Azure Support for 4 Projects.

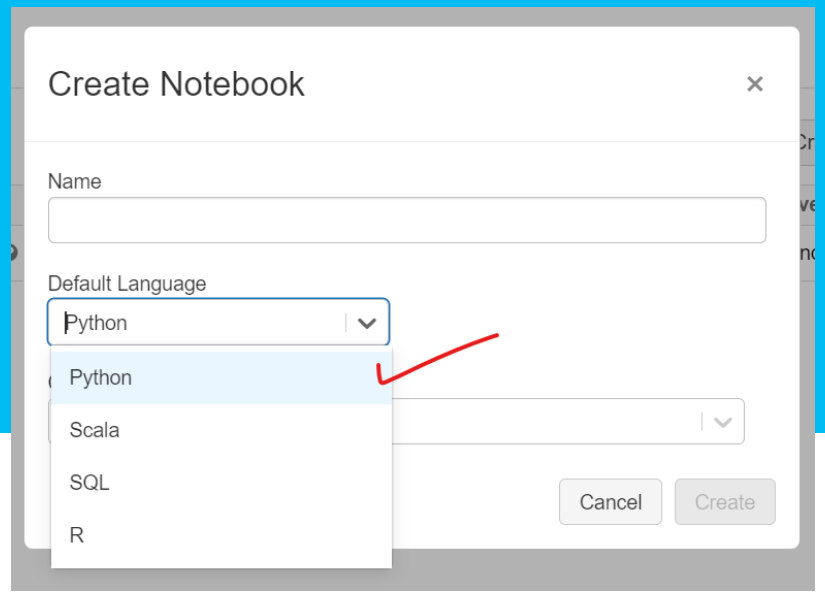
- Python for Apache Spark
- Apache Kafka
- Apache Beam
- Apache Cassandra

PySpark in Azure

Python API for Apache Spark

1

1. Use Azure Databricks.
2. Import Python Notebook into Databricks Workspace or code from scratch.
3. PySpark (Native Python API for Apache Spark Programming)
4. Supports:



2

1. Use Spark Pools in Synapse.
2. Synapse Notebook web interface.
3. Exports in .ipynb
4. Supports:

%%pyspark	Python	Execute a **Python** query against Spark Context.
%%spark	Scala	Execute a **Scala** query against Spark Context.
%%sql	SparkSQL	Execute a **SparkSQL** query against Spark Context.
%%csharp	.NET for Spark C#	Execute a **.NET for Spark C#** query against Spark Context.

R currently NA (2021)

Apache Kafka in Azure

Open-source Distributed Event-Streaming

1

Use Confluent on Azure

- Original creators of Apache Kafka.
- Fully Managed Apache Kafka on Azure.
- Created by Confluent; as a Solution on Azure. Comes with managed connectors with Blob, ADLS Gen2, SQL Server etc.

Try on Azure Marketplace > Click here.
Explore Pricing here: Basic, Standard & Dedicated.

Better to use if you're using OR wanting to use features like:

- Kafka client-side end-to-end block compression.
- Log compaction (evicting all except last record of each key from a partition)
- Kafka Streams client library for Java/Scala apps integration
- Confluent's "ksqlDB" product in your solution architecture

Confluent Cloud

2

Use Azure Event Hub for Kafka (using Kafka protocol)

- Event Hubs end-point for Apache Kafka compatible producer & consumer API.
- Supports API clients at v1.0+.

Key difference between: Apache Kafka & Event Hubs:

1. Kafka you either manage on your OWN, or via Confluent. Event Hubs is a managed service from Microsoft.
2. Scale in Event Hub is managed by TU.

Better to use if you're using OR wanting to use features like:

- Competing-consumer queue pattern (including Idempotency Patterns)
- Pub/Sub at a level that allows subs access to incoming messages based on server-evaluated rules other than plain offsets.
- Tracking of lifecycle of a job initiated by a message
- Sidelining faulty messages into a "dead-letter" queue

Excellent document is: Asynchronous messaging options in Azure.

Apache Beam in Azure

Open-source unified model for Batch + Streaming

A pipeline reads from an external source (file/db), writes output to a sink.

- Built-in I/O Transforms supported by Beam is > [Built-in I/O Transforms (apache.org)]

Databases:

Java SDK

Database

These I/O connectors are used to connect to database systems.

Name	Description	Javadoc
CassandraIO	An IO to read from Apache Cassandra.	org.apache.beam.sdk.io.cassandra.CassandraIO
HadoopFormatIO (guide)	Allows for reading data from any source or writing data to any sink which implements Hadoop InputFormat or OutputFormat.	org.apache.beam.sdk.io.hadoop.format.HadoopFormatIO
HBaseIO	A bounded source and sink for HBase.	org.apache.beam.sdk.io.hbase.HBaseIO
HCatalogIO (guide)	HCatalog source supports reading of HCatRecord from a HCatalog-managed source, for example Hive.	org.apache.beam.sdk.io.hcatalog.HCatalogIO
KuduIO	A bounded source and sink for Kudu.	org.apache.beam.sdk.io.kudu
SolrIO	Transforms for reading and writing data from/to Solr.	org.apache.beam.sdk.io.solr.SolrIO
ElasticsearchIO	Transforms for reading and writing data from/to Elasticsearch.	org.apache.beam.sdk.io.elasticsearch.ElasticsearchIO
BigQueryIO (guide)	Read from and write to Google Cloud BigQuery.	org.apache.beam.sdk.io.gcp.bigquery.BigQueryIO
BigTableIO	Read from and write to Google Cloud Bigtable.	org.apache.beam.sdk.io.gcp.bigtable.BigtableIO
DatastoreIO	Read from and write to Google Cloud Datastore.	org.apache.beam.sdk.io.gcp.datastore.DatastoreIO
SnowflakeIO (guide)	Experimental Transforms for reading from and writing to Snowflake.	org.apache.beam.sdk.io.snowflake.SnowflakeIO
SpannerIO	Experimental Transforms for reading from and writing to Google Cloud Spanner.	org.apache.beam.sdk.io.gcp.spanner.SpannerIO
JdbcIO	IO to read and write data on JDBC.	org.apache.beam.sdk.io.jdbc.JdbcIO
MongoDbIO	IO to read and write data on MongoDB.	org.apache.beam.sdk.io.mongodb.MongoDbIO
MongoDbGridFSIO	IO to read and write data on MongoDB GridFS.	org.apache.beam.sdk.io.mongodb.MongoDbGridFSIO
RedisIO	An IO to manipulate a Redis key/value database.	org.apache.beam.sdk.io.redis.RedisIO
DynamoDBIO	Read from and write to Amazon DynamoDB.	org.apache.beam.sdk.io.aws.dynamodb.DynamoDBIO org.apache.beam.sdk.io.aws2.dynamodb.DynamoDBIO
ClickHouseIO	Transform for writing to ClickHouse.	org.apache.beam.sdk.io.clickhouse.ClickHouseIO

Python SDK

Database

These I/O connectors are used to connect to database systems.

Name	Description	pydoc
BigQueryIO (guide)	Read from and write to Google Cloud BigQuery.	apache_beam.io.gcp.bigquery
BigTableIO	Read from and write to Google Cloud Bigtable.	apache_beam.io.gcp.bigtableio module
DatastoreIO	Read from and write to Google Cloud Datastore.	apache_beam.io.gcp.datastore.v1new.datastoreio
MongoDbIO	IO to read and write data on MongoDB.	apache_beam.io.mongodbio

Use Beam to connect with Azure Cosmos DB's API for MongoDB via MongoDbIO connector.

Example Project > https://bit.ly/3zl9Jqe

Connectivity is one thing; Performance is another. Need to test Performance in PoC.

Apache Cassandra in Azure

Open-source NoSQL Distributed Database

1 Inside Azure VM.

Benefits from more mem. Recommendations come from Performance Experiments, access in GitHub.

- Standard_DS14_v2 OR, Standard_DS13_v2 VMs, OR Standard_L16s_v2.
- Data & Commit Logs stored on a striped set of 2/4 1-TB P30.
- Use 1-2 TB data/VM with enough free space for compaction.
- For highest IOPS using premium managed disks, create a stripe-set instead of larger single disks.
- Azure Ultra Disk can also be evaluated for Cassandra workloads that require smaller disk capacity.
- Accelerated Networking on NIC of Cassandra node and VMs running client apps.
- For achieving "low" random-access disk latency for Cassandra read workloads, use Azure managed disks with ReadOnly caching enabled.
- Set Linux block dev read-ahead setting = 8 KB.
- Set strip set (e.g., /dev/md0) to 8 KB read-ahead.
- Commit logs should be on premium managed disks.
- You maintain IaaS and DB.

Additional Best Practices for Cassandra in VM.

2 Managed DBaaS.

Azure Managed Instance for Cassandra.

- Currently in Public Preview (Dt. 26-08-2021).
- Automated deployment, management (patching & node health).
- Support for Hybrid scenarios.
- Support creation of a Multi-Region Cluster.
- Automated scaling of Nodes in Cluster.
- Pricing is flexible, on-demand, with no licensing fees.
- Cassandra repairs are done automatically for you via Cassandra-repear.io.
- SLA: Currently in Public Preview, hence, does not come with SLA. Once comes out in "GA", SLA will be available.

3 DataStax Astra on Azure.

DataStax Astra built on Apache Cassandra.

- OSS, serverless, DBaaS.
- Astra, now in the Microsoft Azure Marketplace!
- Service plan is Pay-Per-Use.
- DataStax Enterprise also available via Marketplace. Accelerates cloud-native and bare-metal performance with specialist workloads including graph, search, analytics etc. Starter bundle (3-node) starts from $5K/mth.

4 Cosmos DB's API for Cassandra DB

Use apps written for Apache Cassandra.

- Use existing Apache Drivers compliant with CQLv4.
- The Cassandra API enables you to interact with data stored in Azure Cosmos DB using the Cassandra Query Language (CQL) , Cassandra-based tools (like cqlsh) and Cassandra client drivers that you're already familiar with.
- No Ops.
- Cosmos DB guarantees low latency reads and writes at 99th percentile.
- Backed by SLAs.
- Use existing Code & tools.
- Throughput across all regions.
- Ability to globally distribute data across all Azure regions. SLA of 99.99% HA within a single region, 99.999% read and write HA across multiple-regions.
- Event Sourcing: Cassandra API provides access to a persistent change log, the Change Feed, which can facilitate event sourcing directly from the database.

DataStax