

# Implementing Real-Time Analysis with Hadoop in Azure HDInsight

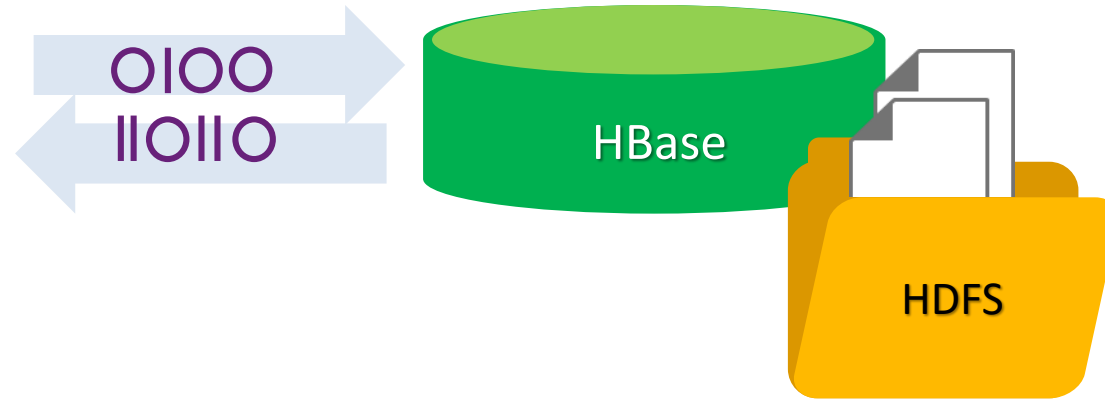
01 | Using HBase for NoSQL Data



Graeme Malcolm | Snr Content Developer, Microsoft

- What is Apache HBase?
- How is HBase Supported in Azure HDInsight?
- How Does HBase Store Data?
- How Do You Work with an HBase Table?
- How Do You Bulk Load Data into HBase?
- How Do You Query HBase Tables from Hive?
- How Do You Query HBase Tables using SQL?
- How Do You Build an HBase Client?

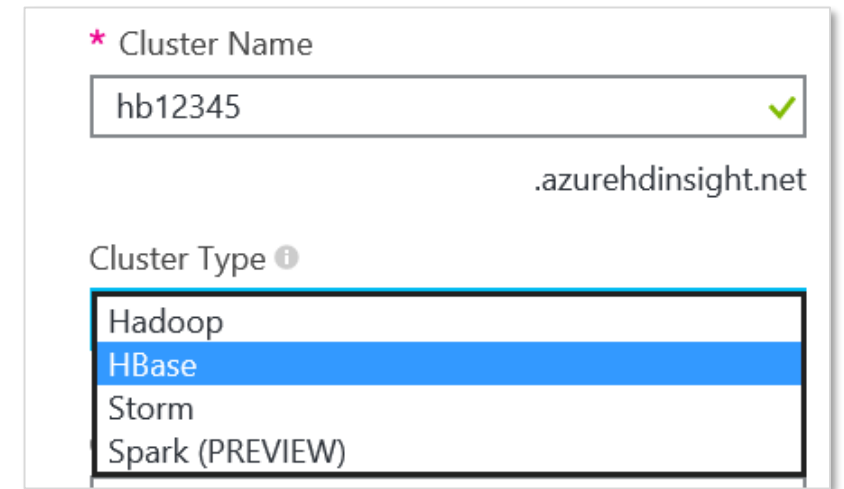
What is Apache HBase?



- A low-latency, NoSQL database built on Hadoop
- Modeled on Google's BigTable
- HBase stores data in StoreFiles on HDFS

How is HBase Supported in Azure HDInsight?

- HDInsight supports an **HBase** cluster type
  - Choose Cluster Type in the Azure Portal
- Can be provisioned in a virtual network



\* Cluster Name

hb12345 ✓

.azurehdinsight.net

Cluster Type ⓘ

Hadoop
<b>HBase</b>
Storm
Spark (PREVIEW)

The screenshot shows a portion of the Azure Portal's cluster creation interface. At the top, there is a 'Cluster Name' field with a red asterisk icon, containing the text 'hb12345' and a green checkmark. Below this is the domain '.azurehdinsight.net'. Further down is the 'Cluster Type' section, which includes an information icon (i) and a dropdown menu. The dropdown menu is open, showing four options: 'Hadoop', 'HBase' (which is highlighted with a blue background), 'Storm', and 'Spark (PREVIEW)'.

# DEMO

---

Provisioning an HBase Cluster

How Does HBase Store Data?



- Data is stored as key-value pairs
- Table schema arranges values into *column families*
- Column family schema is flexible
- Columns are row-specific

readings				
key	sensor		reading	
	id	location	datetime	value
1	Sensor1		2015-01-01	125.9
2	Sensor2		2015-01-01	152.3
3	Sensor1		2015-01-02	87.3
4	Sensor2		2015-01-02	151.8
5	Sensor1	Building 1	2015-01-03	126.3

- Cells in a table are versioned
- Each versioned cell value is indicated by a timestamp

readings				
key	sensor		reading	
	id	location	datetime	value
1	Sensor1		2015-01-01	125.9
2	Sensor2		2015-01-01	152.3
3	Sensor1		2015-01-02	87.3
4	Sensor2		2015-01-02	151.8
5	Sensor1	Building 1	2015-01-03	127.1
				147152436 126.3
				147152442 127.1

How Do You Work with an HBase Table?

```
create 'readings', 'sensor', 'reading'
```

readings		
key	sensor	reading

```
put 'readings', '2', 'reading:value', '157.6'
```

readings				
key	sensor		reading	
	id	location	datetime	value
1	Sensor1		2015-01-01	125.9
2	Sensor2	Building 2	2015-01-01	157.6

get 'readings', '2'

readings				
key	sensor		reading	
	id	location	datetime	value
1	Sensor1		2015-01-01	125.9
2	Sensor2	Building 2	2015-01-01	157.6

COLUMN	CELL
sensor:id	timestamp=142361, value=Sensor2
sensor:location	timestamp=142366, value=Building 2
reading:datetime	timestamp=142363, value=2015-01-01
reading:value	timestamp=142381, value=157.6

```
get 'readings', '2', {COLUMN => [reading:value]}
```

readings				
key	sensor		reading	
	id	location	datetime	value
1	Sensor1		2015-01-01	125.9
2	Sensor2	Building 2	2015-01-01	157.6

COLUMN	CELL
reading:value	timestamp=142379, value=152.3

```
get 'readings', '2', {TIMERANGE => [0,142380]}
```

readings				
key	sensor		reading	
	id	location	datetime	value
1	Sensor1		2015-01-01	125.9
2	Sensor2	Building 2	2015-01-01	157.6

COLUMN	CELL
sensor:id	timestamp=142361, value=Sensor2
sensor:location	timestamp=142366, value=Building 2
reading:datetime	timestamp=142363, value=2015-01-01
reading:value	timestamp=142379, value=152.3



scan 'readings'

readings				
key	sensor		reading	
	id	location	datetime	value
1	Sensor1		2015-01-01	125.9
2	Sensor2	Building 2	2015-01-01	157.6

ROW	COLUMN+CELL
1	column=sensor:id, timestamp=142356, value=Sensor1
1	column=reading:datetime, timestamp=142357, value=2015-01-01
1	column=reading:value, timestamp=142359, value=125.9
2	column=sensor:id, timestamp=142361, value=Sensor2
2	column=sensor:location, timestamp=142366, value=Building 2
2	column=reading:datetime, timestamp=142363, value=2015-01-01
2	column=reading:value, timestamp=142381, value=157.6

```
scan 'readings', {LIMIT => 1}
```

readings				
key	sensor		reading	
	id	location	datetime	value
1	Sensor1		2015-01-01	125.9
2	Sensor2	Building 2	2015-01-01	157.6
3	Sensor1	Building 1	2015-01-02	87.3

ROW	COLUMN+CELL
1	column=sensor:id, timestamp=142356, value=Sensor1
1	column=reading:datetime, timestamp=142357, value=2015-01-01
1	column=reading:value, timestamp=142359, value=125.9

```
scan 'readings', {STARTROW=>'2', STOPROW=>'3' }
```

readings				
key	sensor		reading	
	id	location	datetime	value
1	Sensor1		2015-01-01	125.9
2	Sensor2	Building 2	2015-01-01	157.6

ROW	COLUMN+CELL
2	column=sensor:id, timestamp=142361, value=Sensor2
2	column=sensor:location, timestamp=142366, value=Building 2
2	column=reading:datetime, timestamp=142363, value=2015-01-01
2	column=reading:value, timestamp=142375, value=157.6
3	column=sensor:id, timestamp=142371, value=Sensor1
3	column=sensor:location, timestamp=142372, value=Building 1
3	column=reading:datetime, timestamp=142373, value=2015-01-02

```
delete 'readings', '2', 'sensor:location'
```

readings				
key	sensor		reading	
	id	location	datetime	value
1	Sensor1		2015-01-01	125.9
2	Sensor2		2015-01-01	157.6
3	Sensor1	Building 1	2015-01-02	87.3
4	Sensor2	Building 2	2015-01-02	151.8
5	Sensor1	Building 1	2015-01-03	126.3
6	...			

```
deleteall 'readings', '4'
```

readings				
key	sensor		reading	
	id	location	datetime	value
1	Sensor1		2015-01-01	125.9
2	Sensor2		2015-01-01	157.6
3	Sensor1	Building 1	2015-01-02	87.3
5	Sensor1	Building 1	2015-01-03	126.3
6	...			

```
drop 'readings'
```

readings				
key	sensor		reading	
	id	location	datetime	value
1	Sensor1		2015-01-01	125.9
2	Sensor2		2015-01-01	157.6
3	Sensor1	Building 1	2015-01-02	87.3
5	Sensor1	Building 1	2015-01-03	126.3
6	...			

# DEMO

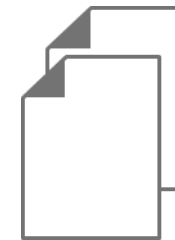
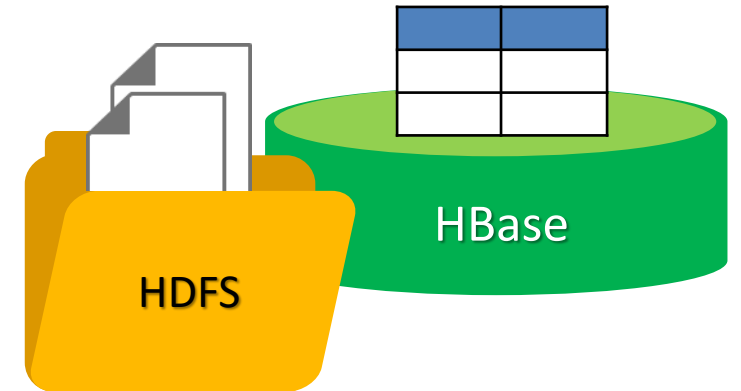
---

Working with an HBase Table

How Do You Bulk Load Data into HBase?



1. Upload data to HDFS  
–in Azure Storage
2. Import into a StoreFile
3. Load The StoreFile to an HBase  
table



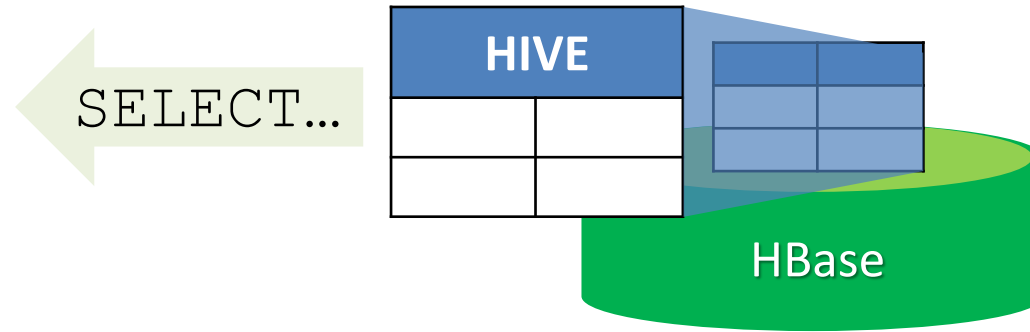
Load into HBase  
Load into HBase

# DEMO

---

Bulk Loading Data into an HBase Table

How Do You Query HBase Tables from Hive?



```
CREATE EXTERNAL TABLE hivetable
(key STRING, col1 STRING, col2 STRING)
STORED BY 'org.apache.hadoop.hive.hbase.HBaseStoragehandler'
WITH SERDEPROPERTIES
('hbase.columns.mapping' = ':key,cf:col1, cf:col2')
TBLPROPERTIES('hbase.table.name' = 'hbtable')
```

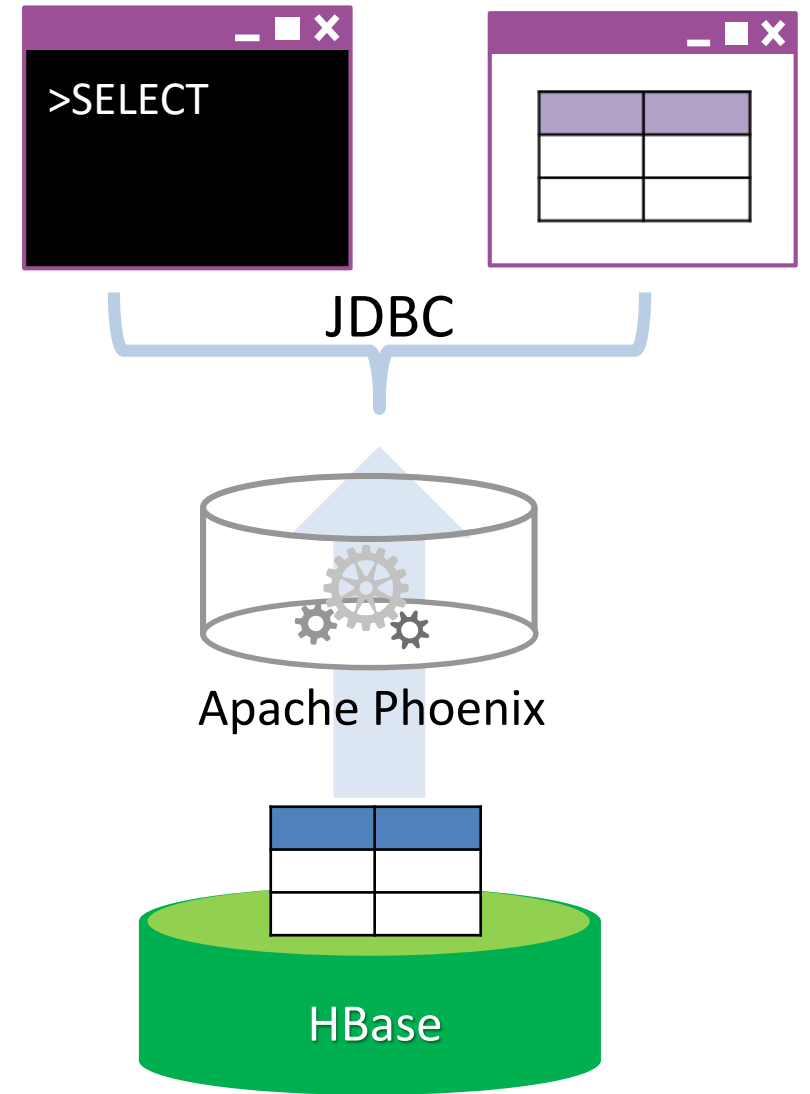
# DEMO

---

Querying HBase from Hive

How Do You Query HBase Tables using SQL?

- Apache Phoenix
  - Relational database engine built on HBase
  - Included in Azure HDInsight
- JDBC Interface
  - Clients connect using JDBC
  - SQLLine client included in HDInsight



# DEMO

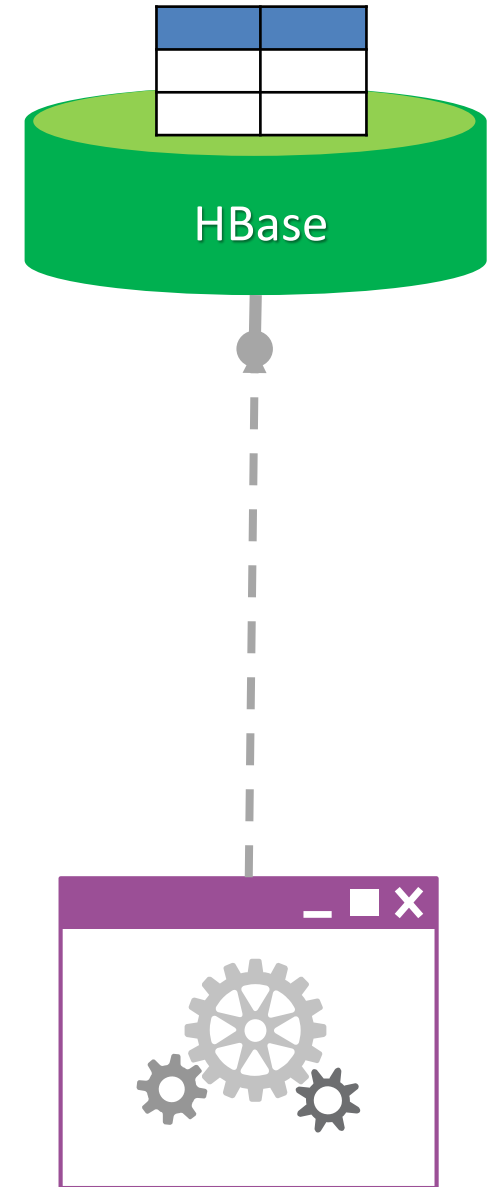
---

Using SQLLine to Query an HBase Table



How Do You Build an HBase Client?

- `org.apache.hadoop.hbase._`
  - Java libraries for HBase
    - `HBaseConfiguration`
    - `HTable`
    - Put, Get, Scan
- **HBase Rest API Client**
  - .NET wrapper around HBase REST API
    - `HBaseClient`
    - `Scanner`
    - `CellSet`
    - `Cell`



# DEMO

---

Creating an HBase Client

- What is Apache HBase?
- How is HBase Supported in Azure HDInsight?
- How Does HBase Store Data?
- How Do You Work with an HBase Table?
- How Do You Bulk Load Data into HBase?
- How Do You Query HBase Tables from Hive?
- How Do You Query HBase Tables using SQL?
- How Do You Build an HBase Client?



# Microsoft

©2014 Microsoft Corporation. All rights reserved. Microsoft, Windows, Office, Azure, System Center, Dynamics and other product names are or may be registered trademarks and/or trademarks in the U.S. and/or other countries. The information herein is for informational purposes only and represents the current view of Microsoft Corporation as of the date of this presentation. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information provided after the date of this presentation. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS PRESENTATION.