什么是云原生？

云原生是在云计算环境中构建、部署和管理现代应用程序的软件方法。现代企业希望构建高度可扩展、灵活且具有弹性的应用程序，可以快速更新以满足客户需求。为此，他们使用现代工具和技术，这些工具和技术本质上支持云基础设施上的应用程序开发。这些云原生技术支持快速、频繁地更改应用程序，而不会影响服务交付，从而为采用者提供了创新的竞争优势。

云原生方法如何使企业受益？

组织在构建云原生软件应用程序时以各种方式获得竞争优势。

提高效率

云原生开发带来了 DevOps 和持续交付（CD）等敏捷实践。开发人员使用自动化工具、云服务和现代设计文化来快速构建可扩展的应用程序。

降低成本

通过采用云原生方法，公司不必投资于昂贵的物理基础设施的采购和维护。这样可以长期节省运营支出。构建云原生解决方案所节省的成本也可能使您的客户受益。

确保可用性

云原生技术使公司能够构建弹性强且高度可用的应用程序。功能更新不会导致停机，公司可以在旺季纵向扩展应用程序资源，以提供积极的客户体验。

什么是云原生应用程序？

云原生应用程序是由多个称为微服务的相互依赖的小型服务组成的软件程序。传统上，开发人员使用包含所有必需功能的单块结构构建整体式应用程序。通过使用云原生方法，软件开发人员将功能分解为更小的微服务。这使得云原生应用程序更加敏捷，因为这些微服务可以独立工作，并且只需极少的计算资源即可运行。

云原生应用程序与传统企业应用程序的对比

传统的企业应用程序是使用不太灵活的软件开发方法构建的。开发人员通常在发布大量软件功能以供测试之前进行工作。因此，传统的企业应用程序需要更长的部署时间，而且无法扩展。

另一方面，云原生应用程序使用协作方法，并且在不同平台上具有高度可扩展性。开发人员使用软件工具在云原生应用程序中大幅度实现构建、测试和部署过程的自动化。您可以立即设置、部署或复制微服务，这是传统应用程序无法实现的操作。

什么是 CNCF？

云原生计算基金会（CNCF）是一个开源基金会，可帮助组织开启云原生之旅。CNCF 成立于 2015 年，支持开源社区开发关键的云原生组件，包括 Kubernetes。Amazon 是 CNCF 的成员。

什么是云原生应用程序架构？

云原生架构结合了开发团队用来构建和运行可扩展的云原生应用程序的软件组件。CNCF 将不可变基础设施、微服务、声明式 API、容器和服务网格列为云原生架构的技术块。

不可变基础设施

不可变基础设施意味着用于托管云原生应用程序的服务器在部署后保持不变。如果应用程序需要更多计算资源，则会丢弃旧服务器，并将应用程序移至新的高性能服务器。通过避免手动升级，不可变基础设施使云原生部署成为一个可预测的过程。

微服务

微服务是小型的独立软件组件，它们作为完整的云原生软件共同运行。每个微服务都侧重于一个小而具体的问题。微服务是松散耦合的，这意味着它们是相互通信的独立软件组件。开发人员通过处理单个微服务来更改应用程序。这样，即使一个微服务出现故障，应用程序仍能继续运行。

API

应用程序编程接口（API）是两个或多个软件程序用来交换信息的方法。云原生系统使用 API 将松散耦合的微服务整合在一起。API 会告诉您微服务想要什么数据以及它能给您带来什么结果，而不是指定实现结果的步骤。

服务网格

服务网格是云基础设施中的一个软件层，用于管理多个微服务之间的通信。开发人员使用服务网格来引入其他功能，而无需在应用程序中编写新代码。

容器

容器是云原生应用程序中最小的计算单元。它们是将微服务代码和其他必需文件打包在云原生系统中的软件组件。通过容器化微服务，云原生应用程序独立于底层操作系统和硬件运行。这意味着软件开发人员可以在本地、云基础设施或混合云上部署云原生应用程序。 开发人员使用容器将微服务与其各自的依赖项（例如主应用程序运行所需的资源文件、库和脚本）打包。

容器的优势

容器的一些优势包括：

- 与传统的应用程序部署相比，您使用的计算资源更少
- 您几乎可以立即部署它们
- 您可以更高效地扩展应用程序所需的云计算资源

什么是云原生应用程序开发？

云原生应用程序开发描述了开发人员如何以及在何处构建和部署云原生应用程序。文化转型对于云原生开发非常重要。开发人员采用特定的软件实践来缩短软件交付时间，并提供满足不断变化的用户期望的准确功能。我们在下面给出了一些常见的云原生开发实践。

持续集成

持续集成（CI）是一种软件实践，在这种实践中，开发人员可以频繁地将更改集成到共享代码库中，而不会出错。小而频繁的更改可以提高开发效率，因为您可以更快发现问题并对其进行问题排查。CI 工具会自动评估每次更改的代码质量，以便开发团队可以更有信心地添加新功能。

持续交付

持续交付（CD）是一种支持云原生开发的软件实践。借助 CD，开发团队可确保微服务随时准备部署到云中。他们使用软件自动化工具来降低进行更改时的风险，例如引入新功能和修复应用程序中的错误。CI 和 CD 协同工作，实现高效的软件交付。

开发运维

DevOps 是一种改善开发和运营团队协作的软件文化。这是一种与云原生模式保持一致的设计理念。DevOps 实践使组织能够加快软件开发生命周期。开发人员和运营工程师使用 DevOps 工具实现云原生开发的自动化。

无服务器

无服务器计算是一种云原生模式，云提供商完全管理底层服务器基础设施。开发人员之所以使用无服务器计算，是因为云基础设施会自动扩展和配置以满足应用程序要求。开发人员只需为应用程序使用的资源付费。当应用程序停止运行时，无服务器架构会自动移除计算资源。

云原生应用程序开发有哪些优点？

更快的开发

开发人员使用云原生方法来缩短开发时间并获得更高质量的应用程序。开发人员无需依赖特定的硬件基础设施，而是使用 DevOps 实践构建随时可部署的容器化应用程序。这使开发人员能够快速响应更改。例如，他们可以在不关闭应用程序的情况下进行多次每日更新。

平台独立性

通过在云中构建和部署应用程序，开发人员可以确保操作环境的一致性和可靠性。他们不必担心硬件不兼容，因为云提供商会解决这个问题。因此，开发人员可以专注于在应用程序中交付价值，而不是设置底层基础设施。

经济高效的运营

您只需为应用程序实际使用的资源付费。例如，如果您的用户流量仅在一年中的特定时间达到峰值，您只需为该时段支付额外费用。您不必预调配一年中大部分时间处于闲置状态的额外资源。

什么是云原生堆栈？

云原生堆栈描述了开发人员用于构建、管理和运行云原生应用程序的云原生技术层。它们分为以下几类。

基础设施层

基础设施层是云原生堆栈的基础。它由操作系统、存储、网络和其他由第三方云提供商管理的计算资源组成。

预调配层

预调配层由分配和配置云环境的云服务组成。

运行时层

运行时层为容器的运行提供云原生技术。包括云数据存储、联网功能和容器运行时（如 containerd）。

编排和管理层

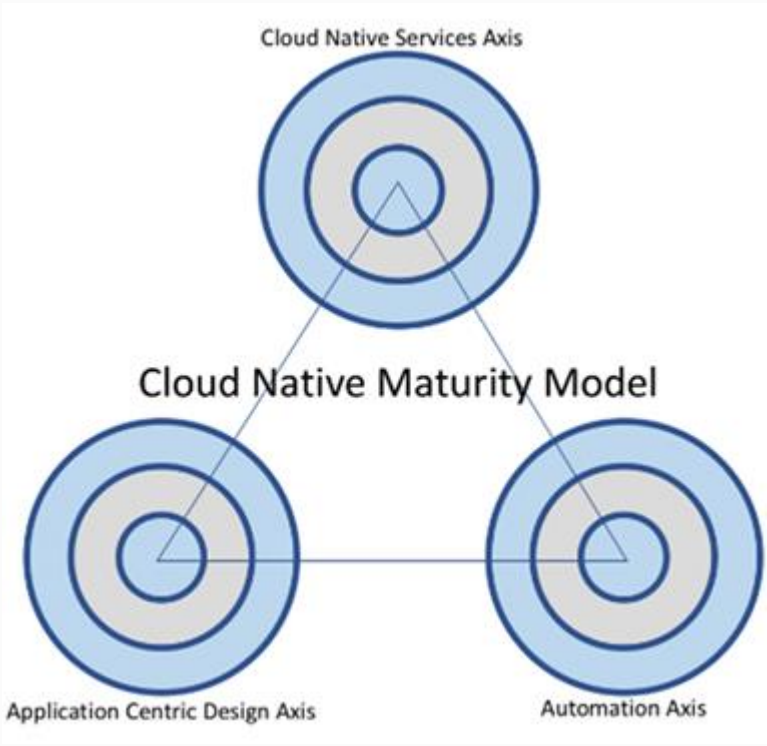
编排和管理负责整合各种云组件，以便它们作为一个单元运行。它类似于操作系统在传统计算中的工作方式。开发人员使用 Kubernetes 等编排工具在不同的机器上部署、管理和扩展云应用程序。

应用程序定义和开发层

此云原生堆栈层由用于构建云原生应用程序的软件技术组成。例如，开发人员使用数据库、消息传递、容器映像等云技术以及持续集成（CI）和持续交付（CD）工具来构建云应用程序。

可观察性和分析工具

可观察性和分析工具监控、评估和改善云应用程序的系统运行状况。开发人员使用工具来监控 CPU 使用率、内存和延迟等指标，以确保应用程序的服务质量不会受到干扰。



什么是云计算？

云计算是指托管在外部数据中心并按使用量付费提供给用户的软件基础设施。公司不必为昂贵的服务器付费并进行维护。相反，他们可以使用云提供商提供的按需云原生服务，例如存储、数据库和分析。

云计算与云原生的对比

云计算是云供应商按需提供的资源、基础设施和工具。而云原生是一种使用云计算模型构建和运行软件程序的方法。

什么是支持云？

支持云的应用程序是以前在本地数据中心运行但已修改为在云端运行的传统企业应用程序。这涉及更改软件模块的一部分以将应用程序迁移到云服务器。因此，您可以从浏览器使用该应用程序，同时保留其原始功能。

云原生与支持云的对比

云原生一词是指从一开始就设计为驻留在云中的应用程序。云原生涉及云技术，例如微服务、容器编排工具和自动扩缩。支持云的应用程序不具备云原生应用程序的灵活性、弹性或可扩展性。这是因为支持云的应用程序即使已迁移到云端，仍保留其整体结构。