

K8S 1.18版本 (<https://www.kubernetes.org.cn/tags/kubernetes1-18>) 1.17 (<https://www.kubernetes.org.cn/tags/kubernetes1-17>) 1.16 (<https://www.kubernetes.org.cn/tags/kubernetes1-16>)
CI/CD (<https://www.kubernetes.org.cn/tags/cicd>) 网络 (<https://www.kubernetes.org.cn/tags/网络>) 存储 (<https://www.kubernetes.org.cn/tags/存储>) 安全 (<https://www.kubernetes.org.cn/tags/安全>)
监控 (<https://www.kubernetes.org.cn/tags/监控>) 运维 (<https://www.kubernetes.org.cn/tags/运维>) 认证 (<https://www.kubernetes.org.cn/tags/认证>) 入门 (<https://www.kubernetes.org.cn/tags/入门>) 问答 (<https://www.kubernetes.org.cn/tags/问答>)
登录 (<https://www.kubernetes.org.cn/wp-login.php?cms=k8s>) | 加入我们 (<https://www.kubernetes.org.cn/%e8%81%94%e7%b3%bb%e6%88%91%e4%bb%ac>)
开发实践 (<https://www.kubernetes.org.cn/practice>) 行业动态 (<https://www.kubernetes.org.cn/news>) 入门教程 (<https://www.kubernetes.org.cn/course>)
安装教程 (<https://www.kubernetes.org.cn/course/install>) 文档下载 (<https://www.kubernetes.org.cn/%E6%96%87%E6%A1%A3%E4%B8%8B%E8%BD>)
QQ/微信群 (<https://www.kubernetes.org.cn/kubernetes交流群>) 视频 (<https://www.kubernetes.org.cn/video>) 活动 (<https://www.kubernetes.org.cn/meetup>)
中文文档 (<http://docs.kubernetes.org.cn>)

Kubernetes CKA实战培训 突击CKA全球认证

官方认证CKA讲师、实操环境实战、现场答疑互动

开班城市：北京 / 上海 / 深圳 / 武汉 / 成都 / 杭州 以及线上直播班

点击查看

(<https://www.kubernetes.org.cn/peixun>)

您目前处于：社区首页 (<https://www.kubernetes.org.cn>) > 微服务 (<https://www.kubernetes.org.cn/microservices>) >

确保微服务最佳运行的5个方面

确保微服务最佳运行的5个方面

(<https://www.kubernetes.org.cn/7889.html>)

2020-06-23 14:59 王延飞 (<https://www.kubernetes.org.cn/author/fly>) 分类：微服务 (<https://www.kubernetes.org.cn/microservices>)
阅读(428) 评论(0)

如今，单体架构模式正在逐渐淡出人们的视线，越来越多的组织开始使用微服务架构模式。并且，这个趋势还要持续流行下去。之所以如此，是因为微服务延续了工程师数十年来一直采用的原则—关注点分离（**Separation of Concerns**，**SoC**）原则。

当组织采用微服务架构后，要想成功运行它们达到预期效果，请牢记以下五个方面。

微服务的挑战

关注点分离（**Separation of Concerns**，**SoC**）是一种设计原则，是使用“关注点”分离来构建软件。在单体应用程序中，这体现在典型的3层体系结构（显示层，业务层和数据层）分离中。其实，这也是职责单一原则的应用。

微服务将这一概念付诸实践，他们将应用程序拆分开，以使该应用程序的单个功能可单独开发部署。这样做的好处是巨大的，但它们也是有代价的—使得运营和运维成本增加。

随着组织将现有应用程序转换为容器，虽然带来了巨大的好处，但维护该应用程序还带来了新的挑战。

挑战1：似乎很难监控整体

尽管单体应用有其自身的挑战，但单体应用的监视相当简单。在容器化的应用程序中，事情要复杂得多。无论你是逐步将单体应用程序分解为微服务，还是从头开始构建新系统，现在都需要监视更多服务。这些都将可能：

- 使用不同的技术/语言
- 部署在不同的机器/容器
- 使用K8s进行编排

使用微服务架构后，组织的软件系统变得高度分散，这也意味着有很多需要监视的地方，并且对集中监控的需求也越来越大。

Kubernetes 1.18 版本

我们很高兴宣布Kubernetes 1.18的交付，这是我们2020年的第一版！Kubernetes 1.18包含38个增强功能：其中15个功能已趋于稳定，beta版本中有11个，alpha版本中有12个。
(<https://www.kubernetes.org.cn/tags/kubernetes1-18>)



关注「K8S中文社区」微信公众号
回复“文档”
获取K8S文档下载链接
回复“加群”
加入K8S微信技术交流群

最新文章



提高微服务安全性的11个方法
2020-06-25 评论(0)
(<https://www.kubernetes.org.cn/7893>)



如何设置Ansible AWS的动态清单
2020-06-24 评论(0)
(<https://www.kubernetes.org.cn/7891>)



Amazon EKS 与 JFrog Artifactory容器化CI/CD发布的最佳实践
2020-06-23 评论(0)
(<https://www.kubernetes.org.cn/7890>)



确保微服务最佳运行的5个方面
2020-06-23 评论(0)
(<https://www.kubernetes.org.cn/7889>)

热门推荐



你的微服务还差个容错机制
2020-03-26
(<https://www.kubernetes.org.cn/7060>)

Kubernetes 1.18GA，15个稳定11个beta，引入kubectrl debug命令
2020-03-26

这意味着不再有任何一套Ops指标可以统一采集和处理所有指标。然而，IT/Ops团队又需要使用这些指标来评估应用程序的总体正常运行时间，因此，团队现在必须处理大量（甚至数千个）度量指标，事件和警报类型，并从中获取到有用的信息。

解决方案： DevOps监视，需要从平面数据模型转变为分层模型，在该模型中，团队能够深入到度量层次结构中，以查看故障来自于哪些微服务，并从那里进入到发生故障的实际容器。从数据存储和可视化的角度来看，这很可能需要对DevOps工具进行重新设计。开源的时间序列数据库和工具（例如，**Prometheus** (<https://prometheus.io/>) 和 **Grafana 7.0** (<https://grafana.com/>)）使这成为一个非常容易实现的目标。

挑战2：跨服务的日志记录

在谈论应用程序的监视时，首先要考虑的事情之一是：日志，日志，日志。服务器每天都会产生相当于“碳排放量”的IT排放量–非结构化文本的GB量规模的数据，最终导致更大的采集，存储和解析工具成本。

即使采用单体架构，你的日志也可能已经使你的工程师有些头疼。使用微服务，日志变得更加分散。一个简单的用户请求现在可能需要经过许多个服务，而且所有这些服务可能都有自己特有的日志记录框架。

要解决软件中出现的问题，你必须从用户请求过程通过的所有服务中提取所有不同的日志，以了解出了什么问题。

解决方案：这里的主要挑战是–需要了解用户单个请求，是如何在不同服务之间“流动”。为了实现这一点，需要对传统的单体应用，在顺序执行期间记录日志的方式进行大量修改。尽管已经出现了许多框架来帮助开发人员进行处理（例如，Jaeger (<https://www.jaegertracing.io/>)），但对于希望将整体重构为微服务的组织而言，转向异步，跟踪驱动的日志记录仍是一项艰巨的努力。

挑战3：部署一项服务可能会破坏另一项服务

单体应用中的一个关键因素是所有代码都在同一时间部署，这意味着应用程序最容易受到攻击的时间段是已知的且相对较短的时间段–应用部署时。

在微服务的世界中，这种因素不再成立，由于微服务与生俱来就相互交织，一个服务的更新变化会导致功能异常或性能问题，而这种问题有时只会在另一个服务中表现出来。

如果部署一项服务出现异常，继而破坏另一项服务，这可能会导致整个应用程序出现意外的不稳定，以及导致组织上的一些摩擦冲突。

解决方案：每当部署相关的微服务前，组织必须在团队之间创建共享的发布日历，并分配资源以紧密测试和监视整个应用程序的行为。

挑战4：寻找问题的根本原因

至此，你已经确定了有问题的服务，采集了要采集的所有数据，包括堆栈跟踪和日志。你可能还拥有某种APM解决方案 (<https://blog.overops.com/appdynamics-vs-new-relic-which-tool-is-right-for-you-the-complete-guide/>)，例如New Relic，AppDynamics或Dynatrace。从那里，你将获得程序运行的相关数据，但是，什么才是问题的根本原因 (<https://blog.overops.com/the-7-key-components-of-true-root-cause-analysis/>)呢？

传统上，这需要有关于每个失败请求的状态的详细信息（即确切地说，为什么失败）。这里的挑战是，这需要开发人员进行大量的预见，以提前知道他们需要什么信息才能解决问题。

解决方案：当程序异常跨越多个服务时，采用什么方法找到根本原因至关重要。团队必须考虑需要使用哪些信息粒子来诊断未来的问题，以及应该将它们记录在什么级别上来兼顾性能和安全性方面的考虑。这是一个艰巨的挑战，而且永无止境。

挑战5：三方代码的版本管理



(<https://www.kubernetes.org.cn/7055>)



英国Monzo银行，用K8s管理1600个微服务实践

2020-03-21
(<https://www.kubernetes.org.cn/7001>)



Java vs. Go 微服务 – 负载测试 (复赛)

2020-03-20
(<https://www.kubernetes.org.cn/6988>)



与时俱进 – 为什么要使用云原生数据库？

2020-03-15
(<https://www.kubernetes.org.cn/6953>)



2020年Service Mesh 三大发展方向

2019-12-13
(<https://www.kubernetes.org.cn/6255>)



(<http://shrsl.com/22gfr>)

社区标签

BoCloud 博文
(<https://www.kubernetes.org.cn/tags/bocloud%e5%8d%9a%e>)

CI/CD
(<https://www.kubernetes.org.cn/tags/cicd>)

CNCF
(<https://www.kubernetes.org.cn/tags/cncf>)

DevOps
(<https://www.kubernetes.org.cn/tags>)

Docker
(<https://www.kubernetes.org.cn/tags>)

etcd (<https://www.kubernetes.org.cn/tags/etcd>)
GO (<https://www.kubernetes.org.cn/tags/go>)

Helm
(<https://www.kubernetes.org.cn/tags/h>)
Istio
(<https://www.kubernetes.org.cn/tags/istio>)

Jenkins
(<https://www.kubernetes.org.cn/tags/jer>)

另一个问题，我们之前提到过，但我们认为值得再次强调，从典型的单体架构过渡到微服务架构，由于超过80%的应用程序代码通常是第三方代码，因此在公司的不同微服务之间共享第三方代码的方式的良好管理，就可能避免出现前所未有的“依赖地狱(dependency hell)”。

考虑一种情况，有些微服务团队使用第三方组件的XY版本，而其他团队则使用XZ。由于不同版本之间缺乏兼容性，或者版本之间的行为发生细微变化，这很可能会就增加软件问题的风险，也会增加软件错误进行故障排除的困难。

任何使用老旧，易受攻击的第三方代码的微服务，都会为黑客提供便利条件。在单体应用环境中，允许不同的团队各自管理代码仓库中的依赖关系可能是可行的。但在微服务体系结构中，这是绝对禁止的。

解决方案：采用微服务的公司必须考虑重新设计他们的代码构建流程，以利用集中化的组件存储库（比如，Artifactory (https://jfrog.com/artifactory/)）来存储第三方和共享的实用程序代码。只能允许微服务团队将自己的代码存储在各自的存储库中。

总结

与大多数技术进步一样，微服务采用了熟悉的概念并将其颠倒过来。微服务重新考虑了大型应用程序的设计，构建和维护方式，他们带来了许多好处，但也带来了新的挑战。所以，组织在采用微服务等新技术时，需要重新思考代码的构建，部署和监控方式。

译文链接：<https://thenewstack.io/5-steps-to-ensure-your-microservices-are-running-optimally/>
(https://thenewstack.io/5-steps-to-ensure-your-microservices-are-running-optimally/)



关注微信公众号，加入社区



(http://www.kubernetes.org.cn/7863.html) (http://www.kubernetes.org.cn/7863.html) (http://www.kubernetes.org.cn/7863.html) (http://www.kubernetes.org.cn/7863.html) (http://www.kubernetes.org.cn/7863.html) (http://www.kubernetes.org.cn/7863.html) (http://www.kubernetes.org.cn/7863.html) (http://www.kubernetes.org.cn/7863.html) (http://www.kubernetes.org.cn/7863.html) (http://www.kubernetes.org.cn/7863.html)

上一篇: 5种主要的软件架构模式 (https://www.kubernetes.org.cn/7863.html)

下一篇: 提高微服务性能的方法 (https://www.kubernetes.org.cn/7893.html)

**KubeOperator**
Kubernetes 集群规划、部署及运营平台

从这里开启您的 Kubernetes 之旅

100% 开源

Web UI

离线部署

一键扩容

应用商店

(https://github.com/KubeOperator/KubeOperator)

标签： 微服务 (https://www.kubernetes.org.cn/tags/%e5%be%ae%e6%9c%8d%e5%8a%a1)

相关推荐

- 提高微服务安全性的11个方法 (https://www.kubernetes.org.cn/7893.html)
- 5种主要的软件架构模式 (https://www.kubernetes.org.cn/7863.html)
- 开放下载！《2020年行业云原生应用报告指南》正式发布 (https://www.kubernetes.org.cn/7799.html)
- Spring Boot 2.3.0配置Graceful-Shutdown，Readiness和Liveness (https://www.kubernetes.org.cn/7788.html)
- Dubbo 迈出云原生重要一步 – 应用级服务发现解析 (https://www.kubernetes.org.cn/7763.html)
- OpenKruise v0.5.0 版本发布，支持无损的流式分批发布策略 (https://www.kubernetes.org.cn/7581.html)
- Apache RocketMQ 的 Service Mesh 开源之旅 (https://www.kubernetes.org.cn/7509.html)

k8s代码解读
(https://www.kubernetes.org.cn/tags/k8s%e4%bb%a3)

kubeadm
(https://www.kubernetes.org.cn/tags/ku)

kubernetes
(https://www.kubernetes.org.cn/tags/kuberenete)

Kubernetes
(https://www.kubernetes.org.c

Kubernetes 1.4 (https://www.kubernetes.org.cn/tags/kubernetes1-4)

Kubernetes 1.5
(https://www.kubernetes.org.cn/tags/kubernetes1-5)

Kubernetes 1.6
(https://www.kubernetes.org.cn/tags/kubernetes1-6)

Kubernetes 1.7
(https://www.kubernetes.org.cn/tags/kubernetes7)

Kubernetes 1.8
(https://www.kubernetes.org.cn/tags/kubernetes1-8)

Kubernetes 1.9
(https://www.kubernetes.org.cn/tags/kubernetes1-9)

Kubernetes 1.10
(https://www.kubernetes.org.cn/tags/kubernetes1-10)

Kubernetes 1.13 (https://www.kubernetes.org.cn/tags/kubernetes1-13)

OpenStack (https://www.kubernetes.org.cn/tags/openstack)

PaaS
(https://www.kubernetes.org.cn/tags/paas)

Pod (https://www.kubernetes.org.cn/tags/pod)

Prometheus
(https://www.kubernetes.org.cn/tags/prometheus)

Rancher (https://www.kubernetes.org.cn/tags/rancher)

Serverless
(https://www.kubernetes.org.cn/tags/serverless)
Service (https://www.kubernetes.org.cn/tags/service)

service mesh
(https://www.kubernetes.org.cn/tags/service-mesh)

云原生
(https://www.kubernetes.org.cn/tags)

企业案例
(https://www.kubernetes.org.cn/tags/%e4)

存储
(https://www.kubernetes.org.cn/tags/%e5%ad%)

安全
(https://www.kubernetes.org.cn/tags/%e5%ae%89%)

容器
(https://www.kubernetes.org.cn/tags/%e6%8d%e5%8a%a1)

- ## 评论 抢沙发

社区交流

85197

提交评论

昵称 (必填)

邮箱 (必填)

网址

容器云平台
(<https://www.kubernetes.org.cn/tags/%e5%ae%b9%e5%99%a8%e4%ba%91%>)

日志
(<https://www.kubernetes.org.cn/tags/%e6%97%a5%e5%bf%9>

灵雀云 Jenkins
(<https://www.kubernetes.org.cn/tags/%e7%81%b5%e9%9b%jenkins>)

网络
([https://www.kubernetes.org.cn/tags/%e7%](https://www.kubernetes.org.cn/tags/%e7%b6%9c%80)

运维
(<https://www.kubernetes.org.cn/tags/%e8%bf%90%e7%bb%b4>)

- Kubernetes v1.18 正式版已发布
(<https://www.kubernetes.org.cn/7055.html>)
- Kubernetes v1.17 正式版已发布
(<https://github.com/kubernetes/kubernetes/releases/tag/v1.17.0>)
- Kubernetes v1.16 正式版已发布
(<https://www.kubernetes.org.cn/5838.html>)
- Kubernetes v1.15 正式版已发布
(<https://www.kubernetes.org.cn/tags/kubernetes-1-15>)
- Kubernetes v1.14 正式版已发布
(<https://www.kubernetes.org.cn/5204.html>)

jessenqiang 3天前说:
老大，麻烦问一下。界面能出来但是里边是空白的，命名空间为空，怎么解决？
(<https://www.kubernetes.org.cn/7189.html#comment-1466>)

刚刚 5 天前说:
时区问题解决了
(<https://www.kubernetes.org.cn/7189.html#comment-1465>)

刚刚 5 天前说:
时区问题解决了
(<https://www.kubernetes.org.cn/7189.html#comment-1464>)

刚刚 5 天前说：
如何更新证书呢，添加节点的时候会报错 I0622
21:56:31.166767 7915 token.go:215] [discovery] Failed to request
cluster-i (<https://www.kubernetes.org.cn/7189.html#comment-1463>)

© 2020 Kubernetes中文社区 粤ICP备16060255号-2 (<http://www.miitbeian.gov.cn/>) 版权说明 (<https://www.kubernetes.org.cn/版权说明>) 联系我们 (<https://www.kubernetes.org.cn/联系我们>) 广告投放 (<https://www.kubernetes.org.cn/广告投放>) 法律声明: 本网站不隶属于谷歌或 Alphabet 公司! kubernetes、kubernetes 标识及任何相关标志均为 Google LLC 公司的商标。

