Contents

[**Implementing Monitoring for Data Factory** 2](#_Toc38751103)

[**Implementing HDInsight Monitoring.** 7](#_Toc38751104)

**Monitoring Microsoft Azure Data pipelines and Processing**

Data pipelines and processing take raw data and enable real, meaningful insights that are usable by the business.

This course talks about monitoring options related to the various data pipeline and processing services that are available in Azure. Here we focus on monitoring Azure Data Factory, HDInsight and Databricks

Data Pipelines also provide a way used to transform and clean data.

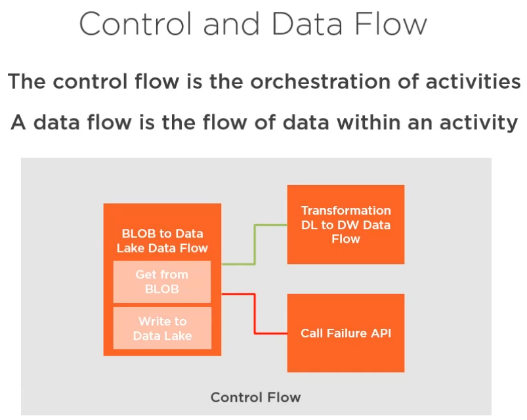
**Why is monitoring Imp:**

Monitoring is required to ensure those pipelines that give us all this great insight are functioning, and therefore that analysis we're getting, that business intelligence, is accurate.

**Control and Data Flow:**

The control flow is the orchestration, the scheduling of activities.   
A data flow is the flow of the actual data within an activity.

A control flow is made up of at least one activity. Within that activity, there shall be distinct steps.

As shown in the diagram there could be a success flow and a failure flow. Activities in failure flow are executed when the pipeline faces any failure.

Control flow (orchestration) is responsible for calling activities that each have a data flow within them.

**Types of Data Flow:**

1. **Wrangling data flows:** Preparing and cleaning data. This typically integrates with things like Power Query, and this is all about data preparation, adding columns, splitting columns etc.
2. **Mapping Data flows:** Transform the data to get it into the shape that we need for that analysis. This includes sorts, joins, merges, inserts, alters, aggregate, lookups, audits, all transformations.

**Problem with ETL approach:** With ETL, the issue was that we only kept the data we knew we needed. This was to optimize the storage. But it left no room to go back to the data later and rethink on the approach as we have new requirement. Today with very cheap lakes of storage, it is now typical to get the data in and then just store everything in its raw format, and then transform and shape as required. This allows us to cater to the requirements of today and as well as tomo.

**Azure Solutions:**

Azure solutions that are used around the control and the data flows are as follows:

1. **Azure Data Factory** for Control Flow. It provides orchestration and scheduling.
2. Data Flow solutions include **Azure Databricks, Azure HDInsight.** Note that even Azure Data Factory is now in the data flow business. It actually uses Databricks behind the flow.

Using Azure Data factory we can create two types of flow: 1) Mapping data flow 2) Wrangling data flow. When we create a dataflow we have the option to choose between the two options.

With the dataflow option, Azure data factory provides both Control and Data flow.

Azure Data Factory – Data flow provides a data flow debug option – Behind the scenes it actually spins up the Databricks cluster where we could debug as we edit. When we are done, the cluster is destroyed within an hour.

## **Implementing Monitoring for Data Factory**

Primary role of Azure Data Factory is to provide the orchestration and scheduling of processes related to big data.

It is a fully managed service with a graphical user interface.

In ADF we create a pipelines. These are the control flows that will interact with the various activities, and those activities will have the data flows.

Azure Data Factory primarily provides control flow. However it also had data flow capabilities that leverage Databricks behind the scenes.

ADF integrate with key data flow services like Databricks and HDInsight, and it has its own native capabilities, i.e. it can perform types of its own compute like performing copy operations, certain types of logic. And so, when compute is required, it leverages something called an integration runtime.

ADF also integrates with SSIS. We can lift and shift my SSIS jobs from SQL Server and directly run them on Data Factory with great compatibility.

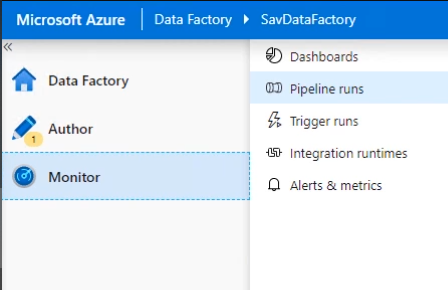
ADF has over 75 of these different connections just built in, so it's great for being able to connect to a whole variety of cloud-based and on-premises based services.

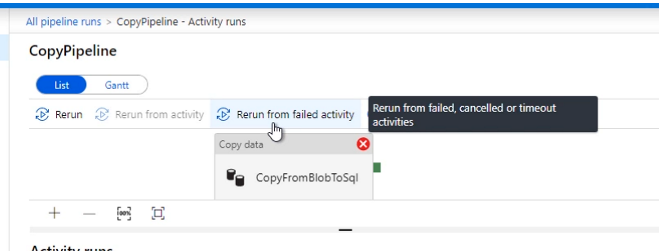
ADF can easily integrate with customer batch services, Databricks, DataLake Analytics, HDinsight. We just drag and drop these onto the graphical interface.

ADF activities run in an integration runtime if they're using some native compute within the Azure Data Factory. There are different types of integration runtime. Simplest one is Azure integration runtime. This is Serverless and compute based. We can also use self-hosted runtime. This is used when we have sources/targets on-premises. This prevents data from having to bounce up to the cloud and back again. Using self-hosted runtime allows us to perform manipulation locally. The control flow might be going to the cloud, but the data can stay within my on‑premises environment. We have another integration runtime service for SSIS jobs.

Types of triggers provided by ADF: schedule‑based, tumbling window, or event‑based (using integration with event grid)

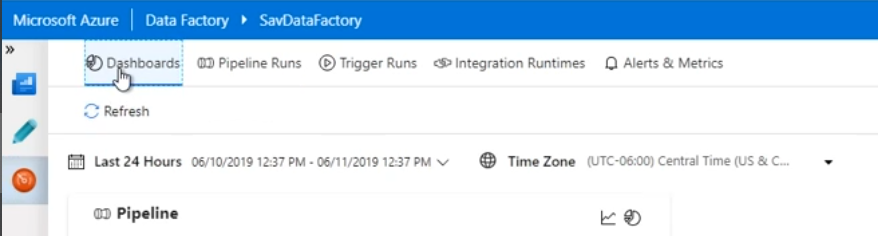
Primarily for ADF monitoring we think about trigger runs, pipeline runs, activity runs.

When a pipeline fails, we also have option to “rerun” the entire pipeline or “rerun from the failed activity”. The 2nd option will not rerun the parts that worked, it will actually jump straight to the point that failed and then carry it on.



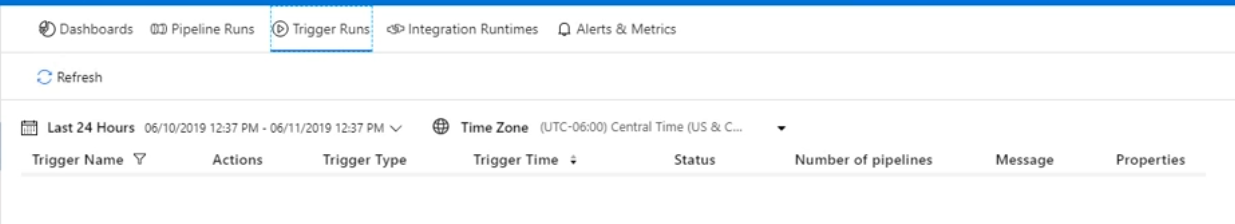
ADF debugger provides little monitoring service as we develop our pipelines. It allows us to run the pipeline, have breakpoints. However that's not really monitoring, but a part of the developer experience, that debug functionality allows us to see exactly what's happening. Bulk of the monitoring comes via monitoring interface.

Following is how the monitoring interface looks like:

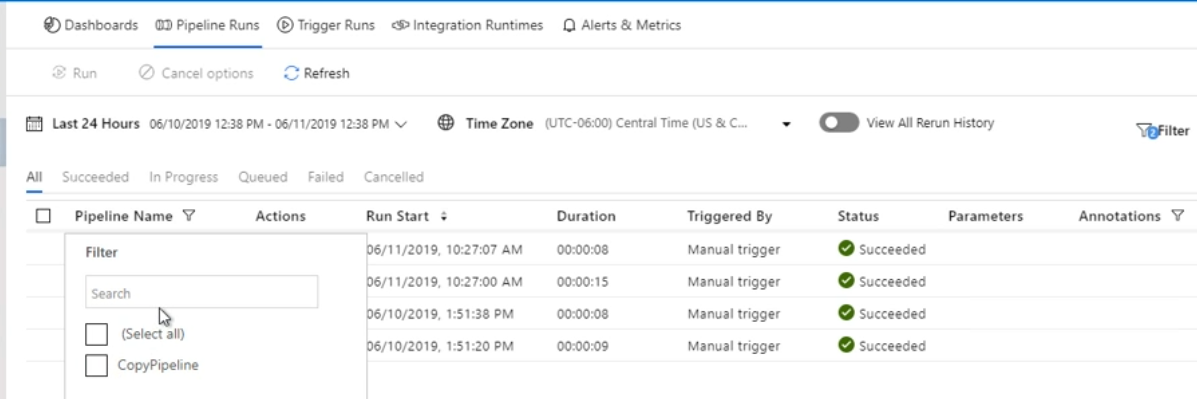


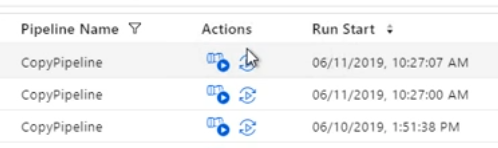
The monitoring interface starts with **Dashboard**. This just gives us a very quick overview of what's happening in the environment. This dashboard will show, by default, the last 24 hours, but we can change that time window.

Under the **“Trigger Runs”**, we can see all the triggers. Triggers may be scheduled or event based.

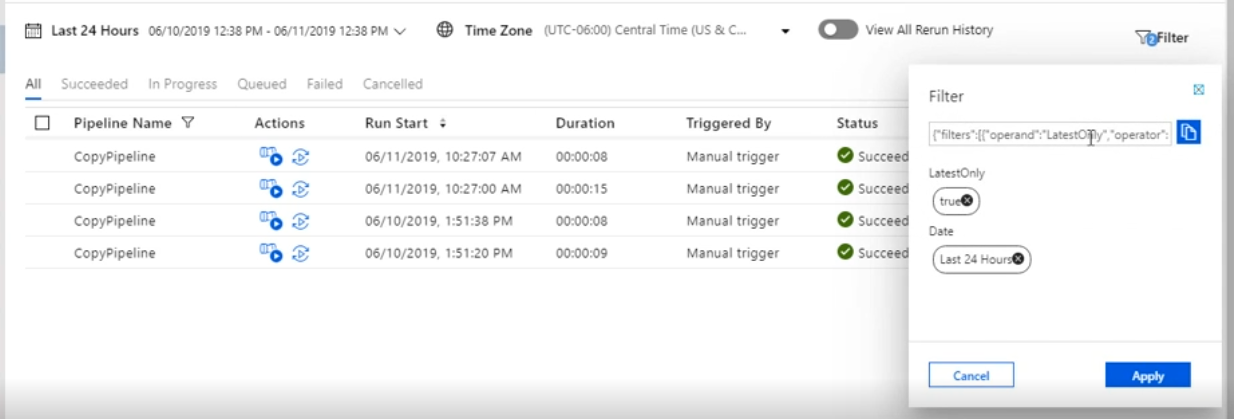


The important tab is “Pipeline Runs”. Here we can see all the executed pipelines. We have options to filter only the pipelines of our interest. For any pipeline execution, we have two options: To “view Activity Runs” OR “Rerun” it. If we click on the “View Activity Runs” we could see all the activities within the pipeline execution.





Note that we can also have overall Filter as shown below:



Viewing activities:

Dashboards Pipeline Runs Trigger Runs Integration Runtimes 
All Pipeline Runs / CopyPipeline - Activity Runs 
Q Alerts & Metrics 
Rerun 
Rerun from activity 
Activity Runs 
Pipeline Run ID 
6f9d493b-790d-4ed9-aa23-7819af4d2637 
All Succeeded In Progress Timed Out Failed Cancelled 
NAME 
CopyFromBlo... 
ACTIVITY TYPE 
Copy 
ACTIONS 
RUN START 
06/11/2019 10:27:08 AM 
INTEGRATION RUNTIME 
DefaultlntegrationRuntime (South Central US) 
USER PROPERTIES 
00:00:08 
Succeeded 
RUNID 
616ä7456-16 

When looking at an activity, we can have a look at the input, output of that activity, amount of data read, the amount of data written, the durations, runId, and integration runtime. We could also click on the little pair of glasses to view details. Following is how it looks for a copy activity:

O Dashboards 
All Pipelin 
Rerun 
Activity 
Pipeline Run 
All Succ 
CopyFrom 
Pipeline Runs Trigger Runs 
Refresfi 
Details 0__ 
Integration Runtimes 
Q Alerts & Metrics 
Learn more on copy performance details from here. 
Succeeded 
Azure Blob Storage 
SQL 
Data written: 
Rows written: 
Azure SQL Database 
Data read: 
Files read: 
Rows read: 
Peak connections: 
52 byte 
4 
Peak connections: 
Throughput: 
84 byte 
4 
10 byte/s 

The “Integration Runtimes” Tab we can see the status of our integration runtimes. By default we shall see the Automatic Azure runtime. If we had SSIS or Self Hosted runtime, we could see them here.

Microsoft Azure Data Factory SavDataFactory 
Dashboards Pipeline Runs Trigger Runs 
Refresh 
Time Zone (UTC-06:00) Central Time (LIS & 
All Self-hosted Azure Azure-SSIS 
o 
Integration Runtimes 
Q Alerts & Metrics 
o 
Created on 
Name 
Auto Resolvelntegrationiu ntime 
Actions 
Type 
Azure 
Sub-type 
Public 
Status 
e Running 
Region 
Auto Resolve 
AutoResoIveIntegrationRuntime 

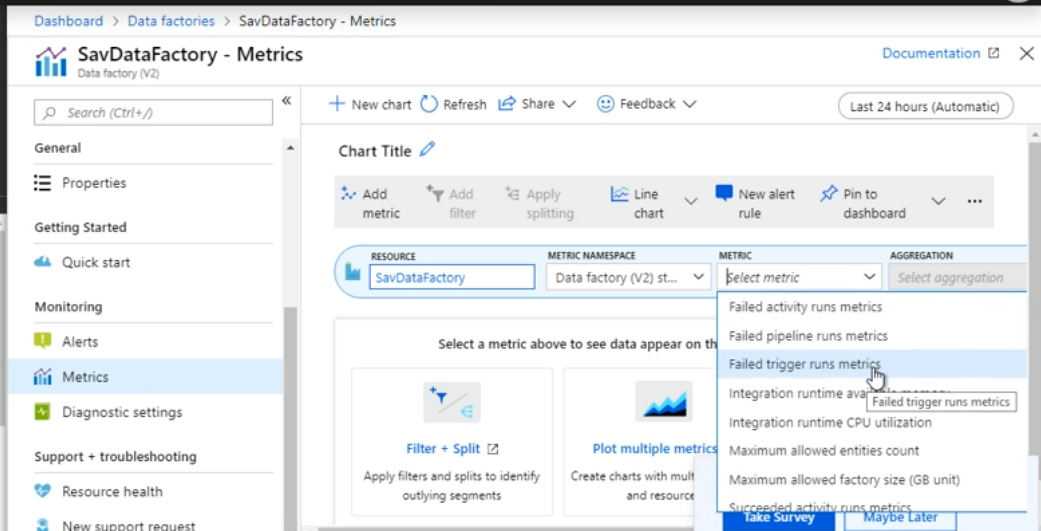
Again under actions, we can see details about activities actually using that integration runtime. Under Actions we can also view the details of the integration runtime as shown below:

Dashboards Pipeline Runs Trigger Runs Integration Runtimes Q Alerts & Metrics 
Integration Runtimes / AutoResolvelntegrationRuntime / Resource Monitor (Details) 
Edit 
e 
STATUS 
Running 
Activities 
Details 
Azure 
REGION 
Auto Resolve 

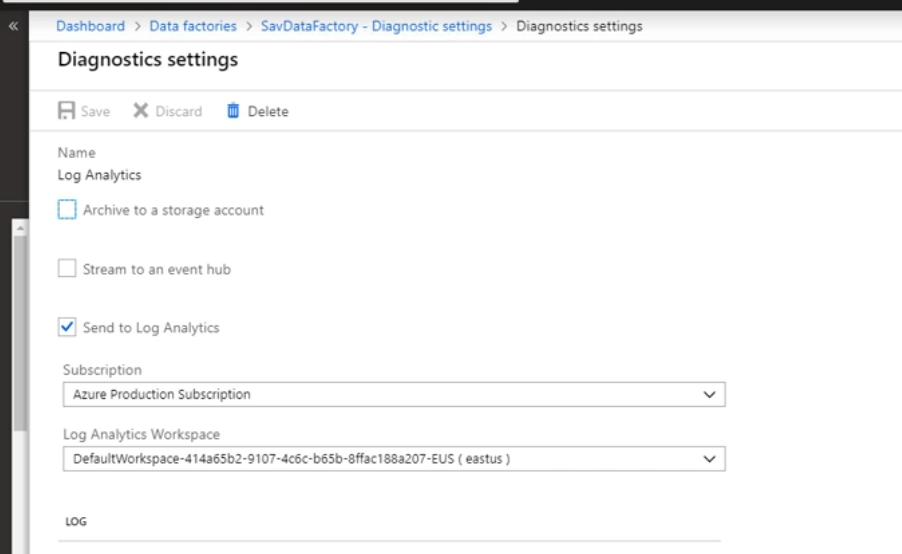
Under “Alerts & Metrics” we can see Metrics and create Alert Rules.

Microsoft Azure I Data Factory SavDataFactory 
Dashboards Pipeline Runs Trigger Runs Integration Runtimes 
Refresh iiii Metrics QC New Alert Rule 
Q Ale 
Metrics 

Clicking on “Metrics” it takes us to Azure Portal where we can view metrics. We can see multiple metrics like Failed activity runs, failed pipeline runs, failed trigger runs, integration runtime available memory, CPU utilization, maximum allowed entities count. And for all of these different metrics we could actually create an alert on of them.



Note that we also have “Diagnostics Settings”, Using this we can send metrics, activity runs, pipeline runs, and Trigger runs to one of the 3 destinations: storage account, an event hub, Log Analytics



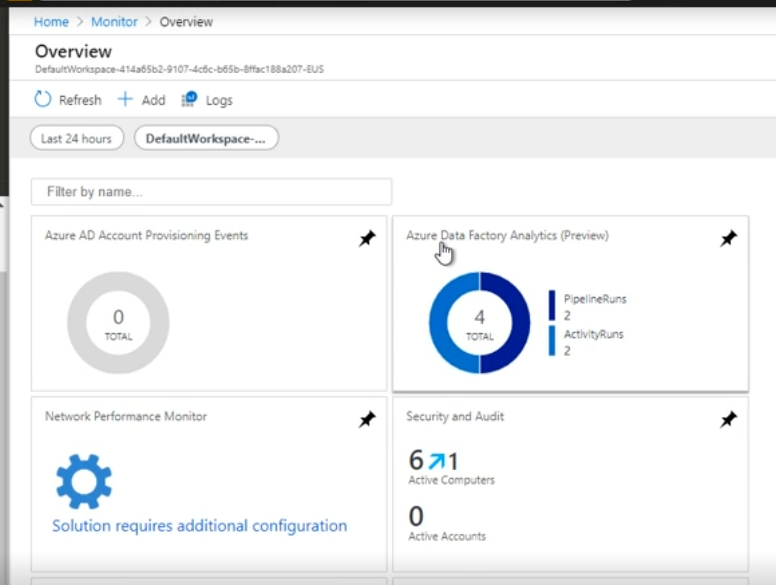
When we send to Log Analytics, we can configure how long to retain the logs using Log Analytics configuration. Once we actually have data from Data Factory being sent to Azure Monitor Logs or Log Analytics, we can create our own views.

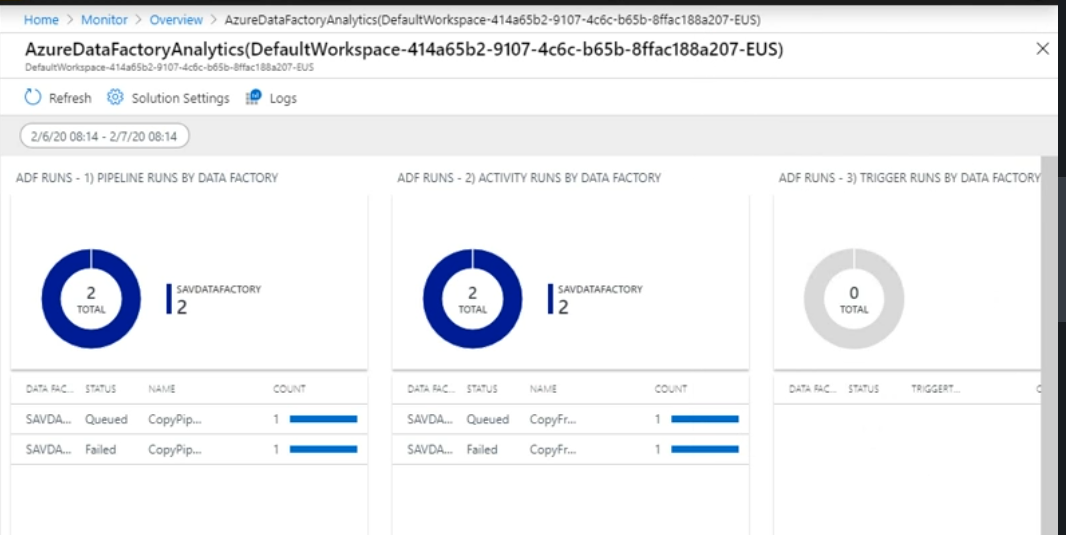
**Azure Data Factory Analytics:**

Microsoft also has their own Azure Data Factory analytics. Using this we can get pre-canned views.  
We can find this under MarketPlace. When we create this, it prompts us to provide the Log Analytics Workspace to which we want to deploy. This would be the one to which we have already configured Data Factory to send the information to.

Products > Azure Data Factory Analytics (Preview) 
Azure Data Factory Analytics (Preview) 
GET IT NOW 
IT & Management Tools 
License Agreement 
Privacy Policy 
Microsoft 
46 (8) 
Overview Reviews 
Azure Data Factory Management Solution Service 
Pack 
This solution provides you a summary of overall health of your 
Data Factory, with options to drill into details and to troubleshoot 
unexpected behavior patterns. 
With rich, out of the box views you can get insights into key 
processing including: 
• At a glance summary of dätä factory pipeline, activity and trigger 
runs 
• Ability to drill into data factory activity runs by type 
• Summary of data factory top pipeline, activity errors 
pre-requisite: To take advantage of this solution. Data Factory 
should enable Log Analytics to push diagnostic data to OMS 
o 
O 
save for later 
o 
O 
O 

Once we have it deployed, we can go to Azure Monitor. Under insights we can see the “Azure Data Factory Analytics”





The native monitoring of an Azure Data Factory instance, the pipelines run, the failures, will only show me the data for that instance. With the “Azure Data Factory Analytics”, if we configured our Data Factory instances to all point to the same Azure Monitor Logs instance, the same Log Analytics workspace, we would now be able to see all of them from this nice single view.

## **Implementing HDInsight Monitoring.**

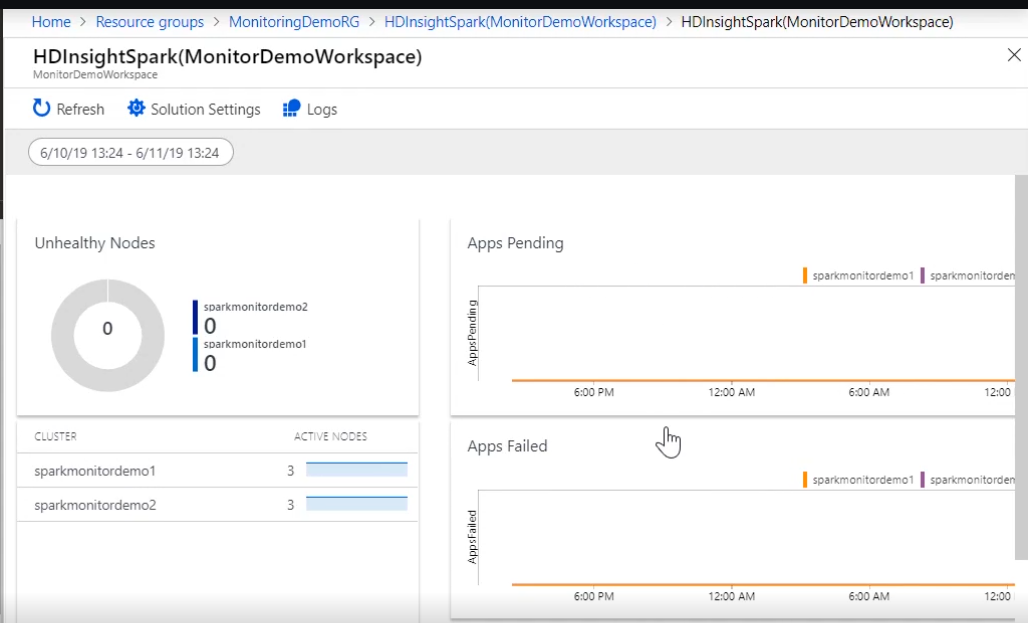
HDInsight is the Azure-managed offering that provides a cloud distribution of Hadoop. HDInsight provides this Hadoop platform on which we can run a number of open-source frameworks that provide different types of big data processing. Different open source frameworks can include Apache Hadoop, Spark, Kafka. Bulk of the monitoring we shall do is within these open-source platforms. They are not actually on Azure. Ex: Ambari: This comes installed on every cluster. There's no additional cost. But that is going to drive most of the interactions we have with these deployed frameworks.

We can still push log data to log analytics through diagnostic settings of the cluster, but most of our interactions would be through Ambari and other open source components.

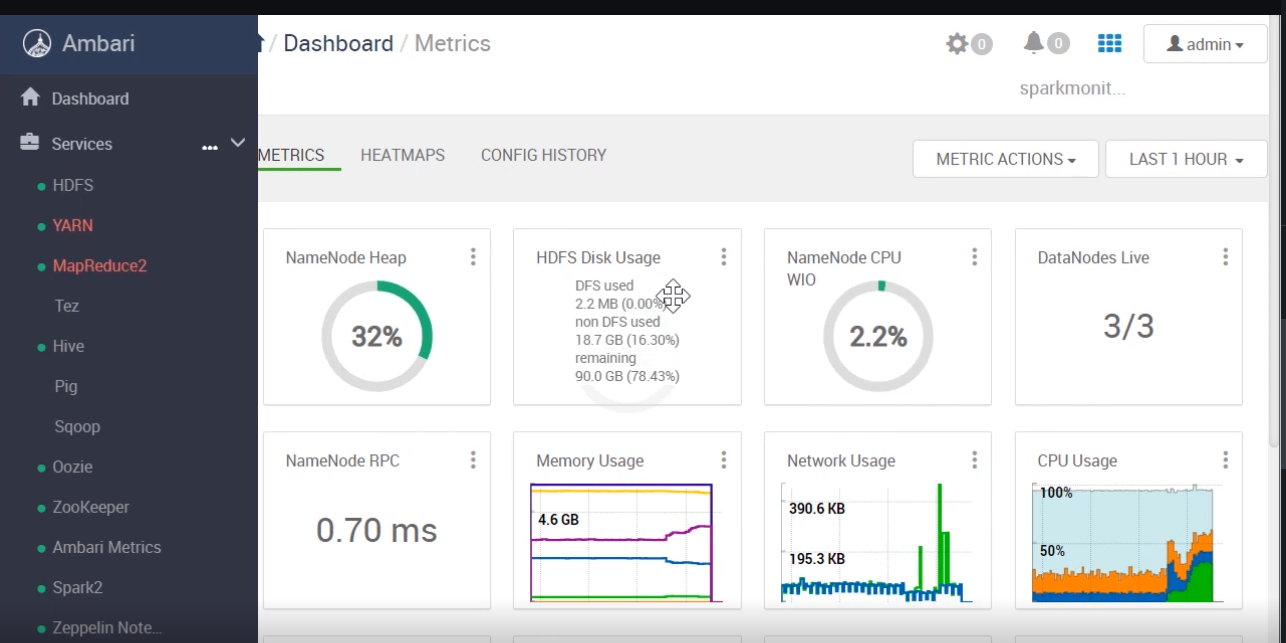
Types of clusters: 1) Batch jobs – Spark Cluster 2) Streaming – Kafka cluster 3) Interactive query – Hive 4) No SQL Storage – Hbase

Depending upon the type of cluster the monitoring strategy changes. For Spark jobs we may be interested in no of success and failure jobs. For Streaming we may be interested in quantity and latency.

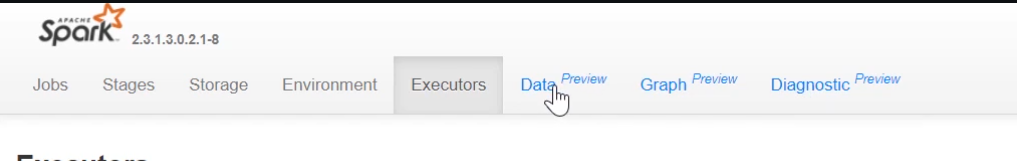
Just like Data factory, Microsoft also provides monitoring solutions for all types of Clusters. We need to configure diagnostic settings to send the diagnostic, the metrics, the logs, to the Log Analytics workspace. And then from these monitoring solutions, we can then monitor multiple clusters. We can search for these monitoring solutions in Azure Market Place.



For details, we need to login to Ambari. However note that with Ambari we can only view details for a single cluster. Following is how the interface looks like:



From the Ambari UI, we can go to the Spark history server UI. The Spark Application UI shows tabs like Jobs, Stages, Storage, Environment, Executors. These are just part of Spark. Apart from this we have other tabs like Data, Graph and Diagnostic as shown below. These are things Microsoft have added on top to bring extra value via HDInsight.



The Graph Tab would be useful as it allows us to playback jobs and stages and see what is happening over time.

**Azure Databricks:**

This is fully managed Apache Spark-based analysis service.

It is fully managed, first-party analytics service. It was really built in conjunction between Microsoft Azure team and the Databricks Company.

So this unified analytics platform can be used for a number of different purposes, primary being machine learning.

Clusters are created that are customer specific. This is not a shared platform. But these dedicated resources can scale from one to thousands. I pay for those nodes that are running for me, but with auto-scale, it can adjust.

Azure Databricks provides tight integration with Azure. Ex: we can leverage Azure Active Directory. There's a native connector for things like SQL Data Warehouse, services like Cosmos DB, Data Lake, even HDInsight.

Large no of language support: SQL, Python, Scala, R etc

We can use Data Factory notebook to call a Databricks notebook to actually go and perform the transformation, the shaping, the modeling of the data.

Databricks provides collaborative workspaces where engineers, scientists, business analysts, can all work together on the data.

Note that Azure Databricks is a fully managed service. So we actually have to go and do some configuration if we want to drive data into Log Analytics. And then from Log Analytics, we could graph the data. We could use things like Grafana, DataDog on top to display the data.