□□□□□ पर एक स्केलेबल एप्लिकेशन बनाना

यह ब्लॉग पोस्ट □□□□□□□-4□ की सहायता से लिखा गया है।

विषय सूची

- □ परिचय
- □ □□□□□ सदस्यता के साथ शुरुआत
- □ □□□□□ □□□□□□□□□□ □□□□□□□ (□□□) के साथ एप्लिकेशन डिप्लॉय करना
  - □□□ क्लस्टर बनाना और प्रबंधित करना
  - एप्लिकेशन डिप्लॉय करना
- □ □□□□ से लॉग्स प्राप्त करना
- □ □□□□□ □□□□□□□□□□ □□□□□□□□ के साथ मॉनिटरिंग और डायग्नोस्टिक्स
- □ □□□□□ □□□□□□□ □□□□□□□□ (□□□) का उपयोग
- □ □□□□□ □□□□□ □□□□ के साथ रियल-टाइम डेटा इंगेशन
- □ □□□□□ □□□ □□□□□□□□□□ □□□□□□□□ के साथ □□□□ प्रबंधित करना
- □ □□□□□ □□□ □□□□□□□□□□ का उपयोग
- □ □□□□□ □□□□□ □□□□□□□□ (□□□) के साथ लॉग्स क्वेरी करना
- □ प्रोएक्टिव मॉनिटरिंग के लिए अलर्ट सेट करना
- □ निष्कर्ष

परिचय

क्लाउड कंप्यूटिंग की दुनिया में, □□□□□□□□□ □□□□□ एप्लिकेशन बनाने, तैनात करने और प्रबंधित करने के लिए एक मजबूत प्लेटफॉर्म के रूप में उभरा है। हमारे हाल के प्रोजेक्ट में, हमने □□□□□ □□□□□□□□□□□□□, □□□□□ □□□□□□□□□□ □□□□□□□ (□□□), □□□□□□□□□□□ □□□□□□□□, □□□□□□□ □□□□□□□□ (□□□), □□□□□ □□□□, □□□ □□□□□□□□□□ □□□□□□□□, और □□□ □□□□□□□□□ जैसी कई □□□□□ सेवाओं का उपयोग करके एक स्केलेबल और मॉनिटर किए गए एप्लिकेशन इंफ्रास्ट्रक्चर बनाया। यह ब्लॉग पोस्ट हमारे दृष्टिकोण, उपयोग किए गए टूल्स, सर्वोत्तम प्रथाओं, और क्लस्टर प्रबंधन, लॉग्स प्राप्त करने, और लॉग्स क्वेरी करने के विस्तृत चरणों को रेखांकित करता है।

□□□□□ सब्सक्रिप्शन के साथ शुरुआत करना

एक □□□□□ सब्सक्रिप्शन □□□□□ सेवाओं तक पहुंचने का आपका प्रवेश द्वार है। यह एक कंटेनर की तरह काम करता है जो आपके सभी संसाधनों को रखता है, जैसे कि वर्चुअल मशीन, डेटाबेस, और □□□□□□□□□□ क्लस्टर।

1. □□□□□ सब्सक्रिप्शन सेट करना:
    - साइन अप: यदि आपके पास □□□□□ अकाउंट नहीं है, तो □□□□□ पोर्टल पर जाकर साइन अप करें।
    - सब्सक्रिप्शन बनाएं: "सब्सक्रिप्शन" सेक्शन पर नेविगेट करें और एक नया सब्सक्रिप्शन बनाएं। यह आपका बिलिंग और प्रबंधन कंटेनर होगा।
2. संसाधन संगठन:
    - संसाधन समूह: अपने संसाधनों को उनके जीवनचक्र और प्रबंधन मानदंडों के आधार पर संसाधन समूहों में व्यवस्थित करें।
    - टैग: अतिरिक्त मेटाडेटा और आसान संसाधन प्रबंधन और बिलिंग के लिए टैग का उपयोग करें।

□□□□□ □□□□□□□□□□ □□□□□□□□ (□□□) के साथ एप्लिकेशन डिप्लॉय करना

□□□□□ □□□□□□□□□□ □□□□□□□□ (□□□) एक प्रबंधित □□□□□□□□□□ सेवा है जो कंटेनरीकृत एप्लिकेशन को तैनात करने, प्रबंधित करने और स्केल करने को सरल बनाती है।

□□□ क्लस्टर बनाना और प्रबंधित करना

1. □□□□□ पोर्टल में □□□ क्लस्टर बनाना:
    - सेटअप: □□□□□ पोर्टल में □□□ खोजें और एक नया □□□□□□□□□□ क्लस्टर बनाएं।
    - कॉन्फ़िगरेशन: अपने क्लस्टर का आकार चुनें, नोड पूल्स को कॉन्फ़िगर करें, और नेटवर्किंग सेटअप करें।
    - प्रमाणीकरण: सुरक्षित पहुंच नियंत्रण के लिए □□□□□ □□□□□□ □□□□□□□□□ (□□□) का उपयोग करें।
    - मॉनिटरिंग: सेटअप प्रक्रिया के दौरान मॉनिटरिंग और लॉगिंग को सक्षम करें।
2. □□□□□ □□□ का उपयोग करके □□□ क्लस्टर बनाना:

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

3. अपने □□□ क्लस्टर का प्रबंधन:
    - क्लस्टर को स्केल करना:

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

 क्लस्टर को अपग्रेड करना:

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

एप्लिकेशन डिप्लॉय करना

1. □□□□□□□□□□ मैनिफेस्ट्स का उपयोग करना: अपने डिप्लॉयमेंट्स, सर्विसेज और अन्य □□□□□□□□□□ ऑब्जेक्ट्स के लिए □□□□ फ़ाइलें लिखें।

```yaml
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. □□□□□□□□ के साथ डिप्लॉय करना:

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

3. □□□□ □□□□□□□: □□□□□□□□□□ एप्लिकेशन और वर्जन कंट्रोल के लिए □□□□ का उपयोग करें।

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

पॉड्स से लॉग्स प्राप्त करना

1. एक □□□ से जुड़ना और लॉग्स प्राप्त करना:

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

   - लॉग्स को स्ट्रीम करने के लिए:

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

2. लॉगिंग के लिए साइडकार का उपयोग करना:

   - अपने पॉड स्पेसिफिकेशन में एक लॉगिंग साइडकार कंटेनर बनाएं ताकि लॉग्स को एक केंद्रीकृत लॉगिंग सेवा पर भेजा जा सके।

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

□□□□□ □□□□□□□□□□□ □□□□□□□□ **के साथ मॉनिटरिंग और डायग्नोस्टिक्स**

□□□□□□□□□□□ □□□□□□□□ आपके एप्लिकेशन के लिए शक्तिशाली मॉनिटरिंग और डायग्नोस्टिक क्षमताएं प्रदान करता है।

1. एप्लिकेशन इनसाइट्स सेटअप करना:

   - इंटीग्रेशन: अपने एप्लिकेशन कोड में □□□□□□□□□□□ □□□□□□□□ □□□ जोड़ें।
   - इंस्ट्रुमेंटेशन कुंजी: अपने एप्लिकेशन को □□□□□□□□□□□ □□□□□□□□ रिसोर्स से प्राप्त इंस्ट्रुमेंटेशन कुंजी के साथ कॉन्फ़िगर करें।

2. प्रदर्शन ट्रैकिंग:

   - मेट्रिक्स: प्रतिक्रिया समय, विफलता दर और एप्लिकेशन निर्भरताओं की निगरानी करें।
   - लाइव मेट्रिक्स स्ट्रीम: तत्काल जानकारी के लिए वास्तविक समय के प्रदर्शन मेट्रिक्स देखें।

3. डायग्नोस्टिक्स और समस्या निवारण:

   - एप्लिकेशन मैप: निर्भरताओं को दृश्यमान बनाएं और प्रदर्शन में बाधाओं की पहचान करें।
   - ट्रांजैक्शन डायग्नोस्टिक्स: सेवाओं के बीच अनुरोधों को ट्रेस करने के लिए वितरित ट्रेसिंग का उपयोग करें।

□□□□□ □□□□□□□□ □□□□□□□□□ **(□□□) का उपयोग**

□□□□□ □□□ कस्टम एप्लिकेशन और सेवाओं को चलाने की लचीलापन प्रदान करते हैं जो कंटेनराइज़्ड नहीं हैं।

1. वर्चुअल मशीन प्रोविजनिंग:
   - □ □□□ बनाएं: □□□□□ पोर्टल में, नई वर्चुअल मशीनें बनाएं और उचित आकार और ऑपरेटिंग सिस्टम चुनें।
   - □ नेटवर्क कॉन्फ़िगरेशन: ट्रैफ़िक को नियंत्रित करने के लिए वर्चुअल नेटवर्क, सबनेट और सुरक्षा समूह सेट अप करें।
2. □□□ को कॉन्फ़िगर करना:
   - □ सॉफ़्टवेयर इंस्टॉलेशन: आवश्यक सॉफ़्टवेयर और डिपेंडेंसीज़ इंस्टॉल करें।
   - □ सुरक्षा: नियमित रूप से पैच और अपडेट लागू करें, फ़ायरवॉल कॉन्फ़िगर करें, और नेटवर्क सुरक्षा समूह (□□□□) का उपयोग करें।
3. □□□ का प्रबंधन:
   - □ बैकअप और पुनर्स्थापना: □□ बैकअप के लिए □□□□□ □□□□□□ का उपयोग करें।
   - □ मॉनिटरिंग: □□□□□ □□□□□□□ का उपयोग करके □□ प्रदर्शन की निगरानी करें।

□□□□□ □□□□□ □□□□ **के साथ रियल-टाइम डेटा इंजेशन**

□□□□□ □□□□□ □□□□ एक बड़ा डेटा स्ट्रीमिंग प्लेटफॉर्म और इवेंट इंजेक्शन सेवा है जो प्रति सेकंड लाखों इवेंट्स को प्राप्त और प्रोसेस करने में सक्षम है।

1. इवेंट हब्स सेट करना:
   - □ इवेंट हब नेमस्पेस बनाएं: □□□□□ पोर्टल में, अपने इवेंट हब्स को रखने के लिए एक इवेंट हब नेमस्पेस बनाएं।
   - □ इवेंट हब्स बनाएं: नेमस्पेस के अंदर, अपने डेटा स्ट्रीम को कैप्चर करने के लिए एक या अधिक इवेंट हब्स बनाएं।
2. डेटा इंगेस्ट करना:
   - □ प्रोड्यूसर्स: अपने एप्लिकेशन या सेवाओं को कॉन्फ़िगर करें ताकि वे □□□□□ □□□□ को इवेंट भेज सकें, जिसके लिए कई भाषाओं (जैसे .□□□, □□□□, □□□□□□) में उपलब्ध □□□□ का उपयोग किया जा सकता है।
   - □ पार्टीशन्स: इवेंट प्रोसेसिंग को स्केल करने के लिए पार्टीशन्स का उपयोग करें, जिससे उच्च थ्रूपुट और समानांतरता सुनिश्चित हो सके।
3. इवेंट्स को प्रोसेस करना:
   - □ उपभोक्ता: इवेंट्स को पढ़ने और प्रोसेस करने के लिए उपभोक्ता समूहों का उपयोग करें। □□□□□ इवेंट्स को प्रोसेस करने के लिए कई विकल्प प्रदान करता है, जिनमें □□□□□ □□□□□□ □□□□□□□□□□, □□□□□ □□□□□□□□□□, और □□□□□ □□□□ □□□ का उपयोग करके कस्टम प्रोसेसिंग शामिल हैं।
4. इवेंट हब्स की निगरानी:
   - □ मेट्रिक्स: □□□□□ पोर्टल के माध्यम से थ्रूपुट, लेटेंसी और इवेंट प्रोसेसिंग मेट्रिक्स की निगरानी करें।
   - □ अलर्ट्स: किसी भी समस्या, जैसे उच्च लेटेंसी या ड्रॉप्ड मैसेजेस, के बारे में सूचित करने के लिए अलर्ट्स सेट करें।

□□□□□ □□□ □□□□□□□□□□ □□□□□□□□ **के साथ** □□□□ **का प्रबंधन**

□□□□□ □□□ □□□□□□□□□□ □□□□□□□□ मौजूदा बैक-एंड सेवाओं के लिए सुसंगत और आधुनिक □□□ गेटवे बनाने का एक तरीका प्रदान करते हैं।

1. □□□ □□□□□□□□□□ सेटअप करना:
   - □ □□□ □□□□□□□□□□ सेवा बनाएं: □□□□□ पोर्टल में, □□□ □□□□□□□□□□ खोजें और एक नई सेवा बनाएं।
   - □ □□□□ कॉन्फ़िगर करें: □□□□□□□ स्पेसिफिकेशन, □□□□□ □□□□□□□□□, या अन्य बैकएंड से □□□□ को परिभाषित और आयात करें।
2. □□□□ को सुरक्षित करना:
   - □ प्रमाणीकरण और अधिकार प्रबंधन: अपने □□□□ को सुरक्षित करने के लिए □□□□□2, □□□ वैलिडेशन और अन्य तंत्रों का उपयोग करें।
   - □ दर सीमित करना और थ्रॉटलिंग: अपने □□□□ को दुरुपयोग से बचाने के लिए नीतियों को लागू करें।
3. मॉनिटरिंग और एनालिटिक्स:
   - □ □□□ इनसाइट्स: उपयोग को ट्रैक करें, प्रदर्शन की निगरानी करें, और लॉग का विश्लेषण करें।
   - □ डेवलपर पोर्टल: डेवलपर्स को आपके □□□□ को खोजने और उपयोग करने के लिए एक पोर्टल प्रदान करें।
4. लाइफसाइकल प्रबंधन:
   - □ वर्जनिंग और रिविज़न: अपने □□□□ के विभिन्न वर्जन और रिविज़न को सहजता से प्रबंधित करें।
   - □ पॉलिसी प्रबंधन: अनुरोधों के ट्रांसफॉर्मेशन, वैलिडेशन और रूटिंग के लिए पॉलिसीज़ लागू करें।

और प्रतिक्रियाएँ।

□□□□□ □□□ **डेटाबेस का उपयोग करना**

□□□□□ □□□ □□□□□□□□ एक पूरी तरह से प्रबंधित रिलेशनल डेटाबेस है जिसमें अंतर्निहित बुद्धिमत्ता, उच्च उपलब्धता और स्केलेबिलिटी शामिल है।

1. □□□□□ □□□ डेटाबेस सेट करना:
   - □ □□□ डेटाबेस बनाएं: □□□□□ पोर्टल में, □□□ डेटाबेस पर नेविगेट करें और एक नया डेटाबेस बनाएं।
   - □ डेटाबेस कॉन्फ़िगर करें: डेटाबेस का आकार, प्रदर्शन स्तर सेट करें और नेटवर्किंग सेटिंग्स कॉन्फ़िगर करें।
2. □□□ डेटाबेस से कनेक्ट करना:
   - □ कनेक्शन स्ट्रिंग्स: अपने एप्लिकेशन को □□□ डेटाबेस से कनेक्ट करने के लिए प्रदान की गई कनेक्शन स्ट्रिंग्स का उपयोग करें।
   - □ फ़ायरवॉल नियम: अपने एप्लिकेशन या लोकल मशीन से एक्सेस की अनुमति देने के लिए फ़ायरवॉल नियम कॉन्फ़िगर करें।
3. डेटाबेस प्रबंधन:

- □ बैकअप और पुनर्स्थापना: अपने डेटा की सुरक्षा के लिए स्वचालित बैकअप और पॉइंट-इन-टाइम पुनर्स्थापना का उपयोग करें।
- □ स्केलिंग: अपनी प्रदर्शन आवश्यकताओं के आधार पर डेटाबेस को ऊपर या नीचे स्केल करें।

4. मॉनिटरिंग और परफॉर्मेंस ट्यूनिंग:
   - □ क्वेरी परफॉर्मेंस इनसाइट्स: क्वेरी परफॉर्मेंस को मॉनिटर और ऑप्टिमाइज़ करें।
   - □ ऑटोमैटिक ट्यूनिंग: परफॉर्मेंस को सुधारने के लिए ऑटोमैटिक ट्यूनिंग फीचर्स को सक्षम करें।

□□□□□ □□□□□ □□□□□□□□□ (□□□) के साथ लॉग्स को क्वेरी करना

□□□□□ □□□□□ □□□□□□□□□ (□□□) का उपयोग □□□□□ □□□□□□□ □□□□ को क्वेरी करने के लिए किया जाता है, जो आपके लॉग डेटा में शक्तिशाली अंतर्दृष्टि प्रदान करता है।

1. बेसिक □□□ क्वेरी:

```
//
LogTableName
| where TimeGenerated > ago(1h)
| project TimeGenerated, Level, Message
```

2. डेटा को फ़िल्टर और एकत्रित करना:

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

3. टेबल्स को जोड़ना:

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

4. क्वेरी के आधार पर अलर्ट बनाना:
   - □ □□□□□ पोर्टल में, □□□ □□□□□□□□□□ वर्कस्पेस पर नेविगेट करें।
   - □ `Logs` पर क्लिक करें और अपना □□□ क्वेरी दर्ज करें।
   - □ क्वेरी परिणामों के आधार पर अलर्ट बनाने के लिए `New alert rule` पर क्लिक करें।

प्रोएक्टिव मॉनिटरिंग के लिए अलर्ट सेट करना

□□□□□ □□□□□□ आपको अपने संसाधनों के स्वास्थ्य और प्रदर्शन के बारे में सूचित रहने में मदद करते हैं।

1. अलर्ट बनाना:
    - मेट्रिक अलर्ट: □□□ उपयोग, मेमोरी उपयोग, और प्रतिक्रिया समय जैसे मेट्रिक्स के आधार पर अलर्ट सेट करें।
    - लॉग अलर्ट: □□□ का उपयोग करके लॉग खोज क्वेरी के आधार पर अलर्ट बनाएं।
2. कार्यों को कॉन्फ़िगर करना:
    - एक्शन ग्रुप्स: यह परिभाषित करें कि किसे सूचित किया जाएगा और कैसे (ईमेल, एसएमएस, वेबहुक)।
    - एकीकरण: स्वचालित घटना प्रबंधन के लिए □□□□□□□□□□ जैसे □□□□ टूल्स के साथ एकीकरण करें।
3. अलर्ट्स का जवाब देना:
    - डैशबोर्ड: अलर्ट्स का केंद्रीकृत दृश्य प्रदान करने के लिए □□□□□ डैशबोर्ड सेट करें।
    - ऑटोमेशन: कुछ अलर्ट्स का स्वचालित रूप से जवाब देने के लिए □□□□□ ऑटोमेशन का उपयोग करें।

निष्कर्ष

□□□□□ □□□□□□□□□□□□□, □□□, □□□□□□□□□□□□ □□□□□□□□□, □□□□□□□□ □□□□□□□□□, □□□□□ □□□□, □□□ □□□-□□□□□□□□ □□□□□□□□□, और □□□ □□□□□□□□□□ का उपयोग करके, हमने एक स्केलेबल, मजबूत और मॉनिटर किए गए एप्लिकेशन इंफ्रास्ट्रक्चर का निर्माण किया। □□□□□ के व्यापक टूल्स ने यह सुनिश्चित किया कि हम अपने एप्लिकेशन को कुशलतापूर्वक डिप्लॉय, प्रबंधित और मॉनिटर कर सकें। इस सेटअप ने न केवल हमारे एप्लिकेशन के प्रदर्शन को बेहतर बनाया, बल्कि हमें अपने संसाधनों को सक्रिय रूप से बनाए रखने और अनुकूलित करने के लिए आवश्यक जानकारी भी प्रदान की।