

Starting at 9:33 AM (3 mins for people to join in!)

Design Storage Solutions

- data (as of DP-200)
- configuration or implementation

NOT included

- design principles
- AVAILABILITY and FAULT

TOLERANCE

- back/recovery/ Disaster management- RTO, RPO

-> IoT Reference Architecture

-> AI Architecture (Conversational bots, cognitive services)

Please activate Skillpipe and Azure-Pass

Case study:

1. Website: Web App
2. OLTP: Storage Account
 1. Container for web app
 2. Container for OLTP app
3. OLTP: SQL Database- Server based vCore or Serverless model?
 1. What kind of replication?
4. Data Analysis:
 1. Many sources (SQL db, others...) -> Data Lake -> Synapse Analytics-> Azure Analysis Service -> power BI
5. Orchestration/ETL-> Data Factory
6. Customer Support Service:
Chatbot
7. Social Media Analysis-> STREAM Analytics + ML/BL
8. Connected Bicycles-> IoT

Reference Architecture

9. Monitoring and Troubleshooting-> Log Analytics, Monitor

Resuming at 11:05 AM

Lambda and Kappa

Lambda-> 2 layers-> 1 batch
(batch + serving layer)
-> 1 speed

Kappa-> all types of inputs-> 1
layer (speed layer)

Apache Beam, Kubeflow

Sensors-> (moisture + humidity)

Stationary-> CSV

1. Batch
2. Stream

1. Data Lake (central repo)
 1. Batch- DataFactory
 2. Stream- IoT Hub
2. Process the data:
 1. DataBricks, Synapse, HDInsight
 2. Real time- stream analytics

blahblah@shantanupandeylive.onmicrosoft.com
Pikachu123!

Data is ALWAYS encrypted-> you or Azure?

- > encryption keys
- > Key encryption Keys
- > every key row will be encrypted

separately

TDE-> KEK-> it will replaced DEK->

1. OS-> Data Protection API

(windows)

1. Service key

2. SQL server-> Service Master Key

3. USER

1. Create Master Key with
password='abc123!'

2. Create certificate mycerti with
subject='my DEK';

3. Create database encryption
key with encryption =

AES_128 ENCRYPTION BY SERVER CERTIFICATE
mycerti;

4.

5. Alter database mydb set
encryption ON;

Resuming at 3:45 pm

Agenda- last module + demo on->
LUIS/QnA Maker

LUIS-> break down natural
language into programmable tokens

My order #123 should be cancelled
Hey! Your company is awesome!
My delivery hasn't reached yet!

Data Scientist + Data Engineer-> a
ML model/service is trained
[DataBricks-> from Blob Storage, to
Cosmos]
- High Concurrency

Data Engineer + AI Engineer -> data model to production

- Stream Analytics -> from Cosmos

Processing->

ML Service

-> output->

Synapse or SQL DB

Security:

- Structured
 - SQL DB
 - Synapse
 - Firewall
 - TDE
 - TLS
 - VNet
- Cosmos Core SQL
 - 4 hour backup policy
 - Geo-redundancy

- Consistency models
- Unstructured
 - Storage account
 - Access Keys- key rotation
 - Shared Access Signatures
 - Read/write/del/create/list permissions
 - IP level permissions, CIDR ranges
 - Start and end date
 - Best to provide media assets to websites!
- HBase (HDInsight)
 - CLUSTER MANAGEMENT
 - Security:
 - Spark, Hbase, IQ, Kafka, Storm, MLwithR
 - Own responsibility
 - Linux OS
 - Production: OS is hardened,

- Cosmos Mongo/Gremlin/Cassandra/Tables
 - MongoDB-> 30 minute timeout
 - MongoDB Enterprise security!
- Azure Active Directory
 - KeyVault (secrets)

DataBricks-> CSV/JSON-> Shared Access Signature

Oauth-> AAD-> tenant id, app id....

`fs.azure.account.oauth2.client.`

`Fs.azure.account.key ->`

`dbutils.fs.mount(`

`mount_point="", source="",`

`extra_configs={`

`"fs.azure.account.key.storageurl`

`..." : key vault token`

})

Without using key vault:

"fs.azure.sas.stroageurl....":

sp=r&st=2021-05-20T15:16:09Z&se=2021-05-20T23:16:09Z&spr=https&sv=2020-02-10&sr=b&sig=cFahdnCPFiQF9Xu9zp2yELkqTiUNqNkKHb22fw9sE%2F8%3D

- Official course
- Docker- hosting apps as container- Azure Container Instance
- Kubernetes- managing containers
 - Azure Kubernetes Service
- Host ML model in Kubernetes
- Automation-> human mistake
- Monitoring kubernetes- cluster autoscaler

(PySpark, Scala, SQL)

- Python
 - Quickly build prototype
 - Precompiled C++

Python-> ML and DE

Python/java/app-> RAM, CPU, IOPS

10 MB RAM, 1 GHz CPU, 100 IOPS

1 app on 1 machine-> wasting rest of the resources

TOTAL machine size that ALL apps would require

Orders, Inventory, Sales -> 1 VM

Orders had scalability requirement-

> scale inc. on entire VM

CONTAINERS

VM-> hardware black box

Container-> OS black box

LIBRARIES

C++-> iostream

Python-> api-> numpy, flask, tensor
flow

VM/Node (Machine) -> N POD

1 POD -> N CONTAINERS

1 Container-> single instance of
app

Each pod could have diff app

2 Machine/nodes/VM-> 100 pods

50 pods-> orders app

20 pods-> sales app

30 pods-> inventory app

Containers-> each with app instances

1. requirements.txt -> ALL dependencies for your program

1. C++-> iostream

2. Python-> flask, num

flask

numpy

2. How to build container-> Dockerfile

Docker and kubernetes configs:
<https://github.com/a-forty-two/cog-21-04-06/>

1. Build Dockerfile and requirementex.txt
 2. sudo docker build -t imgname:tag .
 3. Sudo docker login
 4. Sudo docker tag localname username/reponame
 5. Sudo docker push username/reponame
- <image available in hub.docker.com>

1. Deploy AKS cluster
2. az aks get-credentials — resource-group name —name kubecuster
3. Kubectl get pods (0 pods)

4. Kubectl get services (only 1 service running)
5. Create config.yaml (check git repo)
6. Kubectl apply -f config.yaml
7. Kubectl get services (to get ext. IP address)
8. Kubectl get pods (check when pods are running)

```
apiVersion: v1
kind: Service
metadata:
  name: norepo
spec:
  selector:
    app: norepo
  ports:
    - protocol: "TCP"
      port: 80
      targetPort: 5001
  type: LoadBalancer
```

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: norepo
```

```
spec:
  selector:
    matchLabels:
      app: norepo
  replicas: 2
  template:
    metadata:
      labels:
        app: norepo
    spec:
      containers:
        - name: norepo
          image: afortytwo/norepo:latest
          ports:
            - containerPort: 5001
```