



Kafka on Azure

Options and considerations

Anagha Khanolkar
Cloud Solution Architect, Advanced Analytics & AI

Agenda

1. Kafka on Azure –options
2. Kafka – newer features overview
3. HDInsight Kafka – overview
4. Confluent Kafka – overview
5. Discussion – HDInsight versus Confluent

Kafka on Azure

Options

Options

1. HDInsight Kafka

Hortonworks distribution based Kafka PaaS

2. Confluent Enterprise

Confluent's IaaS marketplace offering

3. Kafka on IaaS – virtual machines, containers

4. Kafka API for Azure Event Hub

Newer kids on the block

Kafka services



Newer Kafka features/services

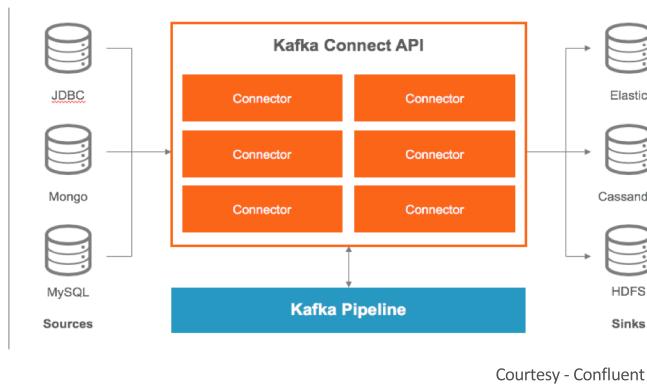
1. KafkaConnect API
2. Kafka Streams API
3. Confluent REST proxy
4. KSQL
5. Confluent Schema registry
6. Log compaction



Kafka Connect

Value proposition

-  Fault tolerant
-  Manage hundreds of data sources and sinks
-  Preserves data schema
-  Part of Apache Kafka project
-  Integrated within Confluent Platform's Control Center



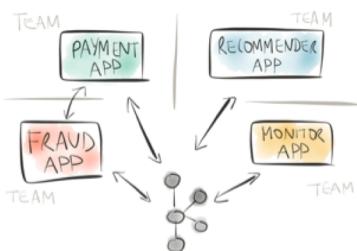
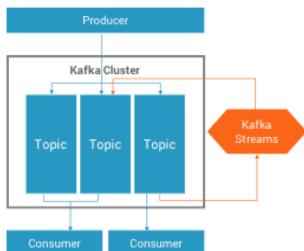
Courtesy - Confluent

- Connector based, codeless integration from/to Kafka and supported sources/sinks
- Configured batch integration – push with automated offset management
- Service is open source, connectors may/may not be
- Can be installed on any Kafka cluster
- Scalable (horizontal)
- Fault tolerant
- Automatically load-balanced
- *Extensible*
- CDC configurable for RDBMS'
- In heavy traffic scenarios, dedicated instances recommended



Kafka Streams

Value proposition



- ✓ Write standard Java applications
- ✓ No separate processing cluster required
- ✓ Elastic, highly scalable, fault-tolerant
- ✓ Equally viable for small, medium, & large use cases

- ✓ Exactly-once processing semantics
- ✓ Develop on Mac, Linux, Windows
- ✓ Deploy to containers, VMs, bare metal, cloud
- ✓ Fully integrated with Kafka security

Courtesy - Confluent

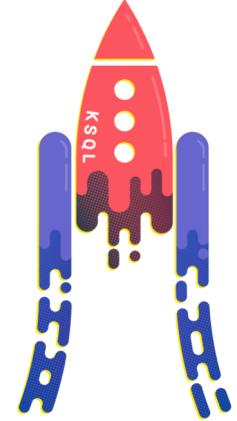
- You don't need a Spark cluster to stream ingest from Kafka – write a Java app instead with Kafka streams
- Open source
- Bring your own infrastructure
- Scalable (horizontal)
- Fault tolerant
- Exactly once processing semantics
- Integrated with Kafka security
- Battle tested and in use in production

```
KStream<User, PageViewEvent> pageViews = builder.stream("pageviews-topic");
KTable<Windowed<User>, Long> viewsPerUserSession = pageViews
    .groupByKey()
    .count(SessionWindows.with(TimeUnit.MINUTES.toMillis(5)), "session-views");
```



Value proposition

- SQL like construct to process streams in a topic
- Confluent open source/enterprise
- Can be installed on non-Confluent Kafka clusters; install on VMs separately
- Use in real-time analytics
E.g. Run KSQL against a topic receiving credit card transactions to detect fraud in real time and write fraudulent transactions to a separate topic against which real-time actioning is configured



1 Streaming ETL

```
CREATE STREAM vip_actions AS
  SELECT userid, page, action
  FROM clickstream c
  LEFT JOIN users u ON c.userid =
    u.user_id
  WHERE u.level = 'Platinum';
```

2 Anomaly Detection

```
CREATE STREAM possible_fraud AS
  SELECT card_number, count(*)
  FROM authorization_attempts
  WINDOW TUMBLING (SIZE 5 SECONDS)
  GROUP BY card_number
  HAVING count(*) > 3;
```

3 Monitoring

```
CREATE TABLE error_counts AS
  SELECT error_code, count(*)
  FROM monitoring_stream
  WINDOW TUMBLING (SIZE 1 MINUTE)
  WHERE type = 'ERROR'
  GROUP BY error_code;
```

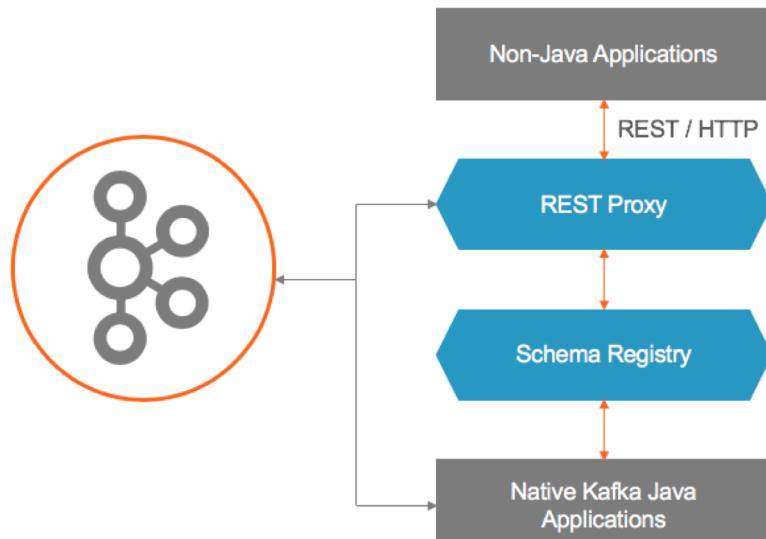
Courtesy - Confluent



Confluent REST proxy

Value proposition

- Provides universal access to Kafka from any network connected device via HTTP
- Is a Confluent Open Source service
- Can be installed on any Kafka cluster
- Dedicated instances recommended for production/high usage scenarios
- Horizontally scalable
- Bring your own load balancer



Courtesy - Confluent



Confluent Schema registry

About

- **Schema registry provides centralized management of schemas**
 - Stores a versioned history of all schemas
 - Provides a RESTful interface + command line tools for storing and retrieving Avro schemas
 - Producers and consumers transparently interact with schema registry minimally
 - A globally unique ID representing the Avro schema is sent with each message; Producers serialize data and prepend the schema ID; Consumers use the schema ID to deserialize the data

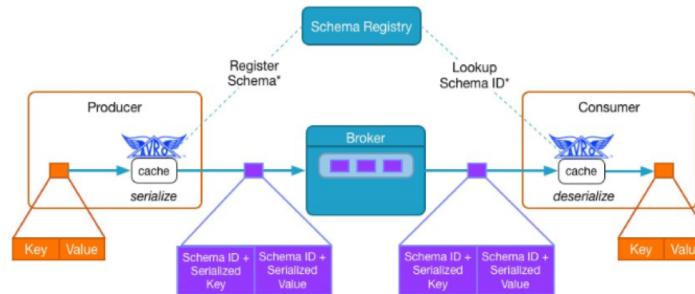
Confluent offering (open source and Enterprise)



Confluent Schema registry

About

- **Consumers use the schema ID to deserialize the data**
 - Schema Registry communication is only on the first message of a new schema
 - Producers and Consumers cache the schema/ID mapping for future messages



- **Can be made HA with single master architecture. In a high traffic scenario, deploy on a separate node**



Confluent Schema registry

What do I need to do in my code?

```
1 Properties props = new Properties();
2 props.put(ProducerConfig.BOOTSTRAP_SERVERS_CONFIG, "broker1:9092");
3 // Configure serializer classes
4 props.put(ProducerConfig.KEY_SERIALIZER_CLASS_CONFIG,
5           io.confluent.kafka.serializers.KafkaAvroSerializer.class);
6 props.put(ProducerConfig.VALUE_SERIALIZER_CLASS_CONFIG,
7           io.confluent.kafka.serializers.KafkaAvroSerializer.class);
8 // Configure schema repository server
9 props.put(AbstractKafkaAvroSerDeConfig.SCHEMA_REGISTRY_URL_CONFIG,
10            "http://schemaregistry1:8081");
11 // Create the producer expecting Avro objects
12 KafkaProducer<Object, Object> avroProducer = new KafkaProducer<Object, Object>(props);
13 // Create the Avro objects for the key and value
14 CardSuit suit = new CardSuit("spades");
15 SimpleCard card = new SimpleCard("spades", "ace");
16 // Create the ProducerRecord with the Avro objects and send them
17 ProducerRecord<Object, Object> record = new
18 ProducerRecord<Object, Object>(
19     "my_avro_topic", suit, card);
20 avroProducer.send(record);
```

Sample – courtesy Confluent



Confluent Schema registry

FAQ

- **Can we live without the avro schema registry?** Absolutely
- **Other registry solutions?** Are available
- **Can the Confluent Avro schema registry be installed on any Kafka cluster?**
Its been tried and has been successful

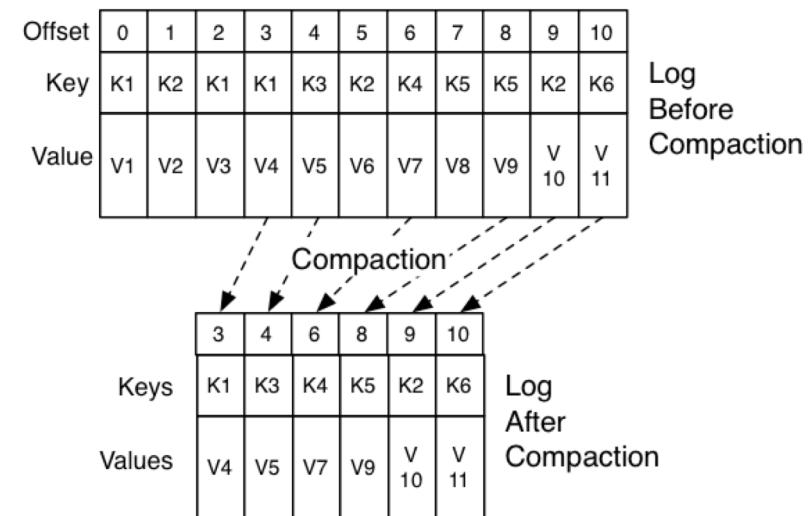
Sample – courtesy Confluent



Log compaction

About

- Log compaction is a mechanism to give finer-grained per-record retention, rather than the coarser-grained time-based/size-based retention. The idea is to selectively remove records where we have a more recent update with the same primary key. This way the log is guaranteed to have at least the last state for each key.
- This retention policy can be set per-topic, so a single cluster can have some topics where retention is enforced by size or time and other topics where retention is enforced by compaction.



The compaction is done in the background, and does not block reads, and can be throttled to use ONLY configured I/O throughput, so as to not impact producer/consumer operations.

HDInsight Kafka



Azure HDInsight

Hortonworks PaaS | Workload based clusters with disaggregated compute and storage



Open Source

- 100% Apache Open Source
- The most popular open source frameworks
- Part of the Hortonworks HDP distribution



Managed

- 99.9% availability SLA
- Cluster Health Monitoring
- Integration with Azure Log Analytics
- Highly optimized for Azure



Secure & Compliant

- Role based access control
- Azure AD & Kerberos based authentication
- Strong VNET and service endpoint support
- The most trusted and compliant platform



Productive

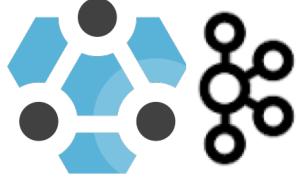
- Works with the tools developers already have
- Special extensions for advanced debugging and diagnostics



Lift & Shift

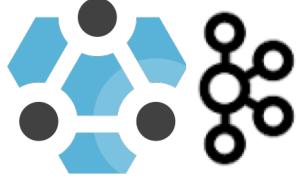
- Move workloads from on-prem or other clouds without code changes
- Curated application platform for wide variety of use cases





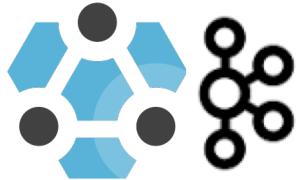
About HDI Kafka

- **Fully managed:**
Kafka clusters (Hortonworks) deployable in 10 minutes
- **Configurable performance:**
Choice of premium (SSD) or standard disks for brokers
- **Scalable:**
Scale brokers via portal, powershell, Azure CLI
- **Securable:**
Perimeter protection (gateway, Vnet, NSGs), authentication (Kerberos), authorization (Ranger) and auditing (Ranger);



Azure HDInsight Kafka

- **Operations:**
Alerting and predictive cluster maintenance through Azure Operations Management Suite
- **Disaster recovery**
MirrorMaker
- **Rack awareness for Kafka on Azure**
Tooling to ensure partition replicas are spread across fault domains
- **Compliance**
HIPAA, SOC, PCI



Azure HDInsight Kafka

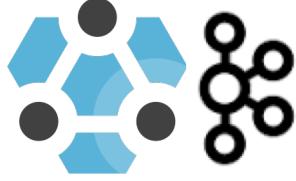
- **Versions:**

Apache Kafka	HDInsight 4.0 (preview)	HDInsight 3.6 (GA)
1.0.1		1.1, 1.0, 0.10.1

- **SLA: 99.9%**

- **New features:**

Service	Comment	Installed/ Supported?	Workaround
Avro schema registry	Confluent open source	No	Install on edge nodes and use
KafkaConnect API	Kafka open source	No	Install on edge nodes and use
Kafka Streams API	Kafka open source	Yes	
KSQL	Confluent open source	No	Install on edge nodes and use
REST proxy	Confluent open source	No	Install on edge nodes and use



Provisioning options

- Azure portal

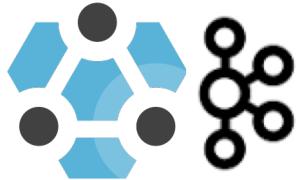
<https://docs.microsoft.com/en-us/azure/hdinsight/kafka/apache-kafka-get-started>

- PowerShell

<https://docs.microsoft.com/en-us/azure/hdinsight/kafka/apache-kafka-quickstart-powershell>

- ARM template

<https://docs.microsoft.com/en-us/azure/hdinsight/kafka/apache-kafka-quickstart-resource-manager-template>



Customize clusters with...

- Bootstrapping

Use Powershell to bootstrap clusters with configuration files located in blob storage

<https://docs.microsoft.com/en-us/azure/hdinsight/hdinsight-hadoop-customize-cluster-bootstrap?toc=%2Fen-us%2Fazure%2Fhdinsight%2Fkafka%2FTOC.json&bc=%2Fen-us%2Fazure%2Fbread%2FToc.json>

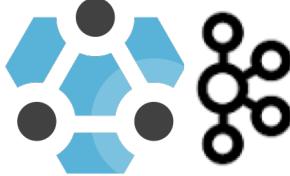
- Script actions

Leverage to install/configure cluster at creation/live cluster

A few script actions available out of the box for use

Create your script (bash), store in accessible storage (blob storage), attach to the cluster

The screenshot shows the 'Script actions' blade in the Azure portal. The left sidebar lists various cluster management options: Access control (IAM), Tags, Diagnose and solve problems, Quick start, Tools, Settings, Cluster size, Quota limits, SSH + Cluster login, Data Lake Storage Gen1, Storage accounts, Applications, and Script actions. The 'Script actions' option is highlighted with a red box. The main area displays the 'anaghaespdemo - Script actions' blade. It includes a search bar, a 'Submit new' button, and two sections: 'PERSISTED SCRIPT ACTIONS' (which shows 'No script actions found') and 'SCRIPT ACTION HISTORY' (which shows a single entry: 'HDI_MONITORING_UNINSTALL_f5d3a1' with a green status indicator). A red arrow points from the 'Script actions' list in the sidebar to the 'HDI_MONITORING_UNINSTALL_f5d3a1' entry in the history list.



Add edge nodes...

Why?

Add edge nodes to run services like KafkaConnect, KSQL etc.

Any limits?

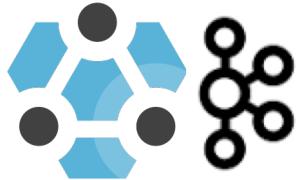
You can add as many edge nodes as needed

What's out of the box?

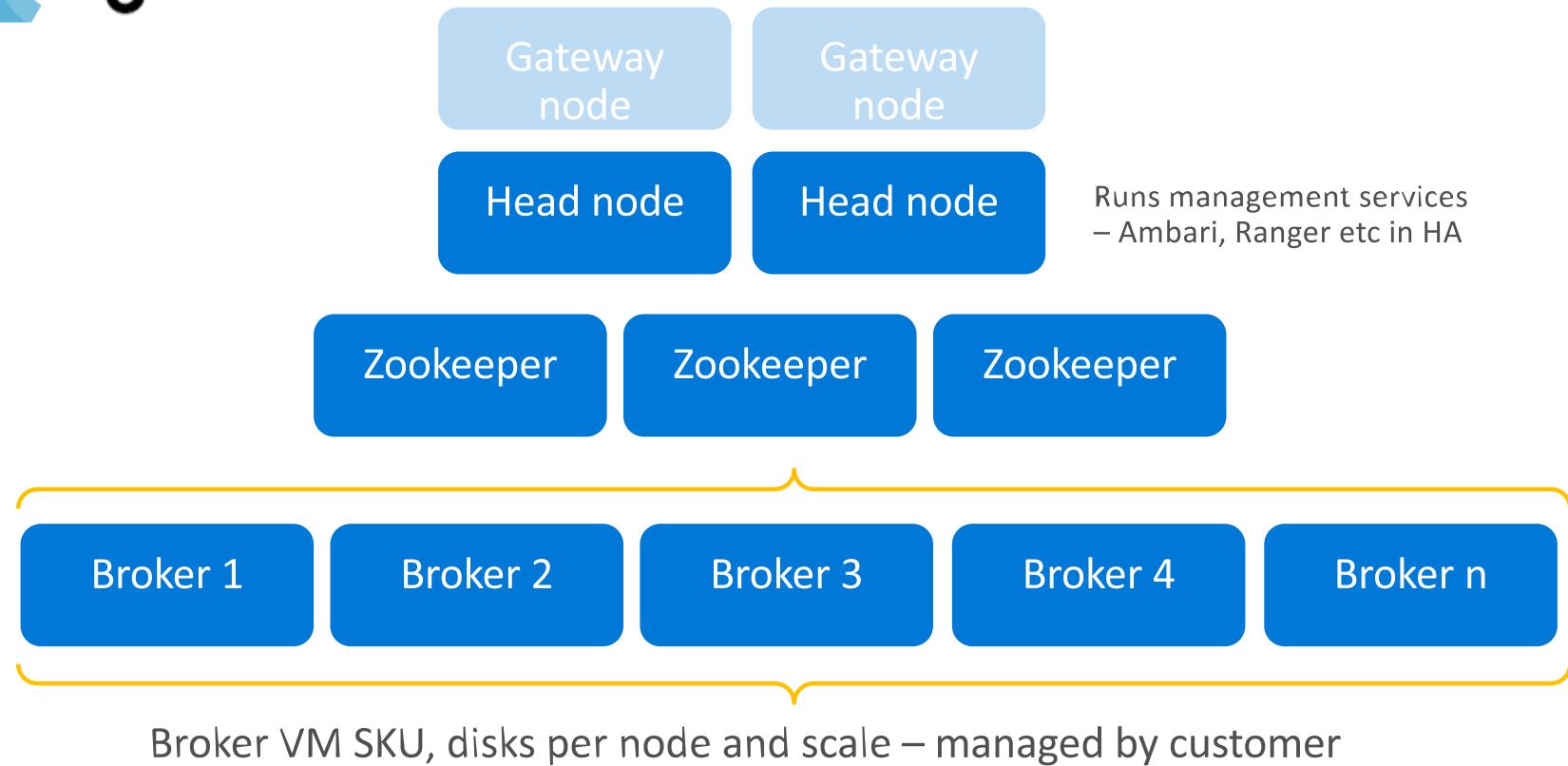
They are automatically domain joined with ESP clusters, and come with all Kafka client libraries preinstalled

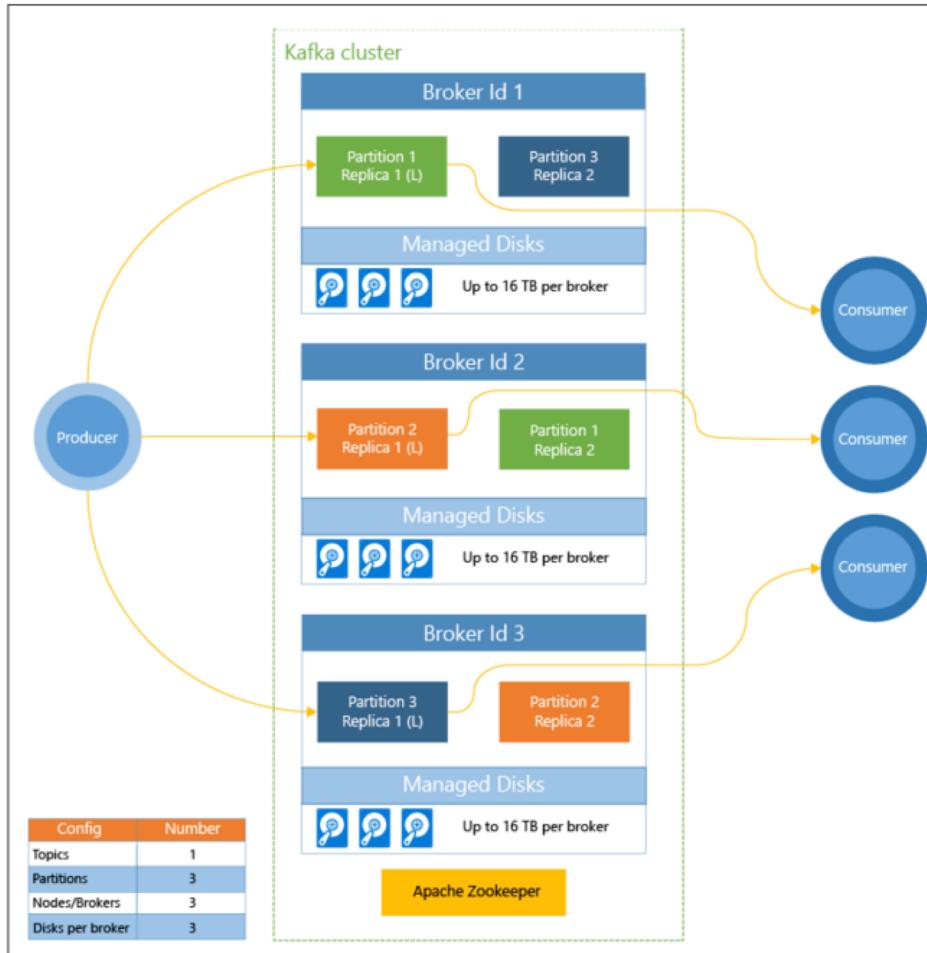
How to provision?

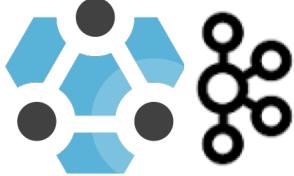
<https://docs.microsoft.com/en-us/azure/hdinsight/hdinsight-apps-use-edge-node#add-an-edge-node-when-creating-a-cluster>



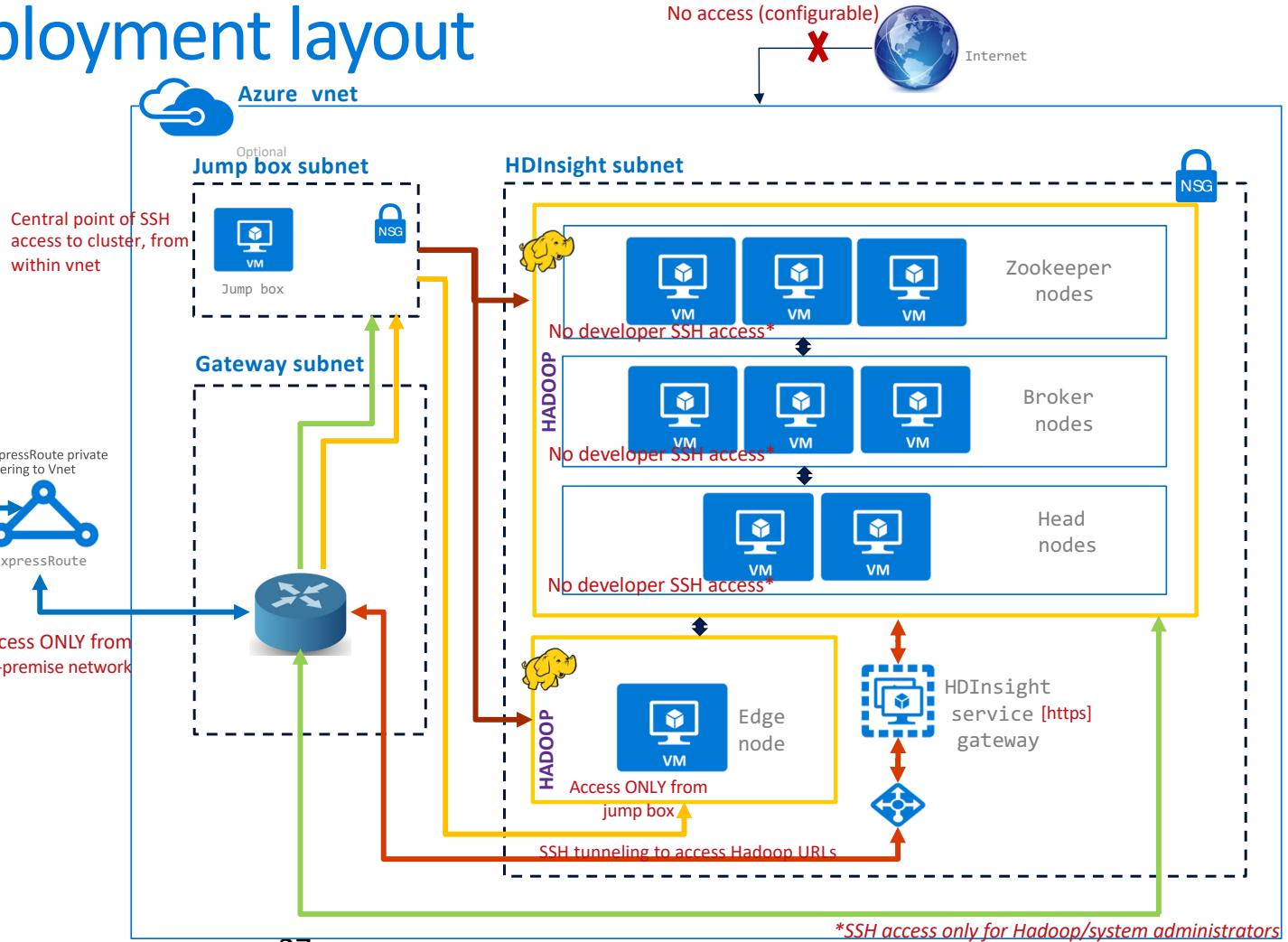
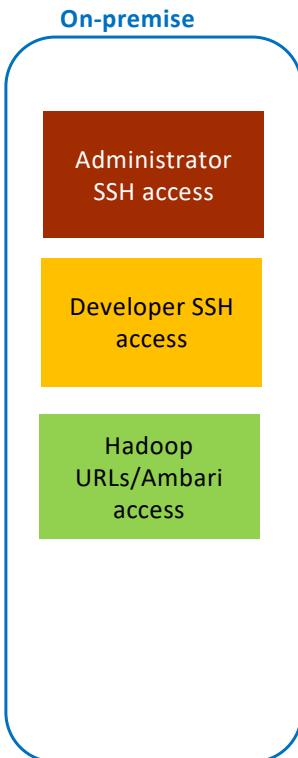
Topology

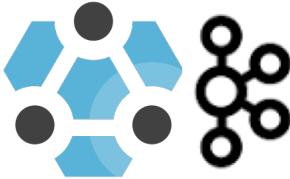




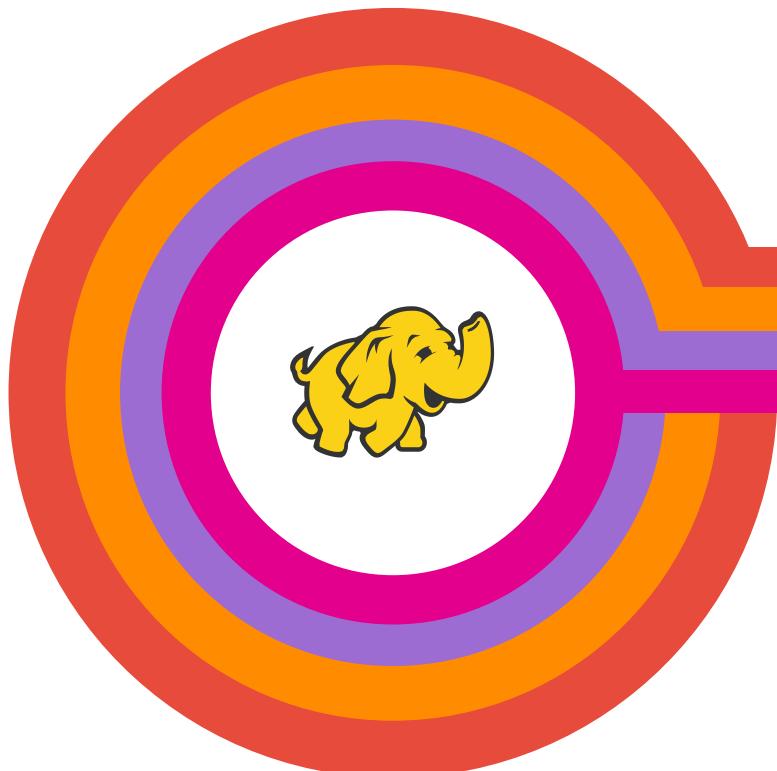


Deployment layout





Security



PERIMETER

Isolate clusters within VNETs
Leverage network security groups as needed
Restrict outbound traffic using NVAs*

AUTHENTICATION

Kerberos with Azure Active Directory Domain Services

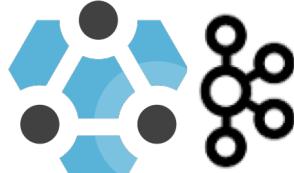
AUTHORIZATION

Apache Ranger for fine grained RBAC

DATA PROTECTION

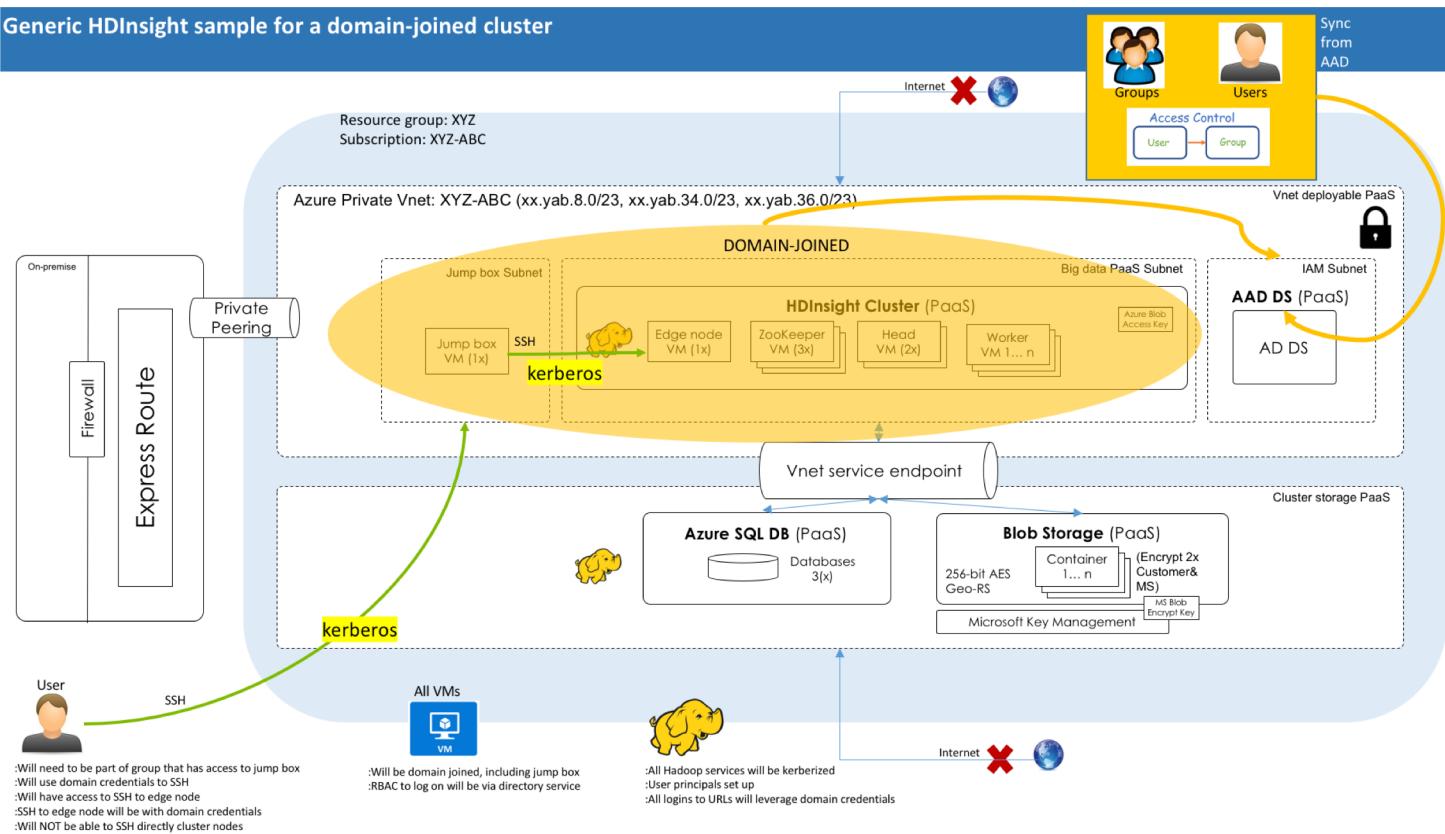
Encryption on-the-wire with HTTPS enforced
Encryption at Rest using Azure Key Vault

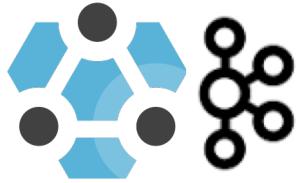
Auditing of all data operations and configuration changes



Security – Authentication

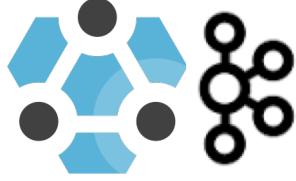
Generic HDInsight sample for a domain-joined cluster





Security – Authorization with Ranger

A screenshot of the Ranger Service Manager interface. The top navigation bar includes tabs for 'Ranger' (selected), 'Access Manager', 'Audit', and 'Settings', along with a user icon for 'admin'. Below the navigation is a toolbar with 'Import' and 'Export' buttons. The main area is titled 'Service Manager' and lists four services: 'HDFS', 'SOLR', 'KAFKA', and 'WASB'. Each service entry has a folder icon, a '+' button, and three small icons for edit, delete, and refresh. The 'KAFKA' entry contains the text 'anaghaeispdemo_kafka' and features a small green checkmark icon. The 'WASB' entry also has a similar set of icons.



Security – Authorization with Ranger

Ranger Access Manager Audit Settings admin

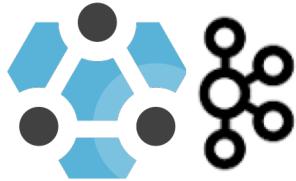
Service Manager > anaghaespdemo_kafka Policies

List of Policies : anaghaespdemo_kafka

Search for your policy...

Add New Policy

Policy ID	Policy Name	Status	Audit Logging	Groups	Users	Action
1	all - transactionalid	Enabled	Enabled	--	hdisp1efdab480c1114c, rangerlookup, kafka, dhruv	
2	all - topic	Enabled	Enabled	--	hdisp1efdab480c1114c, rangerlookup, kafka, dhruv	
5	sales_crud	Enabled	Enabled	--	alice	
6	marketing spend	Enabled	Enabled	--	bob, alice	



Security – Authorization with Ranger

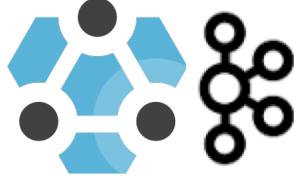
Edit Policy

Policy Details :

Policy Type	Access
Policy ID	5
Policy Name *	<input type="text" value="sales_crud"/> enabled
topic *	<input type="text" value="* sales*"/> include
Audit Logging	YES
Description	<input type="text"/>

Allow Conditions :

Select Group	Select User	Policy Conditions	Permissions	Delegate Admin	
<input type="button" value="Select Group"/>	<input type="text" value="alice"/>	Add Conditions <input type="button" value="+"/>	Publish Consume Create <input type="button" value=""/>	<input type="checkbox"/>	<input type="button" value="x"/>
<input type="button" value="+"/>					



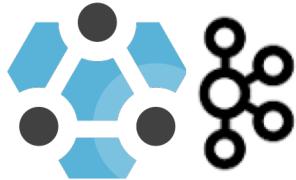
Security – Encryption

At rest:

- Storage account is secured with SSE with AES 256 bit encryption, Microsoft managed keys in Azure Key Vault with annual key rotation on Microsoft schedule
- Want further security and control?
Azure disk encryption can be leveraged with BYOK support, secured by Azure Key Vault. Note: Every additional layer of encryption impacts performance.

In transit:

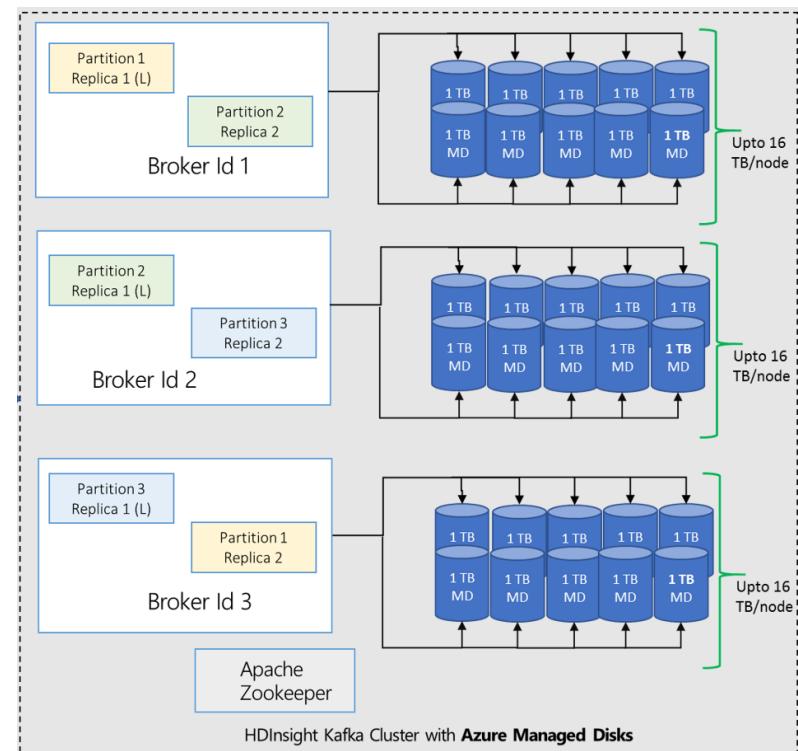
- Not TLS secured out of the box, requires the same diligence you would with an IaaS Kafka cluster
- On the roadmap
- DIY: http://kafka.apache.org/090/documentation.html#security_ssl

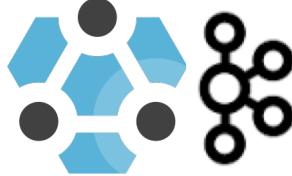


Security – Encryption

Bring Your Own Key

- Default - Managed Disks are encrypted via Azure SSE (Microsoft Managed)
- BYOK – Use Azure Key Vault to store your keys
- User Assigned Managed Identity to access the AKV
- Simple setup during cluster creation
- Transparent Disk Encryption – no change in your client applications





Security – Auditing

Demo

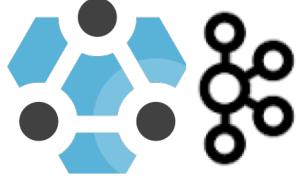
Ranger Access Manager Audit Settings

Access Admin Login Sessions Plugins Plugin Status

START DATE: 10/31/2018

Last Updated Time : 10/31/2018 01:27:02 AM

Policy ID	Event Time ▾	User	Service	Resource	Access Type	Result	Access Enforcer	Client IP	Cluster Name
			Name / Type	Name / Type					
1	10/31/2018 01:27:02 AM	kafka	anaghaespdemo_kafka kafka		kafka_admin	Allowed	ranger-acl	10.11.0.9	anaghaespdemo
1	10/31/2018 01:27:02 AM	kafka	anaghaespdemo_kafka kafka		kafka_admin	Allowed	ranger-acl	10.11.0.11	anaghaespdemo
1	10/31/2018 01:27:01 AM	kafka	anaghaespdemo_kafka kafka		kafka_admin	Allowed	ranger-acl	10.11.0.11	anaghaespdemo
1	10/31/2018 01:27:01 AM	kafka	anaghaespdemo_kafka kafka		kafka_admin	Allowed	ranger-acl	10.11.0.9	anaghaespdemo
1	10/31/2018 01:27:01 AM	kafka	anaghaespdemo_kafka kafka		kafka_admin	Allowed	ranger-acl	10.11.0.14	anaghaespdemo
1	10/31/2018 01:27:01 AM	kafka	anaghaespdemo_kafka kafka		kafka_admin	Allowed	ranger-acl	10.11.0.14	anaghaespdemo



High availability

Management services and nodes:

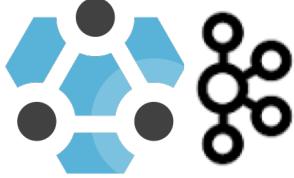
Management/head nodes/gateway nodes are configured for high availability

Kafka:

Is a master-less architecture – and is therefore HA

Ranger:

Is configured HA



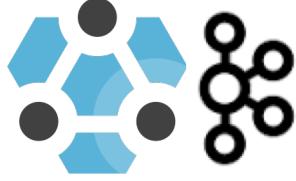
High availability

Azure storage:

- Azure storage is highly available – 3 copies of each disk are available in the same data center and are seamlessly served up upon failure of one
- Rebalancing tool from HDInsight helps with rebalancing across nodes/fault and update domains

Brokers:

- HDInsight Kafka service self-heals – replaces brokers upon failure
- Kafka framework detects and automatically remediates broker failures - with less than replication factor checks and re-replication of logs



Disaster Recovery

Apache Mirrormaker is the solution for disaster recovery

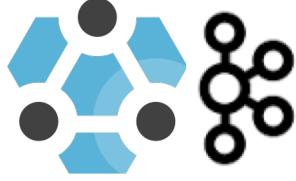
Considerations:

1. Enable global Vnet peering between source and destination clusters with non-overlapping address spaces
2. Leverage private IPs for communication

What about Kerberos?

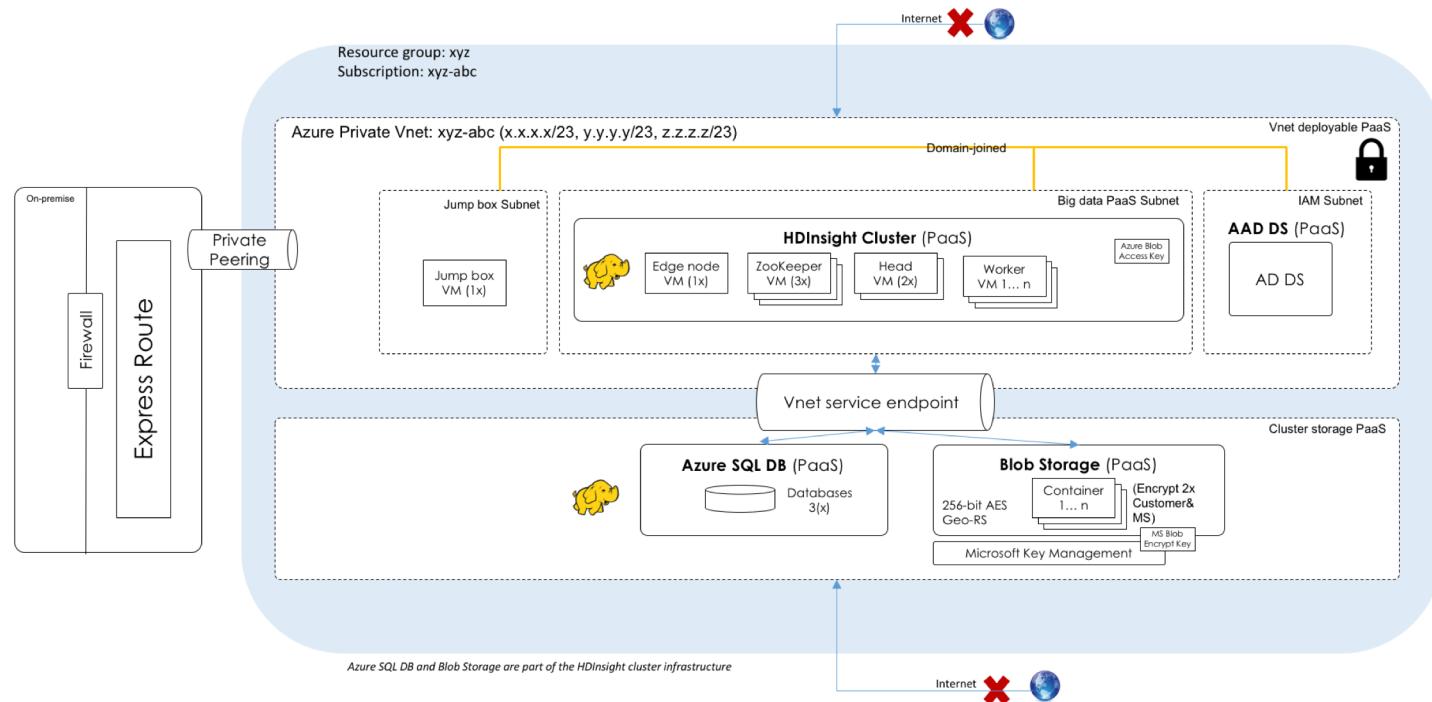
Very cumbersome – establish cross realm trust and more..

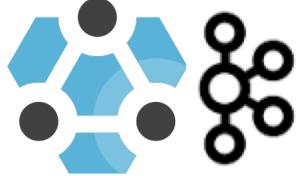
Let us know and we will help



Connecting from on-premise

Configure ExpressRoute private peering into the Kafka Vnet to be able to connect with private IPs





Tuning, specific to Azure

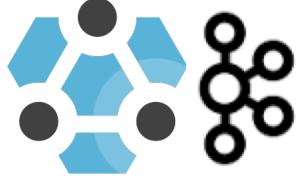
HDI Kafka leverages the latest and greatest of Azure infrastructure and follow best practices – leverages fault domains and update domains

Note: We recommend using an Azure region that contains three fault domains, and using a replication factor of 3. If you must use a region that contains only two fault domains, use a replication factor of 4 to spread the replicas evenly across the two fault domains.

Kafka is not aware of fault domains. When you create a topic in Kafka, it may store all partition replicas in the same fault domain. To solve this problem, HDInsight provides the [Kafka partition rebalance tool](https://github.com/hdinsight/hdinsight-kafka-tools). (<https://github.com/hdinsight/hdinsight-kafka-tools>)

Rebalance partition replicas for your topic at the following times:

- When a new topic or partition is created
- When you scale up a cluster



Monitoring by Microsoft

HDInsight Cluster Dashboard Cluster Search Property Rule HDI Oncall Customer Dashboard Report ashishth@microsoft.com Dashboard Health Feedback

ashishth-Presto

Properties

name	value
ClusterDnsName	ashishth-Presto
PoolStatus	PoolMigratesToPoolCapabilityDisabled
PooledClusterDnsNameUsed	share
CreatedDate	2017-05-16T20:56:30.517Z
HdiDeploymentId	5ad4bc835bda4a038b8cb9554b954cc2 share
UserSubscriptionId	d68b1168-d835-4066-8c45-7d2e0d713c082
InternalSubscriptionId	46c5358b-aac2-475f-94fb-15c420d96c78
IsSecureHadoop	False (valid if cluster created after 11/5/2016)
State	Running
Location	East US 2
ApiVersion	2015-03-01-preview
HdiVersion	3.6.1000.0.10049273
FrontDoorType	ARM
ResourceGroup	rg-presto
CustomerName	AZURE10 CSIT
headnode.vmSize	Standard_D12_V2
headnode.count	2
workernode.vmSize	Standard_D14
workernode.count	16
zookeepernode.vmSize	Medium
zookeepernode.count	3
gateway.vmSize	Medium
gateway.count	2
storage	ramoha02store.blob.core.windows.net,ashishthtpcds36store.blob.core.windows.net,ashishthrest002.blob.core.windows.net,tpcdsashishthstore.blob.core.windows.net
vnet.metrics	[]
oozie.db	ServerName=d630qym524.database.windows.net, DbName=v365ad4bc835bda4a038b8cb9554b954cc2ooziertastore, UserName=v365ad4bc835bda4a038b8cb9554b954cc2ooziertastore, login
ambari.db	Role=headnode, ServerName=q2qgj6x2.database.windows.net, DbName=v365ad4bc835bda4a038b8cb9554b954cc2AmbariDb, UserName=v365ad4bc835bda4a038b8cb9554b954cc2AmbariDbLogin
clusterType	

TSG(Preview) Unavailable(Preview)

AmbariAlerts (Recent 1 hour, 1 distinct alerts)

Service(HDFS), Component(NAMENODE), Label(NameNode Heap Usage (Daily)), DefinitionName(increase_nn_heap_usage_daily)(11)
The variance for this alert is 36MB which is 23% of the 160MB average (32MB is the limit)

Watchdog (Recent 1 hour)

ResourceHealth	NodeManagersUp	Oozie	DataNodesUp	RStudio	JobSubmissionPing	SSHTCPPing	CertRollover	GatewayTCPPing
10 successes 0 errors								

EdgenodeSSH
10 successes 0 errors

Probes (Recent 1 hour)

DataLake	Oozie	Hdfs	Ssh	AmbariAlert	HdinsightZookeeper
22 ok 0 standby 0 failed 0 exception 0 timeout 22 not running 0 not installed	22 ok 0 standby 0 failed 0 exception 0 timeout 0 not running 0 not installed	22 ok 0 standby 0 failed 0 exception 0 timeout 0 not running 0 not installed	22 ok 0 standby 0 failed 0 exception 0 timeout 0 not running 0 not installed	11 ok 11 standby 0 failed 0 exception 0 timeout 0 not running 0 not installed	22 ok 0 standby 0 failed 0 exception 0 timeout 0 not running 0 not installed

Zookeeper	NetworkConnectivity	Wasp	JobHistory	ResourceManager	AmbariServer	AmbariDb
22 ok 0 standby 0 failed 0 exception 0 timeout 0 not running 0 not installed	55 ok 0 standby 0 failed 0 exception 0 timeout 0 not running 0 not installed	22 ok 0 standby 0 failed 0 exception 0 timeout 0 not running 0 not installed	22 ok 11 standby 0 failed 0 exception 0 timeout 0 not running 0 not installed	11 ok 11 standby 0 failed 0 exception 0 timeout 0 not running 0 not installed	11 ok 11 standby 0 failed 0 exception 0 timeout 0 not running 0 not installed	22 ok 0 standby 0 failed 0 exception 0 timeout 0 not running 0 not installed

Daily Availability (Recent 7 days)

Ambari	ODBC	Oozie	Templeton	Hbase	Gateway	MapReduce	Total
100 %	100 %	100 %	100 %	100 %	100 %	100 %	100 %

Weekly Availability (Recent 4 weeks)

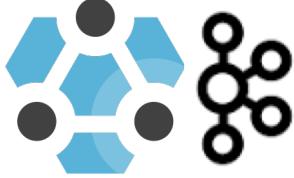
Ambari	ODBC	Oozie	Templeton	Hbase	Gateway	MapReduce	Total
100 %	100 %	100 %	100 %	100 %	100 %	100 %	100 %

Crud Events

time	state
2017-05-16T20:56:32.5606779Z	Accepted
2017-05-16T20:56:51.1647414Z	ClusterStorageConfiguration
2017-05-16T20:58:07.9649262Z	AzureVMConfiguration
2017-05-16T21:01:57.6979916Z	HdinsightConfiguration
2017-05-16T21:08:02.0742475Z	Operational
2017-05-16T21:08:37.0356446Z	Running

Scale Events

An internal view of our service monitoring



Monitoring – Log Analytics

Enable OMS and monitor cluster

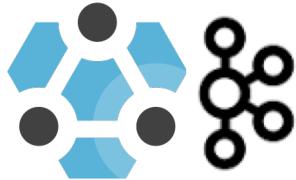
<https://docs.microsoft.com/en-us/azure/hdinsight/hdinsight-hadoop-oms-log-analytics-tutorial?toc=%2Fen-us%2Fazure%2Fhdinsight%2Fr-server%2FTOC.json&bc=%2Fen-us%2Fazure%2Fbread%2Ftoc.json#enable-log-analytics-by-using-the-portal>

Install cluster monitoring solution

<https://docs.microsoft.com/en-us/azure/hdinsight/hdinsight-hadoop-oms-log-analytics-tutorial?toc=%2Fen-us%2Fazure%2Fhdinsight%2Fr-server%2FTOC.json&bc=%2Fen-us%2Fazure%2Fbread%2Ftoc.json#install-hdinsight-cluster-management-solutions>

Query log analytics to monitor cluster

<https://docs.microsoft.com/en-us/azure/hdinsight/hdinsight-hadoop-oms-log-analytics-use-queries?toc=%2Fen-us%2Fazure%2Fhdinsight%2Fr-server%2FTOC.json&bc=%2Fen-us%2Fazure%2Fbread%2Ftoc.json>



Patching

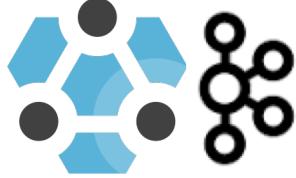
Default patching:

- A virtual machine in the cluster can only reboot for patches at most, once within a 30-day period.
- The reboot occurs starting at 12AM UTC.
- The reboot process is staggered across virtual machines in the cluster, so the cluster is still available during the reboot process.
- The first reboot for a newly created cluster will not happen sooner than 30 days after the cluster creation date.

It's overridable through customization:

- Use a script action to configure your schedule:

Enable/disable automatic reboots	0 or 1. A value of 0 disables automatic reboots while 1 enables automatic reboots.
Frequency	7 to 90 (inclusive). The number of days to wait before rebooting the virtual machines for patches that require a reboot.
Day of week	1 to 7 (inclusive). A value of 1 indicates the reboot should occur on a Monday, and 7 indicates a Sunday. For example, using parameters of 1 60 2 results in automatic reboots every 60 days (at most) on Tuesday.
Persistence	When applying a script action to an existing cluster, you can mark the script as persisted. Persisted scripts are applied when new workernodes are added to the cluster through scaling operations.



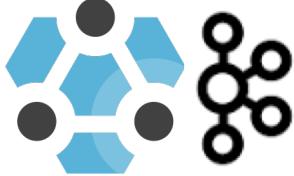
Sizing – bottlenecks are usually...

Typical bottlenecks in Big Data Systems

1. CPU
2. Memory
3. Network
4. Disks + Storage

Kafka uses Filesystem Cache, hence CPU and Memory often not a bottleneck for ingesting millions of events/sec.

Network and Disk often do become the bottleneck – to balance the total throughput, tradeoffs between these often need to be made



Sizing – infrastructure considerations

Managed disks:

Premium managed disks provide guaranteed throughput, but are expensive relative to standard managed disks. Choose based on your budget and performance needs.

<https://docs.microsoft.com/en-us/azure/virtual-machines/linux/premium-storage#scalability-and-performance-targets>

With premium disks – factor in max VM disk throughput and premium managed disk throughput; Attach disks that stay within VM max disk throughput limits to prevent throttling.

Compute:

The VM SKU determines network bandwidth, disk throughput, number of disks, and TYPE of managed disks supported that can be attached, CPU and RAM. Choose SKUs with an 's' in the SKU for support for premium managed disks. Due to replication – there is a lot of east-west traffic. Choose a SKU with sufficient network bandwidth.



Requirements:

Message Rate	10,000 messages/sec
Message size	150 KB upper-bound
Replica count	3
Retention Policy	12 hours

Known Inputs through iPerf Runs:
D12V2 VMs Network limit: 450 MBps

Total Throughput:

$$10,000 \text{ messages/sec} * 150 \text{ KB / message} * 3 \text{ replicas} = 4500 \text{ MB/sec total throughput}$$

Nodes needed from network throughput perspective

$$(4500 \text{ MB/sec} / 450 \text{ MBps}) = \mathbf{10 \text{ Nodes}} \text{ to accommodate the network bandwidth}$$

Nodes needed from storage perspective:

$$4500 \text{ MB/sec} * 12 \text{ hours} = 194.4 \text{ TB / retention policy.}$$

Since each VM can attach 16 disks of 1 TB each.

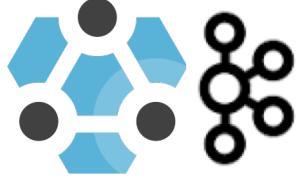
$$194.4 \text{ TB} / 16 \text{ disks per node} \Rightarrow 13 \text{ nodes from storage perspective}$$

Final # nodes needed:

$$\text{MAX}(\#\text{Network Nodes}, \#\text{Storage Nodes}) \Rightarrow$$

13 nodes with 16 managed disks on each node

MICROSOFT CONFIDENTIAL – INTERNAL ONLY

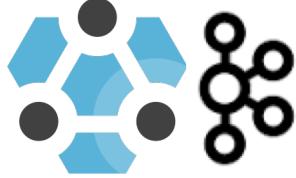


Upgrades

Don't happen automatically.

Keep up with what's new in HDInsight, and plan upgrades.

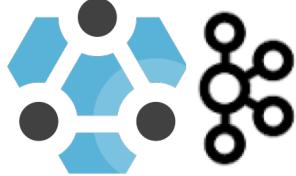
<https://docs.microsoft.com/en-us/azure/hdinsight/hdinsight-upgrade-cluster?toc=%2Fen-us%2Fazure%2Fhdinsight%2Fkafka%2FTOC.json&bc=%2Fen-us%2Fazure%2Fbread%2Ftoc.json>



Pricing

For 5 broker Kafka cluster of DS14v2 SKU (max disk throughput of 768 MBps), with 4x1 TB disks per node, with 2 head nodes and 3 zookeepers, price below for premium versus standard managed disks.

Microsoft Azure Estimate for East US 2					
Service type	Description	Monthly HDI cost	Monthly security cost	Total Monthly cost	Annual cost
HDInsight with premium managed disks	Kafka Component: 2 D12V2 (4 cores, 28 GB RAM) Head nodes x 730 Hours, 5 D14V2 (16 cores, 112 GB RAM) Region nodes x 730 Hours, 3 A1 (1 cores, 1.75 GB RAM) Zookeeper nodes x 730 Hours, 1 D4V2 (8 cores, 28 GB RAM) Edge nodes x 730 Hours, 20 Standard disks	\$9,059.06	\$642.00	\$9,701.06	\$116,412.76
HDInsight with standard managed disks	Kafka Component: 2 D12V2 (4 cores, 28 GB RAM) Head nodes x 730 Hours, 5 D14V2 (16 cores, 112 GB RAM) Region nodes x 730 Hours, 3 A1 (1 cores, 1.75 GB RAM) Zookeeper nodes x 730 Hours, 1 D4V2 (8 cores, 28 GB RAM) Edge nodes x 730 Hours, 20 Standard disks	\$7,420.66	\$642.00	\$8,062.66	\$96,751.96



Databricks Spark - integration

With Vnet injection:

Vnet injection is very new in Databricks.

With Vnet injection, you just need to deploy HDInsight and Databricks in the same Vnet and you are good to go.

Without Vnet injection:

Databricks:

Peer the Databricks Vnet to the Kafka Vnet

Kafka:

Peer the Kafka Vnet with the Databricks Vnet

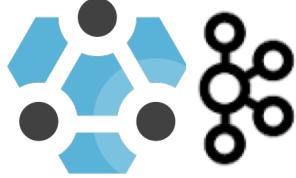
Enable IP broadcasting on your Kafka cluster

Configure listeners to listen on all network interfaces

Restart Kafka

In your Spark code:

Use private IPs of the brokers



HDInsight Spark - integration

Same Vnet:

Use private IPs of brokers or FQDN

Different Vnets:

HDI-Spark:

Peer the HDI-Spark Vnet to the Kafka Vnet

Kafka:

Peer the Kafka Vnet with the HDI-Spark Vnet

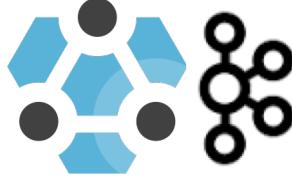
Enable IP broadcasting on your Kafka cluster

Configure listeners to listen on all network interfaces

Restart Kafka

In your Spark code:

Use private IPs of the brokers



Resources

Service Overview: <https://azure.microsoft.com/en-us/services/hdinsight/>

Docs Landing Page: <https://docs.microsoft.com/en-us/azure/hdinsight/kafka/apache-kafka-introduction>

IoTHub Connector: <https://docs.microsoft.com/en-us/azure/hdinsight/kafka/apache-kafka-connector-iot-hub>

Create BYOK clusters: <https://docs.microsoft.com/en-us/azure/hdinsight/kafka/apache-kafka-byok>

Create HDInsight ESP clusters: <https://docs.microsoft.com/en-us/azure/hdinsight/domain-joined/apache-domain-joined-configure-using-azure-adds>

Configure Ranger policies for Kafka: <https://docs.microsoft.com/en-us/azure/hdinsight/domain-joined/apache-domain-joined-run-kafka>

Azure HDInsight Blog: <https://azure.microsoft.com/en-us/blog/tag/hdinsight/>

HDI Kafka Customer Success



"Toyota manufactures millions of cars running globally, and building a connected car platform to process real-time data at Toyota scale is a monumental challenge. To process events at Toyota's scale, technologies such as Kafka need to be leveraged. Since HDInsight is the only managed platform that provides Kafka as a managed service with a 99.9% SLA, Toyota was able to leverage the scalable technology of Kafka, Storm and Spark on Azure HDInsight. Using the HDInsight platform, we were able to deploy enterprise grade streaming pipelines to process events from millions of cars every second. This is just scratching the surface - the future of global connected cars on Azure HDInsight is bright, and we are excited for what's in store."

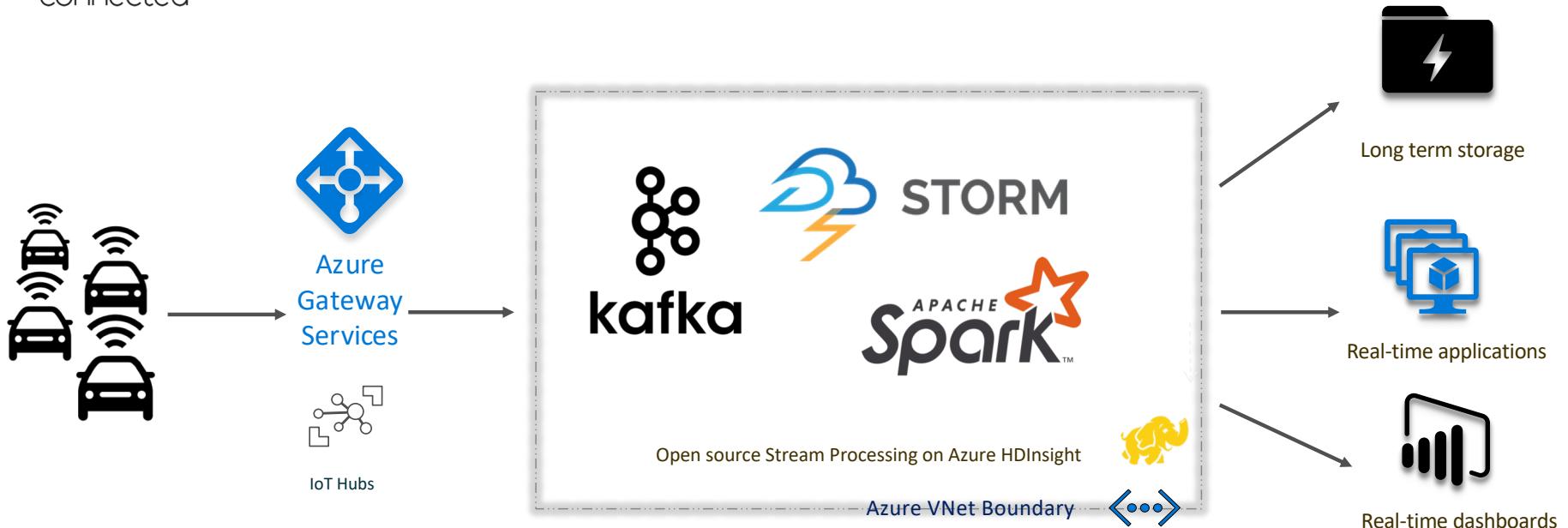
--Vijay Chemuturi, Chief Product Owner, Toyota Connected

<https://azure.microsoft.com/en-us/blog/announcing-public-preview-of-apache-kafka-on-hdinsight-with-azure-managed-disks>



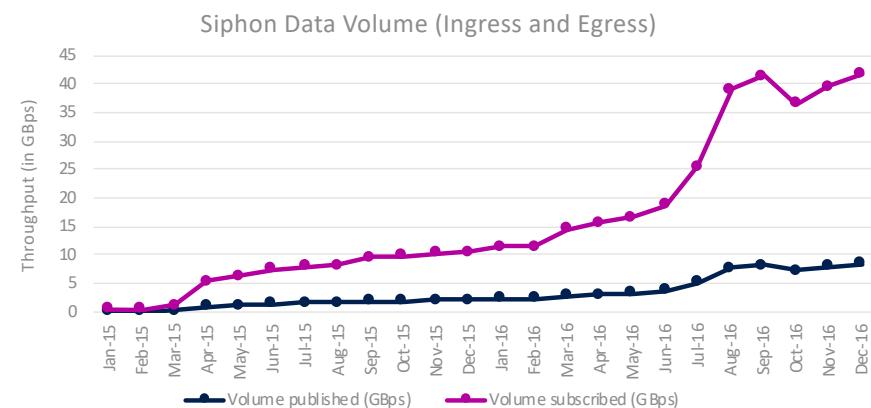
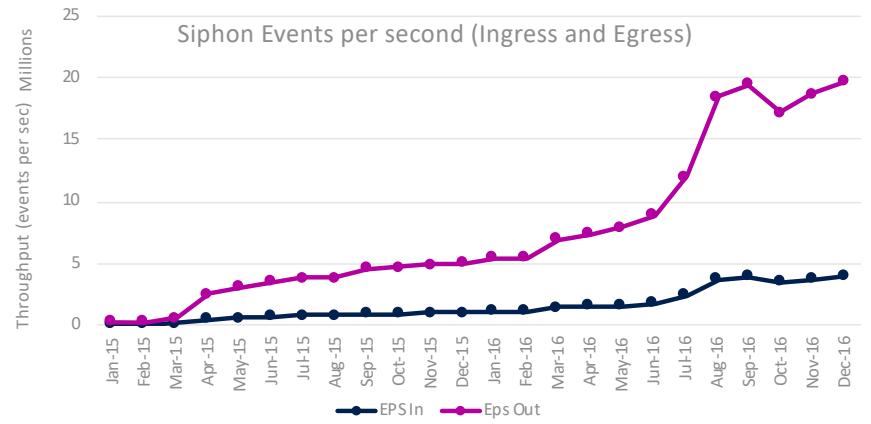
TOYOTA
connected

Connected Car Architecture powered by Azure HDInsight



MICROSOFT CONFIDENTIAL – INTERNAL ONLY

Siphon Usage



8 million

EVENTS PER SECOND PEAK INGRESS

800 TB (10 GB per Sec)

INGRESS PER DAY

1,800; 450

PRODUCTION KAFKA BROKERS; TOPICS

15 Sec

99th PERCENTILE LATENCY

KEY CUSTOMER SCENARIOS

Ads Monetization (Fast BI)

O365 Customer Fabric NRT – Tenant & User insights

BingNRT Operational Intelligence

Presto (Fast SML) interactive analysis

Delve Analytics

Kafka API for Azure Event Hub

About

Kafka API for Azure Event Hub

Event Hubs, is a fully managed, serverless, Azure streaming PaaS that is Kafkaesque.

- provides a Kafka endpoint that can be used by your existing Kafka based applications as an alternative to running your own Kafka cluster.
- supports Apache Kafka protocol 1.0 and later, and works with your existing Kafka applications
- supports streaming into Kafka API for Event Hub via MirrorMaker
- no code change, just configuration change

Kafka API for Azure Event Hub

- Throughput unit concept for provisioning scale – 1 streaming unit = 1 MBps, or 1,000 events/second of ingress
- Auto-scale when limits are hit with auto-inflate feature
- SSL for encryption
- SAS token based authentication

Additional security through event publisher

An event publisher - defines a virtual endpoint for an event hub. The publisher can only be used to send messages to an event hub. It is not possible to receive messages from a publisher.

- Event Hubs capture to archive data to object storage

Kafka API for Azure Event Hub

Not supported:

- Idempotent producer
- Transaction
- Compression
- Size-based retention
- Log compaction
- Adding partitions to an existing topic
- HTTP Kafka API support
- Kafka Connect
- Kafka Streams