

Azure でスケーラブルなアプリケーションを構築する

このブログ記事は、ChatGPT-4o の助けを借りて書かれました。

目次

- はじめに
- Azure サブスクリプションの開始
- Azure Kubernetes Service (AKS) を使用したアプリケーションのデプロイ
 - AKS クラスターの作成と管理
 - アプリケーションのデプロイ
- Pod からのログ取得
- Azure Application Insights を使用した監視と診断
- Azure Virtual Machines (VMs) の活用
- Azure Event Hubs を使用したリアルタイムデータインジェスト
- Azure API Management Services を使用した API の管理
- Azure SQL Databases の活用
- Kusto Query Language (KQL) を使用したログのクエリ
- プロアクティブな監視のためのアラート設定
- 結論

はじめに

クラウドコンピューティングの世界において、Microsoft Azure はアプリケーションの構築、デプロイ、管理のための堅牢なプラットフォームとして際立っています。私たちの最近のプロジェクトでは、Azure Subscription、Azure Kubernetes Service (AKS)、Application Insights、Virtual Machines (VMs)、Event Hubs、API Management Services、SQL Databases など、いくつかの Azure サービスを活用して、スケーラブルで監視可能なアプリケーションインフラストラクチャを構築しました。このブログ記事では、私たちのアプローチ、使用したツール、ベストプラクティス、クラスターの管理、ログの取得、ログのクエリに関する詳細な手順を概説します。

Azure サブスクリプションの始め方

Azure サブスクリプションは、Azure サービスにアクセスするための入り口です。仮想マシン、データベース、Kubernetes クラスターなど、すべてのリソースを保持するコンテナとして機能します。

1. Azure サブスクリプションの設定:

- ・サインアップ: Azure アカウントをお持ちでない場合は、Azure ポータルでサインアップから始めてください。
- ・サブスクリプションの作成: 「サブスクリプション」セクションに移動し、新しいサブスクリプションを作成します。これが請求と管理のコンテナとなります。

2. リソースの整理:

- ・リソースグループ: リソースをそのライフサイクルと管理基準に基づいてリソースグループに整理します。
- ・タグ: 追加のメタデータやリソース管理、課金を容易にするためにタグを使用します。

Azure Kubernetes Service (AKS) を使用したアプリケーションのデプロイ

Azure Kubernetes Service (AKS) は、コンテナ化されたアプリケーションのデプロイ、管理、スケーリングを簡素化するマネージド Kubernetes サービスです。

AKS クラスターの作成と管理

1. Azure Portal での AKS クラスターの作成:

- ・セットアップ: Azure ポータルで AKS を検索し、新しい Kubernetes クラスターを作成します。
- ・構成: クラスターのサイズを選択し、ノードプールを構成し、ネットワークを設定します。
- ・認証: セキュアなアクセス制御のために Azure Active Directory (AAD) を使用します。
- ・監視: セットアッププロセス中に監視とログ記録を有効にします。

2. Azure CLI を使用した AKS クラスターの作成:

```
az aks create \
--resource-group myResourceGroup \
--name myAKSCluster \
```

```
--node-count 3 \
--enable-addons monitoring \
--generate-ssh-keys
```

3. AKS クラスターの管理:

- クラスターのスケーリング:

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

- クラスターのアップグレード:

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

アプリケーションのデプロイ

1. Kubernetes マニフェストの使用: デプロイメント、サービス、その他の Kubernetes オブジェクトの YAML ファイルを作成します。

```
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. kubectl を使用したデプロイ:

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

3. Helm Charts: Kubernetes アプリケーションとバージョン管理には Helm を使用します。

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

Pod からのログ取得

1. Pod にアタッチしてログを取得する:

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

- ログをストリーミングするには:

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

2. サイドカーを使用したロギング:

- ポッド仕様にロギング用のサイドカーコンテナを作成し、ログを集中ロギングサービスに送信します。

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

Azure Application Insights による監視と診断

Application Insights は、アプリケーションの強力な監視および診断機能を提供します。

1. Application Insights の設定:

- ・統合: アプリケーションコードに Application Insights SDK を追加します。
- ・インストルメンテーションキー: Application Insights リソースから取得したインストルメンテーションキーをアプリケーションに設定します。

2. パフォーマンスの追跡:

- ・メトリクス: 応答時間、障害率、アプリケーションの依存関係を監視します。
- ・ライブメトリクストリーム: リアルタイムのパフォーマンスマetriクスを表示し、即座に洞察を得ることができます。

3. 診断とトラブルシューティング:

- ・アプリケーションマップ: 依存関係を可視化し、パフォーマンスのボトルネックを特定します。
- ・トランザクション診断: 分散トレーシングを使用して、サービス間のリクエストを追跡します。

Azure Virtual Machines (VMs) の活用

Azure VMs は、コンテナ化されていないカスタムアプリケーションやサービスを実行する柔軟性を提供します。

1. 仮想マシンのプロビジョニング:

- ・VM の作成: Azure ポータルで新しい仮想マシンを作成し、適切なサイズとオペレーティングシステムを選択します。
- ・ネットワーク設定: 仮想ネットワーク、サブネット、セキュリティグループを設定してトラフィックを制御します。

2. VM の設定:

- ・ソフトウェアのインストール: 必要なソフトウェアと依存関係をインストールします。
- ・セキュリティ: 定期的にパッチとアップデートを適用し、ファイアウォールを設定し、ネットワークセキュリティグループ (NSGs) を使用します。

3. VM の管理:

- ・バックアップと復元: Azure Backup を使用して VM のバックアップを行います。
- ・監視: Azure Monitor を使用して VM のパフォーマンスを監視します。

Azure Event Hubs を使用したリアルタイムデータインジェスト

Azure Event Hubs は、ビッグデータストリーミングプラットフォームであり、イベントインジェクションサービスです。毎秒数百万のイベントを受信し、処理することが可能です。

1. Event Hubs の設定:

- Event Hub 名前空間の作成: Azure ポータルで、Event Hub を収容するための Event Hub 名前空間を作成します。
- Event Hubs の作成: 名前空間内に、データストリームをキャプチャするための 1 つ以上の Event Hubs を作成します。

2. データの取り込み:

- プロデューサー: アプリケーションやサービスを設定して、複数の言語（例: .NET、Java、Python）で利用可能な SDK を使用してイベントを Event Hubs に送信します。
- パーティション: パーティションを使用してイベント処理をスケールし、高いスループットと並列処理を確保します。

3. イベントの処理:

- コンシューマー: イベントを読み取り、処理するためにコンシューマーグループを使用します。Azure は、Azure Stream Analytics、Azure Functions、および Event Hubs SDK を使用したカスタム処理など、いくつかの処理オプションを提供しています。

4. Event Hubs の監視:

- メトリクス: Azure ポータルを通じてスループット、レイテンシ、イベント処理メトリクスを監視します。
- アラート: 高レイテンシやメッセージのドロップなどの問題を通知するアラートを設定します。

Azure API Management サービスでの API 管理

Azure API Management Services は、既存のバックエンドサービスに対して一貫性のあるモダンな API ゲートウェイを作成する方法を提供します。

1. API Management の設定:

- API Management サービスの作成: Azure ポータルで API Management を検索し、新しいサービスを作成します。
- API の設定: OpenAPI 仕様、Azure Functions、またはその他のバックエンドから API を定義してインポートします。

2. API のセキュリティ確保:

- ・認証と認可: OAuth2、JWT 検証などのメカニズムを使用して API を保護します。
- ・レート制限とスロットリング: API が悪用されないようにするためのポリシーを実装します。

3. モニタリングと分析 :

- ・API Insights: 使用状況の追跡、パフォーマンスの監視、ログの分析を行います。
- ・デベロッパー ポータル: 開発者が API を発見し利用するためのポータルを提供します。

4. ライフサイクルの管理:

- ・バージョン管理とリビジョン: API の異なるバージョンやリビジョンをシームレスに管理します。
- ・ポリシー管理: リクエストの変換、検証、ルーティングのためのポリシーを適用します。

そしてレスポンス。

Azure SQL データベースの活用

Azure SQL Database は、組み込みのインテリジェンス、高可用性、スケーラビリティを備えた完全管理型のリレーショナルデータベースです。

1. Azure SQL Database の設定:

- ・SQL Database の作成: Azure ポータルで SQL Databases に移動し、新しいデータベースを作成します。
- ・データベースの設定: データベースのサイズ、パフォーマンスレベルを設定し、ネットワーク設定を構成します。

2. SQL データベースへの接続:

- ・接続文字列: 提供された接続文字列を使用して、アプリケーションを SQL データベースに接続します。
- ・ファイアウォールルール: アプリケーションまたはローカルマシンからのアクセスを許可するようにファイアウォールルールを設定します。

3. データベースの管理:

- ・バックアップと復元: 自動バックアップとポイントインタイムリカバリを使用してデータを保護します。
- ・スケーリング: パフォーマンスのニーズに応じてデータベースをスケールアップまたはスケールダウンします。

4. 監視とパフォーマンスチューニング:

- クエリパフォーマンスインサイト: クエリのパフォーマンスを監視し、最適化します。
- 自動チューニング: パフォーマンスを向上させるために自動チューニング機能を有効にします。

Kusto Query Language (KQL) を使用したログのクエリ

Kusto Query Language (KQL) は、Azure Monitor ログをクエリするために使用され、ログデータに対する強力な洞察を提供します。

1. 基本的な KQL クエリ:

```
// 特定のテーブルからレコードを取得する
LogTableName
| where TimeGenerated > ago(1h)
| project TimeGenerated, Level, Message
```

2. データのフィルタリングと集計:

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

3. テーブルの結合:

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

4. クエリに基づくアラートの作成:

- Azure ポータルで、Log Analytics ワークスペースに移動します。
- ログをクリックし、KQL クエリを入力します。
- クエリ結果に基づいてアラートを作成するには、新しいアラートルールをクリックします。

プロアクティブな監視のためのアラート設定

Azure アラートは、リソースの健全性とパフォーマンスに関する情報を常に把握するのに役立ちます。

1. アラートの作成:

- メトリックアラート: CPU 使用率、メモリ使用率、応答時間などのメトリックに基づいてアラートを設定します。
- ログアラート: KQL を使用したログ検索クエリに基づいてアラートを作成します。

2. アクションの設定:

- アクショングループ: 誰がどのように通知を受けるか（メール、SMS、ウェブフック）を指定するためにアクショングループを定義します。
- 統合: ServiceNow などの ITSM ツールと統合して、自動化されたインシデント管理を実現します。

3. アラートへの対応:

- ダッシュボード: Azure ダッシュボードを設定して、アラートの一元化されたビューを提供します。
- 自動化: Azure Automation を使用して、特定のアラートに自動的に対応します。

結論

Azure Subscription、AKS、Application Insights、Virtual Machines、Event Hubs、API Management Services、SQL Databases を活用することで、スケーラブルで堅牢かつ監視可能なアプリケーションインフラストラクチャを構築しました。Azure の包括的なツールセットにより、アプリケーションのデプロイ、管理、監視を効率的に行うことができました。このセットアップは、アプリケーションのパフォーマンスを向上させるだけでなく、リソースを積極的に維持・最適化するために必要な洞察を提供しました。