

Erstellung einer skalierbaren Anwendung auf Azure

Dieser Blogbeitrag wurde mit Unterstützung von ChatGPT-4o verfasst.

Inhaltsverzeichnis

- Einführung
- Erste Schritte mit einem Azure-Abonnement
- Bereitstellen von Anwendungen mit Azure Kubernetes Service (AKS)
 - Erstellen und Verwalten eines AKS-Clusters
 - Bereitstellen von Anwendungen
- Abrufen von Logs aus Pods
- Überwachung und Diagnose mit Azure Application Insights
- Nutzung von Azure Virtual Machines (VMs)
- Echtzeit-Datenerfassung mit Azure Event Hubs
- Verwalten von APIs mit Azure API Management Services
- Nutzung von Azure SQL-Datenbanken
- Abfragen von Logs mit der Kusto Query Language (KQL)
- Einrichten von Warnungen für proaktive Überwachung
- Fazit

Einführung

In der Welt des Cloud Computing sticht Microsoft Azure als eine robuste Plattform für das Erstellen, Bereitstellen und Verwalten von Anwendungen hervor. In unserem jüngsten Projekt haben wir mehrere Azure-Dienste genutzt, darunter Azure Subscription, Azure Kubernetes Service (AKS), Application Insights, Virtual Machines (VMs), Event Hubs, API Management Services und SQL-Datenbanken, um eine skalierbare und überwachte Anwendungsinfrastruktur zu schaffen. Dieser Blogbeitrag skizziert unseren Ansatz, die verwendeten Tools, Best Practices und detaillierte Schritte zur Verwaltung von Clustern, zum Abrufen von Protokollen und zum Abfragen von Protokollen.

Erste Schritte mit einem Azure-Abonnement

Ein Azure-Abonnement ist Ihr Zugangspunkt zu Azure-Diensten. Es fungiert als Container, der alle Ihre Ressourcen wie virtuelle Maschinen, Datenbanken und Kubernetes-Cluster enthält.

1. Einrichten eines Azure-Abonnements:

- Registrierung: Wenn Sie noch kein Azure-Konto haben, beginnen Sie mit der Registrierung im Azure-Portal.
- Abonnement erstellen: Navigieren Sie zum Abschnitt "Abonnements" und erstellen Sie ein neues Abonnement. Dies wird Ihr Abrechnungs- und Verwaltungscontainer sein.

2. Ressourcenorganisation:

- Ressourcengruppen: Organisieren Sie Ihre Ressourcen in Ressourcengruppen basierend auf ihrem Lebenszyklus und Managementkriterien.
- Tags: Verwenden Sie Tags für zusätzliche Metadaten und eine einfachere Ressourcenverwaltung und Abrechnung.

Bereitstellen von Anwendungen mit Azure Kubernetes Service (AKS)

Azure Kubernetes Service (AKS) ist ein verwalteter Kubernetes-Dienst, der die Bereitstellung, Verwaltung und Skalierung von containerisierten Anwendungen vereinfacht.

Erstellen und Verwalten eines AKS-Clusters

1. Erstellen eines AKS-Clusters im Azure-Portal:

- Einrichtung: Suchen Sie im Azure-Portal nach AKS und erstellen Sie einen neuen Kubernetes-Cluster.
- Konfiguration: Wählen Sie die Clustergröße, konfigurieren Sie Node Pools und richten Sie das Netzwerk ein.
- Authentifizierung: Verwenden Sie Azure Active Directory (AAD) für eine sichere Zugriffskontrolle.
- Überwachung: Aktivieren Sie die Überwachung und Protokollierung während des Einrichtungsprozesses.

2. Erstellen eines AKS-Clusters mit der Azure CLI:

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

3. Verwaltung Ihres AKS-Clusters:

- Skalieren des Clusters:

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

- Aktualisieren des Clusters:

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

Anwendungen bereitstellen

1. Verwendung von Kubernetes-Manifesten: Schreiben Sie YAML-Dateien für Ihre Deployments, Services und andere Kubernetes-Objekte.

```
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. Bereitstellen mit kubectl:

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

3. Helm Charts: Verwenden Sie Helm zur Verwaltung von Kubernetes-Anwendungen und zur Versionskontrolle.

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

Abrufen von Logs aus Pods

1. Anhängen an einen Pod und Abrufen von Logs:

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

- Um Logs zu streamen:

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

2. Verwendung eines Sidecars für die Protokollierung:

- Erstellen Sie einen Sidecar-Container für die Protokollierung in Ihrer Pod-Spezifikation, um Protokolle an einen zentralisierten Logging-Dienst zu senden.

```

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

```

Überwachung und Diagnose mit Azure Application Insights

Application Insights bietet leistungsstarke Überwachungs- und Diagnosefunktionen für Ihre Anwendungen.

1. Einrichten von Application Insights:

- Integration: Fügen Sie das Application Insights SDK zu Ihrem Anwendungscode hinzu.
- Instrumentierungsschlüssel: Konfigurieren Sie Ihre Anwendung mit dem Instrumentierungsschlüssel aus Ihrer Application Insights-Ressource.

2. Leistungsverfolgung:

- Metriken: Überwachen Sie Antwortzeiten, Ausfallraten und Anwendungsabhängigkeiten.
- Live-Metriken-Stream: Sehen Sie sich Echtzeit-Leistungsmetriken an, um sofortige Einblicke zu erhalten.

3. Diagnose und Fehlerbehebung:

- Anwendungsübersicht: Visualisieren Sie Abhängigkeiten und identifizieren Sie Leistungsengpässe.
- Transaktionsdiagnose: Nutzen Sie verteilte Ablaufverfolgung, um Anfragen über Dienste hinweg zu verfolgen.

Nutzung von Azure Virtual Machines (VMs)

Azure VMs bieten die Flexibilität, benutzerdefinierte Anwendungen und Dienste auszuführen, die nicht containerisiert sind.

1. Bereitstellung von virtuellen Maschinen:

- Erstellen von VMs: Im Azure-Portal neue virtuelle Maschinen erstellen und die passende Größe sowie das Betriebssystem auswählen.
- Netzwerkkonfiguration: Virtuelle Netzwerke, Subnetze und Sicherheitsgruppen einrichten, um den Datenverkehr zu steuern.

2. Konfiguration von VMs:

- Software-Installation: Installieren Sie die erforderliche Software und Abhängigkeiten.
- Sicherheit: Wenden Sie regelmäßig Patches und Updates an, konfigurieren Sie Firewalls und verwenden Sie Network Security Groups (NSGs).

3. Verwaltung von VMs:

- Backup und Wiederherstellung: Verwenden Sie Azure Backup für VM-Sicherungen.
- Überwachung: Überwachen Sie die VM-Leistung mit Azure Monitor.

Echtzeit-Datenerfassung mit Azure Event Hubs

Azure Event Hubs ist eine Big-Data-Streaming-Plattform und ein Ereigniserfassungsdienst, der in der Lage ist, Millionen von Ereignissen pro Sekunde zu empfangen und zu verarbeiten.

1. Einrichten von Event Hubs:

- Erstellen eines Event Hub-Namespace: Erstellen Sie im Azure-Portal einen Event Hub-Namespace, um Ihre Event Hubs zu beherbergen.
- Erstellen von Event Hubs: Erstellen Sie innerhalb des Namespaces einen oder mehrere Event Hubs, um Ihre Datenströme zu erfassen.

2. Datenerfassung:

- Produzenten: Konfigurieren Sie Ihre Anwendung oder Dienste so, dass sie Ereignisse über die für mehrere Sprachen verfügbaren SDKs (z. B. .NET, Java, Python) an Event Hubs senden.
- Partitionen: Nutzen Sie Partitionen, um die Ereignisverarbeitung zu skalieren und so einen hohen Durchsatz und Parallelität zu gewährleisten.

3. Verarbeiten von Ereignissen:

- Verbraucher: Verwenden Sie Verbrauchergruppen, um Ereignisse zu lesen und zu verarbeiten. Azure bietet mehrere Optionen für die Verarbeitung, darunter Azure Stream Analytics, Azure Functions und die benutzerdefinierte Verarbeitung mit dem Event Hubs SDK.

4. Überwachung von Event Hubs:

- Metriken: Überwachen Sie den Durchsatz, die Latenz und die Ereignisverarbeitungsmetriken über das Azure-Portal.
- Warnungen: Richten Sie Warnungen ein, um Sie über Probleme wie hohe Latenz oder verlorene Nachrichten zu benachrichtigen.

Verwalten von APIs mit Azure API Management Services

Azure API Management Services bieten eine Möglichkeit, konsistente und moderne API-Gateways für bestehende Backend-Dienste zu erstellen.

1. Einrichten der API-Verwaltung:

- Erstellen eines API Management-Dienstes: Suchen Sie im Azure-Portal nach API Management und erstellen Sie einen neuen Dienst.
- Konfigurieren von APIs: Definieren und importieren Sie APIs aus OpenAPI-Spezifikationen, Azure Functions oder anderen Backends.

2. Sicherung von APIs:

- Authentifizierung und Autorisierung: Verwenden Sie OAuth2, JWT-Validierung und andere Mechanismen, um Ihre APIs zu sichern.
- Ratenbegrenzung und Drosselung: Implementieren Sie Richtlinien, um Ihre APIs vor Missbrauch zu schützen.

3. Überwachung und Analysen:

- API-Einblicke: Nutzung verfolgen, Leistung überwachen und Protokolle analysieren.
- Entwicklerportal: Bieten Sie ein Portal an, in dem Entwickler Ihre APIs entdecken und nutzen können.

4. Lebenszyklusverwaltung:

- Versionierung und Revisionen: Verwalten Sie verschiedene Versionen und Revisionen Ihrer APIs nahtlos.
- Richtlinienverwaltung: Wenden Sie Richtlinien für die Transformation, Validierung und Weiterleitung von Anfragen an.

und Antworten.

Nutzung von Azure SQL-Datenbanken

Azure SQL-Datenbank ist eine vollständig verwaltete relationale Datenbank mit integrierter Intelligenz, hoher Verfügbarkeit und Skalierbarkeit.

1. Einrichten der Azure SQL-Datenbank:

- Erstellen einer SQL-Datenbank: Navigieren Sie im Azure-Portal zu SQL-Datenbanken und erstellen Sie eine neue Datenbank.
- Konfigurieren der Datenbank: Legen Sie die Datenbankgröße, das Leistungs niveau fest und konfigurieren Sie die Netzwerkeinstellungen.

2. Verbindung zur SQL-Datenbank herstellen:

- Verbindungszeichenfolgen: Verwenden Sie die bereitgestellten Verbindungszeichenfolgen, um Ihre Anwendung mit der SQL-Datenbank zu verbinden.
- Firewall-Regeln: Konfigurieren Sie Firewall-Regeln, um den Zugriff von Ihrer Anwendung oder Ihrem lokalen Rechner zu ermöglichen.

3. Verwaltung der Datenbank:

- Backup und Wiederherstellung: Nutzen Sie automatisierte Backups und Point-in-Time-Wiederherstellung, um Ihre Daten zu schützen.
- Skalierung: Skalieren Sie die Datenbank je nach Ihren Leistungsanforderungen hoch oder herunter.

4. Überwachung und Leistungsoptimierung:

- **Query Performance Insights:** Überwachen und optimieren Sie die Abfrageleistung.
- **Automatic Tuning:** Aktivieren Sie automatische Optimierungsfunktionen, um die Leistung zu verbessern.

Abfragen von Protokollen mit der Kusto Query Language (KQL)

Die Kusto Query Language (KQL) wird verwendet, um Azure Monitor Logs abzufragen und bietet leistungsstarke Einblicke in Ihre Protokolldaten.

1. Grundlegende KQL-Abfrage:

```
// Datensätze aus einer bestimmten Tabelle abrufen
LogTableName
| where TimeGenerated > ago(1h)
| project TimeGenerated, Level, Message
```

2. Filtern und Aggregieren von Daten:

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

3. Tabellen verbinden:

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

4. Erstellen von Warnungen basierend auf Abfragen:

- Navigieren Sie im Azure-Portal zum Log Analytics-Arbeitsbereich.
- Klicken Sie auf Logs und geben Sie Ihre KQL-Abfrage ein.
- Klicken Sie auf New alert rule, um eine Warnung basierend auf den Abfrageergebnissen zu erstellen.

Einrichten von Warnmeldungen für proaktive Überwachung

Azure Alerts helfen Ihnen, über den Zustand und die Leistung Ihrer Ressourcen informiert zu bleiben.

1. Erstellen von Warnungen:

- Metrik-Warnungen: Richten Sie Warnungen basierend auf Metriken wie CPU-Auslastung, Speichernutzung und Antwortzeiten ein.
- Log-Warnungen: Erstellen Sie Warnungen basierend auf Log-Suchabfragen mit KQL.

2. Konfiguration von Aktionen:

- Aktionsgruppen: Definieren Sie Aktionsgruppen, um festzulegen, wer benachrichtigt wird und auf welche Weise (E-Mail, SMS, Webhook).
- Integration: Integrieren Sie ITSM-Tools wie ServiceNow für ein automatisiertes Incident-Management.

3. Reaktion auf Benachrichtigungen:

- Dashboards: Richten Sie Azure-Dashboards ein, um eine zentrale Ansicht der Benachrichtigungen bereitzustellen.
- Automatisierung: Nutzen Sie Azure Automation, um automatisch auf bestimmte Benachrichtigungen zu reagieren.

Fazit

Durch die Nutzung von Azure Subscription, AKS, Application Insights, Virtual Machines, Event Hubs, API Management Services und SQL-Datenbanken haben wir eine skalierbare, robuste und überwachte Anwendungsinfrastruktur aufgebaut. Die umfassende Suite von Tools von Azure stellte sicher, dass wir unsere Anwendungen effizient bereitstellen, verwalten und überwachen konnten. Dieses Setup verbesserte nicht nur die Leistung unserer Anwendungen, sondern verschaffte uns auch die Einblicke, die wir benötigten, um unsere Ressourcen proaktiv zu pflegen und zu optimieren.