RED HAT® OPENSHIFT

redhat.

白皮书

高管指南：选择云原生开发平台

简介

在云原生应用的规划之初，任何企业都必须考虑整个时间跨度：从选择开发平台，到应用达到真正意义上的生产级，以及可在云中进行交付，都要同时兼顾。这可能是一个漫长的过程。在这个过程中，您需要做出很多决策。这些决策可能有助于取得进展，也可能会成为绊脚石。

例如，在转向云原生开发初期， 如果开发人员还未了解应用的部署位置就开始选择工具和框架，则很容易出现效率低下的情况。企业开发人员会希望可以选择运行时、框架和语言，而企业则需要采用涵盖应用的整个生命周期的标准，以便削减运营成本，降低风险，并满足合规性要求。无论是单一的云基础架构提供商还是最新的架构模式，企业还要尽力避免供应商捆绑。

此外，鉴于云开发学习曲线十分陡峭，因此，不能留待开发周期的后期，才考虑构建功能强大且可扩展和复原的应用。动态扩展的容量富有弹性正是迁移至云的主要驱动因素，因此，事后才想起可复原性可能为时已晚。

为了制定有效的云原生应用战略，就要做到全盘考虑，从开发工具和云平台，到部署自动化和运营，无所不包。全面的云原生开发方案的好处在于，它有利于获取经验指导，可避免不必要的弯路。

了解云原生开发

云原生开发就是快速、灵活且敏捷地应对变化。这种应变是通过更频繁的部署来实现的，而这些部署可显著减少应变的前置时间。

许多关于云原生应用的讨论都会回到“十二要素应用”这一主张上，这是一组从构建和操作软件即服务（SaaS）应用的过程中形成的原则。其目标是：

- 缩短开发人员的学习曲线。
- 构建非常适合在云平台上部署的应用。
- 通过持续部署，最大限度提升敏捷性。
- 使应用无需进行重大更改即可向上扩展。

通过与客户合作，红帽发现了以下有关开发云原生应用的关键要素：

- **基于服务的架构。**其模式可能是微服务，也可能是任何松散耦合的模块化架构模型。服务代表的是一项相当独立且易于理解的业务活动。其目标是拥有可以轻松更新或替换的服务。与全面的应用相比，明确定义的服务测试起来更容易，也更彻底。
- **容器。**Linux® 容器使用 Docker 镜像，是常用的打包模型，也是实现可移植性以及隔离的自包含执行环境。容器支持高级自动化功能，这使得云平台极具吸引力。

- **DevOps 自动化。**一组旨在整合开发和运营的协作式流程和实践。其目标是加快部署频率，提供质量更高的版本，从而缩短上市时间，降低风险，并提升用户满意度。持续集成/持续交付（CI/CD）与 DevOps 所带来的改进息息相关。另一个重要目标，是通过检测和监控来了解性能，并确保获得优质的最终用户体验。
- **基于 API 的通信。**进程间的通信只能在网络上通过应用编程接口（API）使用基于协定的干净接口进行。这样可以避免发生会限制更改的意外耦合，即常见的故障来源。当应用不再驻留于同一数据中心的混合环境中，这种通信方式的重要性就愈加突出。

了解微服务

微服务架构是一种吸引人的架构模式，可用于解决诸多软件开发难题。应用可分为一组实施特定业务功能的松散连接的服务。其宗旨就是加快创新步伐，它的理论基础在于，相比单体式应用，了解、改进、测试和部署单个微服务要简单得多。由于每个微服务的用途和功能都会得到明确界定，因此，实施自动化测试和持续交付也就更加实际。

云原生应用的要求

云原生架构对应用提出了许多技术要求。其中包括：

- **部署需要实现自动化。**为了加快发布频率，需要引入自动化功能，这样一来，只需单击按钮，即可执行部署过程，以供审核。需要自动化的构建和部署管道，并且必要时能够快速回滚到上一版本。
- **配置信息必须与应用分离开来。**应用在特定环境中运行所需的配置详细信息应存储在部署环境中。这便可在包括开发、测试和生产在内的所有环境中使用相同的应用镜像。此外，还需要安全凭据存储，因为应用将需要通过网络与数据源和其他组件进行安全通信。
- **应用必须进行动态查找。**这对实现跨环境可移植性、高可用性和动态扩展（如果使用负载平衡）来说不可或缺。
- **永久性数据需要单独的数据存储。**进程和容器必须是无状态的。进程可能会发生崩溃并重启，有可能会发生在其他计算机上，时间只有短短的几秒钟。数据、会话信息和日志都必须保留在外部数据存储中。

开始进行云原生开发时，您需要考虑以下问题：

- 对于构建和部署要在云平台上运行的容器化应用，您的企业有多少经验？开发人员是否了解将永久性数据和配置与应用分开管理的必要条件？
- 构建进行云原生开发的开发环境将需要多长时间？开发环境是否与生产云环境相符？
- 构建 CI/CD 并将其有效地集成到开发和部署流程中需要多长时间？

选择云原生开发平台时，必须考虑好上述的所有事项。

微服务注意事项

微服务与云原生应用密切相关。事实上，如果没有云平台、容器和 DevOps 自动化功能，很难成功实施微服务。敏捷性是推动实施微服务的主要因素。它需要使用较小的应用组件来实现，这些组件可在降低风险的情况下进行更频繁的发布。但是，曾经的单个应用可能会变成十多个具有独立发布周期的微服务。下面是改用微服务时需要考虑的一些因素：

- 为每个新的微服务置备新的开发、测试和生产环境将需要多长时间？
- 构建和部署自动化功能是否足以支持所需次数的发布？
- 应用将如何搜索微服务的运行位置？是否可以根据需要灵活地重定位微服务，从而实现可用性或可扩展性？

- 是否可在不招致影响最终用户的停机的情况下发布微服务?
- 诊断问题的难度如何? 跨不同服务和潜在混合环境跟踪事件的难度怎样?
- 一个服务出现的问题是否能得到有效控制，或者这些问题是否会升级为多重故障?
- 鉴于微服务具有分布式特性，因此，还需要其他软件基础架构组件才能执行相应的功能，如服务搜索、客户端负载平衡、永久性状态管理、分布式跟踪和复原能力。
- 开发人员是否需要实施这些服务，或者他们是否可在网上找到所需组件?
- 如果目前部署的每个微服务都包含这些组件，那维护负担有多大? 如何更新组件，以解决错误和安全漏洞问题?
- 这些是应当内置在云运行时或平台中的商品服务吗?

对于着手构建微服务的开发人员而言，面临的首要问题之一是需要使用哪种类型的运行时、框架和语言。要回答这一问题，需要考虑如下事项:

- 开发人员是否需要了解新的框架才能开始构建?
- 在改用微服务时，企业是否可以利用现有的 Java™ EE 代码和专业技术?
- 能否使用新的开发模式，例如反应式编程? 平台能否提供所需的选项，以应对新的需求，例如由手机创建的事件驱动型工作负载以及物联网（IoT）?
- 为了使用最佳工具来完成工作，拥有使用不同的运行时、框架或语言实施的不同微服务是否合理? 这些微服务的配置、部署或防护方式是否有所不同?
- 需要多长时间才能在实际的公共云环境中测试新的微服务?

微服务非常吸引人。但是，微服务和分布式应用架构可能很难掌握。因此，务必要选择可以去除繁杂、简化开发的平台。

对现有单体式应用进行现代化改造

虽然将现有的单体式应用重新编写成微服务对开发人员来说可能很有吸引力，但是，这种方法可能不一定经济实用。不管怎样，这些应用仍需要维护，还有积压的改进有待实施。由于发布周期漫长，因此，积压情况也可能很难减少。通过将这些应用重新托管到云平台，就可以更轻松地实施 CI/CD，以及滚动发布，如蓝/绿部署或 Canary 部署。这样，开发人员便可风险更低地实现更频繁的发布。云托管的应用可以作为“快速发展的单体式应用”，以便开发人员解决积压的改进问题。

另外，在云中重新托管应用，可以有效地开启实施微服务。云具有很大的灵活性，这使在容器中一起部署新的微服务和现有的应用变得更加简单。此后，开发人员可以实施从单体式应用中迁移功能的微服务。

考虑到将现有的应用提升和转移到云所带来的种种好处，下面是选择云原生开发平台时的一些注意事项:

- 开发平台是否提供用于现有应用现代化和新绿地开发的选项?
- 云平台是否能够支持尚不完全属于云原生类型的应用?
- 有哪些选项可用于迁移现有的 Java EE 应用?
- 开发平台是否支持加速交付单体式应用?

对于云原生应用，应用服务器可作为可行的运行时环境，但是，在迁移到云原生平台时，传统应用服务器的角色发生了改变。传统上，应用服务器提供了运行时和部署环境，以及适用于系统群集上运行的中央域的管理服务。但是，很多内含的操作工具，例如管理控制台，已不再需要，而且还适得其反。云平台之所以有吸引力，原因是它们包含自动化管理功能，支持动态扩展和持续交付。如果将传统应用服务器连同其管理组件一起打包在容器中，这可能会使开发人员无法充分利用这些功能。需要考虑的方面有：

- 平台是否包含与云平台的管理功能集成的 Java EE 运行时选项？
- 是否提供有助于将传统应用服务器上运行的应用迁移到基于云的现代化 Java EE 环境的工具？
- 对于并未使用 Java EE 的所有功能的应用，例如 Web 应用，是否还有其他可能更适用于云的替代方案？

有了正确的平台和工具，企业便可从现有的应用中获得更多的价值，并且迁移到云原生和微服务模型。

支持内部环境和多云

大多数的 IT 企业都不愿意受制于单一公共云基础架构提供商。此外，还有很多企业相信，即使再过几年，至少有一半的应用仍会在现场运行。由于大多数的企业都有这两项要求，因此，需要考虑的因素有：

- 是否需要更改应用，以便使用来自不同提供商的云基础架构？那部署、运营和监控呢？在每个云平台上，这些活动是否会有所不同？
- 如何最大限度地缩小公共云基础架构与现场系统之间的差距？公共云中使用的云平台是否也可以在现场或企业首选的基础架构即服务（IaaS）上运行？

胸中有全局

在评估云原生开发平台时，还应考虑所做的选择将对其他领域产生哪些影响：

- 与云开发和部署平台结合使用的开发工具是否现成可用？如果没有，开发人员尝试配置和集成其工具将需要多长时间？
- 该平台是否为开发人员提供了一种规定性或指导性方法来提高工作效率？
- 是否提供一系列基于云且适用于消息传递、数据存储和业务流程或规则管理的中间件服务，而且这些服务可在云平台上运行？
- 是否提供可与应用集成的预构建第三方容器？
- 是否提供培训和咨询服务？可用的内部资源是否具有应用现代化方面的经验？

选择合适的云开发平台的重要性

对于正着手进行云开发的开发人员，按照教程操作并从网上选择软件组件，每次解决一个问题，是一件自然而然的事。首先，他们需要了解如何汇编足够多的运行时环境，从而将应用代码构建到容器中。然后，他们必须确定还要购买其他哪些软件组件，才能构建全面的应用。这个过程可能非常缓慢，而且存在诸多缺点：

- 学习曲线和所需的集成工作量可能会导致开发人员工作效率下降，而这一情况很难向企业解释。凡是新就任的开发人员，都需要了解已在内部创建的堆栈。
- 任何选定的软件组件都需要维护，以确保这些组件没有已知错误和安全漏洞。对于开源组件，还存在它们是否适合企业使用的问题。
- 据很多企业称，开发人员已不再为配置和部署等功能编写代码，因为这些功能并不会直接创造任何商业价值。更糟的是，其中很多功能都关系到部署平台的选择，并且是应当属于部署平台的商品服务。

RED HAT® OPENSHIFT

redhat.

也许，汇编一系列组件时遇到的最大问题是，无法缩短开发和运营的整个学习曲线。替代方案是选择旨在处理各阶段任务的云开发平台，从选择运行时到开始开发，一直到生产部署。

迁移到云原生应用，并非朝夕可成。对很多企业而言，这是一个累积经验、逐步改进的过程。考虑到云原生开发所需的一切资源，涵盖开发到部署的端到端方案显然更有可能成功。 红帽就提供了这样一个平台，其中包含红帽® OpenShift 应用运行时和红帽 OpenShift。

红帽 OPENSHIFT 应用运行时和红帽 OPENSHIFT

红帽 OpenShift 应用运行时旨在简化云原生应用的开发。一组精选的集成运行时和框架，提供了指导性方法和规范性体验，可用于快速启动开发。全面简化的开发和部署平台之所以适合，是因为 OpenShift 应用运行时基于红帽 OpenShift 并进行了相应的优化。而红帽 OpenShift 是专为混合云设计的容器应用平台。

红帽 OpenShift 为开发人员和运营团队提供了自助服务平台，用于构建和运行容器化应用。通过 OpenShift，用于新的微服务或应用的环境，可在几分钟内就置备就绪。使用 OpenShift，可以显著缩短部署周期，并大幅降低风险。只需点击几下，即可提供功能强大的自动化 CI/CD 构建和部署管道。通过将红帽 OpenShift 应用运行时及红帽 OpenShift，配合红帽 开发工具和红帽 咨询使用，企业可在降低风险的前提下，以更少的时间迁移到云原生应用。

为了让开发人员获得合适的工具来完成工作，可以选择 OpenShift 应用运行时中的运行时。对于微服务开发，开发人员可以选用 Java EE，从而充分利用现有的专业技术、较新的 Java 微配置文件标准（经过演变可满足微服务的要求），或事件驱动型框架（用于构建反应式微服务，以针对高并发性、低延迟工作负载进行扩展）。Node.js 运行时是针对经常与移动和 Web 应用结合使用的 JavaScript 后端服务所提供。为了迁移现有的应用，这套解决方案还提供了基于红帽 JBoss® 企业应用平台的运行时，以及基于现代模块化云就绪型架构的 Java EE 7 认证应用服务器。所有的运行时都已通过红帽的测试和验证。

为了便于开发人员快速入门，我们通过红帽开发人员网站提供了 Launch，这是一款基于云的免费 SaaS 工具，运行于红帽 OpenShift 在线版上。在选择了示例应用和运行时后，开发人员可以获得完整的代码库，该代码库可基于 OpenShift 在线版在云中构建和运行。作为教育工具，相同的示例应用适用于所有的运行时，这使得开发人员可以轻松地比较可用架构模式的优点。

如需了解更多，请访问 https://www.redhat.com/zh/technologies/cloud-computing/openshift/application-runtimes

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