进程的安全方法类似。目前最佳实践依旧是放弃特权，进一步的方案则是创建尽可能少的特权容器。

容器应该作为普通用户运行，而不是 root 用户，并利用 Linux 中可用的多种级别的安全特性确保容器的安全：Linux 命名空间，安全增强的Linux（SELinux），cgroups，capabilities 和安全计算模式（seccomp）。

2. 容器内容安全（使用可信源）

当说到安全性的时候，对于容器内容来说意味着什么呢?通常应用程序和基础设施都是由现有组件组建而成，且部分来自于开源软件，例如 Linux 操作系统，Apache Web 服务器，PostgreSQL 和 Node.js。同样基于容器的各种软件包版本现也一应俱全，所以您不需要单独制作。但是，与从外部源下载的任何代码一样，您需要知道包的起源、它们是由谁创建，以及它们内部是否存在恶意代码。

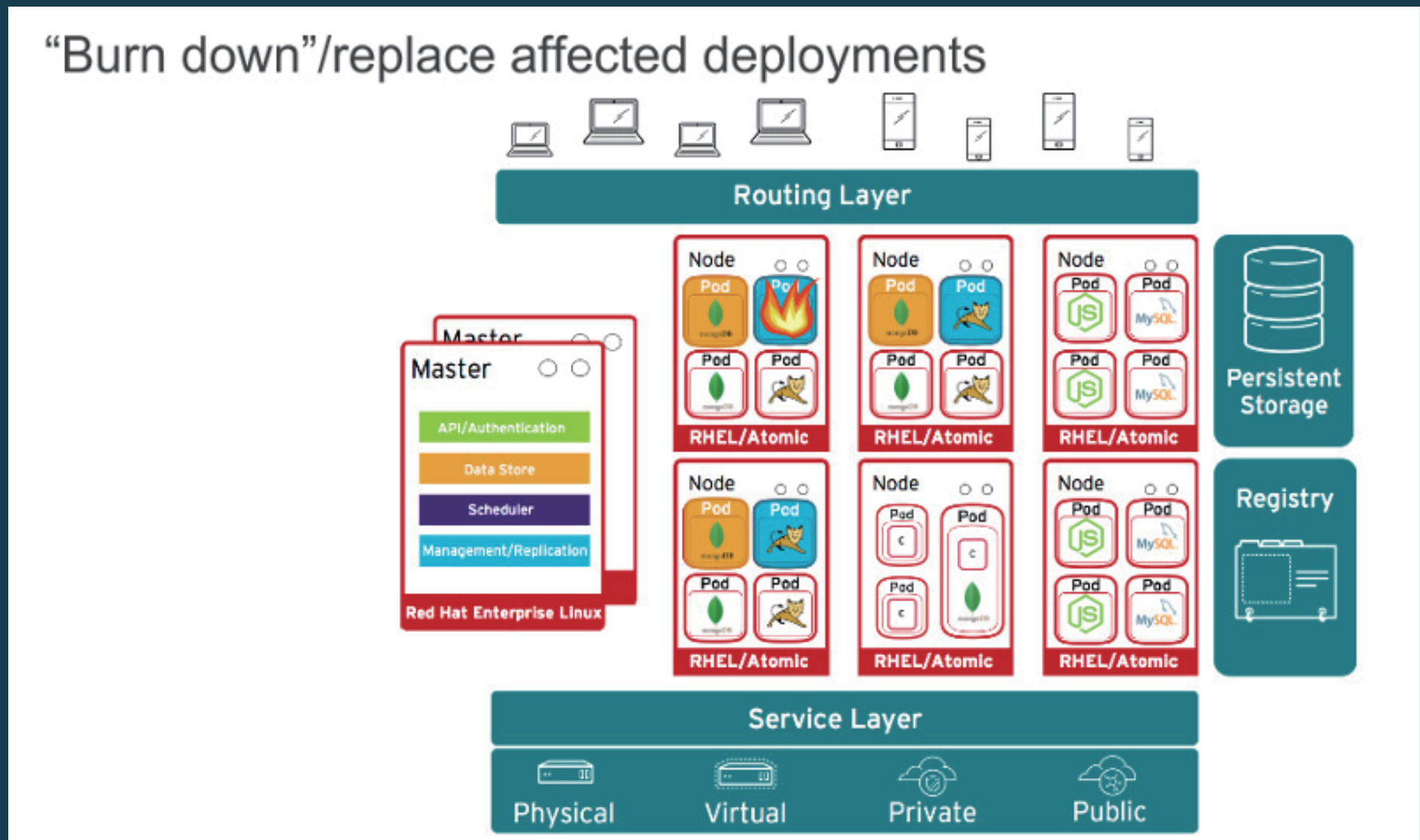


3. 容器注册 (容器镜像安全访问)

如您团队在构建容器时是基于下载的公共容器镜像，那么对它的访问控制和更新下载是管理的关键，对于管理容器镜像、内建镜像及其他类型的二进制文件也应如此。许多私有仓库注册服务器支持存储容器镜像，建议选择一个私有的、存储使用的容器镜像自动化策略的注册服务器。

4. 容器构建过程的安全

在一个容器化的环境里，软件的构建是整个生命周期的一个阶段，应用程序代码需要与运行库集成。管理此构建过程是确保软件栈安全的关键。坚持“一次构建，到处部署（build once, deploy everywhere）”的理念，确保构建过程的产品正是生产中部署的产品。这一点对于维护容器持续稳定也非常重要，换句话说，不要为运行的容器打补丁；而是应该重新构建、重新部署它们。



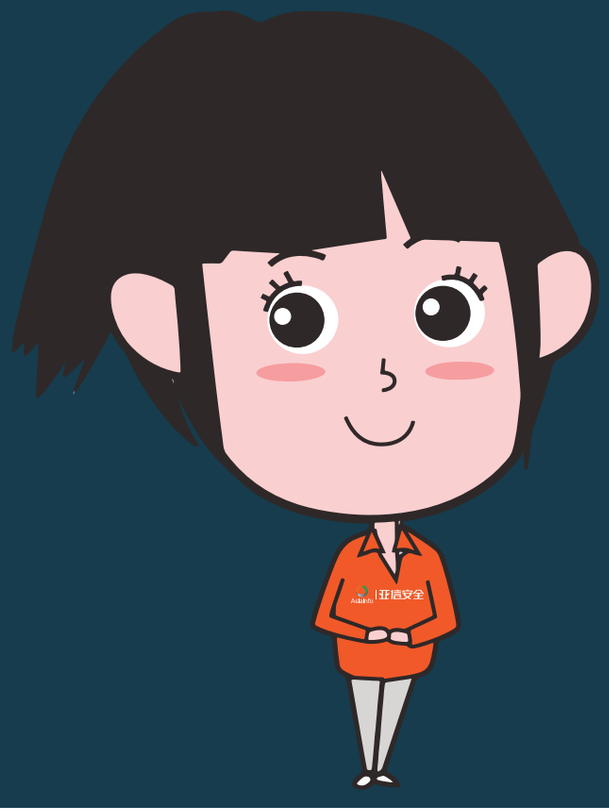
5. 控制集群中可部署的内容

为了防备在构建过程中发生任何问题，或者在部署一个镜像后发现漏洞，需要增加以自动化的、基于策略的部署的另一层安全性。

6. 容器编排：加强容器平台安全

通常情况下应用程序不会在单个容器中交付，即使是简单的应用程序通常有一个前端，后端和数据库。在容器中部署现代微服务应用，通常意味着多容器部署，有时在同一主机上有时分布在多个主机或节点。以下为规模部署需要注意的方面：a. 哪些容器应该部署到哪个主机上；b.哪个主机容量更大；c.哪些容器需要相互访问；d.如何控制对共享资源的访问和管理；e.如何自动扩展应用能力以满足需求？

AsiaInfo | 亚信安全



7. 网络隔离

在容器部署现代微服务应用程序往往意味着在多个节点分布式部署多个容器。考虑到网络防御，则需要一种在集群中隔离应用程序的方法，如通过网络命名空间，每个容器集合（称为“POD”）获得自己的IP和端口绑定范围，从而在节点上隔离 POD 网络，来自不同命名空间（项目）的 POD 不能将包发送到或接收来自不同项目的 POD 、服务的数据包。

8. 存储

对于有状态和无状态的应用程序来说，容器是非常有用的。保护存储是保证有状态服务的关键要素。容器平台应提供多样化的存储插件，包括网络文件系统（NFS），AWS Elastic Block Stores，GlusterFS，iSCSI，RADOS（CEPH）、Cinder 等等。

9. API管理，终端安全和单点登录 (SSO)

保护应用程序安全包括管理应用程序和 API 身份验证和授权。Web SSO 功能是现代应用程序的关键部分。当开发者构建他们自己的应用时，容器平台可以提供各种容器服务给他们使用。API 是微服务应用的关键组成部分，所有 API 平台都应该提供各种 API 认证和安全的标准选项，它们可以单独使用或组合使用，发布证书和控制访问。这些选项包括标准的 API 密钥、应用ID、密钥对和 OAuth 2.0。