# 云原生应用白板设计会议学生指南

### ****内容概览****

##### 摘要和学习目标

##### 步骤1：客户案例调研

* 客户情况
* 客户的需求
* 客户异议
* 常见情况的信息图

##### 步骤2：设计概念验证解决方案

##### 步骤3：提出解决方案

##### 总结

##### 其他参考

### ****摘要和学习目标****

在此白板设计会议中，您将学习与在Azure中构建和部署容器化应用程序有关的选择，围绕此问题的关键决策以及解决方案的其他方面，包括提升和移动应用程序部分以减少应用程序更改的方法。

在本设计课程结束时，您将能够更好地设计针对Azure Kubernetes服务（AKS）的解决方案，并为容器化应用程序定义DevOps工作流。

### ****步骤1：客户案例调研****

**结果**

分析您的客户需求。

时限：15分钟

指导：与所有参与者一起，讲师提供客户案例研究的概述以及技术提示。

1. 与您的同桌参与者和培训师见面。
2. 阅读学生指南中步骤1-3的所有说明。
3. 作为表团队，请查看以下客户案例研究。

**客户情况**

Fabrikam Medical Conferences提供适合医学界的会议网站服务。他们始于10年前，为一个小型会议组织者建立了一些会议场所。从那时起，口碑相传，Fabrikam Medical Conferences现在是一个著名的行业品牌。现在，他们每年处理100多个会议，并且还在不断增长。

医疗会议通常是低预算的网站，因为高端会议通常在100至1500位与会者之间。同时，会议所有者具有重大的客户化要求，并快速在会议中实现。这些更改可能会影响从UI到后端的系统各个方面，包括会议注册和付款条件。

Fabrikam的工程VP Arthur Block拥有一个由12名开发人员组成的团队，负责他们客户站点的开发，测试，部署和运营管理的各个方面。由于客户的需求，他们在开发和DevOps工作流程的效率和可靠性方面存在问题。

会议站点当前在本地托管，具有以下拓扑和平台实现：

* 会议网站使用MEAN堆栈（Mongo，Express，Angular，Node.js）构建。
* 网站和API是基于托管在Linux服务器上的微服务构建的。
* 本地数据后端是MongoDB；在单​​独的Linux服务器群集上运行。
* 关系型数据存储在PostgreSQL中，运行在Linux服务器上。

客户被视为“租户”，每个租户都被视为唯一的部署，由此发生以下情况：

* 每个租户在MongoDB集群中都有一个具有自己的collection的数据库，而在PostgreSQL中则有一个数据库。
* 将获取最新的功能会议代码库的副本，并将其配置为指向租户数据库。
* 这包括一个网站代码库和一个管理网站代码库，用于输入会议内容，例如演讲者，会议，讲习班，赞助商和与会者的会议反馈。
* 进行修改以支持客户要求的UI样式，图形，布局和其他自定义需求。
* 会议所有者有权访问管理网站以输入事件详细信息。
* 他们每年都会在每个会议上继续使用此管理站点。
* 他们可以添加每个新事件并在每个会议之间隔离发言人，会议，研讨会和其他详细信息。
* 租户的应用代码（会议和管理网站）被部署到一组特定的负载平衡Linux服务器。每组服务器托管一组特定的租户，并且根据租户的Scale要求进行分配。
* 新会议流程被启用后，不可避免的要求更改网站页面，样式，注册要求以及任意数量的自定义请求。

Arthur痛苦地意识到，这家小企业已经发展成为更大的企业，已经有机地发展成为应该用于会议的完全多租户应用程序套件。但是，团队很难达到这个目标。他们会不断地为每个租户更新代码，并尽最大努力将改进内容合并到核心代码中，以便用于召开新会议。变革的步伐很快，预算很紧，而且他们根本没有时间停止和重组核心代码库来支持客户所需的所有灵活性。

Arthur希望朝着这个方向迈出一步，并牢记以下目标：

* 进行更改时，减少单个租户中引入的回归测试。
* 代码的问题之一是，它具有跨功能的许多依赖关系。看似简单的代码区域更改会带来布局，响应能力，注册功能，内容刷新等问题。
* 为避免这种情况，他想重新设计核心代码，以便注册，电子邮件通知和模板，内容和配置彼此之间以及与前端完全分开。
* 理想情况下，对单个区域的更改将不再需要对该站点进行完整的回归测试；但是，鉴于他们管理的站点数量，这是不太现实的。
* 改善DevOps生命周期。
* 在会议的整个生命周期中，招募新租户，为现有租户启动新站点以及管理所有实时租户所花费的时间非常低效。
* 通过减速客户招募，管理已部署站点和监视运行状况的Workloads，随着公司不断增长，公司可以控制成本和开销。这可能会留出时间来改善他们希望为长期增长而构建的多租户平台。
* 增加对系统操作和运行状况的可见性。
* 该团队对已部署的网站几乎没有总体健康监控。

尽管实现多租户是新代码设计的目标，但即使有了这一点，Arthur仍然认为，始终需要为一次性定制需求的特定租户提供代码的定制化副本。Arthur认为Docker容器可能是满足其短期DevOps和开发敏捷性需求的一个很好的解决方案，而且一旦达到多数多租户应用程序解决方案，它也是正确的方向。

**客户的需求**

1. 减少部署新会议租户的时间，复杂性和成本。
2. 提高会议租户更新网站的可靠性。
3. 在Azure上的为Docker容器架构策略选择合适的服务。服务平台选择应：

* 使部署和管理基础架构变得容易。
* 提供工具以帮助他们监视和管理容器的运行状况和安全性。
* 使管理不同租户的弹性资源需求变得更加容易，从而使他们不再需要将租户分配给特定的负载均衡集群。
* 提供平台中立的解决方案，以使特定的内部部署或云环境不会成为新的依赖项。

1. 以最小的应用程序代码更改将数据从本地MongoDB迁移到Azure Cosmos DB。
2. 将关系数据从PostgreSQL本地数据库迁移到Microsoft Azure
3. 继续使用Git存储库进行源代码控制，并将其集成到CI / CD工作流程中。
4. 首选具有以下方面的完整的运营管理工具套件：

* 在开发和初始POC工作期间用于手动部署和管理的UI。
* 集成CI / CD自动化的API。
* 容器调度和编排。
* 健康监控和警报，可视化状态。
* 容器镜像扫描。

1. 为单个租户完成建议解决方案的实现，以培训团队并完善流程。
2. 借助AI增强与会者会话反馈，以防止发布不适当的内容，并进行实时语言翻译，以更好地适应不断增长的全球会议出席人数。

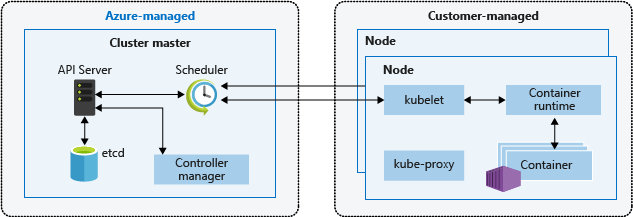
**客户异议**

1. 有多种方法可以在Azure上部署Docker容器。这些选择如何比较？每种选择的动机是什么？
2. Azure中是否有一个选项可以提供易于管理和迁移到的容器编排平台功能，还可以满足我们的Scale和管理工作流要求？
3. 我们听说Azure Cosmos DB与MongoDB兼容。这将提供最小化代码更改的迁移吗？
4. 我们知道Microsoft提供带有预先构建的AI模型的认知服务。哪些型号提供了我们希望用于增强会议网站的功能？

**常见情况的信息图**

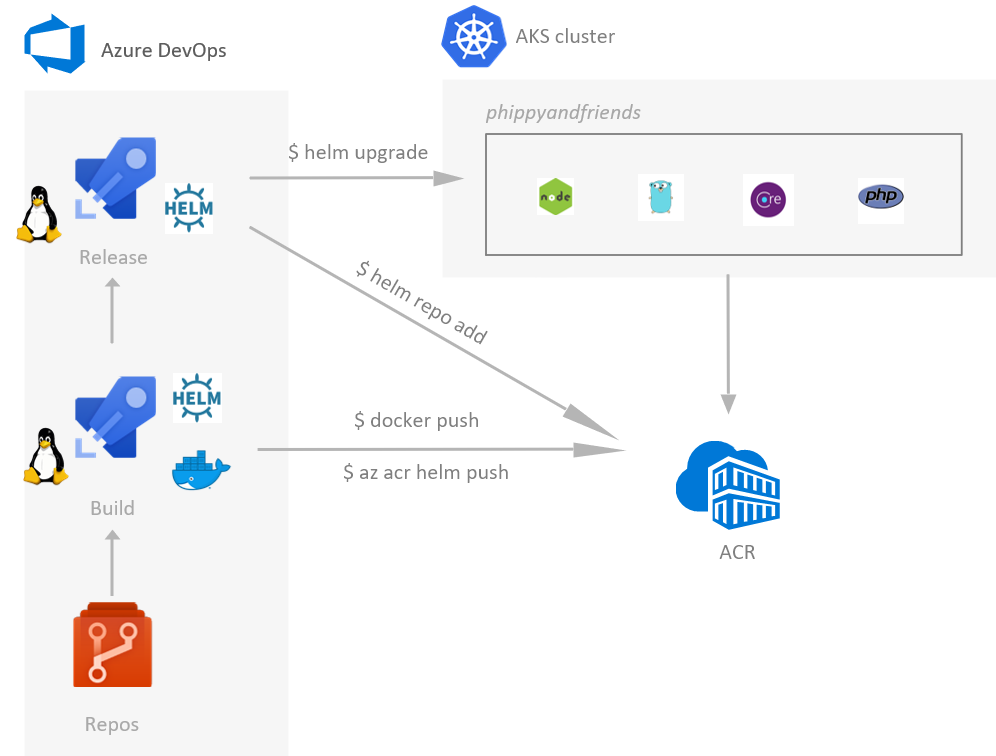
Kubernetes\_\_体系结构

**注意**：此图说明了Kubernetes拓扑，说明了Azure管理的主Node以及客户可以在其中集成和部署应用程序的代理Node。

[](https://github.com/Azure/OCPCHINATECH/blob/master/OCPPTSCSAHandsonLabs/MCW-Cloud-native-applications/Whiteboard%20design%20session/media/azure-kubernetes-components.png)

[https://docs.microsoft.com/zh-CN/azure/aks/intro-kubernetes](https://docs.microsoft.com/en-us/azure/aks/intro-kubernetes)

使用\_\_Azure DevOps\_\_的\_\_CI / CD\_\_到\_\_Azure Kubernetes\_\_服务

[](https://github.com/Azure/OCPCHINATECH/blob/master/OCPPTSCSAHandsonLabs/MCW-Cloud-native-applications/Whiteboard%20design%20session/media/azure-devops-aks.png)

<https://cloudblogs.microsoft.com/opensource/2018/11/27/tutorial-azure-devops-setup-cicd-pipeline-kubernetes-docker-helm/>

### ****步骤2：设计概念验证解决方案****

**结果**

设计一个解决方案，并准备以15分钟的对话形式将解决方案介绍给目标客户。

时间范围：60分钟

**业务需求**

方向：让所有参与者都在您旁边，回答以下问题，并在白板上列出答案：

1. 您应该向谁提出这个解决方案？谁是您的目标客户群？谁是决策者？
2. 您需要使用解决方案解决哪些客户业务需求？

**设计**

方向：让所有与会者都在您的旁边，在白板上回答以下问题：

总体概览架构

1. 根据客户的情况，您将为单个会议租户提出什么样的容器作为新的微服务体系结构的一部分？
2. 在不涉及细节的情况下（以下各节将讨论特定的细节），图解您对容器平台，应部署的容器（对于单个租户）和数据层的初步构想。

在\_\_Azure\_\_上选择容器平台

1. 列出将容器部署到Azure的潜在平台选择。
2. 您会推荐哪个？为什么？
3. 描述客户如何配置其Azure Kubernetes服务（AKS）环境以开始其POC。

容器，发现和负载平衡

1. 描述开发人员在构建POC时将遵循的高级手动步骤，以在Azure Kubernetes Service（AKS）上构建映像和运行容器。摘要中包括以下组件：

* 包含其源代码的Git存储库。
* Docker映像注册表（Image Registry）。
* 生成Docker映像并推送到注册表（Registry）的步骤。
* 使用Kubernetes仪表板运行容器。

1. 客户对Docker映像注册表和容器扫描有哪些选择，您会推荐什么？
2. 客户将如何配置网站容器，以便可以从Azure Kubernetes服务（AKS）在端口80/443上公开访问它们？
3. 说明Azure Kubernetes服务（AKS）如何将请求路由到位于端口80/443上的同一Node上的多个网站容器

可伸缩性注意事项

1. 向客户说明Azure Kubernetes服务（AKS）及其预先配置的缩放集如何支持群集Autoscale。

自动化\_\_DevOps\_\_工作流程

1. 描述GitHub Actions如何帮助客户自动化其连续的集成和部署工作流以及Azure Kubernetes Service（AKS）基础架构。
2. 描述建议的方法，以使Azure Kubernetes服务（AKS）Node保持最新，最新的安全修补程序或受支持的Kubernetes版本。

**准备**

方向：所有同组与会人员都在您的旁边：

1. 确定未通过建议的解决方案解决的任何客户需求。
2. 确定您的解决方案的好处。
3. 确定您将如何回应客户的反对。

准备给客户15分钟的口头演讲风格的演示文稿。

### ****步骤3：提出解决方案****

**结果**

以15分钟的演讲形式向目标客户群体介绍解决方案。

时限：30分钟

**推介会**

方向：

1. 与另一张桌子的学员配对。
2. 一组人代表Microsof团队，另一组代表客户。
3. Microsoft团队向客户介绍了他们提出的解决方案。
4. 客户从异议列表中提出异议之一。
5. Microsoft团队对此异议做出了回应。
6. 客户团队向Microsoft团队提供反馈。
7. 两组人切换角色并重复步骤2-6。

### ****总结****

时限：15分钟

方向：与较大的小组重新开会，以听取主持人/中小型企业分享案例研究的首选解决方案。

**其他参考**

Azure Kubernetes服务（AKS）

<https://docs.microsoft.com/azure/aks/intro-kubernetes/>

Kubernetes

<https://kubernetes.io/docs/home/>

AKS常见问题

<https://docs.microsoft.com/azure/aks/faq>

AutoscaleAKS

<https://github.com/kubernetes/autoscaler>

AKS群集Autoscale器

<https://docs.microsoft.com/azure/aks/cluster-autoscaler>

升级AKS集群

<https://docs.microsoft.com/azure/aks/upgrade-cluster>

Azure管道

<https://docs.microsoft.com/azure/devops/pipelines/>

集装箱安全

<https://docs.microsoft.com/azure/container-instances/container-instances-image-security/>

镜像隔离

<https://github.com/Azure/acr/tree/master/docs/preview/quarantine/>

集装箱监控解决方案

<https://docs.microsoft.com/azure/azure-monitor/insights/containers>

Azure Cosmos数据库

<https://docs.microsoft.com/azure/cosmos-db/introduction>

PostgreSQL的Azure数据库

<https://azure.microsoft.com/services/postgresql/>

Azure认知服务

<https://azure.microsoft.com/services/cognitive-services/>

窗体顶端

Go

窗体底端

* © 2022 GitHub, Inc.
* [Terms](https://docs.github.com/en/github/site-policy/github-terms-of-service)
* [Privacy](https://docs.github.com/en/github/site-policy/github-privacy-statement)
* [Security](https://github.com/security)
* [Status](https://www.githubstatus.com/)
* [Docs](https://docs.github.com/)
* [Contact GitHub](https://support.github.com/?tags=dotcom-footer)
* [Pricing](https://github.com/pricing)
* [API](https://docs.github.com/)
* [Training](https://services.github.com/)
* [Blog](https://github.blog/)
* [About](https://github.com/about)

Loading complete