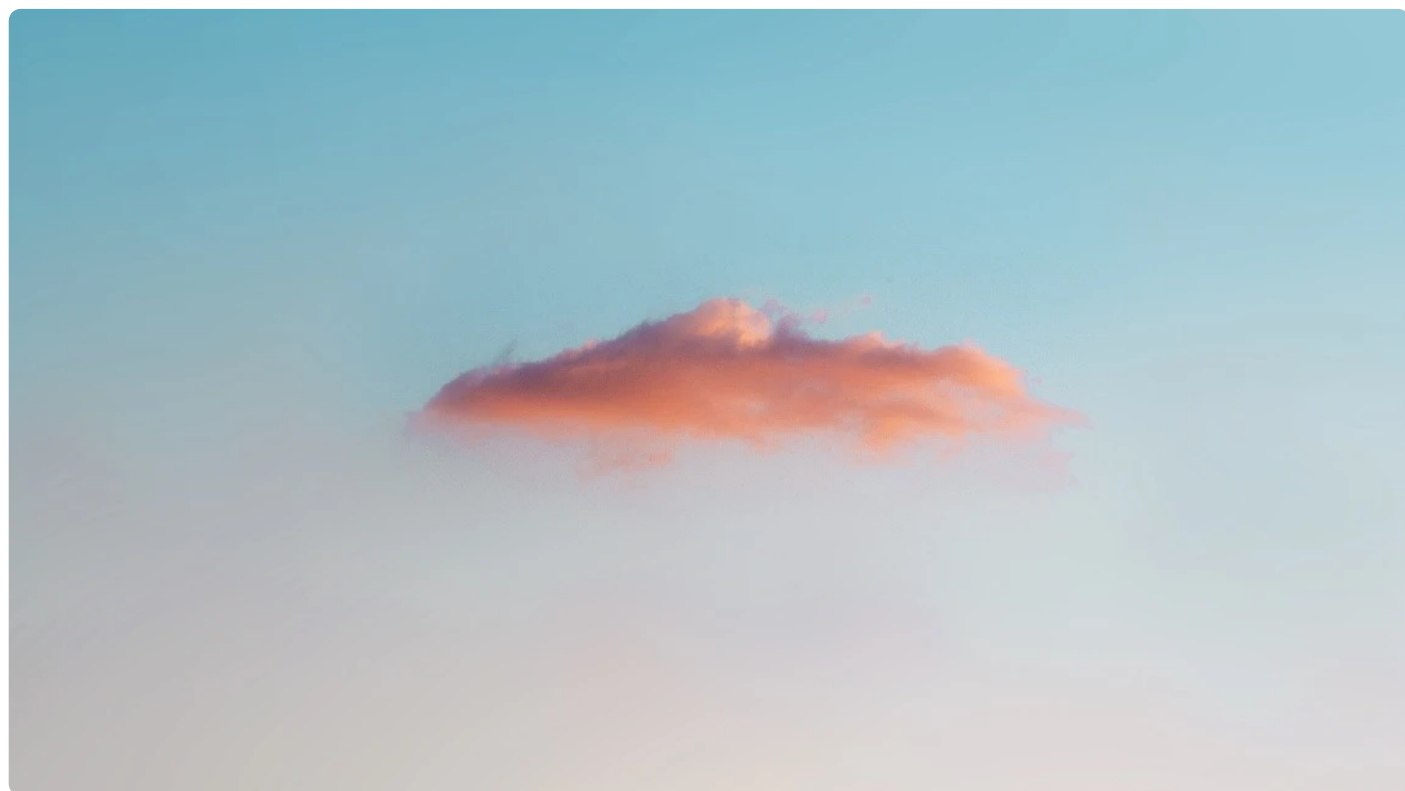


08 | 云上运维：云端究竟需不需要运维？需要怎样的运维？

2020-03-20 何恺铎 来自北京

《深入浅出云计算》



你好，我是何恺铎。

谢谢你的努力和坚持，我们已经学习了 IaaS 篇中的大多数内容。今天 IaaS 部分的最后一讲，我们来谈谈云上的运维工作。

云端需要运维吗？

既然要谈运维，我们得先回答这个必要性的问题。许多人都觉得，因为云服务大多都具有了非常高的可靠性和自动化程度，所以在云时代，运维就不那么重要了，甚至是可以省略的事情了。

这种观点有意无意地散播，其实会造成一些负面的影响。**开发者会容易轻视运维工作的重要性，忽略架构设计中运维友好性问题；而从事运维方向的工程师们，可能更会有点儿焦虑，甚至于担心未来的职业生涯。**

但很显然，这是一种误解。云端当然需要运维，而且云上运维很重要。因为不管在什么样的运行环境下，运维的本质和需求都没有消失，一样要为业务保驾护航，要保证系统的正常运作、应对突发情况等等。

云时代的运维，正确的理解应该是这样的：**云不但没有消灭运维，反而是助推了运维的发展。**

这是因为，云的引入能够让我们在更高的层面去思考和解决问题。比如说，云端基础设施的存在，可以让运维从偏硬件服务器、偏物理机房的日常繁琐工作中解脱出来，更多地基于云在**软件的层面**，进行部署、监控、调整。而云上的高质量、高可用的服务，也能避免我们重复建设，不用自己造轮子，也大大减轻了运维负担。

注意：底层的机房运维、基础架构运维仍然会继续存在，但会向头部的云供应商大规模集中。这属于云厂商的运维视角，是另一个宏大的话题，我们这里不多做讨论。

所以，云其实是提高了运维的效率，改变了运维的形态。

与此同时，由于云上运维的**软件属性**显著增强了，它就自然地 and 研发会有更强的融合。近期 DevOps 理念和云原生热潮的兴起，就说明了这一点。许多工作，你慢慢地会分不清它究竟是属于运维还是研发，因为两者的界限正在模糊。

另外，由于云独有的一些特点，它也会带来一些新的运维工作。比如我们课程中一直在涉及的**成本控制**，这也是云时代新运维所应当关注和包含的重要事项。因为云的成本消耗是动态、时刻发生着的，这 and 传统运维中的各类实时监控的对象，在形态上非常接近。

所以，云端需要运维吗？答案已经不言而喻了。

云时代的运维利器

工欲善其事，必先利其器。为了做好扎实的云上运维，首先我给你的一个建议是，你需要掌握云的**命令行工具**。现在几乎每个云都推出了自己的命令行工具，比如 AWS CLI、Azure CLI、阿里云 CLI 等等。

在前面各讲的例子中，为了便于你学习和理解，我都使用了公有云的网站门户来进行操作。但如果是在生产环境，你需要对很大规模的资源池逐个进行调整，或者同一件事情，你需要在不同时间反复地操作很多遍，那你就很可能需要将这些操作脚本化、程序化，这就需要用到云的命令行工具了。

虽然命令行工具有一定的学习曲线，但如果你熟悉了以后，其实是可以干脆利落地表达一个操作的。比如说，如果你要创建在 [第 6 讲](#) 的实验中，使用的虚拟机 “vm1-in-vpc1”，你就可以使用下面的 **aliyun ecs 命令** 来轻松表达：

 复制代码

```
1 [client@clientVM ~]$ aliyun ecs CreateInstance --ImageId ubuntu_18_04_x64_20G_ali
2 {
3     "InstanceId": "i-uf6hn8z47kqve3xxxxxx",
4     "RequestId": "222DA83B-0269-44BF-A303-00CB98E4AB07"
5 }
6 [client@clientVM ~]$ aliyun ecs StartInstance --InstanceId i-uf6hn8z47kqve3xxxxxx
7 {
8     "RequestId": "8E4C43CA-8F36-422C-AEF1-14ED5023856D"
9 }
```


现在各个云的 CLI 基本上都进化到了第二代，相比第一代，CLI 在易用性和表达能力上都有了很大的提升，你不妨学习尝试一下。而且这些 CLI 都能和 Shell 编程进行比较好的融合，你可以通过脚本组合多个关联的操作。

小提示：除了命令行工具，各云还都提供了开发者工具包（SDK）。如果你的资源调度逻辑相当复杂，或者需要与你自己的程序集成，那么你可以考虑使用相应语言的 SDK，来进行云上的一些资源管理操作。

如果你要频繁地在云上部署一套包含众多资源项的复杂系统，你还有另外一个得力的帮手：**资源编排类云服务**。属于这个领域的服务包括有 AWS CloudFormation、Azure 的 ARM Template、阿里云资源编排服务（ROS）等等，它们都可以通过使用一个 JSON 格式的文本文件，来描述和定义一个系统中所有的组件，以及它们互相之间的关系。

这个 JSON 文件，就是一个可以自动部署、可复用的单元了。这其实就是“基础设施即代码”（Infrastructure as Code）理念在云端的实现。

下面我给出了一个 Azure 的 ARM Template 的配置文件局部示例，可以让你有一个直观的感受：

 复制代码

```
1 {
2   "$schema": "https://schema.management.azure.com/schemas/2015-01-01/deployment
3   "contentVersion": "1.0.0.0",
4   "parameters": {
5     "adminUsername": {
6       "type": "string",
7       "metadata": { "description": "This is the username you wish to assign to
8     },
9     ...
10  },
11  "variables": {
12    "nicName": "VMNic",
13    "addressPrefix": "10.0.0.0/16",
14    "imagePublisher": "Canonical",
15    ...
16  },
17  "resources": [
18    {
19      "apiVersion": "2015-05-01-preview",
20      "type": "Microsoft.Network/publicIPAddresses",
21      "name": "[variables('publicIPAddressName')]",
22      "location": "[parameters('location')]",
23      "properties": { "publicIPAllocationMethod": "[variables('publicIPAddressT
24    },
25    {
26      "apiVersion": "2015-05-01-preview",
27      "type": "Microsoft.Network/virtualNetworks",
28      "name": "[variables('virtualNetworkName')]",
29      "location": "[parameters('location')]",
30      "dependsOn": [
31        "[resourceId('Microsoft.Network/networkSecurityGroups', variables('netw
32      ],
33      "properties": { ... }
34    },
35    {
36      "apiVersion": "2017-03-30",
37      "type": "Microsoft.Compute/virtualMachines",
38      "name": "[variables('vmName')]",
```

```
39     "location": "[parameters('location')]",
40     "dependsOn": [ "[concat('Microsoft.Network/networkInterfaces/', variables"
41     "properties": {
42         "hardwareProfile": { "vmSize": "[parameters('vmSize')]" },
43         "networkProfile": {
44             "networkInterfaces": [
45                 { "id": "[resourceId('Microsoft.Network/networkInterfaces',variable"
46             ]
47         },
48         ...
49     }
50 },
51 ...
52 ]
53 }
```

注：这个文件是用于配置单机 WordPress 网站的模板，这里略去了许多内容，其全貌可以参见 [这个链接](#)。

这类资源编排服务，理论上能够**支持云上所有服务的组合，而且配置节点互相能够引用**，功能十分强大。它还具有一定的灵活性，一般都有**输入参数字段**，允许你在部署时动态决定一些选项和参数值，还可以**自定义结果输出字段**，方便部署完成后告诉你一些结果信息。

在这类资源编排部署系统的帮助下，我们云端部署类的工作可以得到极大的自动化。

云运维由哪些工作组成？

有了趁手的工具之后，我们下一个需要讨论的问题就是，**云时代的运维具体有哪些重要的工作呢？哪些是和传统运维一脉相承的事情，哪些又是在云环境下所特有的内容呢？**

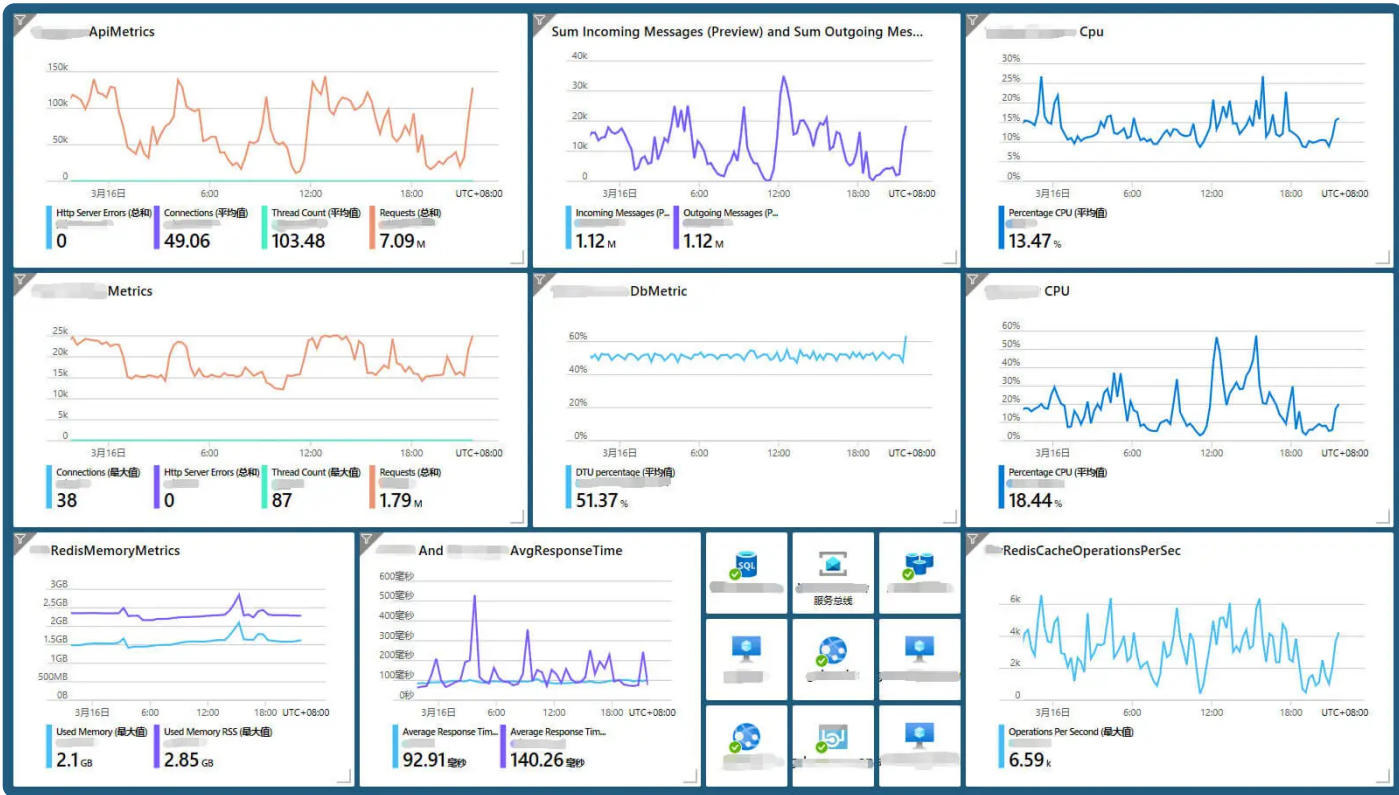
现在，我就和你一起来简单梳理一下。

首先，在云端，传统的运维工作仍然存在，其中包括你所熟知的监控、部署、升级、备份等等。只是操作手段会有所不同，比如在云上，我们可以利用前面说到的命令行工具和资源模板来进行部署。

监控一直是运维最核心的工作之一。几乎所有的云端服务都自带有一定的监控功能，默认提供了不少内置的维度指标和可视化图表，这些开箱即用的图表你要充分利用好，它们能够很好地帮助你了解相关服务的状态。

那么，如果自带的监控不够用怎么办？其实这些默认的统计监控的背后，往往都是由云的一个**大型统一监控服务**来支撑的，如 AWS 的 CloudWatch 和 Azure 的 Monitor 等等。你可以好好研究一下这类统一监控服务，通过它可以满足你更深度的自定义监控需求。

另外，这些你精心选择和设置的监控项，还能够和云上的仪表盘服务，以及报警服务联动，轻松实现运营监控的“大屏”和问题的实时报警。



Azure 上的自定义监控仪表盘示例

这里我还想再多谈一谈**备份**。

备份是一个简单但又很容易被我们忽视的事项。即便是在云端，尽管云厂商已经做了许多如三副本之类的防护措施，但还是会存在出故障的可能，所以我们仍然需要做好备份，尤其是重要

数据的备份。总之，我们在云上需要创造多层次的冗余，而备份在创造冗余方面也承担着重要的角色，有的时候，它会是我们的最后保障。

在 IaaS 的虚拟机层面做备份，你的得力助手会是**镜像**和**快照**。

镜像我们在 [上一讲](#) 中已经接触过了，它可以用来恢复虚拟机；快照则是云磁盘级别对应的备份概念，它可以帮助你某块磁盘某一时刻的状态进行封存和恢复，你还可以定期定时为一些重要磁盘自动生成快照。

<input type="checkbox"/>	云盘ID/云盘名称	云盘种类(全部)	云盘状态(全部)	付费类型(全部)	云盘属性(全部)	操作
<input type="checkbox"/>	d-uf6eg68odug2okqe6byi my-data-drive	高效云盘 20GiB (1960 IOPS)	待挂载	按量付费	数据盘	创建快照 重新初始化磁盘 设置自动快照策略 更多
<input type="checkbox"/>	d-uf64a7xie8evn2xbcli7 my-system-drive	高效云盘 40GiB (2120 IOPS)	使用中	按量付费	系统盘	创建快照 重新初始化磁盘 设置自动快照策略 更多

注意：不要小看镜像和快照这样简单基础的操作，像在 [第 5 讲](#) 中提到过的创业公司严重事故，就完全可以通过简单的磁盘快照进行避免。因为快照的存储本身不依赖于云盘，这就是额外的冗余。

除了虚拟机和磁盘层面，文件层面的备份同样重要而有效。而且文件的备份最好还能以异地的方式来存储。云上的对象存储可以在这方面肩负重任，我在 PaaS 篇中会做专门讲解。

其次，你的运维工作中很可能包含**迁移**。

这是带有云端特色的运维任务，因为**只要不是在云上创建的全新业务，传统业务在逐步上云的过程中一定会面临迁移工作。**

迁移显然是非常大的一个话题，有些复杂的迁移项目，持续的时间可能长达几个月。这里我想告诉你两点**最核心**的建议：

第一，在生产业务切换过来之前，一定要对云上的新架构、新方案进行充分而深入的 POC 测试，不可操之过急。对于复杂场景，可能要通过不断地实践，才能够逐步进化出完善的云

上解决方案。

第二，对于一些虚拟机、数据库等独立的软硬件单元，许多云厂商都提供了官方的迁移服务或工具，支持离线甚至在线迁移，妥善使用可以事半功倍。比如 AWS 的主机迁移服务 SMS (Server Migration Service)、数据库迁移服务 DMS (Database Migration Service) 和阿里云的数据传输服务 DTS (Data Transmission Service) 等。

所以，当你遇到一些迁移场景时，不妨先查一查云厂商是否有官方的支持。由于迁移类服务能够直接为厂商导流获客，所以云厂商一般都会比较重视，往往能给你提供相当好的用户体验。

再次，云上的运维会包含和云厂商进行对接的工作。

毕竟我们的大厦是建立在云厂商所提供的基础设施之上的。云虽然已经高度成熟，但作为一个高度复杂的系统，也总难免会有不按你所期望进行工作的时候，或者极为偶尔也会出些小 Bug，这时和云厂商的对接渠道就显得尤为重要了。

所以，我们的运维团队中需要有相应的角色对云的工单机制，以及技术支持侧的对接方式了然于胸，以备不时之需。你也要熟读文档，要吃透云计算的许多特性，这样才能更准确地与客服沟通，更快地寻求到对口的帮助，最后解决好问题。

最后，云上运维会具有很强的管理属性。

这里的管理，指的不仅仅是对云上资源的管理，更要深入到流程和制度的管理层面。比如对于云资源的命名、开通、清理等日常操作的规范，各类云上安全的控制和最佳实践，所有云资源的负责人、所属资源组和权限体系等等。这些都需要有效的管理手段，才能避免资源在云上的野蛮生长。

所以，高明的云上运维，既要为应用开发赋能，要足够高效，也要有适当的管理和约束。我们团队的组织架构和分工，最好也能够配合和适应这个需要。

好在云厂商也在不断推出和完善与云上管理相关的配套服务，比如说，Azure Policy 能够限定只有某类型号的资源可以被创建，还可以扫描和检查各种最佳实践是否得到了应用；再比如，

AWS CloudTrail 能够对账户内的操作进行监控和审计。如果你的组织内用户（团队成员）较多，就值得好好探索研究一下这一类的云服务。

当然，管理层面还有一项重要事务，就是我们多次提到的**成本管理**。公司或团队中，应当有专人对成本进行监控和分析，以此提升每一位用户的成本意识。我自己曾使用的实践，是按月来组织资源的使用方进行成本消耗的回顾，分析资源使用的上升、下降趋势及其主要原因，同时还会检查月度账单明细，以杜绝成本浪费。

课堂总结与思考

今天这一讲，与其说是教程，不如说是和你一起探讨云上运维的相关要点。因为篇幅所限，今天我主要总结介绍了那些最重要的，和你最需要了解的内容，没有办法深入探究每一个与运维相关的细节。但你必须知道这些事务的存在，明白云上运维需要做哪些事情，这样在你需要的时候，才能有针对性地去查找资料，找到怎么做这些事情的方法。

当前业界的一个重要趋势是，运维和开发的边界正在模糊。所以我在前面提到的诸多运维工作，可能是由开发者来负责，也可能是运维人员来承担。这要根据你们公司和部门的具体情况来决定。但至少，这些工作很重要，无论由什么角色来完成，总是需要有人来扎实落地的。

所以从个人视角来看，作为开发者，你应该学习和掌握一些运维的知识和技巧，让自己变得更加全面和综合；如果作为运维人员，你也应该学习了解现代软件构建和系统架构方面的知识，尤其是学习云、掌握云，为云端架构的全面到来做好准备。

今天留给你的思考题是：

如果要执行一些云上的 CLI 命令，你当然可以在自己的机器上安装命令行工具包，但其实你还可以使用不少云都提供的非常方便的“Cloud Shell”。那你知道什么是 Cloud Shell，以及要如何使用它吗？

前面讲到云上资源管理时，我提到了“资源组”的概念。你知道资源组是什么吗？它起到什么作用呢？

好，至此我们课程 IaaS 部分的 8 篇内容就全部结束了，希望你有所收获。下一讲，我们将进入精彩的 PaaS 世界。欢迎你留言与我交流，咱们下期再见。

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

精选留言 (15)



何恺铎 置顶

2020-03-20

[上讲问题参考回答]

1. “会话保持”用于保证客户端的多次连接都能够到达同一台后端机器，这样在负载均衡之后的应用服务可轻松实现和同一个客户的持续“会话”(Session)。在实现层面，负载均衡器主要通过 IP 地址信息或 HTTP 协议中的 cookie 来标记和定向客户端请求。
2. 包年包月虚拟机，和竞价示例虚拟机，都可以参与到弹性伸缩类服务中。前者常用作“保底”的机器，后者推荐作为按需创建的机型，这样结合起来可以最大化节约成本。

共 1 条评论 >

👍 18



摇滚诗人M

2020-03-21

都是一些概念性介绍的文章，请问有在深入一点的关于技术实现层面的文章吗？

作者回复：你好。在我们专栏的开篇词中已经提到了，这个专栏专注于“用云”而不是“做云”。如果你想了解云计算的实现，可以看看极客时间上刘超老师的Linux专栏，里面对虚拟化有精彩的介绍。如果想要更加深入，可以考虑直接啃一啃KVM、OpenStack这样的开源项目，学习文档和源码。

回到“用云”的角度，也请不要忽视专栏讨论的诸多“概念”。这些概念的诞生初衷、技术本质、产品形态和应用场景，正是云计算的核心知识体系，是我们需要了解和掌握的。

共 2 条评论 >

👍 15



一步

2020-03-20

Cloud Shell 基于云的 Shell，可以直接在浏览器上或者app内使用的命令行进行操作

作者回复: 是的, 省去了创建虚拟机、安装云上CLI的麻烦。



👍 7



我来也

2020-03-20

资源组, 正好是我最近在用的东西。

可以把不同的云资产归宿到不同的组, 可以针对组设置一些规则, 还可以方便的统计资源组内的整体费用支出。

有了它, 就不再是笼统的一本糊涂账了。

知道那部分花的多了, 可以有针对性的优化。

作者回复: 没错, 善用资源组, 能让云上资源井井有条。



👍 7



leslie

2020-03-20

不知道老师是否有这种感觉: 上云之后似乎只要规划、部署和迁移。可能现实中确实看到了许多, 觉得监控和备份-用好云厂商的就够了? 不知道老师对此如何去看待, 至少个人是觉得不够。

可能课程一路学习一路发现问题吧: CLI应当是更合乎云系统/环境的定制化编程, 就像老师开课时让我们寻找操作系统时去选择云厂商的linux系统。其实如果明白且会shell, 换手其实很快的, 这个就像真正掌握SQL语言, 关系型数据库之间的轮换使用只要几天就够了; 曾经有过3种以上关系型数据库全部在生产中使用的经历。

云上的资源组: 主机监控中我们会看到许多硬件情况; 资源组最简单的概念就是你所有拥有资源的组合, 包括cpu、硬盘、内存, 让你去明白这些东西在某种状态下的使用率以及是否合理。不过其实个人去看过不少; 其中解决的层次只能是Public的东西。

概念的模糊其实源自去过分追求某个名字: 就像有些觉得自己有敏捷开发有问题, 不断努力, 几年后还不满意, 要做DevOps; 可是咨询过后发现其实已经做成了90的DevOps了, 只需稍微调整一下就好了。为了做而做反而会去迷失其本性的东西。

谢谢老师今天的分享: IaaS篇让我受益良多; 至少我明确的知道要做什么了。

作者回复: 赞总结。

关于监控和备份, 云厂商的功能是否足够取决于你的具体需求。但一个事实是, 许多用户对云这方面的内置能力还都没有充分地了解和使用的。



👍 5



夜空中最亮的星

2020-03-20

老师这一篇打消了我心中的恐惧，运维不是在消失，而是在升级



👍 3



艾利特-G

2020-04-01

资源组是成本管理里的重要逻辑单位。

云上的每个资源对象都应该被纳入相应的资源组。比如我公司有2个事业部，就可以创建2个资源组，各自使用的计算，存储，网络类资源在财务上就和事业部关联起来了。



👍 2



mrtwenty

2020-03-21

老师，阿里云的oss能否存放数据库的备份文件，我想做一个定时任务，从ecs服务器里面将数据库到处备份到oss里。

作者回复: 当然是可以的，对象存储的重要应用场景之一就是文件备份。第10讲我们会专门讨论对象存储。



👍 2



Eric

2020-03-21

老师，怎么看云管理服务商，cloud MSP的发展？上云后，客户对云上运维的需求少了很多，很少的人，做更多的事情，运维的路变窄了

作者回复: 我比较看好Cloud MSP的市场和业务。因为总体云计算的盘子在增长，而且对于许多传统企业来说这是个新生事物，需要帮助和指导，这里就有Cloud MSP的空间了。

另外，从商业模式上来讲，Cloud MSP如果有一些自研的产品，产品加咨询的方式来售卖，会更有发展潜力。



👍 2



2020-03-20

对运维提出来更高要求



Geek_761876

2022-06-26

对于迁移，老师有推荐的比较干货的课程活资料吗？



胖子

2020-04-13

资源编排类云服务与k8s是何关系？ 里面包涵应用的自动化部署吗？

共 1 条评论 >



戴斌

2020-03-21

等待期待更新



丁乐洪

2020-03-21

专栏看下来，跟运维相关内容比较多，期待专栏有跟开发部署等相关内容



一步

2020-03-20

我认为现在的云上运维要充分的了解云提供的各种服务，以及这些服务怎么组合在对外提供服务（必须要熟读文档）

