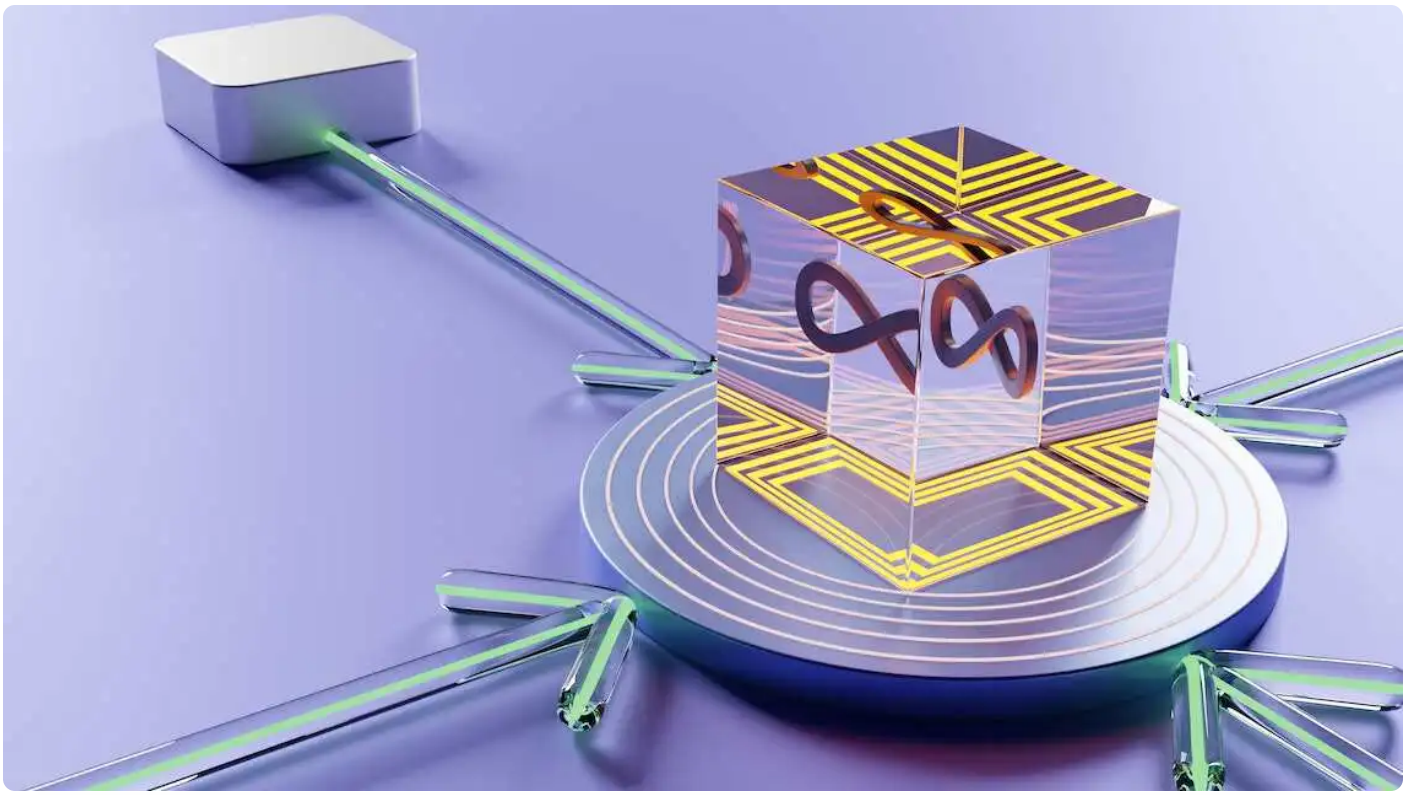
# 加餐04丨谈谈容器云与和CaaS平台

2023-01-09 LMOS 来自北京



《计算机基础实战课》

课程介绍 >



讲述：陈晨

时长 12:07　大小 11.07M

你好，我是 LMOS。

在前面几节课程中，我们学习了解了 IAAS、PAAS 以及大数据相关的实现基础，这节课我们学习另外一个云计算相关的概念，就是 CaaS。

CaaS 也是我持续关注的一个主题，刚好和你分享分享。作为加餐，我们可以轻松一点，把重点放在了解它大概是什么，又能给我们提供什么样的支持，最后，我还会分享几个优秀的开源项目。

## 容器即服务——CaaS

CaaS 其实是个简称，全称是 Containers-as-a-Service，中文是容器即服务。CaaS 是一款云服务，可以帮助使用者基于容器的抽象来管理和部署应用，以此实现关键 IT 功能的自动化。

想理解一个新概念，我们不妨和熟悉的概念来关联、比较一下。CaaS 调度的基本单元是容器。说起容器我们应该都不陌生，它是云原生资源和微服务资源的常见部署形式。

那容器都有哪些优势呢？我画了一张表格来梳理。

| 优势 | 说明 |
| --- | --- |
| 可移植性强 | 用容器开发的应用可移植性强，足够灵活，可以轻松在不同 CaaS 供应商之间移动工作负载 |
| 可扩展性好 | 能根据实际需求扩展或缩减容器数量 |
| 更高的资源利用率 | 容器不需要单独的操作系统，所需资源比虚拟机少，成本也更低 |
| 安全性高 | 由于容器之间是彼此隔离的，所以当一个容器遭到破坏的时候其他容器并不会受到影响 |
| 速度快 | 容器自主性更强，启动和停止只要几秒钟就足够，这给运维和开发人员带来了更快、更流畅的使用体验 |



如果你想更深入地了解容器，还可以看看第一季第四十四节课的内容。

IT 运维团队可以在 CaaS 上对容器云集群进行管理和编排。容器提供了一致的环境，方便研发人员快速开发和交付可以在任何地方运行的云原生应用，这样也就实现了对资源的自动化运维和管理。研发团队则可以按照自己的需求，申请或自助使用资源。

CaaS 通常被认为是 IaaS 的一种以容器为载体的子集，CaaS 介于 IaaS 和 PaaS 之间，它起到了屏蔽底层系统 IaaS，支撑并丰富上层应用平台 PaaS 的作用。

这里又一次体现了分层的思想（关于分层，我们这一季前面第四十一节课讨论过）。有了 CaaS，就可以将底层的 IaaS 封装成一个大的资源池，我们只要把自己的应用部署到这个资源池中，不再需要关心资源的申请、管理以及与业务开发无关的事情。

## 有了 Kubernetes 为什么还需要 CaaS

常见的 CaaS 平台都是基于原生的 Kubernetes，提供 Kubernetes 集群进行完整的全生命周期管理。

从根本上说，Kubernetes 和 CaaS 都与容器管理相关，不过 Kubernetes 是容器平台，而 CaaS 是订阅型服务。二者不同之处在于一个是基础设施解决方案，而另一个是管理解决方案。当我们需要大规模运行生产工作负载时，二者都至关重要。

Kubernetes 集群能够提供各种资源，支持开发者高效开发，用户选择和灵活性是它与生俱来的优势。与传统 PaaS 系统不同的是，Kubernetes 能够支持多种工作负载。容器出现故障时，其还能够自我修复或重新启动，替代及淘汰无法在必要时响应的容器。作为容器级别运作的平台，Kubernetes 会提供部分 PaaS 常见功能，但这些都不是 Kubernetes 的内建功能。

作为订阅型服务，Caas 提供了部署、扩展和平衡负载，并将日志记录、监控和警报解决方案集成为可选插件。CaaS 提供商通常会使用 Kubernetes 平台来管理容器，借助 Kubernetes 提供平衡负载、自动装载存储系统、打包功能，还能描述已部署应用的预期状态。

不过，直接使用 Kubernetes 会有很多痛点，主要是**使用复杂度、存储、网络、安全等方面的问题。**

首先是**使用复杂度**。Kubernetes 作为一个编排引擎，本身就有很高的复杂度和学习门槛。像声明式 API、CRD、Operator 等概念，对于传统应用开发者来说也属于新鲜事物。对于开发者，他们更关注的是，怎样屏蔽底层复杂度，还有如何实现对业务快速上线的支持。

而对于应用的运维管理人员来说，他们希望厂商能提供对基础设施（IaaS）和 Kubernetes 统一管理的能力，来帮助他们运维好开发者所编写的应用。这种让不同用户只需关心自己事情的能力，是降低 Kubernetes 使用门槛的关键所在。

另外，Kubernetes 的工作负载由多个对象组成，在别的技术中很简单的操作，在 Kubernetes 的语境中可能就会变得复杂。所以对于一些技术能力不足的用户来说，哪怕只是安装、部署、使用过程中遇到一些小阻碍，可能都没有能力自行排查和解决问题。

所以，这给 CaaS 创造了机会，好的 CaaS 产品需要保证操作的简单、不出错，同时提供排查异常情况的方式，比如明确的错误码。

我们再看看**存储**方面的痛点。现在容器化的应用越来越广泛，复杂的大规模容器的容器应用也越来越常见。最初容器只是用于隔离资源的简单无状态的业务单体，发展至今，越来越多的企业和应用将生产级别、复杂度高和高性能计算的有状态应用通过容器的方式管理部署。

应用迭代快、服务更新频繁是云原生应用的重要特征，也是云原生应用场景中绕不开的强需求。虽然 Kubernetes 在许多方面非常有用，例如可伸缩性、可移植性和管理能力，但受限于其架构设计思想，原生 Kubernetes 缺乏对存储状态的支持，因此持久化存储一直以来都是容器技术的一大挑战。

Pod 和容器可以自我修复和复制，而且在不断动态变化的过程中，它们的生命周期是十分短暂的。如何让持久化存储应对不断变化的容器、保证容器的可移植性，这个问题就变得很复杂。

此外，存储技术本来就门类众多，例如私有云、公有云、裸金属等，因此用户对于在不同存储上面的迁移也是需要考虑的问题。

最后，我们再来看看**网络**方面的问题。Kubernetes 将网络建立在 pod 级别，每个 pod 都可以获取一个 IP 地址，但需要确保 pod 之间的连接性以及 node 无需 NAT（网络地址转换）就可与 pod 进行连接。这种模型的优点是无论 pod 是否在同一台物理机上，所有 pod 都会通过 IP 直接访问其他 pod。

如果用户之前有一定的虚拟化经验，这种模型不会带来过多技术迁移的负担，如果应用程序之前在虚拟机中工作，那么几乎可以保证它可以在 Kubernetes 上运行的 pod 中工作。

另外，不同的应用程序对网络要求差异会很大。与存储同理，Kubernetes 不是在单个解决方案中解决所有这些需求，而是将网络从 Kubernetes 本身中抽象出来，允许供应商构建特定的解决方案来满足不同的需求，这就要提到 CNI（容器网络接口）的概念。

用户对网络要求不同，相应地，主流的 CNI 也各具特点，用户如何能选择到最适合自己业务的 CNI 也需要仔细考虑。

## CaaS 平台具有哪些功能

分析了 Kubernetes 的使用痛点，我们也简单聊聊，一个企业级的容器云平台需要具备哪些能力。

CaaS 平台首先要满足 **Kubernetes 集群的基本调度和生命周期管理**，这是最基础的能力。CaaS 平台可以自动化完成 Kubernetes 集群的部署、扩容、升级，无需人工操作。通过不同的 IaaS provider 插件，可以将 Kubernetes 集群部署在 IaaS 服务或其他云服务上。

CaaS 平台还要具备 **Kubernetes 集群高可用的调度能力**，HA deploy 通过部署多 master/etcd 节点实现高可用，当 IaaS 支持高级放置策略时，也支持将 master/etcd 节点放置于不同的节点上，进一步提升可用性。

当发现 Kubernetes 集群节点健康状态异常时，可以自动将其隔离并创建新节点加入集群，以保证集群服务能力始终符合预期。而升级 Kubernetes 集群时将使用滚动升级策略，保证集群中应用无需停止服务。还要支持多种 Kubernetes 版本，所以无需同时升级所有集群。

**容器网络与安全能力**也很关键，对于容器网络，Kubernetes 提供了 CNI 的能力，而 CaaS 平台需要支持 Calico 等主流开源 CNI。当然，也可以根据需要推出自己的 CNI，提供 Pod 网络接口管理、IPAM、Service（ClusterIP/NodePort based on Kube-proxy/iptables）、NetworkPolicy 功能。

内部网络的对外暴露一般通过 Ingress，将服务暴露到 Kubernetes 集群之外。负载均衡支持开箱即用的 MeteralLB，也支持用户自己配置已有的 Load Balance 方案。

CaaS 还要提供监控、告警、日志管理、分析、可视化在内的一系列**可观测性功能**，展示所有 Kubernetes 集群资源消耗的统计数据。

Kubernetes 集群的监控指标将被实时采集，用户可以定制可视化面板的展示和基于监控指标的告警规则，同时支持电子邮件、短信的实时通知方式。Kubernetes 集群、节点、pod、container 等资源的日志将被聚合到 **logging** 中，提供日志搜索、限流、归档等功能。

除了上述功能，CaaS 还需要具备以下功能，我同样梳理了表格。

| 功能 | 描述 |
| --- | --- |
| 持久化存储（Storage） | 支持市面上的 CSI 插件，支持 NFS 存储、文件存储、块存储等，也可以根据需要对特定的存储实现自身的 CSI 插件 |
| 多租户（Multi-tenancy） | 支持将 Kubernetes 集群整体或其中的一部分 namespace 划分给特定租户。通过与 Everoute 微分段功能集成，多个租户共同使用一个 Kubernetes 集群时也能彼此网络隔离 |
| 应用商店（Application Store） | 帮助应用完成构建、上架。应用商店中既包含验证并发布的推荐应用，也支持用户上架内部应用 |
| registry 和 image 管理（Registry） | 支持通过 harbor 与用户自己部署的私有 registry 或是公共 registry 进行集成 |
| CI/CD | 提供“监听git -> 执行测试 -> 构建镜像 -> 构建 Kubernetes manifests -> 部署至Kubernetes 集群”的完整流程，降低运维与研发之间的沟通成本，提升部署质量 |
| 企业级的使用体验 | 提供直观的 UI，用户可以轻松地安装部署、管理 Kubernetes 集群，同时为各个功能提供流畅的操作体验，例如应用商店、监控报警等 |



CaaS 这么强大，支撑它的核心技术就是—— Cluster API（简称 CAPI）。

## CaaS 平台的核心技术——Cluster API

这是 Kubernetes 社区中一个非常开放、活跃和成熟的开源项目，遵循 Apache License v2.0。

Cluster API 项目创建于 2018 年，由 Kubernetes Cluster Lifecycle Special Interest Group 负责管理。Cluster API 吸纳了其他开源的 Kubernetes 部署工具的优点，提供一套声明式的 Kubernetes 风格的 API 以及相关工具来简化 Kubernetes 集群的创建、扩容、缩容、更新配置、升级、删除等完整的 Kubernetes 集群生命周期管理操作。

Cluster API 实现了灵活可扩展的框架，支持在 vSphere、AWS、 Azure、GCP、OpenStack 等多种云平台中部署 Kubernetes 集群。开发人员可以增加新的 Cluster API Cloud Provider 以支持更多的云平台。Cluster API 还支持 Kubernetes 组件参数配置、Kubernetes 控制平面高可用、自动替换故障节点、节点自动伸缩等高级功能。

很多开源项目和商业产品都在使用 Cluster API，比如 VMware Tanzu、Red Hat OpenShift、SUSE Rancher、Kubermatic 等。

一般的云厂商都会基于 Cluster API 框架自主研发的一种 Cluster API Cloud Provider 来适配自身的物理集群。

## 常见的容器云开源项目

接下来，我分享几个 CaaS 的优质开源项目，它们都使用了 Cluster API。

### VMware Tanzu

VMware Tanzu 社区版是一个功能齐全、易于管理的 Kubernetes 平台，适合学习者和用户，特别是在小规模或生产前环境中工作的用户。Tanzu 的主要产品是商业化版本，核心的 TKG 和 TCE 等开源，开源部分主要是 Tanzu 自己在维护。

### Rancher

Rancher 是一个企业级商用 Kubernetes 管理平台。它解决了跨任何基础架构管理多个 Kubernetes 集群的运营和安全挑战，同时为 DevOps 团队提供了运行容器化工作负载的集成工具。Rahcher2.5 版本通过使用 RKE 来创建工作节点，2.6 后的版本也使用了 Cluster API 来创建节点。

### KubeSphere

KubeSphere 是国产厂商青云主导开发的一款开源容器 PaaS 方案，通过社区贡献，目前已经有了上万的 star，社区活动比较活跃 KubeSphere 的后端设计中沿用了 Kubernetes 声明式 API 的风格，所有可操作的资源都尽可能地抽象成为 CR。它还提供了管理集群和 workload 集群的能力，通过一个管理集群来管理多个工作集群。

### OpenShift

红帽 OpenShift 是一个领先的企业级 Kubernetes 平台，在其部署的任何地方都能实现云体验。无论是在云端、本地还是在边缘，红帽 OpenShift 都能让企业轻松选择构建、部署和运行应用的位置，并提供一致的体验。

凭借红帽 OpenShift 的全堆栈自动化运维以及面向开发人员的自助服务置备，团队可以紧密携手合作，更有效地推动创意从开发过渡到生产阶段。

OpenShift 和 Rancher 或者 Kubesphere 不一样，它没有管理集群和 workload 集群这种概念，它不管理其他集群，它不是在现有 Kubernetes 集群上安装套件，而是基于 Kubernetes 内核通过 Operator 设计重新构建了一套集群，它自身就是一个 PaaS 平台，是 Kubernetes 的开箱即用功能完备的企业发行版。

## 思考与总结

在这节课的内容中，我们了解了云计算场景下 IAAS、PAAS 平台之外的又一种概念——CaaS 平台，也就是容器云管理平台。在当今容器化呼声越来越高的场景下，容器云平台呼声也是越来越高，常见的容器云平台依托于 Google 的开源容器集群管理系统—— Kubernetes，扩容了 Kubernetes 的功能，让 Kubernetes 集群的管理变得更容易。

企业级容器云和 Kubernetes 管理平台的结合正在为企业提供更快捷、高效的云计算服务。企业级 CaaS 平台相比 Kubernetes 集群管理有很多优势。建议你在课后体验一下这几个 CaaS 平台，看下他们具有哪些功能，解决了企业云资产管理的哪些痛点。

另外，当前 CaaS 平台中最重要的项目就是 Cluster API，我推荐了使用它的几个优秀开源 CaaS 项目。如果想更加深入地了解容器云相关的知识，你可以阅读上面开源项目的代码以及 Cluster API 的代码。



上一篇　加餐03丨学习攻略（二）：大数据&云计算，究竟怎么学？

下一篇　加餐05丨分布式微服务与智能SaaS

## 精选留言

写留言

> 由作者筛选后的优质留言将会公开显示，欢迎踊跃留言。