

在 Azure 上构建可扩展的应用程序

此博客文章由 *ChatGPT-4o* 协助撰写。

目录

- 简介
- 开始使用 Azure 订阅
- 使用 Azure Kubernetes 服务 (AKS) 部署应用程序
 - 创建和管理 AKS 集群
 - 部署应用程序
- 从 Pods 获取日志
- 使用 Azure Application Insights 进行监控和诊断
- 利用 Azure 虚拟机 (VMs)
- 使用 Azure 事件中心进行实时数据摄取
- 使用 Azure API 管理服务管理 API
- 利用 Azure SQL 数据库
- 使用 Kusto 查询语言 (KQL) 查询日志
- 设置主动监控的警报
- 结论

简介

在云计算领域，Microsoft Azure 作为一个强大的平台，在构建、部署和管理应用程序方面表现突出。在我们最近的项目中，我们利用了多个 Azure 服务，包括 Azure 订阅、Azure Kubernetes 服务 (AKS)、Application Insights、虚拟机 (VMs)、事件中心 (Event Hubs)、API 管理服务以及 SQL 数据库，以创建一个可扩展且受监控的应用程序基础设施。本博客文章概述了我们的方法、使用的工具、最佳实践，以及管理集群、获取日志和查询日志的详细步骤。

开始使用 Azure 订阅

Azure 订阅是您访问 Azure 服务的入口。它作为一个容器，包含了所有的资源，如虚拟机、数据库和 Kubernetes 集群。

1. 设置 Azure 订阅：
 - 注册：如果您还没有 Azure 账户，请先在 Azure 门户 注册。
 - 创建订阅：导航到“订阅”部分并创建一个新订阅。这将是您的计费和管理容器。
2. 资源组织：

- 资源组：根据资源的生命周期和管理标准，将资源组织到资源组中。
- 标签：使用标签添加额外的元数据，以便更容易管理资源和进行计费。

使用 Azure Kubernetes 服务（AKS）部署应用程序

Azure Kubernetes 服务（AKS）是一种托管的 Kubernetes 服务，可以简化容器化应用程序的部署、管理和扩展。

创建和管理 AKS 集群

1. 在 Azure 门户中创建 AKS 集群：

- 设置：在 Azure 门户中搜索 AKS 并创建一个新的 Kubernetes 集群。
- 配置：选择集群大小、配置节点池并设置网络。
- 身份验证：使用 Azure Active Directory（AAD）进行安全访问控制。
- 监控：在设置过程中启用监控和日志记录。

2. 使用 Azure CLI 创建 AKS 集群：

```
az aks create \
  --resource-group myResourceGroup \
  --name myAKSCluster \
  --node-count 3 \
  --enable-addons monitoring \
  --generate-ssh-keys
```

3. 管理您的 AKS 集群：

- 扩展集群：

```
az aks scale \
  --resource-group myResourceGroup \
  --name myAKSCluster \
  --node-count 5
```

- 升级集群：

```
az aks upgrade \
  --resource-group myResourceGroup \
  --name myAKSCluster \
  --kubernetes-version 1.21.2
```

部署应用程序

1. 使用 Kubernetes 清单：编写 YAML 文件来描述您的部署、服务和其他 Kubernetes 对象。

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: myapp
spec:
  replicas: 2
  selector:
    matchLabels:
      app: myapp
  template:
    metadata:
      labels:
        app: myapp
    spec:
      containers:
        - name: myapp
          image: myregistry.azurecr.io/myapp:latest
          ports:
            - containerPort: 80
```

2. 使用 kubectl 进行部署：

```
kubectl apply -f myapp-deployment.yaml
```

3. Helm 图表：使用 Helm 来管理 Kubernetes 应用程序和版本控制。

```
helm install myapp ./mychart
```

从 Pods 获取日志

1. 附加到 Pod 并获取日志：

```
kubectl logs <pod-name>
```

- 实时流日志：

```
kubectl logs <pod-name> -f
```

2. 使用 Sidecar 进行日志记录：

- 在您的 Pod 规范中创建一个日志记录 sidecar 容器，将日志发送到集中日志服务。

```
spec:
  containers:
  - name: myapp
    image: myregistry.azurecr.io/myapp:latest
    ...
  - name: log-shipper
    image: log-shipper:latest
    ...
```

使用 Azure Application Insights 进行监控和诊断

Application Insights 为您的应用程序提供强大的监控和诊断功能。

1. 设置 Application Insights:

- 集成: 将 Application Insights SDK 添加到您的应用程序代码中。
- 仪表化密钥: 使用从 Application Insights 资源获取的仪表化密钥配置您的应用程序。

2. 跟踪性能:

- 指标: 监控响应时间、失败率和应用程序依赖项。
- 实时指标流: 查看实时性能指标以获取即时见解。

3. 诊断和故障排除:

- 应用程序地图: 可视化依赖项并识别性能瓶颈。
- 事务诊断: 使用分布式跟踪来跟踪跨服务的请求。

利用 Azure 虚拟机 (VMs)

Azure VMs 提供了运行自定义应用程序和服务的灵活性, 这些应用程序和服务不能被容器化。

1. 配置虚拟机:

- 创建虚拟机: 在 Azure 门户中创建新的虚拟机并选择适当的大小和操作系统。
- 网络配置: 设置虚拟网络、子网和安全组以控制流量。

2. 配置虚拟机:

- 软件安装: 安装所需的软件和依赖项。
- 安全性: 定期应用补丁和更新, 配置防火墙并使用网络安全组 (NSGs)。

3. 管理虚拟机:

- 备份和恢复: 使用 Azure Backup 进行虚拟机备份。
- 监控: 使用 Azure Monitor 监控虚拟机性能。

使用 Azure 事件中心进行实时数据摄取

Azure 事件中心是一个大数据流平台和事件摄取服务, 能够每秒接收和处理数百万个事件。

1. 设置事件中心：

- 创建事件中心命名空间：在 Azure 门户中创建一个事件中心命名空间，以容纳您的事件中心。
- 创建事件中心：在命名空间内创建一个或多个事件中心以捕获您的数据流。

2. 摄取数据：

- 生产者：配置您的应用程序或服务以使用多个语言的 SDK（例如，.NET、Java、Python）将事件发送到事件中心。
- 分区：使用分区来扩展事件处理，确保高吞吐量和并行处理。

3. 处理事件：

- 消费者：使用消费组来读取和处理事件。Azure 提供了几种处理选项，包括 Azure Stream Analytics、Azure Functions 和使用事件中心 SDK 的自定义处理。

4. 监控事件中心：

- 指标：通过 Azure 门户监控吞吐量、延迟和事件处理指标。
- 警报：设置警报以通知您任何问题，例如高延迟或丢失消息。

使用 Azure API 管理服务管理 API

Azure API 管理服务提供了一种为现有后端服务创建一致且现代的 API 网关的方法。

1. 设置 API 管理：

- 创建 API 管理服务：在 Azure 门户中搜索 API 管理并创建一个新服务。
- 配置 API：定义并导入来自 OpenAPI 规范、Azure Functions 或其他后端的 API。

2. 保护 API：

- ** 身份验证和授权

：使用 OAuth2、JWT 验证和其他机制来保护您的 API。 - 速率限制和限流 **：实施策略以保护您的 API 免受滥用。

3. 监控和分析：

- API 洞察：跟踪使用情况、监控性能并分析日志。
- 开发者门户：为开发者提供一个发现和使用您的 API 的门户。

4. 管理生命周期：

- 版本控制和修订：无缝管理 API 的不同版本和修订。
- 策略管理：应用策略以转换、验证和路由请求和响应。

利用 Azure SQL 数据库

Azure SQL 数据库是一个完全托管的关系数据库，具有内置的智能、高可用性和可扩展性。

1. 设置 Azure SQL 数据库：

- 创建 SQL 数据库：在 Azure 门户中导航到 SQL 数据库并创建一个新数据库。

- **配置数据库：** 设置数据库大小、性能级别并配置网络设置。
2. **连接到 SQL 数据库：**
 - **连接字符串：** 使用提供的连接字符串将您的应用程序连接到 SQL 数据库。
 - **防火墙规则：** 配置防火墙规则以允许从您的应用程序或本地机器访问。
 3. **管理数据库：**
 - **备份和恢复：** 使用自动备份和时间点恢复来保护您的数据。
 - **扩展：** 根据性能需求上下扩展数据库。
 4. **监控和性能调优：**
 - **查询性能洞察：** 监控和优化查询性能。
 - **自动调优：** 启用自动调优功能以提高性能。

使用 Kusto 查询语言 (KQL) 查询日志

Kusto 查询语言 (KQL) 用于查询 Azure Monitor 日志，提供对日志数据的强大洞察。

1. 基本 KQL 查询：

```
// 从特定表中检索记录
LogTableName
| where TimeGenerated > ago(1h)
| project TimeGenerated, Level, Message
```

2. 过滤和聚合数据：

```
LogTableName
| where TimeGenerated > ago(1h) and Level == "Error"
| summarize Count=count() by bin(TimeGenerated, 5m)
```

3. 连接表：

```
Table1
| join kind=inner (Table2) on $left.UserId == $right.UserId
| project Table1.TimeGenerated, Table1.Message, Table2.AdditionalInfo
```

4. 基于查询创建警报：

- 在 Azure 门户中，导航到 Log Analytics 工作区。
- 点击 日志 并输入您的 KQL 查询。
- 点击 新建警报规则，根据查询结果创建警报。

设置主动监控的警报

Azure 警报帮助您及时了解资源的健康和性能。

1. 创建警报：

- 指标警报：根据 CPU 使用率、内存使用率和响应时间等指标设置警报。
- 日志警报：使用 KQL 创建基于日志搜索查询的警报。

2. 配置操作：

- 操作组：定义操作组以指定通知对象和方式（电子邮件、短信、Webhook）。
- 集成：与 ServiceNow 等 ITSM 工具集成，实现自动化事件管理。

3. 响应警报：

- 仪表板：设置 Azure 仪表板，提供警报的集中视图。
- 自动化：使用 Azure Automation 自动响应某些警报。

结论

通过利用 Azure 订阅、AKS、Application Insights、虚拟机、事件中心、API 管理服务和 SQL 数据库，我们构建了一个可扩展、健壮和受监控的应用程序基础设施。Azure 的全面工具套件确保我们能够高效地部署、管理和监控我们的应用程序。此设置不仅提高了应用程序性能，还为我们提供了维护和优化资源所需的洞察。