

# 在 Azure 上構建可擴展的應用程式

此博客文章是在 ChatGPT-4o 的協助下撰寫的。

---

## 目錄

- 簡介
- 開始使用 Azure 訂閱
- 使用 Azure Kubernetes Service (AKS) 部署應用程式
  - 創建和管理 AKS 集群
  - 部署應用程式
- 從 Pod 中獲取日誌
- 使用 Azure Application Insights 進行監控和診斷
- 利用 Azure 虛擬機 (VMs)
- 使用 Azure Event Hubs 進行實時數據攝取
- 使用 Azure API 管理服務管理 API
- 利用 Azure SQL 數據庫
- 使用 Kusto 查詢語言 (KQL) 查詢日誌
- 設置警報以進行主動監控
- 結論

## 簡介

在雲計算的世界中，Microsoft Azure 作為一個強大的平台，用於構建、部署和管理應用程式。在我們最近的項目中，我們利用了多種 Azure 服務，包括 Azure 訂閱、Azure Kubernetes Service (AKS)、Application Insights、虛擬機 (VMs)、Event Hubs、API 管理服務和 SQL 數據庫，以創建一個可擴展且受監控的應用程式基礎設施。本博客文章概述了我們的方法、使用的工具、最佳實踐，以及管理集群、獲取日誌和查詢日誌的詳細步驟。

## 開始使用 Azure 訂閱

Azure 訂閱是您訪問 Azure 服務的入口。它作為一個容器，包含您所有的資源，如虛擬機、數據庫和 Kubernetes 集群。

### 1. 設置 Azure 訂閱：

- 註冊：如果您沒有 Azure 帳戶，請先在 Azure 門戶註冊。
- 創建訂閱：導航到“訂閱”部分並創建一個新訂閱。這將是您的計費和管理容器。

### 2. 資源組織：

- 資源組：根據其生命週期和管理標準將資源組織到資源組中。
- 標籤：使用標籤進行額外的元數據和更輕鬆的資源管理和計費。

## 使用 Azure Kubernetes Service (AKS) 部署應用程式

Azure Kubernetes Service (AKS) 是一個託管的 Kubernetes 服務，簡化了容器化應用程式的部署、管理和擴展。

### 創建和管理 AKS 集群

#### 1. 在 Azure 門戶中創建 AKS 集群：

- 設置：在 Azure 門戶中搜索 AKS 並創建一個新的 Kubernetes 集群。
- 配置：選擇您的集群大小，配置節點池，並設置網絡。
- 認證：使用 Azure Active Directory (AAD) 進行安全的訪問控制。
- 監控：在設置過程中啟用監控和日誌記錄。

#### 2. 使用 Azure CLI 創建 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：

- 擴展集群：

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

- 升級集群：

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

## 部署應用程式

1. 使用 Kubernetes 清單：為您的部署、服務和其他 Kubernetes 對象編寫 YAML 文件。

```
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 部署：

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

3. Helm Charts：使用 Helm 管理 Kubernetes 應用程式和版本控制。

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

## 從 Pod 中獲取日誌

### 1. 附加到 Pod 並獲取日誌：

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

- 要流式傳輸日誌：

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

### 2. 使用 Sidecar 進行日誌記錄：

- 在您的 Pod 規範中創建一個日誌記錄 sidecar 容器，以將日誌發送到集中式日誌記錄服務。

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

## 使用 Azure Application Insights 進行監控和診斷

Application Insights 為您的應用程式提供了強大的監控和診斷功能。

### 1. 設置 Application Insights：

- 集成：將 Application Insights SDK 添加到您的應用程式代碼中。
- 檢測密鑰：使用來自 Application Insights 資源的檢測密鑰配置您的應用程式。

### 2. 跟踪性能：

- 指標：監控響應時間、失敗率和應用程式依賴項。
- 實時指標流：查看實時性能指標以獲得即時見解。

### 3. 診斷和故障排除：

- 應用程式地圖：可視化依賴項並識別性能瓶頸。
- 事務診斷：使用分佈式跟踪來跟踪跨服務的請求。

## 利用 Azure 虛擬機 (VMs)

Azure 虛擬機提供了運行未容器化的自定義應用程式和服務的靈活性。

### 1. 配置虛擬機：

- 創建虛擬機：在 Azure 門戶中創建新的虛擬機，並選擇適當的大小和操作系統。
- 網絡配置：設置虛擬網絡、子網和安全組以控制流量。

### 2. 配置虛擬機：

- 軟件安裝：安裝所需的軟件和依賴項。
- 安全性：定期應用補丁和更新，配置防火牆，並使用網絡安全組 (NSGs)。

### 3. 管理虛擬機：

- 備份和恢復：使用 Azure 備份進行虛擬機備份。
- 監控：使用 Azure 監控器監控虛擬機性能。

## 使用 Azure Event Hubs 進行實時數據攝取

Azure Event Hubs 是一個大數據流平台和事件攝取服務，能夠每秒接收和處理數百萬個事件。

### 1. 設置 Event Hubs：

- 創建 Event Hub 命名空間：在 Azure 門戶中創建一個 Event Hub 命名空間以容納您的 Event Hubs。
- 創建 Event Hubs：在命名空間內創建一個或多個 Event Hubs 以捕獲您的數據流。

### 2. 攝取數據：

- 生產者：配置您的應用程式或服務以使用多種語言的 SDK（例如 .NET、Java、Python）將事件發送到 Event Hubs。
- 分區：使用分區來擴展事件處理，確保高吞吐量和並行性。

### 3. 處理事件：

- 消費者：使用消費者組來讀取和處理事件。Azure 提供了多種處理選項，包括 Azure Stream Analytics、Azure Functions 和使用 Event Hubs SDK 的自定義處理。

### 4. 監控 Event Hubs：

- 指標：通過 Azure 門戶監控吞吐量、延遲和事件處理指標。
- 警報：設置警報以通知您任何問題，例如高延遲或丟失的消息。

## 使用 Azure API 管理服務管理 API

Azure API 管理服務提供了一種為現有後端服務創建一致且現代 API 網關的方法。

1. 設置 API 管理：
  - 創建 API 管理服務：在 Azure 門戶中搜索 API 管理並創建一個新服務。
  - 配置 API：從 OpenAPI 規範、Azure Functions 或其他後端定義和導入 API。
2. 保護 API：
  - 認證和授權：使用 OAuth2、JWT 驗證和其他機制來保護您的 API。
  - 速率限制和節流：實施策略以保護您的 API 免受濫用。
3. 監控和分析：
  - API 洞察：跟踪使用情況、監控性能並分析日誌。
  - 開發者門戶：提供一個門戶供開發者發現和使用您的 API。
4. 管理生命週期：
  - 版本控制和修訂：無縫管理不同版本和修訂的 API。
  - 策略管理：應用策略以轉換、驗證和路由請求和響應。

## 利用 Azure SQL 數據庫

Azure SQL 數據庫是一個完全託管的關係數據庫，具有內置智能、高可用性和可擴展性。

1. 設置 Azure SQL 數據庫：
  - 創建 SQL 數據庫：在 Azure 門戶中導航到 SQL 數據庫並創建一個新數據庫。
  - 配置數據庫：設置數據庫大小、性能級別並配置網絡設置。
2. 連接到 SQL 數據庫：
  - 連接字符串：使用提供的連接字符串將您的應用程式連接到 SQL 數據庫。
  - 防火牆規則：配置防火牆規則以允許從您的應用程式或本地機器訪問。
3. 管理數據庫：
  - 備份和恢復：使用自動備份和時間點恢復來保護您的數據。
  - 擴展：根據您的性能需求擴展數據庫。
4. 監控和性能調優：
  - 查詢性能洞察：監控和優化查詢性能。
  - 自動調優：啟用自動調優功能以提高性能。

## 使用 Kusto 查詢語言 (KQL) 查詢日誌

Kusto 查詢語言 (KQL) 用於查詢 Azure 監控器日誌，為您的日誌數據提供強大的洞察力。

### 1. 基本 KQL 查詢：

```
// 從特定表中檢索記錄  
LogTableName  
| where TimeGenerated > ago(1h)  
| project TimeGenerated, Level, Message
```

### 2. 過濾和聚合數據：

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

### 3. 連接表：

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

### 4. 基於查詢創建警報：

- 在 Azure 門戶中，導航到 Log Analytics 工作區。
- 點擊 Logs 並輸入您的 KQL 查詢。
- 點擊 New alert rule 以基於查詢結果創建警報。

## 設置警報以進行主動監控

Azure 警報幫助您了解資源的健康狀況和性能。

### 1. 創建警報：

- 指標警報：基於 CPU 使用率、內存使用率和響應時間等指標設置警報。
- 日誌警報：使用 KQL 創建基於日誌搜索查詢的警報。

### 2. 配置操作：

- 操作組：定義操作組以指定誰會收到通知以及如何收到通知（電子郵件、短信、webhook）。
- 集成：與 ITSM 工具（如 ServiceNow）集成以進行自動化事件管理。

### 3. 響應警報：

- 儀表板：設置 Azure 儀表板以提供警報的集中視圖。
- 自動化：使用 Azure 自動化自動響應某些警報。

## 結論

通過利用 Azure 訂閱、AKS、Application Insights、虛擬機、Event Hubs、API 管理服務和 SQL 數據庫，我們構建了一個可擴展、強大且受監控的應用程式基礎設施。Azure 全面的工具套件確保我們能夠高效地部署、管理和監控我們的應用程式。這種設置不僅提高了我們的應用程式性能，還為我們提供了維護和優化資源所需的洞察力。