

Création d'une Application Évolutive sur Azure

Cet article de blog a été rédigé avec l'assistance de ChatGPT-4o.

Table des matières

- Introduction
- Démarrage avec un abonnement Azure
- Déploiement d'applications avec Azure Kubernetes Service (AKS)
 - Création et gestion d'un cluster AKS
 - Déploiement d'applications
- Récupération des logs des pods
- Surveillance et diagnostics avec Azure Application Insights
- Utilisation des machines virtuelles Azure (VMs)
- Ingestion de données en temps réel avec Azure Event Hubs
- Gestion des API avec les services Azure API Management
- Utilisation des bases de données Azure SQL
- Interrogation des logs avec le langage de requête Kusto (KQL)
- Configuration d'alertes pour une surveillance proactive
- Conclusion

Introduction

Dans le domaine du cloud computing, Microsoft Azure se distingue comme une plateforme robuste pour la création, le déploiement et la gestion d'applications. Dans notre projet récent, nous avons exploité plusieurs services Azure, notamment Azure Subscription, Azure Kubernetes Service (AKS), Application Insights, Virtual Machines (VMs), Event Hubs, API Management Services et SQL Databases, pour créer une infrastructure d'application scalable et surveillée. Cet article de blog décrit notre approche, les outils utilisés, les meilleures pratiques, ainsi que les étapes détaillées pour la gestion des clusters, la récupération des logs et l'interrogation des logs.

Premiers pas avec un abonnement Azure

Un abonnement Azure est votre passerelle pour accéder aux services Azure. Il agit comme un conteneur qui regroupe toutes vos ressources, telles que les machines virtuelles, les bases de données et les clusters Kubernetes.

1. Configuration de l'abonnement Azure :

- **Inscription :** Si vous n'avez pas de compte Azure, commencez par vous inscrire sur le portail Azure.
- **Créer un abonnement :** Accédez à la section "Abonnements" et créez un nouvel abonnement. Celui-ci servira de conteneur pour la facturation et la gestion.

2. Organisation des ressources :

- **Groupes de ressources :** Organisez vos ressources en groupes de ressources en fonction de leur cycle de vie et de leurs critères de gestion.
- **Étiquettes :** Utilisez des étiquettes pour ajouter des métadonnées supplémentaires et faciliter la gestion des ressources et la facturation.

Déployer des applications avec Azure Kubernetes Service (AKS)

Azure Kubernetes Service (AKS) est un service Kubernetes managé qui simplifie le déploiement, la gestion et la mise à l'échelle d'applications conteneurisées.

Création et gestion d'un cluster AKS

1. Création d'un cluster AKS dans le portail Azure :

- **Configuration :** Dans le portail Azure, recherchez AKS et créez un nouveau cluster Kubernetes.
- **Configuration :** Choisissez la taille de votre cluster, configurez les pools de nœuds et paramétrez le réseau.
- **Authentification :** Utilisez Azure Active Directory (AAD) pour un contrôle d'accès sécurisé.
- **Surveillance :** Activez la surveillance et la journalisation pendant le processus de configuration.

2. Création d'un cluster AKS avec Azure CLI :

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

3. Gestion de votre cluster AKS :

- Mise à l'échelle du cluster :

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

- Mise à niveau du cluster :

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

Déploiement d'Applications

1. Utilisation des Manifests Kubernetes : Écrivez des fichiers YAML pour vos déploiements, services et autres objets Kubernetes.

```
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. Déploiement avec kubectl :

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

3. Helm Charts : Utilisez Helm pour gérer les applications Kubernetes et le contrôle de version.

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

Récupération des logs depuis les Pods

1. Attacher à un Pod et récupérer les logs :

```
kubectl logs <nom-du-pod>
```

- Pour diffuser les logs en continu :

```
kubectl logs <nom-du-pod> -f
```

2. Utilisation d'un Sidecar pour la journalisation :

- Créez un conteneur sidecar de journalisation dans la spécification de votre pod pour envoyer les journaux vers un service de journalisation centralisé.

```

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

```

Surveillance et diagnostics avec Azure Application Insights

Application Insights offre des fonctionnalités puissantes de surveillance et de diagnostic pour vos applications.

1. Configuration d'Application Insights :

- **Intégration :** Ajoutez le SDK Application Insights à votre code d'application.
- **Clé d'instrumentation :** Configurez votre application avec la clé d'instrumentation provenant de votre ressource Application Insights.

2. Suivi des performances :

- **Métriques :** Surveillez les temps de réponse, les taux d'échec et les dépendances de l'application.
- **Flux de métriques en direct :** Consultez les métriques de performance en temps réel pour des insights immédiats.

3. Diagnostic et dépannage :

- **Carte des applications :** Visualisez les dépendances et identifiez les goulots d'étranglement de performance.
- **Diagnostic des transactions :** Utilisez le traçage distribué pour suivre les requêtes à travers les services.

Utilisation des Machines Virtuelles Azure (VM)

Les machines virtuelles Azure offrent la flexibilité nécessaire pour exécuter des applications et des services personnalisés qui ne sont pas conteneurisés.

1. Provisionnement des Machines Virtuelles :

- **Créer des machines virtuelles :** Dans le portail Azure, créez de nouvelles machines virtuelles et choisissez la taille et le système d'exploitation appropriés.
- **Configuration du réseau :** Configurez des réseaux virtuels, des sous-réseaux et des groupes de sécurité pour contrôler le trafic.

2. Configuration des machines virtuelles :

- **Installation de logiciels :** Installez les logiciels et les dépendances nécessaires.
- **Sécurité :** Appliquez régulièrement les correctifs et les mises à jour, configurez les pare-feux et utilisez les groupes de sécurité réseau (NSGs).

3. Gestion des machines virtuelles :

- Sauvegarde et restauration : Utilisez Azure Backup pour les sauvegardes de machines virtuelles.
- Surveillance : Surveillez les performances des machines virtuelles à l'aide d'Azure Monitor.

Ingestion de données en temps réel avec Azure Event Hubs

Azure Event Hubs est une plateforme de streaming de données massives et un service d'ingestion d'événements capable de recevoir et de traiter des millions d'événements par seconde.

1. Configuration des Event Hubs :

- Créer un Namespace Event Hub : Dans le portail Azure, créez un namespace Event Hub pour héberger vos Event Hubs.
- Créer des Event Hubs : Au sein du namespace, créez un ou plusieurs Event Hubs pour capturer vos flux de données.

2. Ingestion des données :

- Producteurs : Configurez votre application ou services pour envoyer des événements à Event Hubs en utilisant les SDK disponibles pour plusieurs langages (par exemple, .NET, Java, Python).
- Partitions : Utilisez des partitions pour mettre à l'échelle le traitement des événements, garantissant un débit élevé et un parallélisme.

3. Traitement des événements :

- Consommateurs : Utilisez des groupes de consommateurs pour lire et traiter les événements. Azure propose plusieurs options pour le traitement, notamment Azure Stream Analytics, Azure Functions, et un traitement personnalisé à l'aide du SDK Event Hubs.

4. Surveillance des Event Hubs :

- Métriques : Surveillez le débit, la latence et les métriques de traitement des événements via le portail Azure.
- Alertes : Configurez des alertes pour vous informer de tout problème, comme une latence élevée ou des messages perdus.

Gestion des API avec Azure API Management Services

Les services Azure API Management offrent un moyen de créer des passerelles API modernes et cohérentes pour les services back-end existants.

1. Configuration de la Gestion des API :

- Créer un Service de Gestion des API : Dans le portail Azure, recherchez “API Management” et créez un nouveau service.
- Configurer les API : Définissez et importez des API à partir de spécifications OpenAPI, Azure Functions, ou d’autres backends.

2. Sécurisation des API :

- Authentification et Autorisation : Utilisez OAuth2, la validation JWT et d’autres mécanismes pour sécuriser vos API.
- Limitation et Régulation des Requêtes : Mettez en place des politiques pour protéger vos API contre les abus.

3. Surveillance et Analyse :

- API Insights : Suivez l’utilisation, surveillez les performances et analysez les journaux.
- Portail Développeur : Fournissez un portail permettant aux développeurs de découvrir et d’utiliser vos APIs.

4. Gestion du Cycle de Vie :

- **Versioning et Révisions** : Gérez facilement les différentes versions et révisions de vos APIs.
- **Gestion des Politiques** : Appliquez des politiques pour la transformation, la validation et le routage des requêtes.

et réponses.

Utilisation des bases de données Azure SQL

Azure SQL Database est une base de données relationnelle entièrement gérée dotée d’une intelligence intégrée, d’une haute disponibilité et d’une grande évolutivité.

1. Configuration d’Azure SQL Database :

- Créer une base de données SQL : Dans le portail Azure, accédez à SQL Databases et créez une nouvelle base de données.

- Configurer la base de données : Définissez la taille de la base de données, le niveau de performance et configurez les paramètres de réseau.

2. Connexion à la base de données SQL :

- Chaînes de connexion : Utilisez les chaînes de connexion fournies pour connecter votre application à la base de données SQL.
- Règles de pare-feu : Configurez les règles de pare-feu pour autoriser l'accès depuis votre application ou votre machine locale.

3. Gestion de la base de données :

- Sauvegarde et restauration : Utilisez des sauvegardes automatisées et la restauration à un point dans le temps pour protéger vos données.
- Mise à l'échelle : Augmentez ou réduisez la taille de la base de données en fonction de vos besoins de performance.

4. Surveillance et Optimisation des Performances :

- Insights sur les Performances des Requêtes : Surveillez et optimisez les performances des requêtes.
- Réglage Automatique : Activez les fonctionnalités de réglage automatique pour améliorer les performances.

Interrogation des journaux avec le langage de requête Kusto (KQL)

Le langage de requête Kusto (KQL) est utilisé pour interroger les journaux d'Azure Monitor, offrant des informations puissantes sur vos données de journal.

1. Requête KQL de base :

```
// Récupérer les enregistrements d'une table spécifique
LogTableName
| where TimeGenerated > ago(1h)
| project TimeGenerated, Level, Message
```

2. Filtrage et agrégation des données :

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


3. Jointure de tables :

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

4. Création d'alertes basées sur des requêtes :

- Dans le portail Azure, accédez à l'espace de travail Log Analytics.
- Cliquez sur **Logs** et saisissez votre requête KQL.
- Cliquez sur **New alert rule** pour créer une alerte basée sur les résultats de la requête.

Configuration des alertes pour une surveillance proactive

Les alertes Azure vous aident à rester informé sur l'état de santé et les performances de vos ressources.

1. Création d'alertes :

- **Alertes métriques** : Configurez des alertes basées sur des métriques telles que l'utilisation du CPU, l'utilisation de la mémoire et les temps de réponse.
- **Alertes de journaux** : Créez des alertes basées sur des requêtes de recherche dans les journaux en utilisant KQL.

2. Configuration des Actions :

- **Groupes d'Actions** : Définissez des groupes d'actions pour spécifier qui doit être notifié et comment (e-mail, SMS, webhook).
- **Intégration** : Intégrez avec des outils ITSM comme ServiceNow pour une gestion automatisée des incidents.

3. Réponse aux alertes :

- **Tableaux de bord** : Configurez des tableaux de bord Azure pour offrir une vue centralisée des alertes.
- **Automatisation** : Utilisez Azure Automation pour répondre automatiquement à certaines alertes.

Conclusion

En tirant parti des services Azure tels que les Abonnements Azure, AKS (Azure Kubernetes Service), Application Insights, les Machines Virtuelles, les Event Hubs, les services de gestion d'API et les bases de données SQL, nous avons construit une infrastructure d'application évolutive, robuste et surveillée. La suite complète d'outils d'Azure nous a permis de déployer, gérer et surveiller nos applications de manière efficace. Cette configuration a non seulement amélioré les performances de notre application, mais nous a également fourni les informations nécessaires pour maintenir et optimiser nos ressources de manière proactive.