

Microsoft Azure

Azure HDInsight에서 Apache Spark 사용하기

이 문서는 Microsoft Partner인 [파트너사 이름]의 [요청자 이름]님께 제공된 문서입니다.

요약

이 내용들은 표시된 날짜에 Microsoft에서 검토된 내용을 바탕으로 하고 있습니다. 따라서, 표기된 날짜 이후에 시장의 요구사항에 따라 달라질 수 있습니다. 이 문서는 고객에 대한 표기된 날짜 이후에 변화가 없다는 것을 보증하지 않습니다.

이 문서는 정보 제공을 목적으로 하며 어떠한 보증을 하지는 않습니다.

저작권에 관련된 법률을 준수하는 것은 고객의 역할이며, 이 문서를 마이크로소프트의 사전 동의 없이 어떤 형태(전자 문서, 물리적인 형태 막론하고) 어떠한 목적으로 재 생산, 저장 및 다시 전달하는 것은 허용되지 않습니다.

마이크로소프트는 이 문서에 들어있는 특허권, 상표, 저작권, 지적 재산권을 가집니다. 문서를 통해 명시적으로 허가된 경우가 아니면, 어떠한 경우에도 특허권, 상표, 저작권 및 지적 재산권은 다른 사용자에게 허용되지 아니합니다.

© 2017 Microsoft Corporation All right reserved.

Microsoft® 는 미합중국 및 여러 나라에 등록된 상표입니다.

이 문서에 기재된 실제 회사 이름 및 제품 이름은 각 소유자의 상표일 수 있습니다

문서 작성 연혁

1. 이 문서는 2017년 01월 09일 한국 마이크로소프트(유), GPS Korea의 Partner Technical Consultant 송민경에 의해 처음 작성 되었습니다. (Version 1.0.0.0)
2. 이 문서는 2017년 04월 07일 한국 마이크로소프트(유), GPS Korea의 Partner Technical Consultant 송민경에 의해 수정 되었습니다. (Version 1.0.1.0)
3. 이 문서는 2017년 04월 20일 한국 마이크로소프트(유), GPS Korea의 Partner Technical Consultant 송민경에 의해 수정 되었습니다. (Version 1.1.0.0)
4. 이 문서는 2017년 04월 21일 한국 마이크로소프트(유), GPS Korea의 Partner Technical Consultant 송민경에 의해 수정 되었습니다. (Version 1.1.1.0)

목차

1. 사전 준비.....	6
1.1 개요.....	6
1.2 비즈니스 케이스	6
1.3 교육 목표	7
1.4 실습 사전 준비 사항	8
1.5 Azure 관리 포털에서 HDInsight에 접근하기.....	8
2. Azure HDInsight에서 Spark 클러스터 만들기.....	9
2.1 스파크 클러스터 생성.....	9
2.2 스파크 클러스터 사이즈 조정하기	15
3. Azure Blob 스토리지에 데이터 올리기	18
3.1 샘플 데이터 다운로드 받기	18
3.2 스토리지 키 얻기	19
3.3 AzCopy로 파일 업로드 하기.....	21
4. Jupyter에서 분석하기	23
4.1 Jupyter 노트북 생성하기	23
4.2 데이터를 스파크 DataFrame 으로 만들기	27
4.3 Spark SQL로 데이터 쿼리하기.....	29
4.4 Hive Table과 Parquet 파일로 데이터를 저장하기.....	31
5. (응용) Azure SQL DW에서 HDFS 파일 읽어 오기.....	32
5.1 PolyBase 개요.....	32
5.2 Azure SQL DW 생성하기	33
5.3 Azure SQL DW에 접속하기.....	38

5.4	Spark 시스템에 샘플 파일 업로드 하기.....	41
5.5	PolyBase 기능을 활용해 Spark 시스템에 쿼리하기	47
6.	(응용) Power BI에서 연결하기	51
6.1	Power BI 데스크톱 다운로드.....	51
6.2	Power BI 데스크톱에서 Hive Table 연결하기	52

1. 사전 준비

1.1 개요

이번 실습에서는 Apache Spark을 소개하고 스파크 클러스터를 Azure HDInsight위에서 만드는 내용을 진행할 예정입니다. Spark on HDInsight는 인메모리 프로세싱을 지원하고 다른 여러가지 능력을 가지고 있기 때문에 기존 하둡 시스템보다 더 강화된 시스템이라고 볼 수 있습니다.



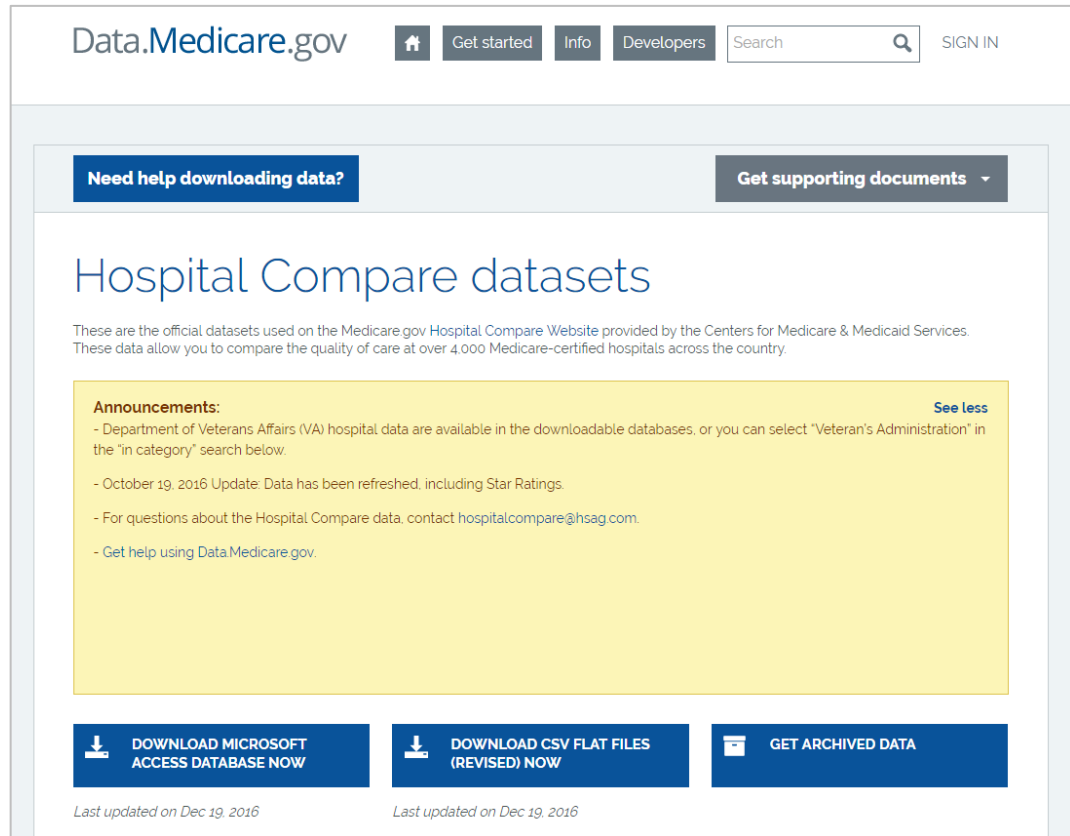
이번 실습에서는 스파크 클러스터를 만들어 보고, Azure Blob storage에 파일을 어떻게 업로드 하고 Jupyter 노트북을 활용해서 어떻게 분석할 수 있는지 확인합니다

1.2 비즈니스 케이스

병원에서는 잠재적인 감염을 예방이 치료의 한 과정으로 다뤄지고 있습니다. 병원 및 헬스케어 공급자들을 비교하기 위해서 healthcare-associated infection (HAI) 측정 지표를 만들어 평가하는데 사용이 됩니다. 따라서 병원에서는 다양한 HAI 값들을 보고해야 할 필요가 있습니다. 게다가 더 나은 환자 돌봄 시스템을 제공하기 위해 HAI의 숫자를 줄인 병원에 대해서는 금전적인 보너스 체계가 더 많이 제공이 됩니다.

이번 실습에서는 헬스케어 공급자의 HAI 관련 통계 데이터 기반의 샘플 데이터셋을 활용할 예정입니다. 이 측정값들은 data.medicare.gov 아래 사이트에서 확인할 수 있습니다.

(<https://data.medicare.gov/data/hospital-compare>)



그들은 이 지표를 활용하여 병원 환경 및 환자 돌봄의 결과를 감염이라는 지표를 추적함으로써 평가받게 됩니다. 각각의 공급자들은 감염 타입 별로 점수를 매기며, Spark DataFrame과 Spark SQL을 활용해 분석을 하게 될 예정입니다.

1.3 교육 목표

이 실습을 마침으로써, Azure HDInsight와 관련 된 개념과 기능에 대해서 직접 HOL 경험을 갖게 됩니다.

- 1) Azure Portal에서 Apache Spark을 만들 수 있다
- 2) AzCopy를 통해 Azure Blob Storage에 파일을 업로드 할 수 있다
- 3) Jupyter 노트북을 생성할 수 있다
- 4) 테이블과 데이터프레임에 있는 데이터를 변형하고 쿼리 할 수 있다
- 5) 스파크를 활용하여 Hive 테이블에 데이터를 저장할 수 있다
- 6) Azure 관리 포털에서 스파크 클러스터를 삭제할 수 있다

1.4 실습 사전 준비 사항

Azure 구독이 필요하며 만약 없다면 아래 링크에서 생성할 수 있습니다:

<https://azure.microsoft.com/en-us/free/>

AzCopy 유틸리티가 필요하며 해당 유틸리티는 아래 사이트에서 다운로드 받을 수 있습니다:

<http://aka.ms/downloadazcopy/>

이 실습에서 사용한 데이터와 파일은 아래 링크에 포함되어 있습니다:

<https://github.com/CortanaAnalyticsLabs/CortanaAnalyticsLabs/raw/master/AzureHDInsightSpark/SparkLab.zip>

Mozilla Firefox or Google Chrome. Jupyter Notebooks are not supported for Internet Explorer or Edge.

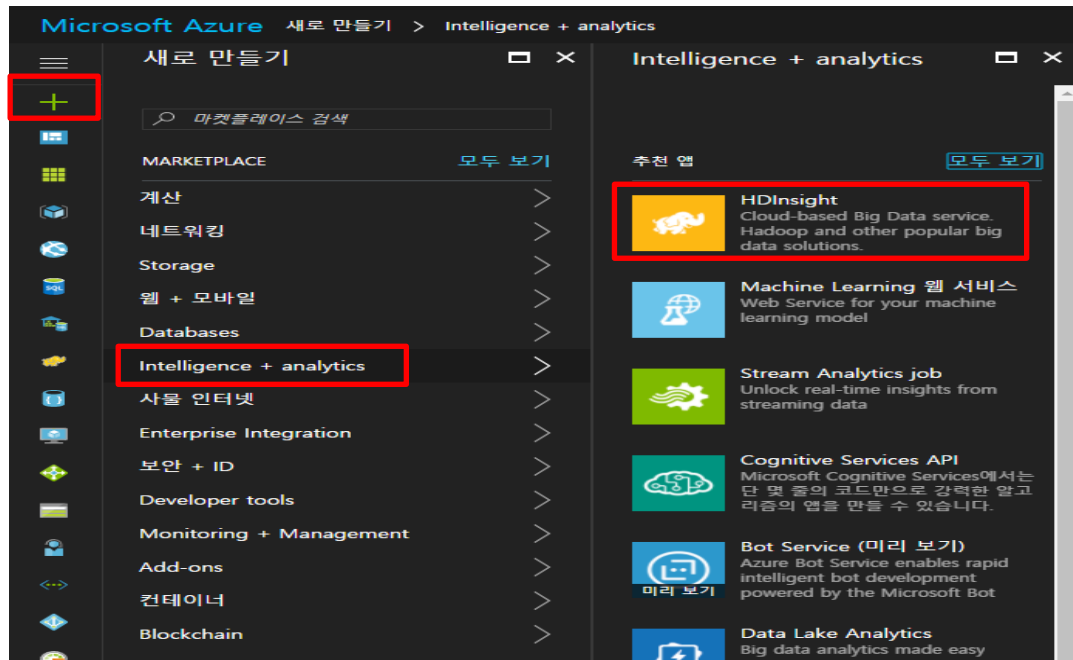
1.5 Azure 관리 포털에서 HDInsight에 접근하기

이번 챕터에서는 Spark를 포함한 Azure HDInsight 클러스터를 생성할 수 있습니다.

2. Azure HDInsight에서 Spark 클러스터 만들기

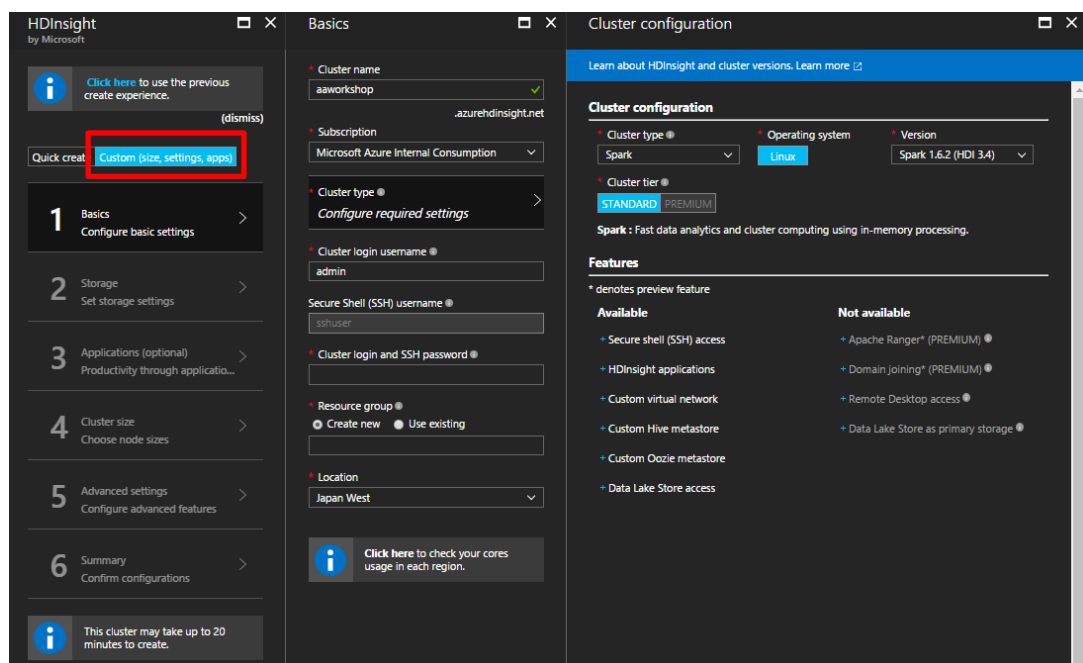
브라우저를 열어Azure Portal: <http://portal.azure.com>에 접속합니다.

+NEW 버튼을 클릭하여 **Intelligence + Analytics** 선택하고 **HDInsight**를 선택해 줍니다.



2.1 스파크 클러스터 생성

Custom cluster 만들기를 클릭하고 클러스터 이름을 입력합니다.



그리고 클러스터 타입은 Spark 클러스터, Version은 1.6.3(HDI 3.5) 버전을 클릭합니다.

Cluster configuration

* Cluster type ⓘ Spark ▼

* Operating system Linux

* Version Spark 1.6.3 (HDI 3.5) ▼

그리고 아래 화면과 같이 클러스터 로그인 및 SSH 비밀번호를 대소문자를 섞어 적어주고 리소스 그룹과 지역을 선택해 줍니다. 이후 Next 버튼을 눌러 다음 단계로 넘어갑니다.

기본

* 클러스터 이름
misongspark
azurehdinsight.net

* 구독
Microsoft Azure Enterprise ▼

* 클러스터 유형 ⓘ
Linux의 Spark 1.6(HDI 3.5) >

* 클러스터 로그인 사용자 이름 ⓘ
admin

* 클러스터 로그인 암호 ⓘ
..... ✓

SSH(보안 셸) 사용자 이름 ⓘ
sshuser

☒ 클러스터 로그인과 동일한 암호 사용 ⓘ

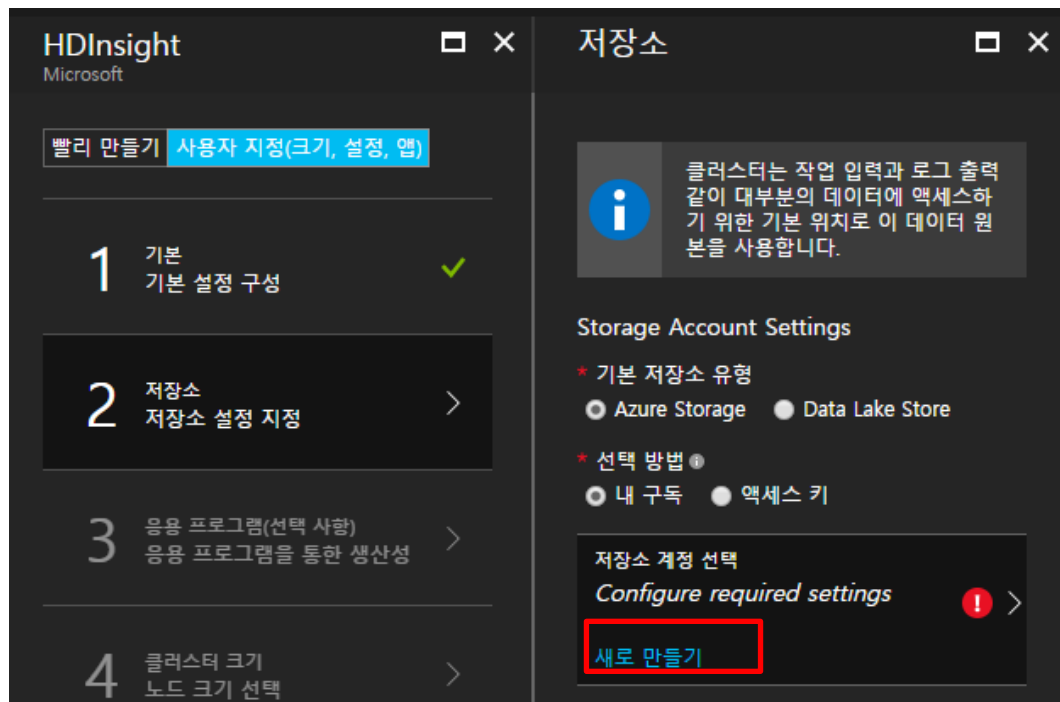
* 리소스 그룹 ⓘ
☐ 새로 만들기 ☒ 기존 그룹 사용
 aaworkshop ▼

* 위치
일본 서부 ▼

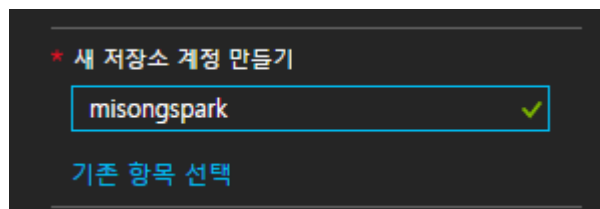
코어 사용량을 보려면 여기를 클릭하세요.

Next

값을 선택하고 저장소(Storage) 카테고리에서 저장소 계정 선택 부분에 [새로 만들기] 버튼을 클릭해 줍니다.



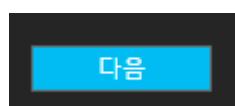
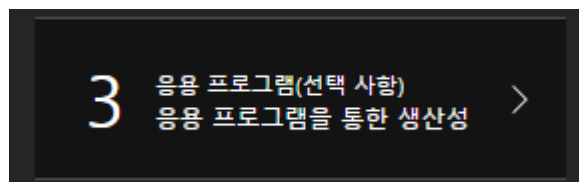
새 저장소 계정 만들기 칸에 적당한 저장소 이름을 넣어 줍니다.



그리고 나머지 값들은 기본값으로 둔 채로 아래와 같이 다음 버튼을 클릭해 줍니다.



응용프로그램 선택 사항은 그대로 두고 다음 버튼을 클릭해 넘어갑니다.



그리고 클러스터 사이즈는 기본 값으로 Worker node 4대를 가져가되 Worker node Size를 클릭하여 D4 이상의 클러스터를 선택해 줍니다.

HDInsight by Microsoft

[Click here to use the previous create experience.](#) (dismiss)

Quick create: Custom (size, settings, apps)

- 1 Basics
Configure basic settings ✓
- 2 Storage
Set storage settings ✓
- 3 Applications (optional)
Productivity through applicatio... ✓
- 4 Cluster size
Choose node sizes >
- 5 Advanced settings
Configure advanced features >
- 6 Summary
Confirm configurations >

Cluster size

To learn more, visit our pricing page. [Learn more](#)

Number of Worker nodes 2 ✓

* Worker node size
D4 (2 nodes, 16 cores) >

* Head node size
D12 (2 nodes, 8 cores) >

WORKER NODES	1491.600 x 2 = 2983.200
HEAD NODES	912.000 x 2 = 1824.000
TOTAL COST	4807.20 KRW/HOUR (ESTIMATED)

24 of 60 cores would be used in Japan West.

This price estimate does not include storage costs, network egress costs, or subscription discounts.

Questions? [Contact billing support.](#)

Note: Clusters with more than 32 Worker nodes require a Head node size with at least 8 cores and 14 GB RAM.

Next

고급 설정의 경우도 마찬가지로 기본 값으로 두고 다음(NEXT) 버튼을 눌러 다음 차례를 진행합니다.

HDInsight by Microsoft

[Click here](#) to use the previous create experience. (dismiss)

Quick create Custom (size, settings, apps)

- 1 Basics
Configure basic settings ✓
- 2 Storage
Set storage settings ✓
- 3 Applications (optional)
Productivity through applicatio... ✓
- 4 Cluster size
Choose node sizes ✓
- 5 Advanced settings
Configure advanced features >
- 6 Summary
Confirm configurations >

Advanced settings

Remote Access

* Secure Shell (SSH) username ⓘ
sshuser

☒ Use same password as cluster login

Script Actions (optional)

Script actions
Optional >

Virtual Network Settings (optional)
Filtered to location and subscription of cluster.

Virtual network
Select or type to filter virtual networks ▾

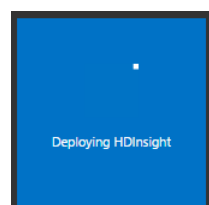
Subnet
Please select a valid virtual network

Next

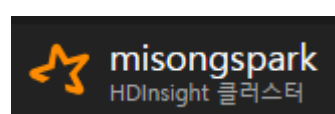
요약 정보가 나타나면 이를 확인하고 Create 버튼을 클릭해줍니다.

[만들기](#) [Download template and parameters](#)

대시보드를 확인하여 배포가 잘 진행되고 있는지 확인합니다.



배포가 완성되고 나면 아래와 같이 Spark 클러스터 개요 페이지로 자동 넘어갑니다.

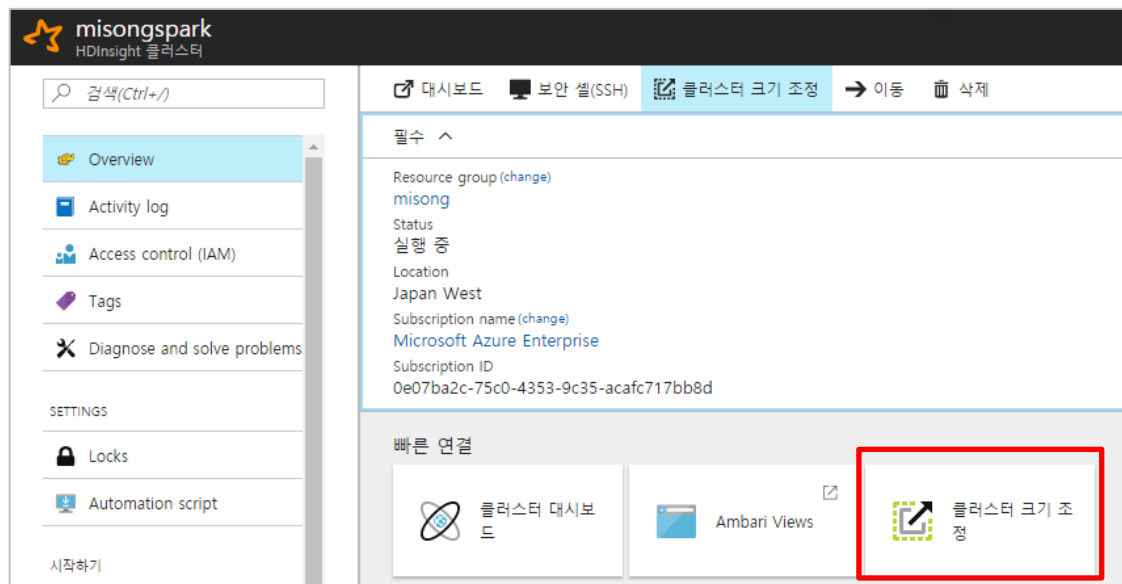


2.2 스파크 클러스터 사이즈 조정하기

클라우드 하둡 클러스터 시스템을 사용하는 가장 큰 이점 중 하나는 작업 클러스터 노드 수를 미리 예측할 필요 없이, 필요할 때 재 조정(늘림 또는 줄임) 할 수 있다는 점입니다. 클라우드 클러스터를 사용하게 되면 클러스터 작업 노드 수를 추가할 때 하드웨어 구매에 소요되는 시간과 비용을 절약할 수 있으며, 분석할 데이터의 크기에 따라 빈번하게 바뀔 수 있는 요구되는 리소스 사양에 맞는 시스템을 수 분 내로 배포할 수 있습니다.

클라우드 클러스터 시스템인 HDInsight는 클러스터 배포 자체도 몇 분 내에 수행할 수 있으며 마찬가지로 크기 조정 또한 몇 분 내에 이뤄질 수 있습니다.

아래 그림과 같이 앞서 만든 클러스터 Portal 화면 개요 상에서 클러스터 크기 조정 버튼을 클릭합니다.



그러면 아래와 같이 클러스터 크기 조정이라는 메뉴가 생기며 여기서 작업자 노드 수를 가능한 크게 변경해 봅니다. (현재 화면에서는 13) 최대 사용 가능한 Spark 클러스터 작업자 노드 수는 처음 클러스터를 배포한 지역과 구독의 상태에 따라 달라질 수 있습니다. 관리 포털 상에서 UI로 신청되는 최대 노드 수 보다 더 큰 노드 수를 사용하기 위해서는 별도로 요청할 필요가 있습니다.

작업자 노드 수를 변경하였으면 저장 버튼을 클릭해 크기조정 작업을 시작합니다.

클러스터 크기 조정

misongspark

작업자 노드 수 ⓘ

13 | ✓

작업자 노드 크기

D3 (노드 2개, 코어 8개)

🔒

헤드 노드 크기

D3 (노드 2개, 코어 8개)

🔒

작업자 노드

$746.400 \times 13 = 9703.200$

헤드 노드

$746.400 \times 2 = 1492.800$

전체 비용

11196.00

KRW/시간(예상)

i

이 클러스터는 Japan West에서 사용할 수 있는 구독의 코어를 토대로 최대 13개의 작업자 노드로 확장할 수 있습니다.

코어 사용량을 보려면 클릭하세요.

이 예상 요금에는 저장소 비용, 네트워크 송신 비용 또는 구독 할인이 포함되어 있지 않습니다.

질문이 있으면 [청구 지원에 문의하세요](#).

자세히 알아보려면 [가격 책정 페이지](#)를 방문하세요.

저장

취소

크기 조정이 시작된 순간부터 크기 조정이 완료된 시간을 체크해 봅니다. 수 분 내에 클러스터 작업자 노드 수를 늘려 대용량 데이터 분석에 맞게 확장할 수 있음을 확인할 수 있습니다.

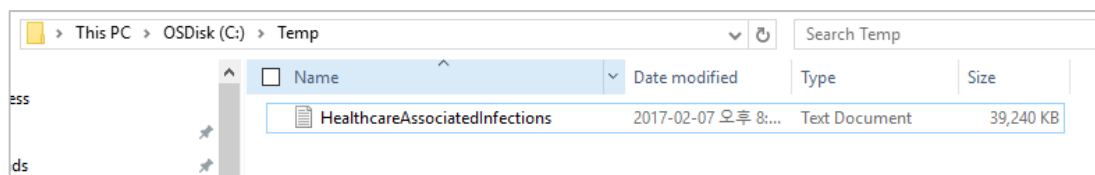
3. Azure Blob 스토리지에 데이터 올리기

3.1 샘플 데이터 다운로드 받기

클러스터가 프로비저닝 되기를 기다리는 동안 스토리지 컨테이너에 샘플 파일을 업로드 해볼 예정입니다. AzCopy라고 하는 명령어 기반 유틸리티를 통해 텍스트 파일을 복사하여 스토리지 컨테이너에 업로드 할 예정입니다. 만약 다른 스토리지 관련 툴이 편리하다고 하면 그렇게 작업할 수 있습니다.

아래 경로에서 SparkLab.zip 파일을 다운로드 받아서 해당 압축 파일을 풀거나 <https://github.com/CortanaAnalyticsLabs/CortanaAnalyticsLabs/raw/master/AzureHDInsightSpark/SparkLab.zip> 또는 배포 된 Spark 실습 파일에서 HealthcareAssociatedInfections.txt 파일과 SparkLab.ipynb 파일을 PC의 적당한 위치에 받아 옮겨 놓습니다.

해당 파일을 풀고, HealthcareAssociatedInfection.txt 파일을 확인합니다. 이 파일은 약 38.3MB 정도를 차지하는 파일입니다. 이 파일은 미국의 헬스케어 공급자들의 속성을 포함하며 이는 다양한 HAI 지표를 점수화 한 데이터 입니다. 이 텍스트 파일을 **C드라이브 밑에 Temp 폴더를 하나 만들어 그 위치에 복사해서 넣어 둡니다.** 만약 Temp 폴더가 없다면 Temp 폴더를 생성해 줍니다.



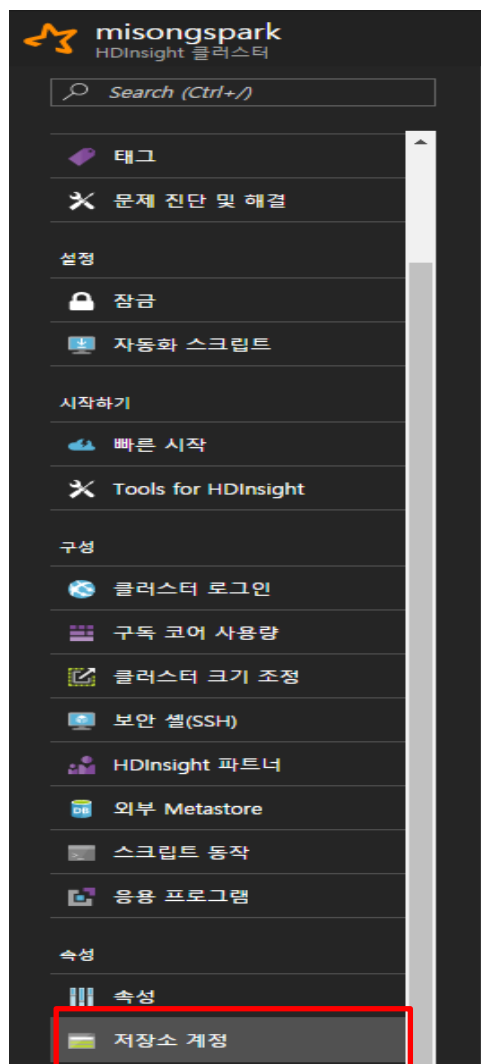
ProviderID	HospitalName	Address	City	State	ZIPCode	CountyName	PhoneNumber	MeasureName	MeasureID
10001	SOUTHEAST ALABAMA MEDICAL CENTER			1108	ROSS CLARK CIRCLE	DOTHAN AL	36301	HOUSTON	3347938701
ComparedToNationalScore Footnote									
CLABSI: Lower Confidence Limit	HAI_1_CI_LOWER	No Different than National Benchmark		0.313					
10001	SOUTHEAST ALABAMA MEDICAL CENTER			1108	ROSS CLARK CIRCLE	DOTHAN AL	36301	HOUSTON	3347938701
CLABSI: Upper Confidence Limit	HAI_1_CI_UPPER	No Different than National Benchmark		3.348					
10001	SOUTHEAST ALABAMA MEDICAL CENTER			1108	ROSS CLARK CIRCLE	DOTHAN AL	36301	HOUSTON	3347938701
CLABSI: Number of Procedures	HAI_1_DOPC_DAYS	No Different than National Benchmark		1887					
10001	SOUTHEAST ALABAMA MEDICAL CENTER			1108	ROSS CLARK CIRCLE	DOTHAN AL	36301	HOUSTON	3347938701
CLABSI: Predicted Cases	HAI_1_ELIGCASES	No Different than National Benchmark		2.439					
10001	SOUTHEAST ALABAMA MEDICAL CENTER			1108	ROSS CLARK CIRCLE	DOTHAN AL	36301	HOUSTON	3347938701
CLABSI: Observed Cases	HAI_1_NUMERATOR	No Different than National Benchmark		3					
10001	SOUTHEAST ALABAMA MEDICAL CENTER			1108	ROSS CLARK CIRCLE	DOTHAN AL	36301	HOUSTON	3347938701
Central line-associated bloodstream infections (CLABSI) in ICUs and select wards							HAI_1_SIR	No Different than National Benchmark	1.23
10001	SOUTHEAST ALABAMA MEDICAL CENTER			1108	ROSS CLARK CIRCLE	DOTHAN AL	36301	HOUSTON	3347938701

3.2 스토리지 키 얻기

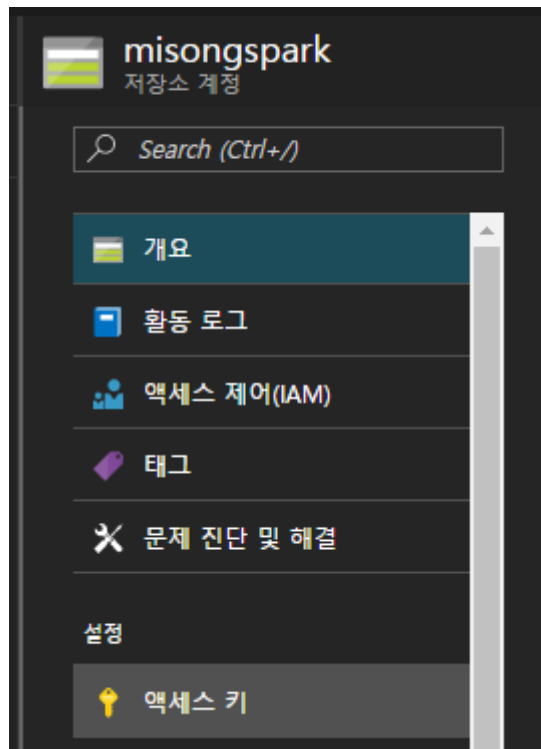
Azure Storage Explorer와는 다르게 AzCopy는 키를 요구합니다. Azure Storage Explorer는 별도의 로그인 창을 띄워 로그인을 받습니다. AzCopy를 사용하기 위해서 스파크 클러스터를 만든 위치의 blob storage의 컨테이너 키를 복사 해 줍니다. Azure 포털에서 검색 버튼을 클릭하여 스파크 클러스터를 생성할 때 만들어 준 스토리지 계정 이름을 검색할 수 있습니다.



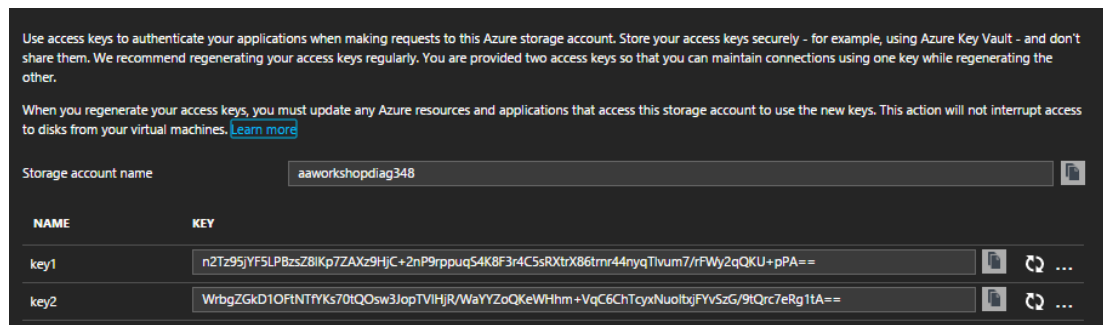
또는 아래 그림과 같이 Spark Cluster 개요 페이지에서 저장소계정(Storage Account)를 찾아 들어가면 저장소 계정을 찾을 수 있습니다.



스토리지 어카운트를 클릭하여 액세스키(Access Key)를 클릭해 줍니다.



그리고 아래 화면과 같은 화면이 나오면 Key 1을 메모장에 복사해 줍니다.

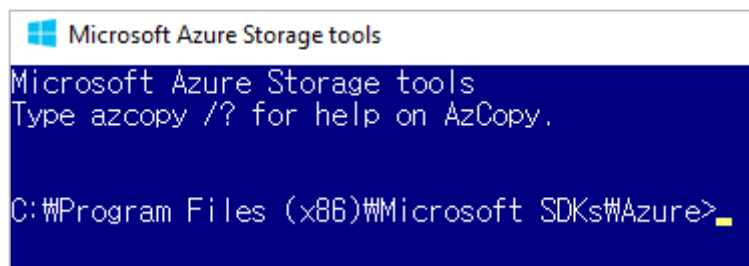
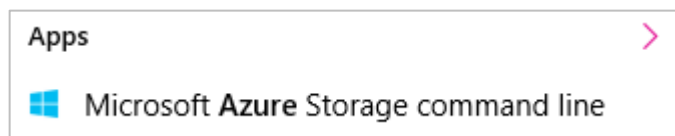


3.3 AzCopy로 파일 업로드 하기

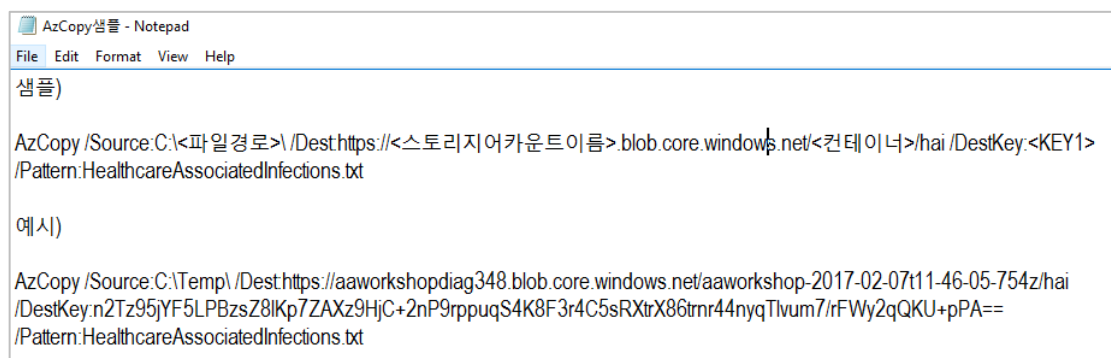
키를 얻었으면 이제 **HealthcareAssociatedInfections.txt** 이 텍스트 파일을 AzCopy 유틸리티를 활용하여 복사해 줍니다.

가장 최신 버전의 AzCopy를 다음 경로에서 다운로드 받을 수 있으며, Microsoft Azure Storage Tool의 MSI 파일이 나오면 이를 실행해 줍니다. <http://aka.ms/downloadazcopy>

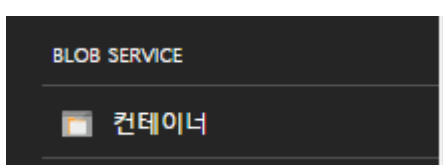
설치가 완료 되면 Microsoft Azure Storage Command Line 어플리케이션을 실행해 줍니다.



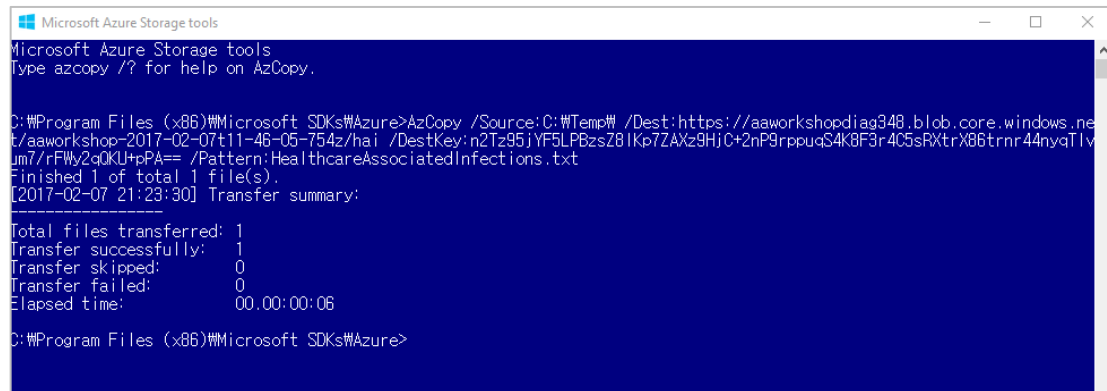
제공된 실습 파일에 있는 AzCopy샘플을 열어 Blob 스토리지 이름 및 Blob 스토리지의 해당하는 컨테이너(Container)명을 적습니다. 아래 규칙과 동일하게 본인의 경로를 적어주면 됩니다.



컨테이너 이름은 마찬가지로 저장소 계정의 개요 페이지에서 아래와 같은 컨테이너 이름을 찾으면 확인하실 수 있습니다.



그리고 해당 명령어를 복사하여 Azure Storage Tools에서 명령어를 입력합니다.

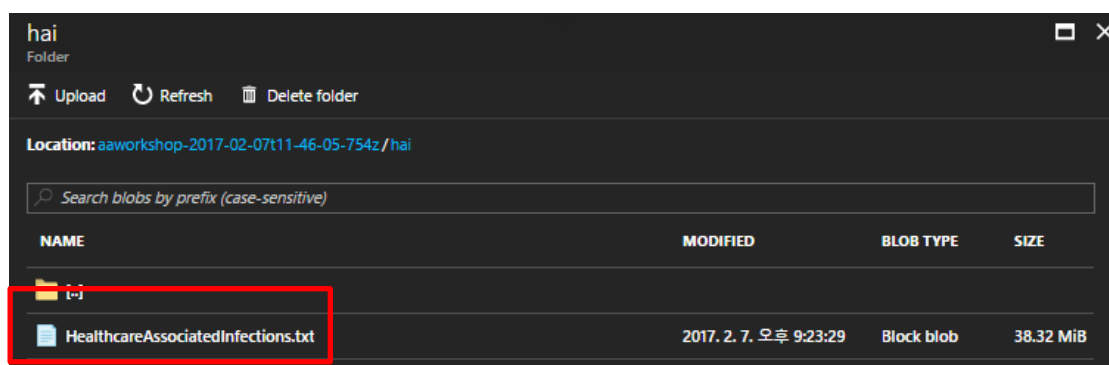
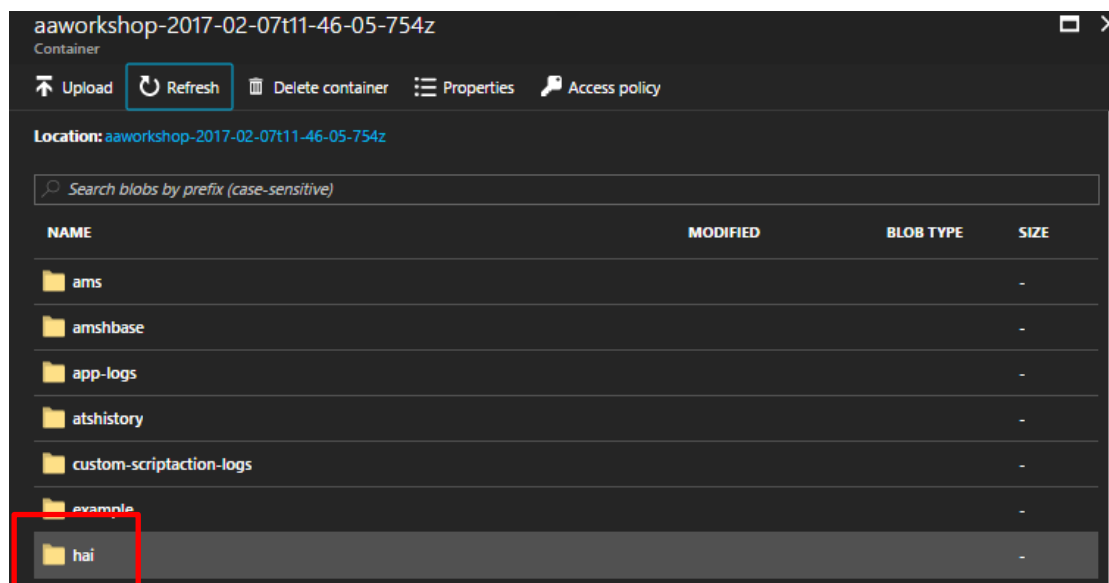


```
Microsoft Azure Storage tools
Type azcopy /? for help on AzCopy.

C:\Program Files (x86)\Microsoft SDKs\Azure>AzCopy /Source:C:\Temp\ /Dest:https://aaworkshopdiag348.blob.core.windows.net/aaworkshop-2017-02-07t11-46-05-754z/hai /DestKey:n2Tz95jYF5LPBzsZ8IKp7ZAXz9HjC+2nF9rppuqS4K8F3r4C5sRXtrX86trnr44nyqTlvum7/rFWy2qKU+pPA== /Pattern:HealthcareAssociatedInfections.txt
Finished 1 of total 1 file(s).
[2017-02-07 21:23:30] Transfer summary:
-----
Total files transferred: 1
Transfer successfully: 1
Transfer skipped: 0
Transfer failed: 0
Elapsed time: 00.00:00:06

C:\Program Files (x86)\Microsoft SDKs\Azure>
```

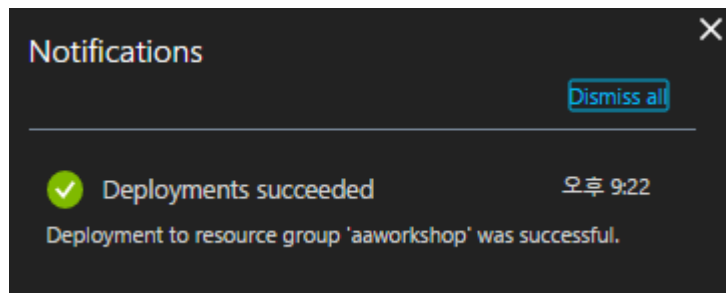
그러면 아래 화면과 같이 Blob 컨테이너의 hai 디렉토리에 들어가면 해당 샘플 파일이 업로드 되었는지 확인할 수 있습니다.



4. Jupyter에서 분석하기

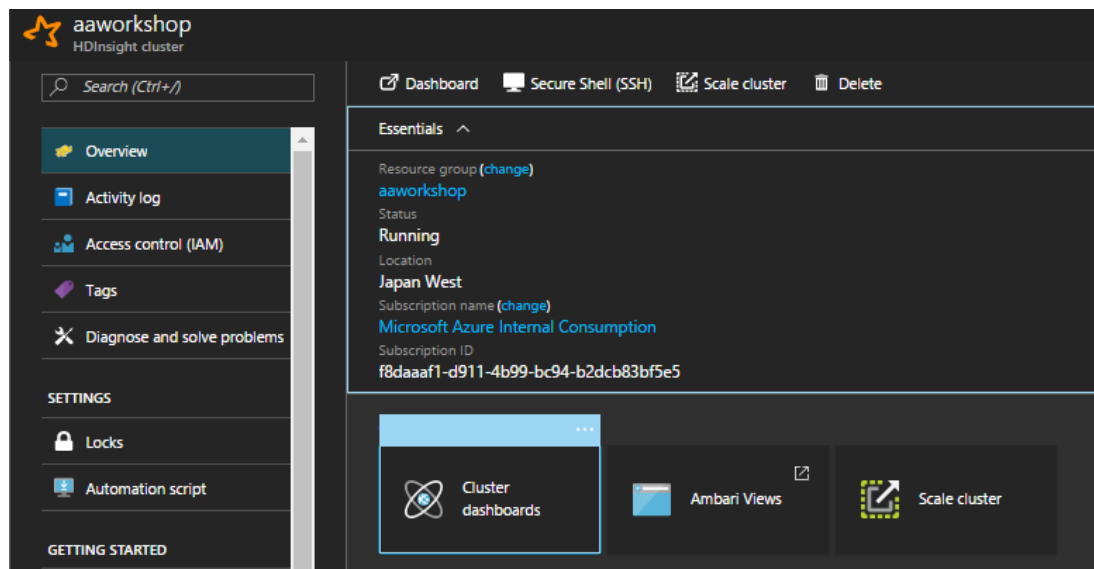
4.1 Jupyter 노트북 생성하기

클러스터가 한 번 만들어지면, 샘플 데이터들이 함께 업로드 되어 있습니다. 아래 화면과 같이 클러스터가 성공적으로 배포되었음을 확인하였으면, 이제 스파크를 통해 데이터를 관리하고 분석할 준비가 되었습니다.

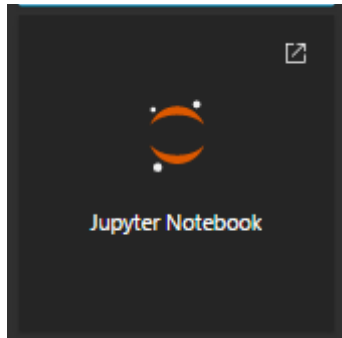


이를 위해서 우리는 Jupyter 노트북을 [Jupyter](#) 활용할 예정입니다. Jupyter 노트북은 다양한 언어를 사용해서 콘텐츠를 공유하고 코딩할 수 있도록 설계된 인기 있는 어플리케이션입니다. Jupyter는 리눅스 용 Spark HDInsight에 배포된 주된 노트북 기능입니다.

Azure Portal 대시보드를 열어 만들어진 Spark HDInsight 클러스터를 클릭합니다. 여기에서 클러스터 대시보드를 클릭해 줍니다.



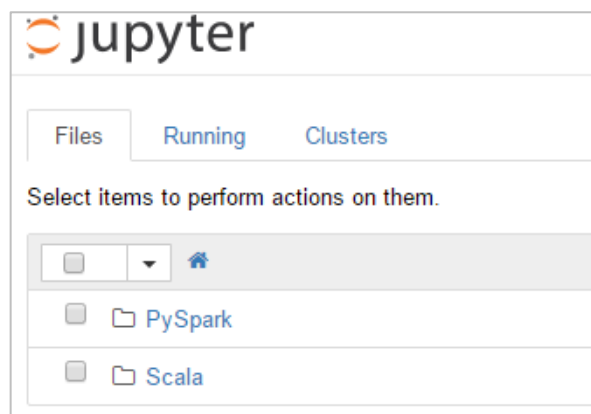
이중에서 Jupyter Notebook을 클릭해 줍니다.



인증 창이 열리면 앞서 지정해준 사용자 이름(admin)과 비밀번호를 입력해 줍니다.

 A light gray dialog box titled "인증 필요" (Authentication Required) with a close button (X) in the top right corner. The text inside says: "https://aaworkshop.azurehdinsight.net에 사용자 이름과 비밀번호를 입력해야 합니다." (You must enter the user name and password for https://aaworkshop.azurehdinsight.net). Below the text are two input fields: "사용자 이름:" (User Name) and "비밀번호:" (Password). At the bottom right are two buttons: "로그인" (Login) and "취소" (Cancel).

그러면 아래와 같이 Jupyter 노트북이 열립니다.



향후에는 Jupyter Notebook에 아래와 같은 링크로 직접 연결할 수도 있습니다.

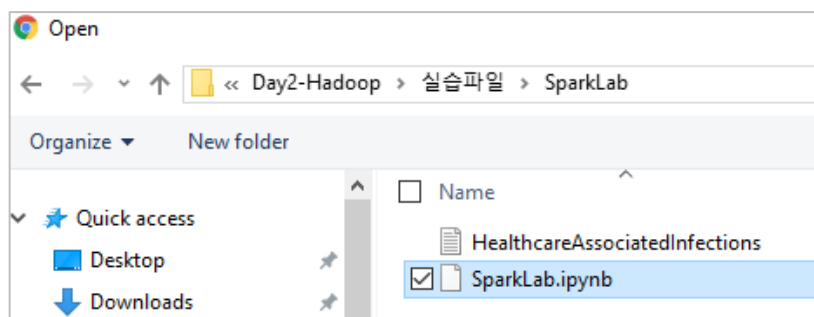
<https://<clustername>.azurehdinsight.net/jupyter>

Jupyter 노트북의 오른쪽 상단에 Upload 버튼을 클릭하여 기존에 다운로드 받아 두었던

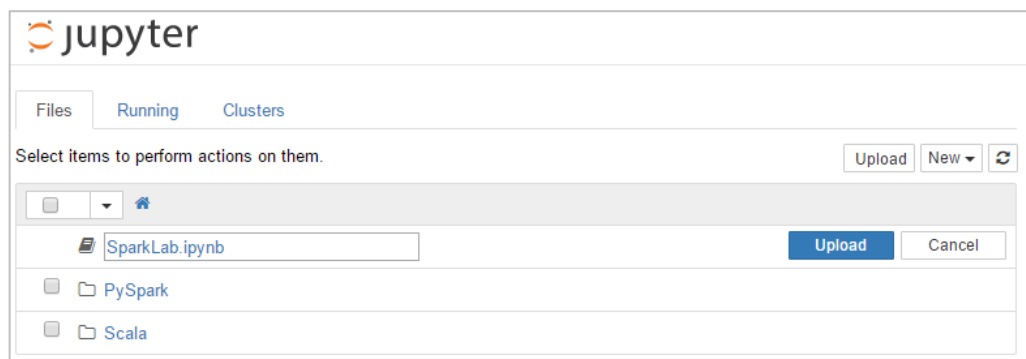
SparkLab.ipynb 파일을 업로드 해줍니다.



기존 경로에 들어가 업로드를 합니다.



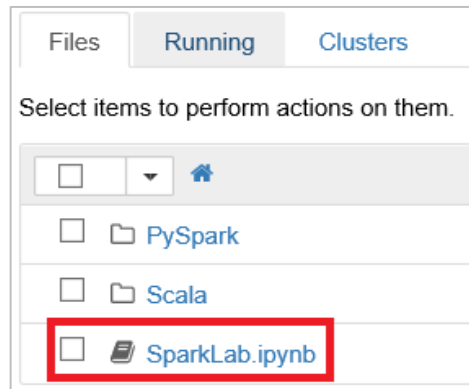
파일이 선택되면 파란색 업로드 버튼을 클릭해줍니다.



업로드가 정상적으로 수행되면 아래와 같이 업로드 된 목록이 나타납니다.




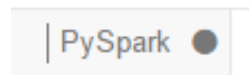
이제 SparkLab.ipynb 파일을 클릭해 줍니다.



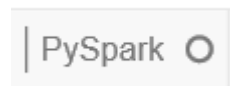
4.2 데이터를 스파크 DataFrame 으로 만들기

한번 SparkLab 노트북을 올리고 나면, 이제 이를 차례로 실행할 수 있게 됩니다. 각각의 코드 블록들은 []: 이러한 형식으로 라벨링이 되어 있습니다. 한번 해당 셀을 호출하게

되면 실행 순서가 나타나게 됩니다.  버튼으로 실행을 시키게 되면 아래와 같이 채워진 원이 나타나게 됩니다.



실행이 완료되면 아래와 같이 원이 비어 있는 상태로 상태가 변하게 됩니다.



첫 번째 셀을 클릭해서 실행되도록 누릅니다. 또는 Shift+Enter를 눌러서 실행할 수도 있습니다. 이번 코드는 스토리지 컨테이너에 업로드한 파일과 관련이 있는 샘플이기 때문에 이를 잘 확인하면서 진행합니다. 첫 번째 행에서는 기존에 해당 위치에 올려 두었던 샘플 파일을 **Resilient Distributed Dataset, or RDD** 로 변환하기 위해서 SparkContext를 만들어 냅니다. 이 작업이 몇 분의 시간이 걸릴 수 있습니다. 실행이 완료되면 코드 블록에 번호가 매겨지며 아래와 같이 결과 창이 나옵니다.

```
In [1]: haiRaw = sc.textFile("wasb://hai/HealthcareAssociatedInfections.txt",16)
        .map(lambda line: [x for x in line.split("#")])
        .filter(lambda r: r[0] != 'ProviderID')
        .map(lambda r: (int(r[0]), str(r[1]), str(r[4]), str(r[8]), str(r[9]), str(r[10]),
        haiRaw.take(1)
```

Creating SparkContext as 'sc'

ID	YARN Application ID	Kind	State	Spark UI	Driver log	Current session?
0	application_1486468767678_0004	pyspark	idle	Link	Link	✓

Creating HiveContext as 'sqlContext'

SparkContext and HiveContext created. Executing user code ...

```
[(10001, 'SOUTHEAST ALABAMA MEDICAL CENTER', 'AL', 'CLABSI: Lower Confidence Limit',
'HA1_CI_LOWER', 'No Different than National Benchmark', '0.313')]
```

다음 셀을 클릭하여 이전과 같이 실행시키도록 합니다.

이 코드는 기존에 만들어 두었던 `haiRaw` RDD로부터 스키마를 매칭하여서 Spark의 또 다른 키 객체인 `DataFrame`을 만들어 냅니다. 이번 셀은 실행이 완료 되어도 아무런 출력 값이 없습니다.

```
In [2]: from pyspark.sql.types import *
        haiSchema = StructType([StructField('ProviderID', IntegerType(), False),
                                StructField('HospitalName', StringType(), True),
                                StructField('State', StringType(), True),
                                StructField('MeasureName', StringType(), True),
                                StructField('MeasureID', StringType(), True),
                                StructField('ComparedToNationalScore', StringType(), True),
                                StructField('MeasureValue', StringType(), True)])
        haiDF = sqlContext.createDataFrame(haiRaw, haiSchema)
```

다음 단계를 진행하고 실행합니다. 이번 코드는 `DataFrame`을 메모리로 캐싱합니다.

```
In [3]: haiDF.cache()

DataFrame[ProviderID: int, HospitalName: string, State: string, MeasureName: string, MeasureID: string, ComparedToNationalScore: string, MeasureValue: string]
```

다음 셀을 수행해 줍니다. 이 코드는 `DataFrame`으로부터 첫 번째 레코드를 보여주는 코드입니다. `show()` 연산은 첫 번째 코드 블록에서 RDD를 가지고 수행했던 `take()` 연산과 동일한 작업을 수행합니다.

```
In [4]: haiDF.show(1)

+-----+-----+-----+-----+-----+-----+
|ProviderID|HospitalName|State|MeasureName|MeasureID|ComparedToNationalScore|MeasureValue|
+-----+-----+-----+-----+-----+-----+
|10001|SOUTHEAST ALABAMA...|AL|CLABSI: Lower Con...|HA1_1_CI_LOWER|No Difference than...|0.313|
+-----+-----+-----+-----+-----+-----+
only showing top 1 row
```

그리고 다음 셀을 또한 수행해 줍니다. 이 코드는 나중에 쿼리에서 사용하기 위해서 `DataFrame` 기반의 `hainfections`라고 하는 임시 테이블을 등록해 줍니다. 이번 코드 블록도 아무런 결과값이 출력되지 않습니다.

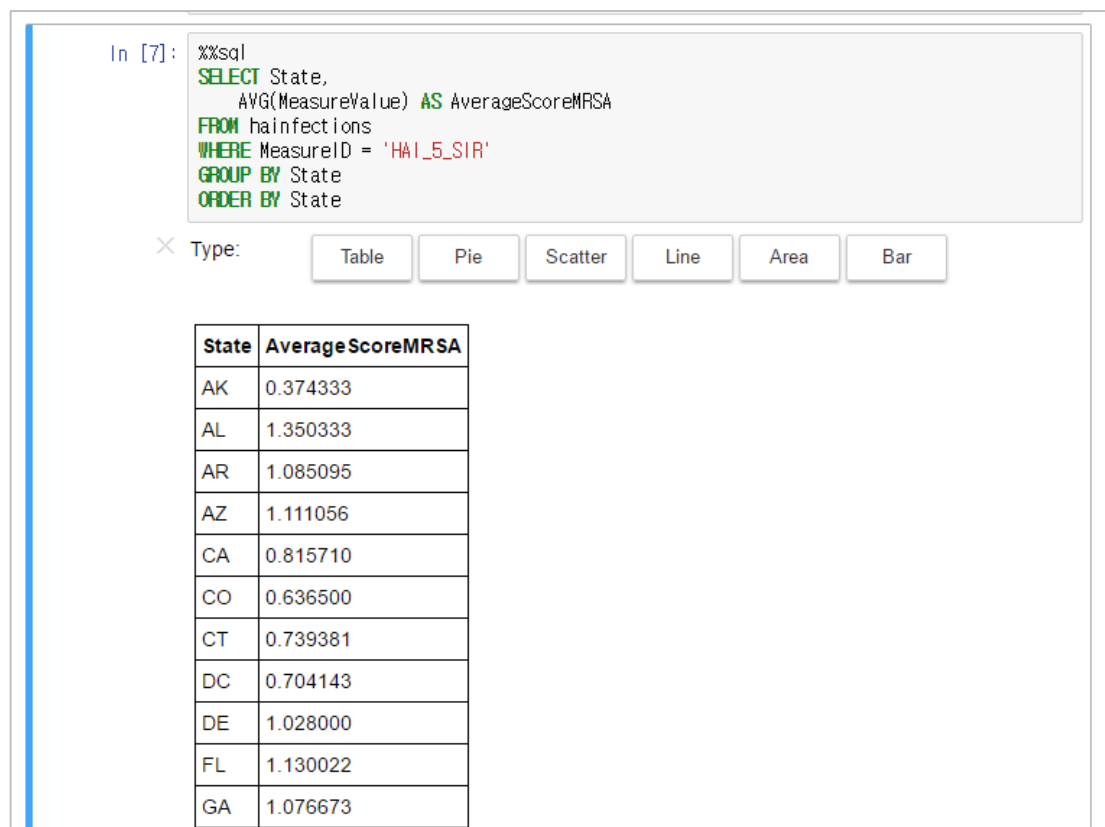
```
In [6]: haiDF.registerTempTable('hainfections')
```

4.3 Spark SQL로 데이터 쿼리하기

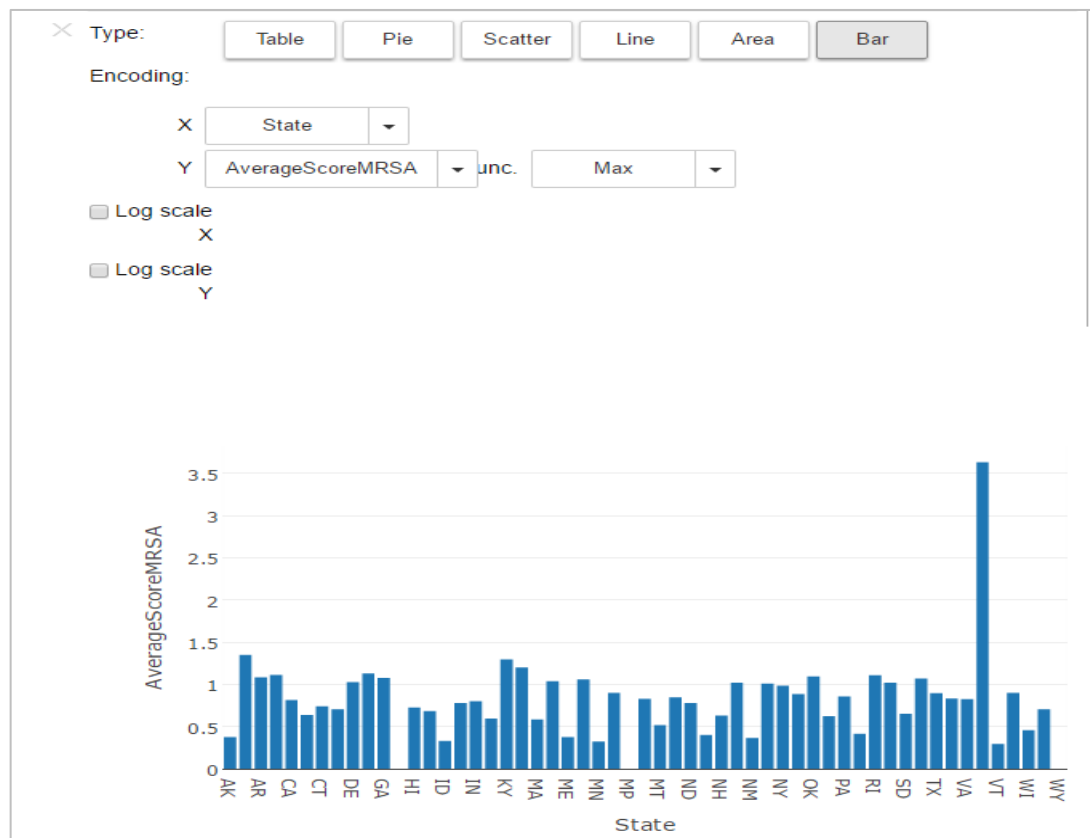
Spark DataFrame들은 또한 필터링, Aggregation 또는 다른 데이터 변경 연산을 지원합니다. DataFrame을 통해 이런식으로 수행하는 연산에 더하여서, Spark SQL을 통해 쿼리하여 비슷한 결과물을 얻을 수도 있습니다. 앞선 단계의 작업에서, 임시 테이블을 등록하고 SQL 명령어를 사용하여 쿼리를 수행할 수 있습니다. `%%sql` “magic” 명령어를 사용하여 Spark Context를 SQL로 변경해주는지 확인할 수 있습니다.

다음 셀을 실행합니다. 다양한 Healthcare 지표 중에서도 MRSA라고 하는 속성이 중요합니다. 이 코드는 기본적인 Spark SQL 쿼리 문으로 MRSA 감염 점수를 주 별로 평균 값을 확인할 수 있도록 합니다. SIR로 라벨링 되어 있는 MeasureID 값을 가진 데이터 셋을 스코어링 해봅니다. 그리고 MRSA를 위한 ID는 `HAI_5_SIR` 입니다.

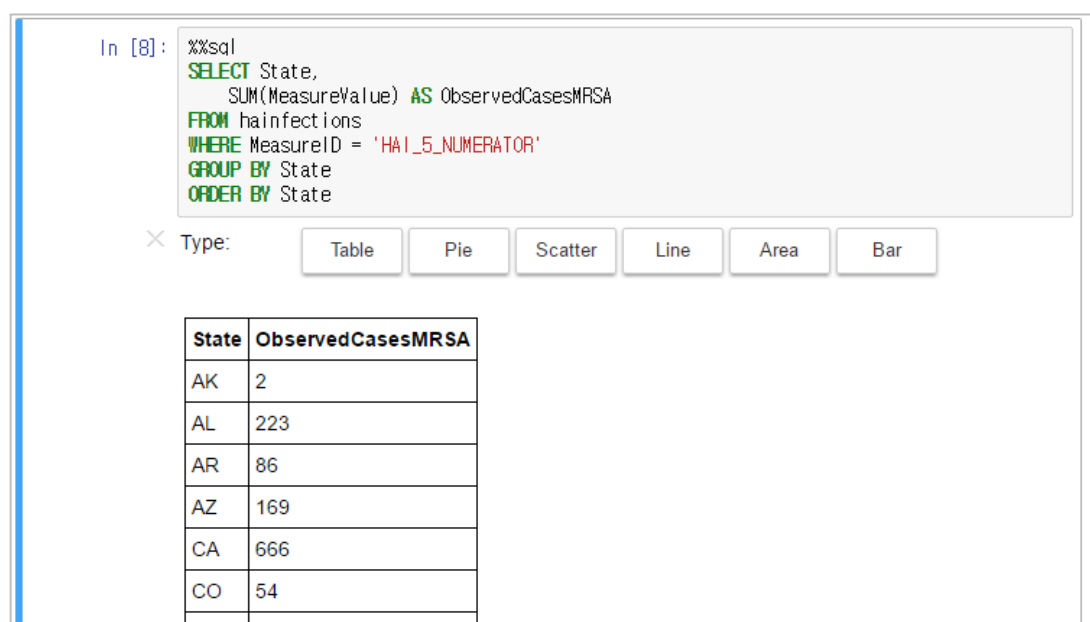
Spark SQL로는 결과 값이 테이블로 보여지게 됩니다. 그러나 DataFrame과 다른 점은, Jupyter 노트북 자체에서 같은 데이터를 다양한 형태의 차트로 보여주는 기능을 가지고 있다는 점입니다. 이는 [Plotly](#)와 결합된 결과 입니다. Bar 옵션을 클릭하여 결과가 바뀌어 보여지는 것을 확인할 수 있습니다.



아래와 같이 Bar로 변경된 결과를 확인할 수 있습니다.



다음 셀을 계속해서 실행해 줍니다. 이 또한 마찬가지로 전 단계 셀과 유사한 작업을 수행합니다. 그러나 측정하는 측정 지표가 달라집니다. MeasureID가 Numerator로 라벨링 된 것과 일치하는 데이터의 감염된 환자 수를 보여줍니다. MRSA를 위한 라벨은 *HAI_5_NUMERATOR* 입니다.



4.4 Hive Table과 Parquet 파일로 데이터를 저장하기

만들어진 결과를 DataFrame과 임시 테이블로 저장하는 것은 가능합니다. 이번 챕터에서는 데이터를 Parquet이라고 하는 최적화된 파일 포맷으로 저장해 봅니다. 한 번 데이터가 Hive Table에 저장되면, 어떤 툴로도 접근하고 연결하는 것이 가능해 집니다. 심지어 Power BI 툴에서는 Hive 테이블로의 연결을 지원하고, Excel 에서도 Hive ODBC를 이용하여 접근할 수 있습니다.

마찬가지로 다음 셀을 클릭하여 실행시킵니다. 이번에는 `%%sql` 매직을 사용하는 대신에, `sqlContext`를 명시적으로 호출합니다. SQL 쿼리의 결과가 `mrSa`라고 하는 DataFrame으로 저장됩니다. 그리고 이는 최적화된 포맷인 Parquet으로 데이터가 저장됩니다. 이 셀 또한 아무런 결과를 출력해주지 않습니다.

```
In [9]: mrSa = sqlContext.sql("SELECT State, #
                                MeasureValue AS ObservedCasesMRSA #
                                FROM hainfections #
                                WHERE MeasureID = 'HAI_5_NUMERATOR'")
mrSa.select("*").write.save("mrSa.parquet", format="parquet")
```

추가로 이 데이터를 Hive Table에 저장하는 것도 가능합니다. 다음 셀을 클릭하여 이를 실행시킵니다. 이 코드는 원래의 raw 데이터와 스키마를 기반으로 DataFrame을 새로 만듭니다. 그리고 **infection**이라는 Hive Table에 저장합니다. 이 결과 또한 아무런 데이터를 출력하지 않습니다.

```
In [11]: hiveHaidF = sqlContext.createDataFrame(haiRaw, haiSchema)
hiveHaidF.write.saveAsTable('infections')
```

그리고 다음 셀을 클릭하고 수행합니다. 이 코드는 `%%sql` 매직을 사용하여 Hive 테이블 목록을 보여 줍니다. 앞 단계에서 만든 `infections` 테이블이 결과로 보여지는 것을 확인할 수 있습니다.

```
In [11]: %%sql
show tables
```

× Type:

Table

Pie

Scatter

Line

Area

Bar

tableName	is Temporary
hainfections	True
hivesampletable	False
infections	False

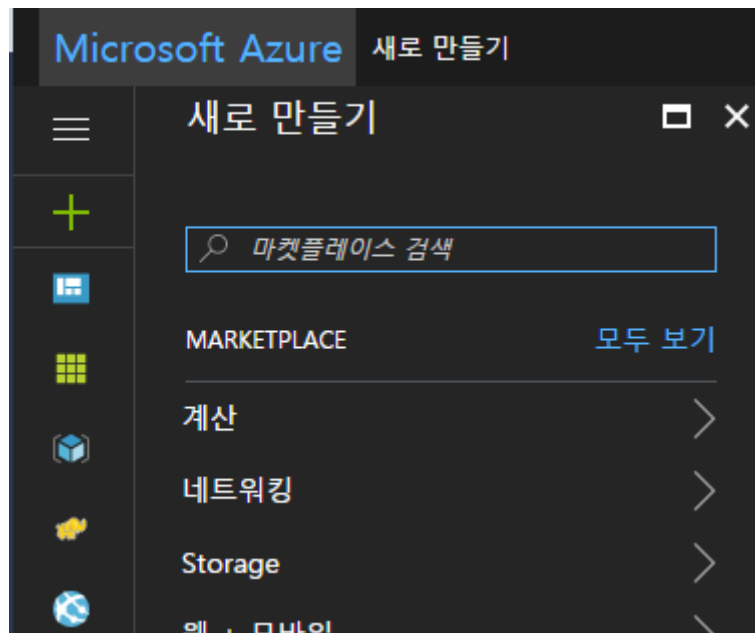
5. (응용) Azure SQL DW에서 HDFS 파일 읽어 오기

5.1 PolyBase 개요

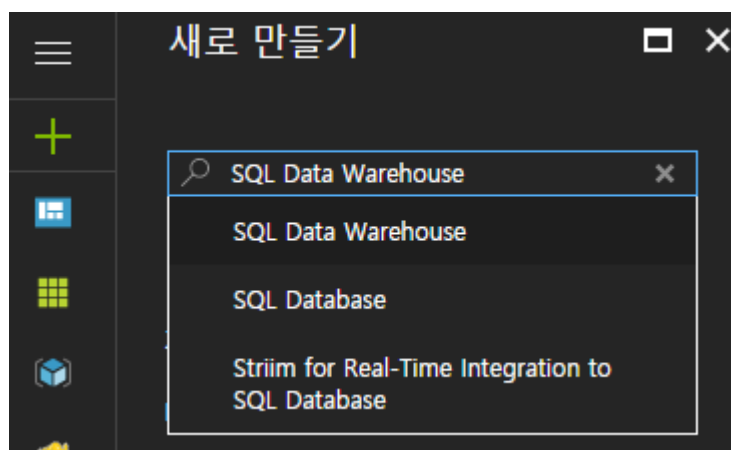
이번 시나리오에서는 앞서 사용했던 하둡 파일 시스템에 저장된 데이터를 Azure SQL Data Warehouse와 연동해보는 작업을 실습해 볼 예정입니다. 먼저 Azure SQL Data Warehouse는 SQL Server의 MPP(Massively Parallel Processing) 기반 병렬 처리 어플라이언스 장비인 APS(Analytics Platform System)을 클라우드 형 서비스로 제공하여 마찬가지로 병렬 처리가 가능한 대용량 DW를 위한 PaaS 서비스입니다. 이 Azure SQL DW 서비스는 PolyBase라는 기능을 제공하여 Hadoop 시스템에 들어있는 비정형, 반정형 데이터를 DBMS에 들어있는 데이터와 함께 연계 분석할 수 있습니다. 이 PolyBase 서비스는 Azure SQL DW 뿐만 아니라 SQL Server(2016 버전) 내에서도 사용할 수 있으며 하둡 파일 시스템 뿐 만 아니라 Azure Blob 스토리지에 있는 외부 데이터에도 쿼리할 수 있습니다.

5.2 Azure SQL DW 생성하기

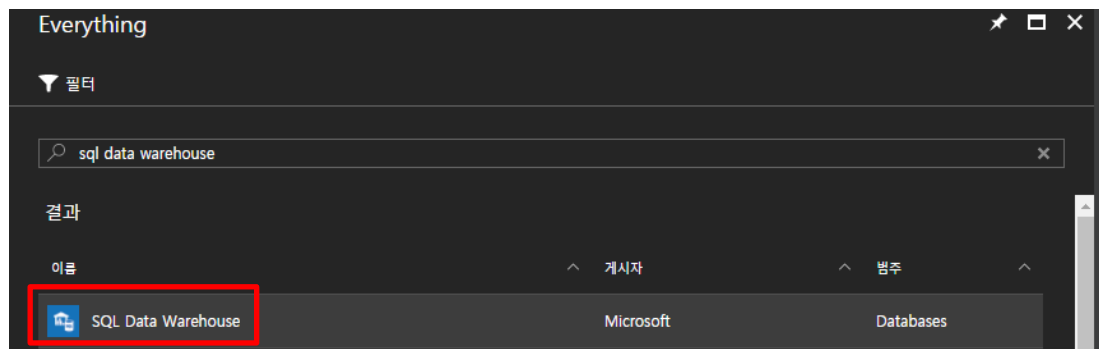
먼저 하단에 있는 데이터베이스를 Azure SQL DW에서 함께 쿼리하기 위해 Azure SQL DW 서비스를 생성해 줍니다. 아래 그림과 같이 <http://portal.azure.com/>에 접속하여 새로 만들기 버튼을 클릭합니다.



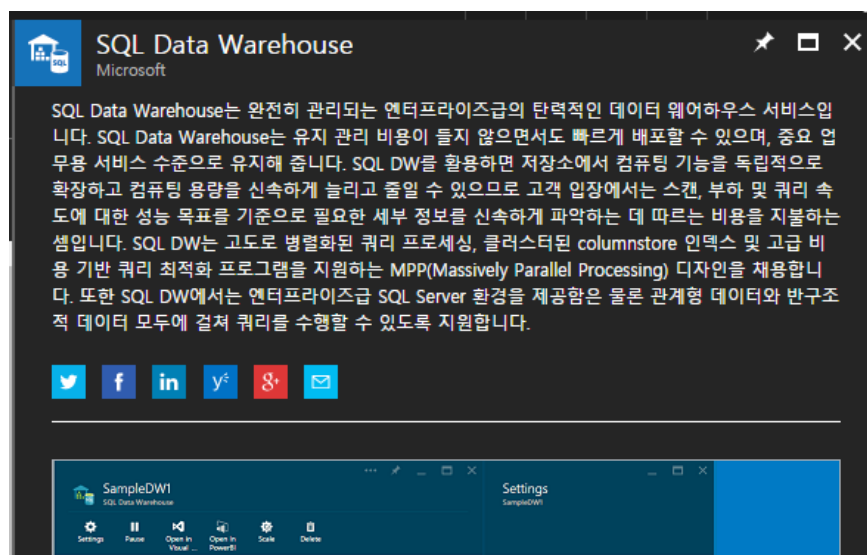
아래 그림과 같이 'SQL Data Warehouse'라는 검색어로 서비스를 검색해 줍니다.



그러면 검색창에 아래와 같이 SQL Data Warehouse를 클릭해 줍니다.



그러면 아래와 같이 SQL Data Warehouse에 대한 개요와 설명이 나타납니다. 만들기 버튼을 클릭하여 구성을 진행해 줍니다.



만들기

그러면 아래와 같이 구성창이 나타나며 여기에 적당한 데이터베이스 이름과 앞서 Spark 클러스터 배포 시 만들었던 리소스 그룹을 사용해주고, 소스 선택 칸에서는 샘플을 선택하여 AdventureWorksDW 샘플로 지정해 줍니다.

SQL Data Warehouse

2017년 2월 1일에 SQL Data Warehouse가 일반 가용성 가격 책정으로 이동합니다.

* 데이터베이스 이름
misongtest ✓

* Subscription
Microsoft Azure Enterprise ▼

* 리소스 그룹 ⓘ
☐ Create new ☐ Use existing
misong ▼

* 소스 선택 ⓘ
샘플 ▼

샘플 선택 ⓘ
AdventureWorksDW ▼

이후 서버를 선택해줍니다. 기존에 만들었던 Azure SQL용 서버가 있다면 아래와 같이 기존에 만들었던 서버가 자동으로 잡히게 되며, 기존에 만들었던 Azure SQL용 서버가 없다면 서버 새로 만들기를 클릭해 줍니다.

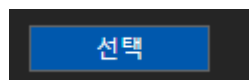
* 서버
hdinsight (대한민국 중부) >

이후 아래 그림과 같이 새 서버 만들기를 클릭하여 새 서버에 적당한 서버 이름과 서버 관리자 이름을 넣어줍니다. 그리고 서버 위치도 가까운 서버 위치 중 적당한 곳을 지정해 줍니다.

The screenshot shows the '새 서버' (New Server) configuration window. The left pane has a button '+ 새 서버 만들기' and a section for 'hdinsight' with the text '대한민국 중부' and 'misong'. The right pane contains the following fields:

- * 서버 이름: hdinsighttest (with a green checkmark)
- * 서버 관리자 로그인: misong (with a green checkmark)
- * 암호: (masked with dots, with a green checkmark)
- * 암호 확인: (masked with dots, with a green checkmark)
- * 위치: 대한민국 중부 (dropdown menu)
- ☒ Azure 서비스의 서버 액세스 허용

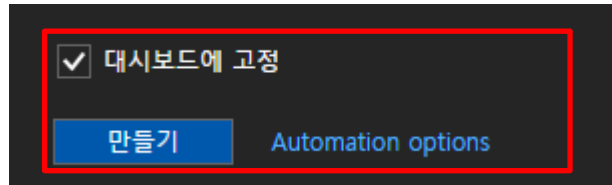
그리고 아래 그림과 같이 선택 버튼을 눌러줍니다.



이후 서버 설정까지 완료 되었으면 아래와 같이 나머지 값들은 기본 값으로 둡니다.

The screenshot shows the '서버' (Server) configuration window. The '서버' section displays 'hdinsighttest (대한민국 중부)' with a right arrow. The '데이터 정렬' (Data Sort) section shows '샘플에 의해 지정됨'. The '성능' (Performance) section shows a slider set to 400. At the bottom, it says '가격 책정을 로드하는 중...' and provides a link to '가격 책정에 대한 자세한 내용을 알아보세요.'

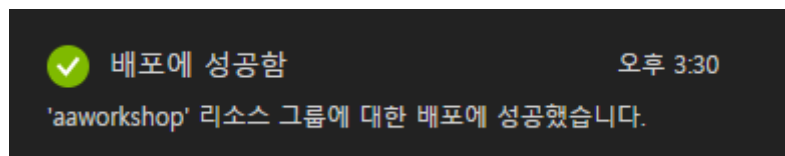
설정이 완료되면 아래 그림과 같이 대시보드에 고정버튼을 클릭하여 포털 메인 화면에 해당하는 Azure SQL DW가 노출될 수 있도록 체크를 클릭해주고 이후 만들기 버튼을 클릭해 줍니다.



그러면 포털 화면에서 아래와 같이 배포 중이라는 메시지가 나타납니다.



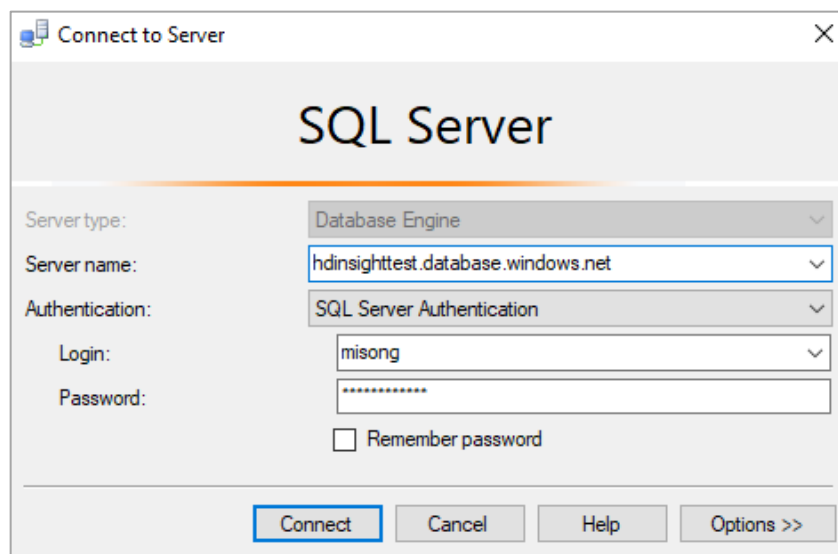
배포가 완료되면 알림 창을 통해 아래와 같이 배포가 성공했다는 메시지가 나타나게 됩니다.



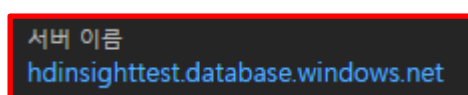
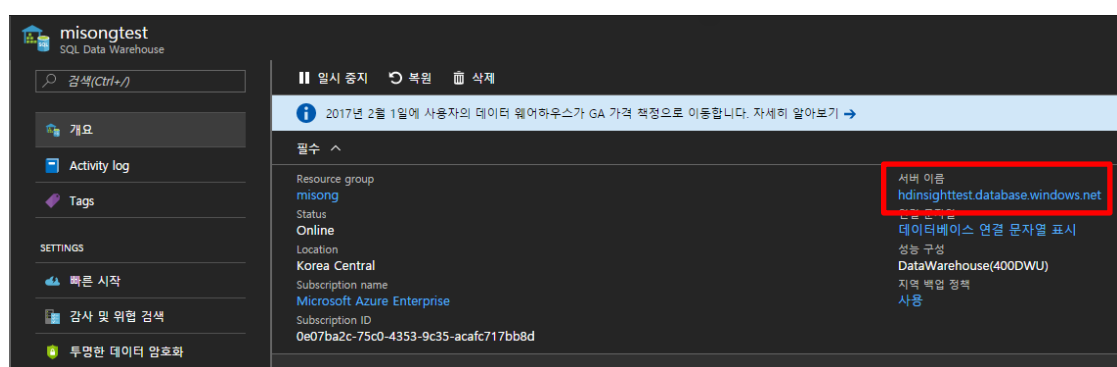
5.3 Azure SQL DW에 접속하기

Azure SQL DW에 쿼리를 하는 방법은 다양합니다. 먼저 SQL Server 온프레미스 버전과 동일하게 SQL Server Management Studio를 설치(<https://msdn.microsoft.com/ko-kr/library/mt238290.aspx>)하거나 또는 Azure 포털 상에서 직접 쿼리를 수행하는 방법이 있습니다.

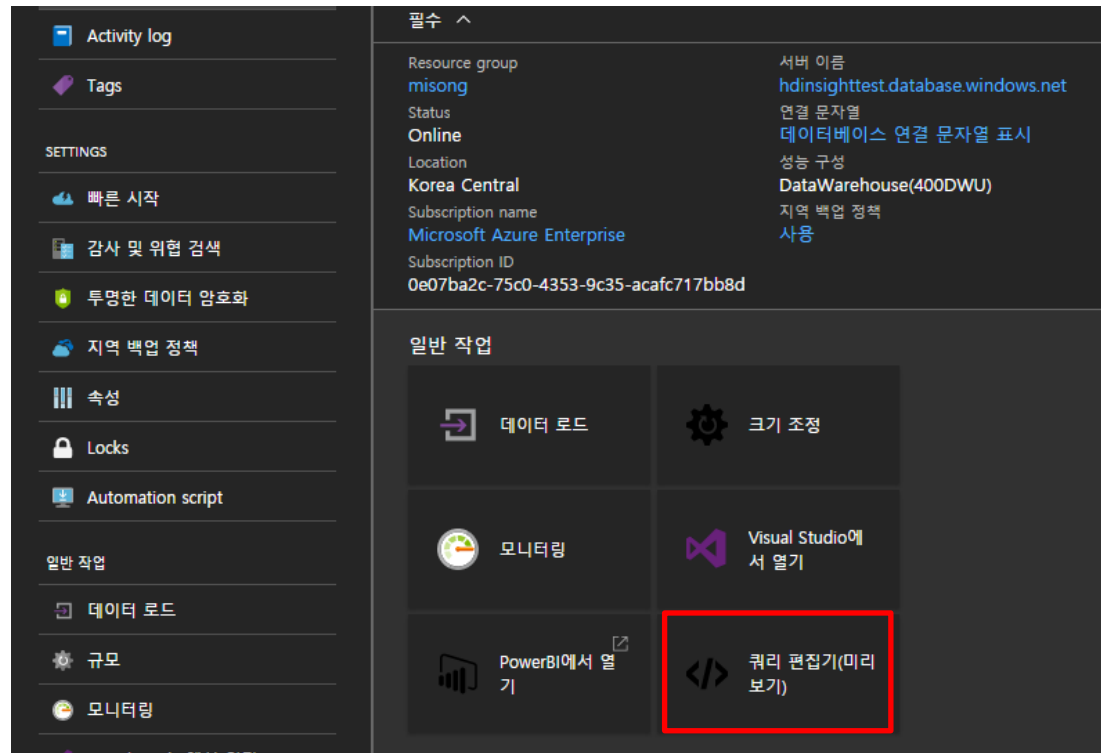
SQL Server Management Studio에서는 아래 그림과 같이 서버 이름에 Azure SQL DW가 제공하는 논리적인 서버 이름을 적어서 연결할 수 있습니다.



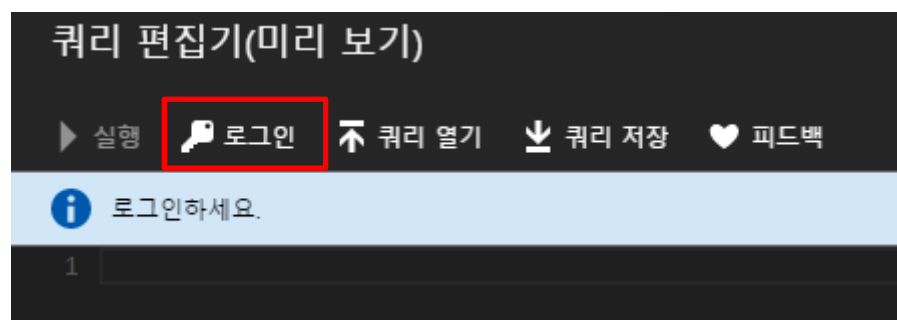
이러한 논리적인 서버의 이름은 Azure SQL DW 포털의 개요 상에서 아래와 같이 확인할 수 있습니다.



또는 포털 상에서 쿼리 편집기를 사용할 수도 있습니다. SQL Server Management Studio가 이미 설치되지 않은 환경에서 실습을 진행할 때에는 아래와 같이 쿼리 편집기를 이용할 예정입니다.



이후 쿼리 편집기에서는 로그인하라는 메시지가 나타납니다. 로그인을 위해 열쇠 모양의 로그인 버튼을 클릭해 줍니다.



그러면 아래 그림과 같이 전에 만들어준 계정이름을 넣고 SQL Server 인증으로 OK 버튼을 클릭해 줍니다.

로그인

인증 형식
SQL Server 인증

* 로그인
misong

* 암호
.....

OK Cancel

그러면 아래와 같이 로그인 중이라는 메시지가 나타납니다.

쿼리 편집기(미리 보기)

▶ 실행 🔑 로그인 ↕ 쿼리 열기 ⬇ 쿼리 저장 ❤ 피드백

i misong(으)로 로그인 중...

1

잠시 뒤 인증이 되었다는 메시지를 확인할 수 있습니다.

쿼리 편집기(미리 보기)

▶ 실행 🔑 로그인 ↕ 쿼리 열기 ⬇ 쿼리 저장 ❤ 피드백

i 다음으로 인증됨 misong

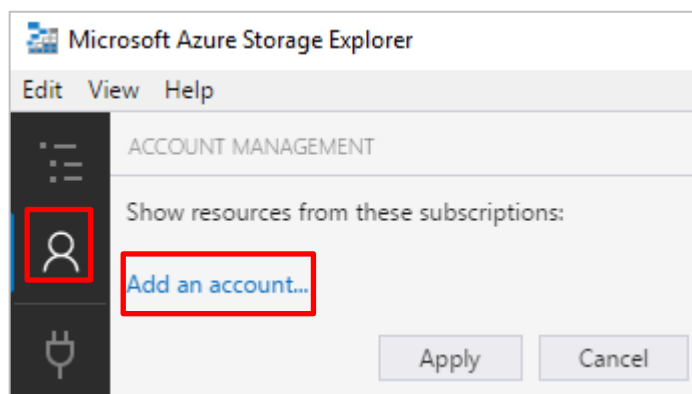
1

5.4 Spark 시스템에 샘플 파일 업로드 하기

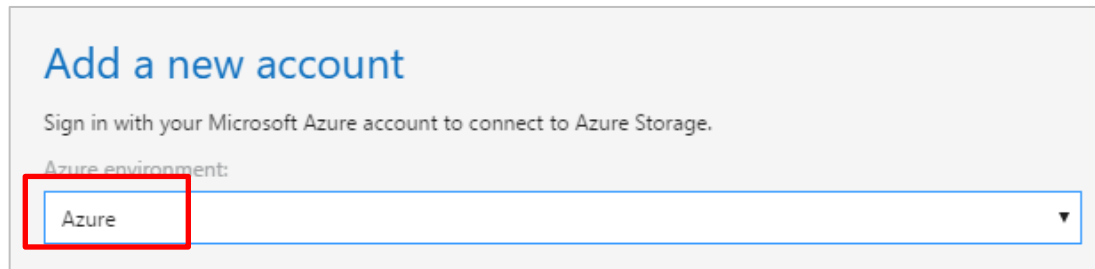
앞서 3장에서는 Spark 클러스터에 AzCopy 유틸리티를 통해 샘플 파일을 업로드 하였습니다. AzCopy 외에 또 한 가지 저장소에 접근할 수 있는 편리한 툴이 바로 Microsoft Azure Storage Explorer 입니다. Storage Explorer는 아래 링크를 통해 다운로드 받을 수 있으며, 만약 Azure 상의 여러 개의 계정과 구독이 있다고 하더라도 이를 통합해서 관리할 수 있다는 장점이 있습니다. <http://storageexplorer.com/> 이 링크에 접속하여 Windows 용 Storage Explorer를 다운로드 해주고 설치해 줍니다.



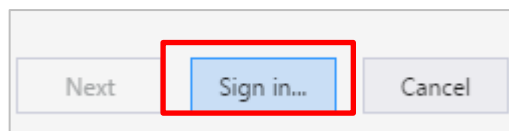
Storage Explorer 설치가 완료되었으면 이를 실행시켜 아래와 같이 사람 모양의 이모티콘을 클릭해 줍니다. 이 버튼을 통해서 계정을 추가해 구독 전체에 대한 리소스를 한 눈에 관리할 수 있습니다.



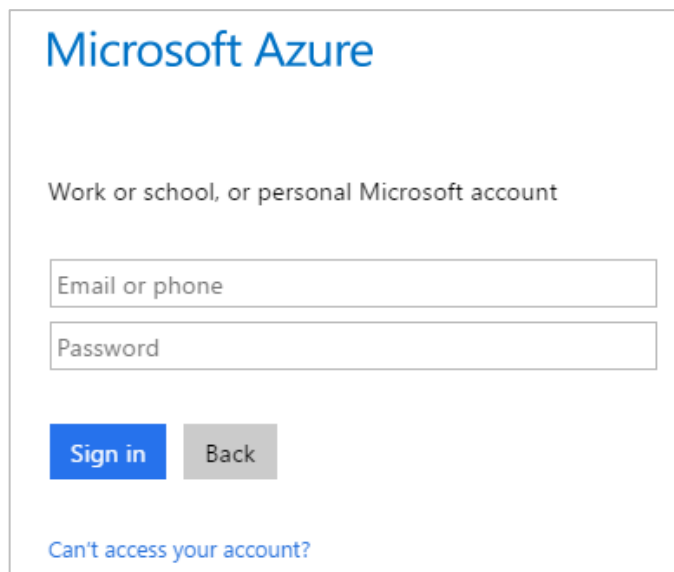
위와 같이 계정 추가하기 버튼을 클릭하면 Azure를 선택하는 화면이 나타납니다.



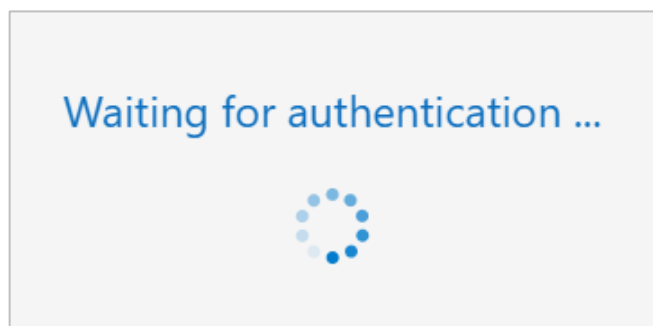
그러면 Azure가 선택된 상태로 Sign in 버튼을 클릭해 줍니다.



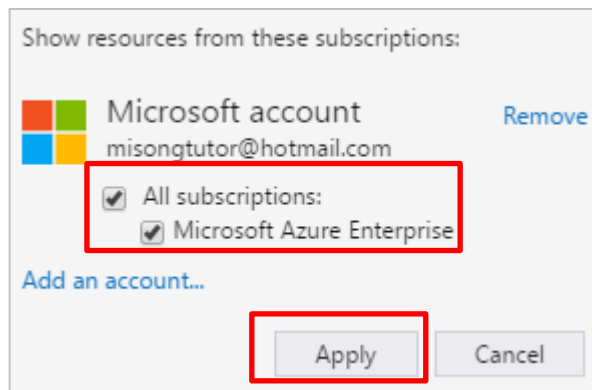
그러면 아래와 같이 Azure 계정 로그인 화면이 나타나고 본인의 Azure 계정과 비밀번호를 입력해 줍니다.



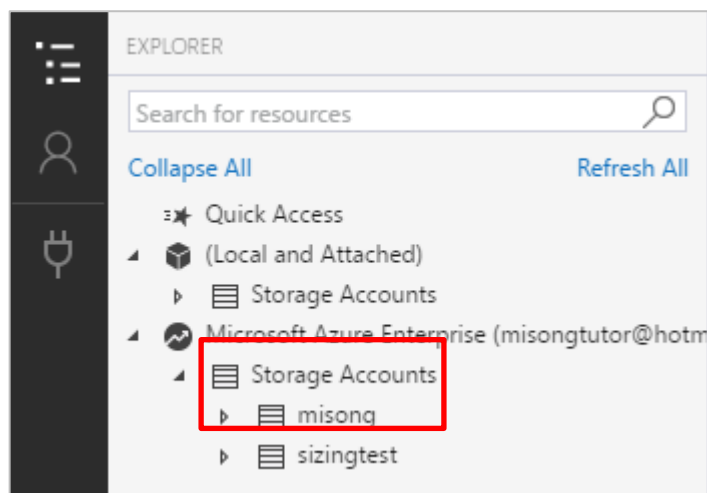
그러면 아래와 같이 로그인이 진행됩니다.



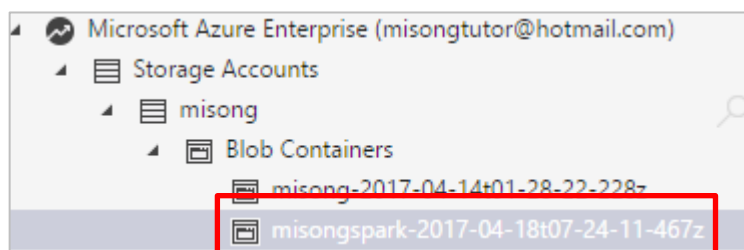
그러면 로그인한 계정과 물려 있는 Azure 구독 명들이 나타나고 이것이 체크된 상태로 Apply 적용 버튼을 클릭해 줍니다.



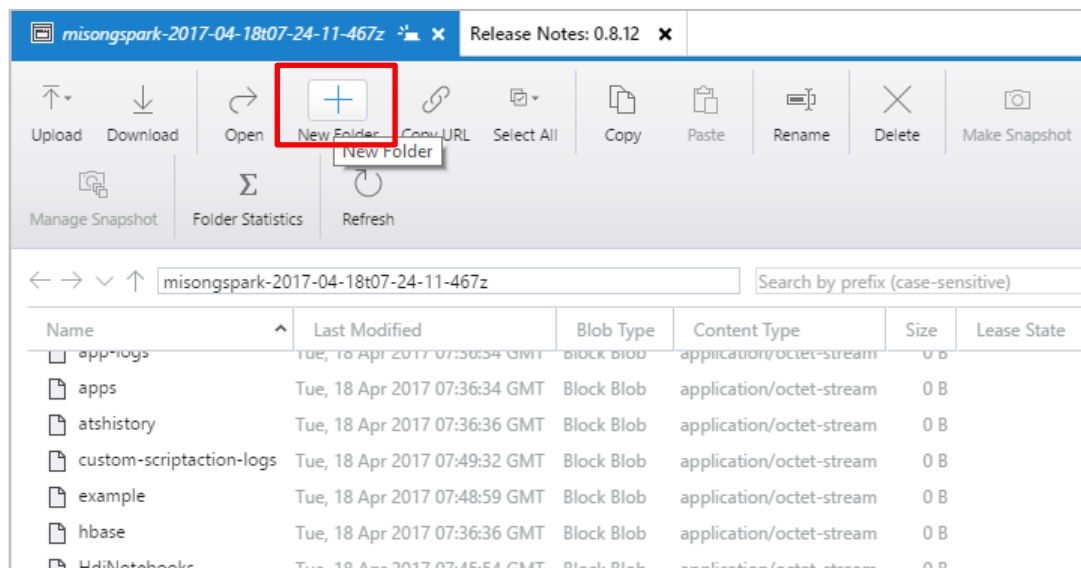
그러면 앞서 만들어진 스토리지 계정 명들이 나타나게 되고 여기서 Spark 클러스터를 만들 때 사용한 스토리지 계정을 선택해 줍니다.



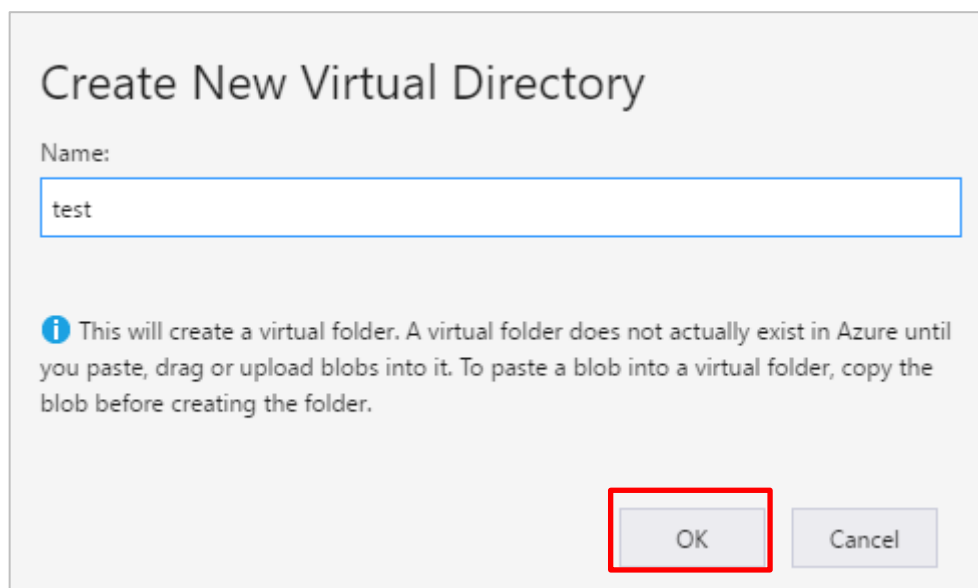
그리고 해당 계정에서 Spark 클러스터에서 사용한 컨테이너 이름을 찾아줍니다. 그리고 이를 클릭합니다.



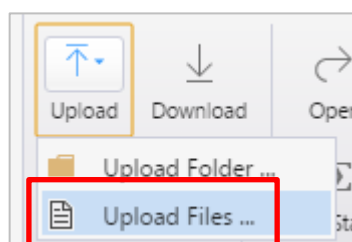
이제 나타난 목록에서 New Folder 새 폴더 만들기를 클릭해줍니다.



아래와 같이 test라는 이름의 폴더 명을 하나 지정하고 OK를 클릭해 줍니다.



그러면 이 test 폴더에 들어가 아래 그림과 같이 업로드 파일 버튼을 클릭해 줍니다.



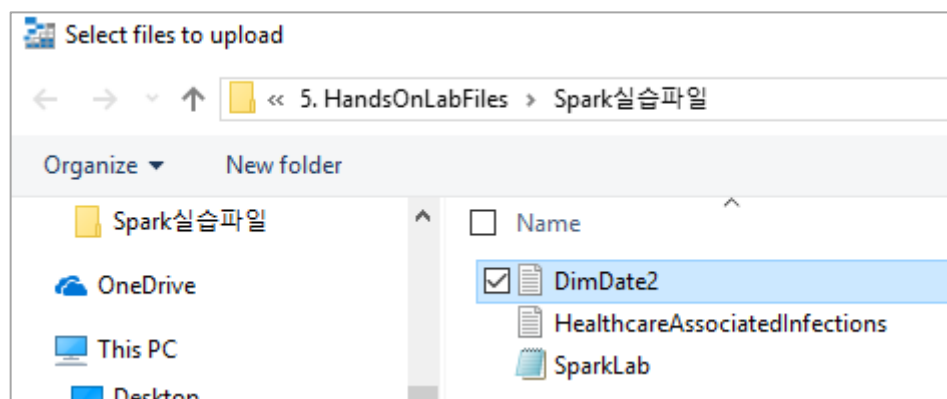
업로드 파일 페이지에서 오른쪽 ... 버튼을 클릭하여 업로드 할 파일을 찾아줍니다.

그리고 주어진 실습 파일 중 DimDate2라는 파일을 선택하여 업로드 해줍니다.
DimDate2에는 아래와 같이 텍스트가 들어 있습니다.

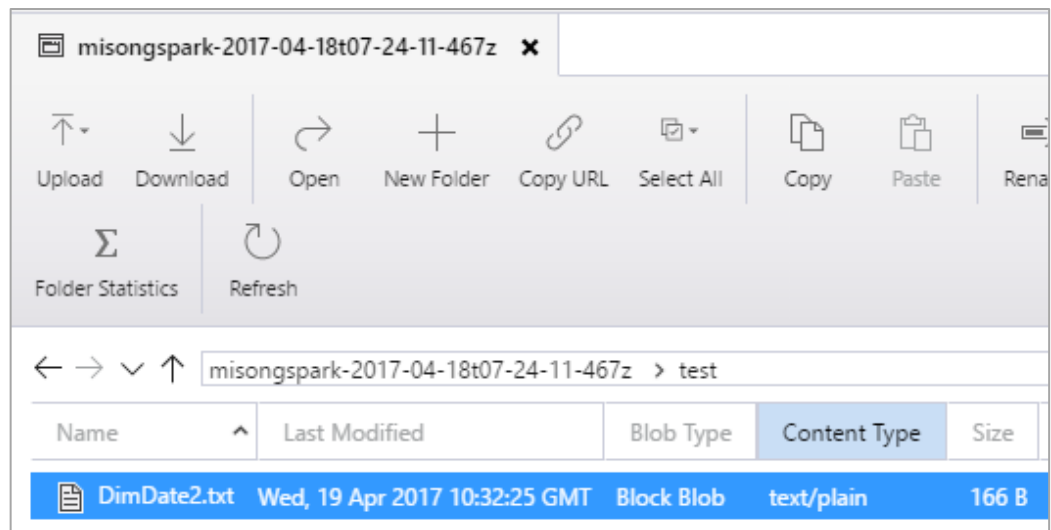
```

20150301,1,3
20150501,2,4
20151001,4,2
20150201,1,3
20151201,4,2
20150801,3,1
20150601,2,4
20151101,4,2
20150401,2,4
  
```

아래와 같이 이 파일을 클릭하여 업로드 해줍니다.



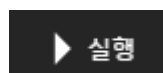
그러면 아래 화면과 같이 test 폴더에 DimDate2.txt 파일이 정상적으로 업로드 된 것을 확인할 수 있습니다.



5.5 PolyBase 기능을 활용해 Spark 시스템에 쿼리하기

먼저 PolyBase 기능을 사용하기 위해서는 서버 마스터 키를 만들어 줍니다. 쿼리 창에 아래와 같이 CREATE MASTER KEY; 를 입력하고 실행 버튼을 클릭해 줍니다. 주의하실 점은 실행 버튼 대신 SSMS를 사용하시던 습관처럼 키보드의 F5버튼을 눌러 실행하게 되면 브라우저이기 때문에 브라우저가 새로 고쳐집니다.

```
1  
2 CREATE MASTER KEY;
```



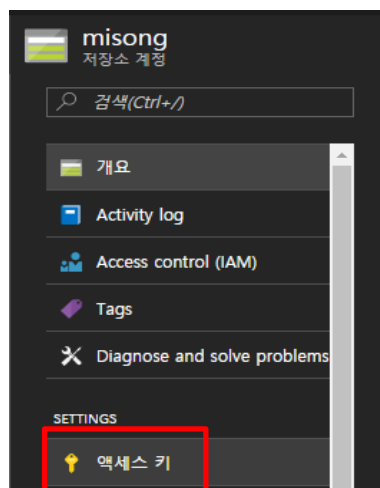
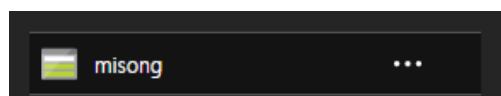
그러면 잠시 뒤 쿼리가 성공했다는 메시지를 확인할 수 있습니다.

```
결과 메시지  
쿼리 성공: 영향을 받는 행: 0.
```

다음으로는 아래와 같이 AzureStorage에 연결할 Credential을 만들어 줍니다.

```
4 CREATE DATABASE SCOPED CREDENTIAL AzureStorageCredential  
5 WITH  
6     IDENTITY = 'user',  
7     SECRET = 'f02R7GS+4YJ4XSQBJR5kUaPNAsk1Enaz5WYTINnYLB95KY8Y/XubV37Hg6HiuHETutm+KQ5hawqx1YHKhdUmXA=='  
8 ;  
9
```

여기서 IDENTITY는 string 값인 이름이 되며 SECRET에다 Azure Storage Account Key를 넣어줍니다. Azure Storage Account key는 저장소 계정 부분의 액세스 키를 선택하여 구할 수 있습니다.



아래 음영이 있는 부분에 해당 액세스 키를 복사 붙여 넣기 해줍니다.

```
4 CREATE DATABASE SCOPED CREDENTIAL AzureStorageCredential
5 WITH
6     IDENTITY = 'user',
7     SECRET = 'f02R7G5+4YJ4XSQB3R5kUaPNAsk1Enaz5WYTIInNYLb95KY8Y/XubV37Hg6HiuHETutm+KQ5hawqx1YHKhdUmXA=='
8 ;
9
```

아래 그림과 같이 CREATE 부터 ;까지 선택하고 실행 버튼을 클릭해 줍니다.

```
4 CREATE DATABASE SCOPED CREDENTIAL AzureStorageCredential
5 WITH
6     IDENTITY = 'user',
7     SECRET = 'f02R7G5+4YJ4XSQB3R5kUaPNAsk1Enaz5WYTIInNYLb95KY8Y/XubV37Hg6HiuHETutm+KQ5hawqx1YHKhdUmXA=='
8 ;
9
```

그러면 아래와 같이 쿼리 성공 메시지가 나타납니다.

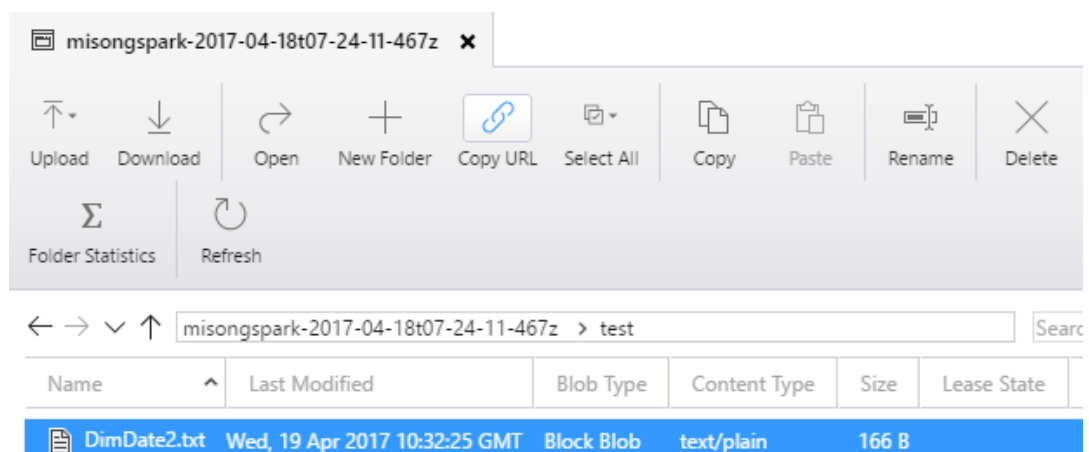
결과 메시지

쿼리 성공: 영향을 받는 행: 0.

이제 외부 데이터 소스를 지정해줄 차례입니다. 아래와 같이 컨테이너명에는 블랍 컨테이너와 스토리지 계정 명에는 스토리지 계정 명으로 대체해 줍니다. 이를 빠르게 확인할 수 있는 방법은

```
CREATE EXTERNAL DATA SOURCE AzureStorage
WITH (
    TYPE = HADOOP,
    LOCATION = 'wasbs://<컨테이너명>@<스토리지계정명>.blob.core.windows.net',
    --https://misong.blob.core.windows.net/misongspark-2017-04-18t07-24-11-467z/test/DimDate2.txt
    CREDENTIAL = AzureStorageCredential
);
```

Storage Explorer에서 아래와 같이 해당 파일을 선택한 뒤 Copy URL 버튼을 클릭하는 것입니다.



그래서 빈 메모장 등에 붙여 넣어보면 아래와 같이 misong이 스토리지 계정 명이며, test 폴더 앞에 위치한 이름이 블랍 컨테이너 명입니다.

<https://misong.blob.core.windows.net/misongspark-2017-04-18t07-24-11-467z/test/DimDate2.txt>

이를 아래 그림과 같이 채워줍니다. 그리고 아래 데이터 소스 만드는 영역을 선택하여 다시 실행 버튼을 클릭해 줍니다.

```
12 CREATE EXTERNAL DATA SOURCE AzureStorage
13 WITH (
14     TYPE = HADOOP,
15     LOCATION = 'wasbs://misongspark-2017-04-18t07-24-11-467z@misong.blob.core.windows.net',
16     --https://misong.blob.core.windows.net/misongspark-2017-04-18t07-24-11-467z/test/DimDate2.txt
17     CREDENTIAL = AzureStorageCredential
18 );
```

▶ 실행

그러면 마찬가지로 쿼리가 성공적으로 완료되었다는 메시지를 확인할 수 있습니다.

결과 메시지

쿼리 성공: 영향을 받는 행: 0.

이제 외부 데이터 파일 포맷을 만듭니다. 아래와 같이 TextFile이라는 파일 포맷을 하나 만들어주고, 쉼표를 통해 구분하게 하는 파일 포맷을 지정해 줍니다. 마찬가지로 이 쿼리도 영역을 선택하여 실행 버튼을 클릭해 줍니다.

```
20 CREATE EXTERNAL FILE FORMAT TextFile
21 WITH (
22     FORMAT_TYPE = DelimitedText,
23     FORMAT_OPTIONS (FIELD_TERMINATOR = ',')
24 );
```

▶ 실행

그러면 아래와 같이 쿼리가 성공적으로 수행됩니다.

결과 메시지

쿼리 성공: 영향을 받는 행: 0.

이후 아래 쿼리를 수행하여 외부 테이블을 만들어 줍니다.

```
CREATE EXTERNAL TABLE dbo.DimDate2External (
    DateId INT NOT NULL,
    CalendarQuarter TINYINT NOT NULL,
    FiscalQuarter TINYINT NOT NULL
)
WITH (
    LOCATION='/test/',
    DATA_SOURCE=AzureStorage,
    FILE_FORMAT=TextFile
);
```

▶ 실행

그러면 아래와 같이 쿼리가 성공적으로 수행됩니다.

결과	메시지
쿼리 성공: 영향을 받는 행: 0.	

만약 여기서 오류가 나는 경우 앞의 Azure Blob Storage의 URL이 잘못되지 않았는지 확인해 보실 수 있습니다.

그리고 아래와 같이 SELECT 쿼리를 수행하게 되면 아래와 같이 UI 상으로 조회된 Spark 클러스터의 데이터를 확인할 수 있습니다.

```
43 SELECT * FROM dbo.DimDate2External;
```

▶ 실행

43 SELECT * FROM dbo.DimDate2External;		
44		
Results Messages		
Search to filter items...		
DATEID	CALENDARQUARTER	FISCALQUARTER
20150301	1	3
20150501	2	4
20151001	4	2
20150201	1	3
20151201	4	2
20150801	3	1
20150601	2	4
20151101	4	2

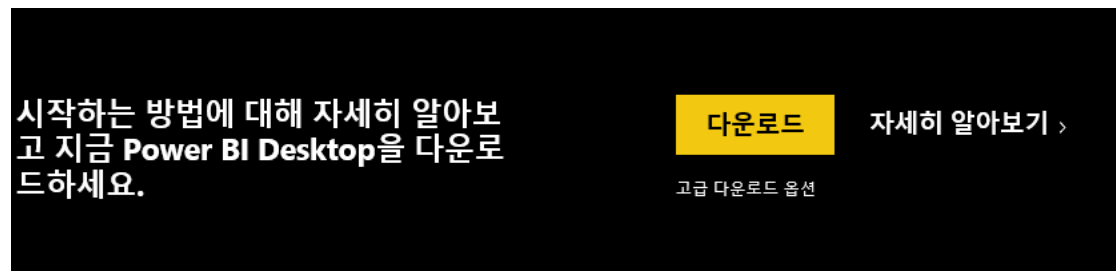
이를 통해 Azure SQL DW에서 Spark 클러스터 시스템에 올라온 데이터를 마치 DBMS와 같이 조회할 수 있음을 확인해 보았습니다.

6. (응용) Power BI에서 연결하기

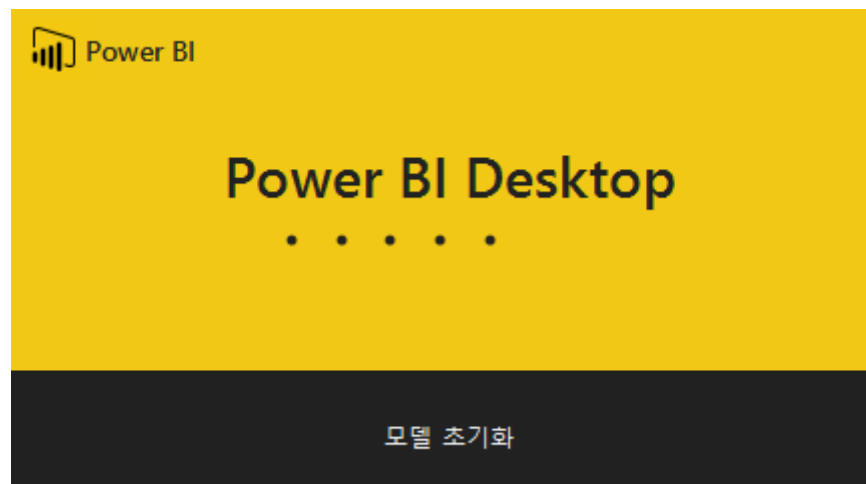
6.1 Power BI 데스크톱 다운로드

먼저 Power BI 솔루션에서 Hive Table에 연결하기 위해서 Power BI Desktop 파일을 다운로드 받습니다. 다운로드는 아래 링크에서 받을 수 있습니다.

<https://powerbi.microsoft.com/ko-kr/desktop/>

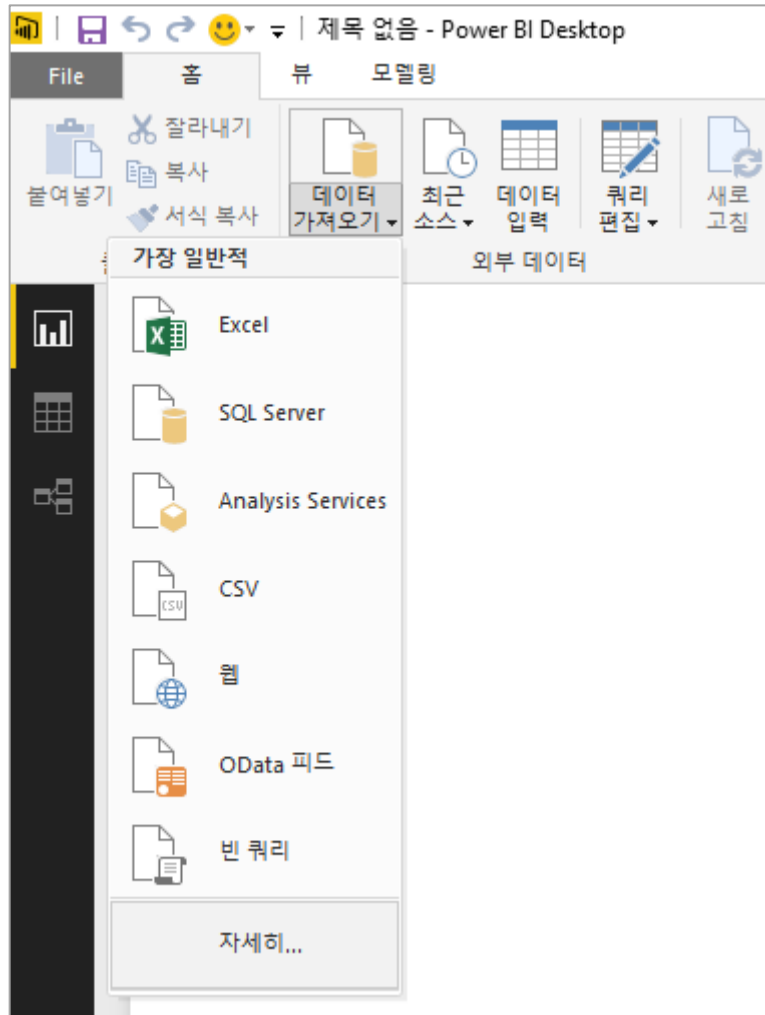


다운로드 받은 파일을 실행하여 Power BI Desktop을 설치해 줍니다. 그리고 설치가 완료되었으면 Power BI Desktop 프로그램을 검색하여 실행시켜 줍니다. 아래 그림은 실행 화면입니다.

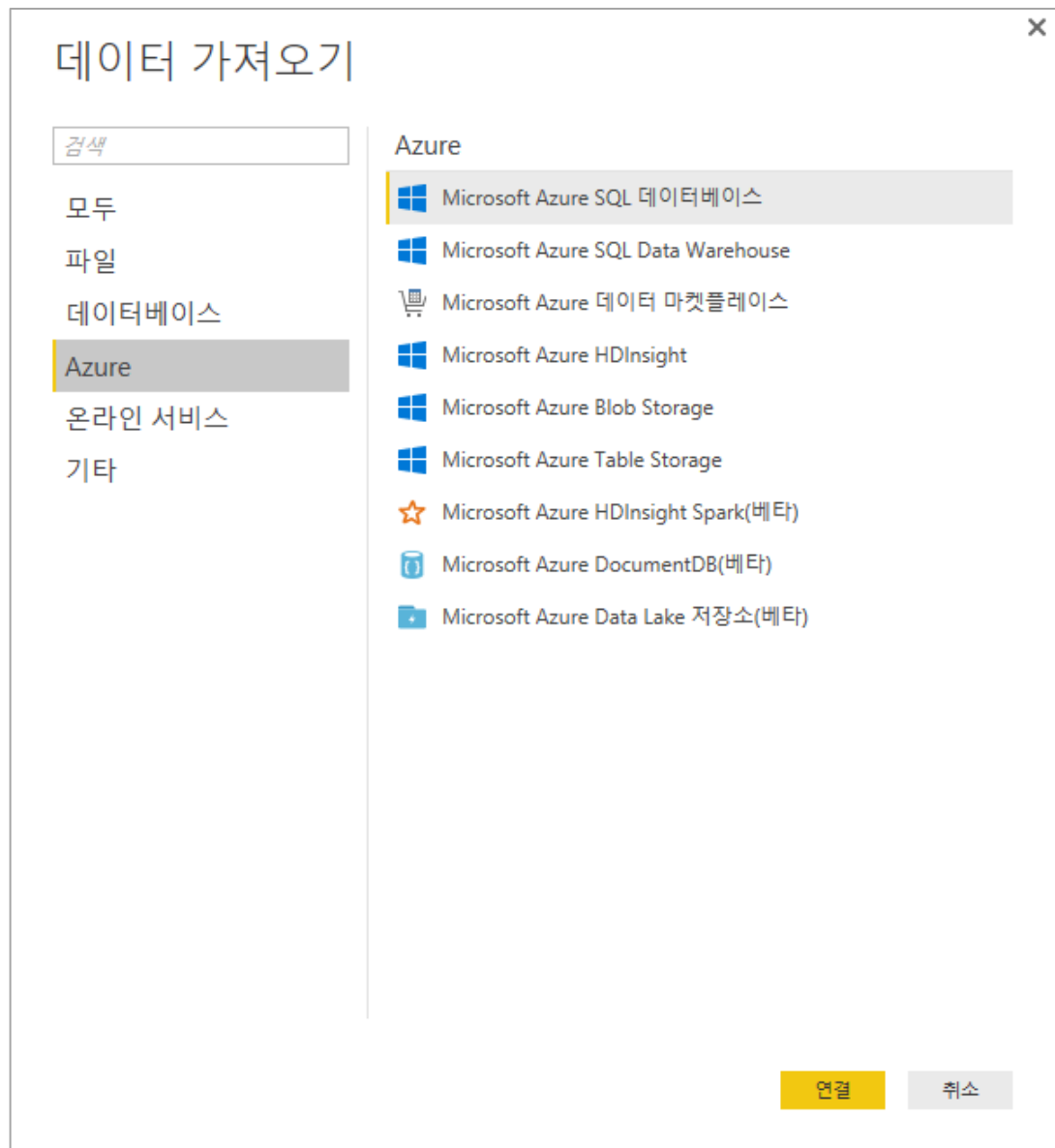


6.2 Power BI 데스크톱에서 Hive Table 연결하기

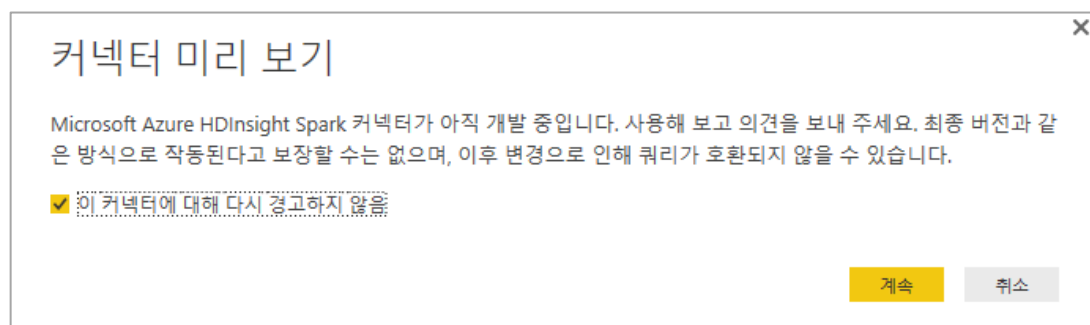
상단의 Home 탭에서 아래 그림과 같이 [데이터 가져오기] 버튼을 클릭해 줍니다.
그리고 나타난 상태에서 [자세히...] 버튼을 클릭해 줍니다.



그리고 아래 가져오기 화면에서 [Azure] > [Microsoft Azure HDInsight Spark] 버튼을 클릭하고 [연결]을 클릭합니다.



그리고 아래 화면과 같이 아직 개발중이라는 경고 창이 나타나며 [계속] 버튼을 클릭해 줍니다.



이는 아직 커넥터가 개발 단계여서 나타나는 메시지이며, 개발이 완료되면 사라질 예정입니다.

그리고 아래 화면처럼 클러스터 URL을 복사하여 붙여넣기 해줍니다. 클러스터 URL 정보는 Spark 대시보드의 개요 부분에서 확인할 수 있습니다. 그리고 확인 버튼을 클릭해 줍니다.

Microsoft Azure HDInsight Spark

Azure Spark 인스턴스의 테이블을 나열합니다.

서버

☒ 가져오기

☐ DirectQuery

확인

취소

aaworkshop

HDInsight cluster

Search (Ctrl+/)

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Dashboard

Secure Shell (SSH)

Scale cluster

Delete

Essentials

Resource group (change)

aaworkshop

Status

Running

Location

Japan West

Subscription name (change)

Microsoft Azure Internal Consumption

Subscription ID

f8daaf1-d911-4b99-bc94-b2dc83bf5e5

Cluster type: HDI version

Spark on Linux (HDI 3.4.1000.0)

URL

<https://aaworkshop.azurehdinsight.net>

Learn more

Documentation

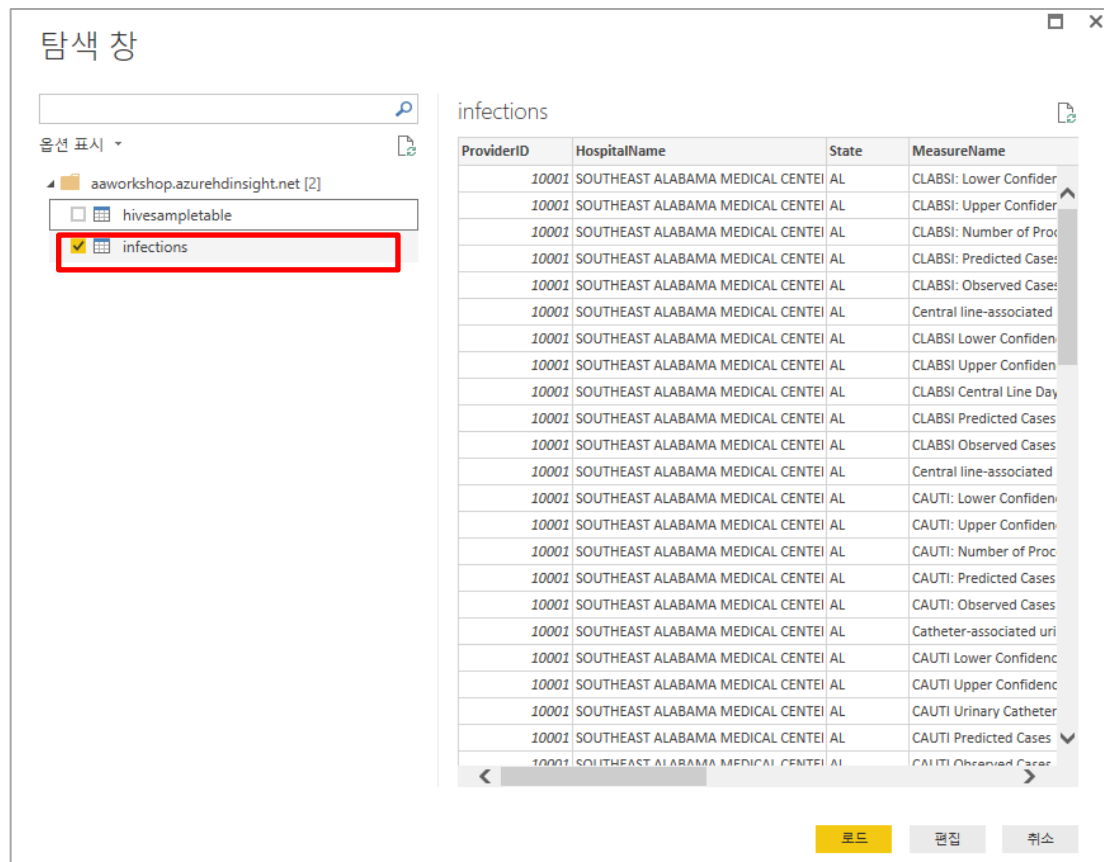
Getting started

Quickstart

Head Nodes, Worker nodes

D12 (x2), D4 (x2)

그러면 샘플 화면 창에 아래와 같이 Hive 테이블 목록이 나타나게 되며, 여기서 Infection이라고 만든 Hive 테이블을 클릭하고 [로드] 버튼을 눌러줍니다.



탐색 창

옵션 표시

aaworkshop.azurehdinsight.net [2]

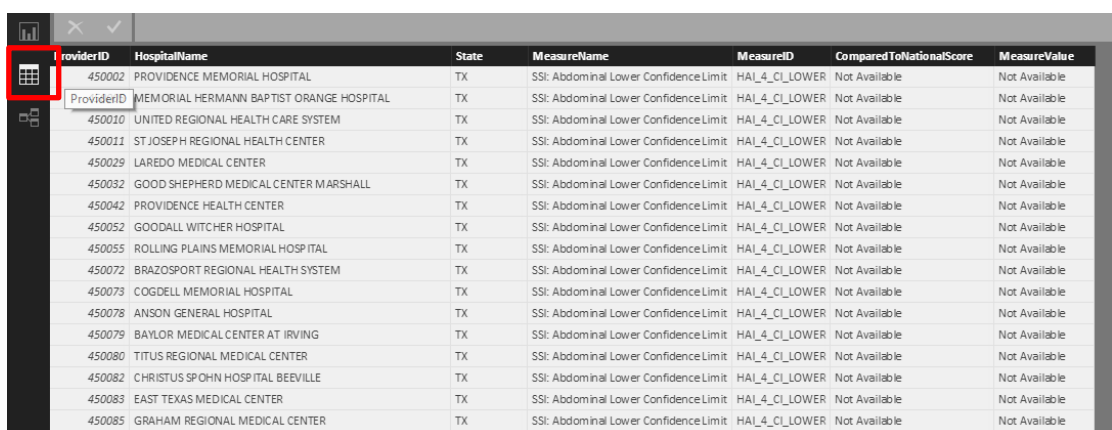
- hivesampletable
- infections**

infections

ProviderID	HospitalName	State	MeasureName
10001	SOUTHEAST ALABAMA MEDICAL CENTER	AL	CLABSI: Lower Confidence
10001	SOUTHEAST ALABAMA MEDICAL CENTER	AL	CLABSI: Upper Confidence
10001	SOUTHEAST ALABAMA MEDICAL CENTER	AL	CLABSI: Number of Pro
10001	SOUTHEAST ALABAMA MEDICAL CENTER	AL	CLABSI: Predicted Cases
10001	SOUTHEAST ALABAMA MEDICAL CENTER	AL	CLABSI: Observed Cases
10001	SOUTHEAST ALABAMA MEDICAL CENTER	AL	Central line-associated
10001	SOUTHEAST ALABAMA MEDICAL CENTER	AL	CLABSI Lower Confidence
10001	SOUTHEAST ALABAMA MEDICAL CENTER	AL	CLABSI Upper Confidence
10001	SOUTHEAST ALABAMA MEDICAL CENTER	AL	CLABSI Central Line Day
10001	SOUTHEAST ALABAMA MEDICAL CENTER	AL	CLABSI Predicted Cases
10001	SOUTHEAST ALABAMA MEDICAL CENTER	AL	CLABSI Observed Cases
10001	SOUTHEAST ALABAMA MEDICAL CENTER	AL	Central line-associated
10001	SOUTHEAST ALABAMA MEDICAL CENTER	AL	CAUTI: Lower Confidence
10001	SOUTHEAST ALABAMA MEDICAL CENTER	AL	CAUTI: Upper Confidence
10001	SOUTHEAST ALABAMA MEDICAL CENTER	AL	CAUTI: Number of Pro
10001	SOUTHEAST ALABAMA MEDICAL CENTER	AL	CAUTI: Predicted Cases
10001	SOUTHEAST ALABAMA MEDICAL CENTER	AL	CAUTI: Observed Cases
10001	SOUTHEAST ALABAMA MEDICAL CENTER	AL	Catheter-associated uri
10001	SOUTHEAST ALABAMA MEDICAL CENTER	AL	CAUTI Lower Confidence
10001	SOUTHEAST ALABAMA MEDICAL CENTER	AL	CAUTI Upper Confidence
10001	SOUTHEAST ALABAMA MEDICAL CENTER	AL	CAUTI Urinary Catheter
10001	SOUTHEAST ALABAMA MEDICAL CENTER	AL	CAUTI Predicted Cases
10001	SOUTHEAST ALABAMA MEDICAL CENTER	AL	CAUTI Observed Cases

로드 편집 취소

그러면 왼쪽 상단의 테이블 모양의 데이터 탭을 클릭하였을 때 로드 된 Infection 테이블의 데이터를 확인할 수 있습니다.



ProviderID	HospitalName	State	MeasureName	MeasureID	ComparedToNationalScore	MeasureValue
450002	PROVIDENCE MEMORIAL HOSPITAL	TX	SSI: Abdominal Lower Confidence Limit	HAL_4_CI_LOWER	Not Available	Not Available
450002	MEMORIAL HERMANN BAPTIST ORANGE HOSPITAL	TX	SSI: Abdominal Lower Confidence Limit	HAL_4_CI_LOWER	Not Available	Not Available
450010	UNITED REGIONAL HEALTH CARE SYSTEM	TX	SSI: Abdominal Lower Confidence Limit	HAL_4_CI_LOWER	Not Available	Not Available
450011	ST JOSEPH REGIONAL HEALTH CENTER	TX	SSI: Abdominal Lower Confidence Limit	HAL_4_CI_LOWER	Not Available	Not Available
450029	LAREDO MEDICAL CENTER	TX	SSI: Abdominal Lower Confidence Limit	HAL_4_CI_LOWER	Not Available	Not Available
450032	GOOD SHEPHERD MEDICAL CENTER MARSHALL	TX	SSI: Abdominal Lower Confidence Limit	HAL_4_CI_LOWER	Not Available	Not Available
450042	PROVIDENCE HEALTH CENTER	TX	SSI: Abdominal Lower Confidence Limit	HAL_4_CI_LOWER	Not Available	Not Available
450052	GOODALL WITCHER HOSPITAL	TX	SSI: Abdominal Lower Confidence Limit	HAL_4_CI_LOWER	Not Available	Not Available
450055	ROLLING PLAINS MEMORIAL HOSPITAL	TX	SSI: Abdominal Lower Confidence Limit	HAL_4_CI_LOWER	Not Available	Not Available
450072	BRAZOSPORT REGIONAL HEALTH SYSTEM	TX	SSI: Abdominal Lower Confidence Limit	HAL_4_CI_LOWER	Not Available	Not Available
450073	COGDELL MEMORIAL HOSPITAL	TX	SSI: Abdominal Lower Confidence Limit	HAL_4_CI_LOWER	Not Available	Not Available
450078	ANSON GENERAL HOSPITAL	TX	SSI: Abdominal Lower Confidence Limit	HAL_4_CI_LOWER	Not Available	Not Available
450079	BAYLOR MEDICAL CENTER AT IRVING	TX	SSI: Abdominal Lower Confidence Limit	HAL_4_CI_LOWER	Not Available	Not Available
450080	TITUS REGIONAL MEDICAL CENTER	TX	SSI: Abdominal Lower Confidence Limit	HAL_4_CI_LOWER	Not Available	Not Available
450082	CHRISTUS SPOHN HOSPITAL BEEVILLE	TX	SSI: Abdominal Lower Confidence Limit	HAL_4_CI_LOWER	Not Available	Not Available
450083	EAST TEXAS MEDICAL CENTER	TX	SSI: Abdominal Lower Confidence Limit	HAL_4_CI_LOWER	Not Available	Not Available
450085	GRAHAM REGIONAL MEDICAL CENTER	TX	SSI: Abdominal Lower Confidence Limit	HAL_4_CI_LOWER	Not Available	Not Available

그리고 다시 대시보드 탭으로 넘어옵니다.



아래 화면처럼 오른쪽 패널에서 데이터를 시각화 하고 싶은 몇 가지 데이터들을 클릭해보고 이를 다양한 그래프로 표현해 봅니다.

