

Azure Service

Azure Scale Analytics Workshop

Day 2

Cosmos DB Real-Time Advanced Analytics

이 문서는 데이터솔루션과 마이크로소프트가 공동으로 제작한 자료입니다.

실습 환경 설정

○ 요구 사항

- Microsoft Azure 구독 (Microsoft 이외의 구독은 유료 구독이어야 합니다).
- 중요** :이 실습의 OAuth 2.0 액세스 구성 요소를 완료하려면 Azure 구독 내에 Azure Active Directory 내에 앱 등록 및 서비스 주체를 만들 수 있는 권한이 있어야 합니다.

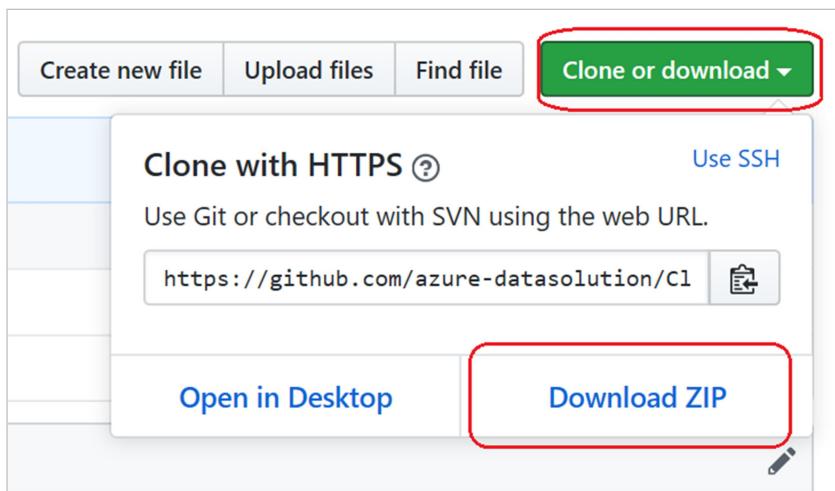
○ 실습 전에

[소요 시간 : 20 분]

작업 1 : 스타터 파일 다운로드

랩 솔루션에서 처리하기 위해 실시간 지불 데이터와 랩에서 사용되는 데이터 파일을 전송하는 지불 데이터 생성기가 포함 된 스타터 프로젝트를 다운로드 하십시오.

- LabVM에서 GitHub 저장소의 .zip 사본을 다운로드하십시오.
- 웹 브라우저에서 [Cloud Scale Analytics Hands on Lab repo](#) 로 이동하십시오.
(GitURL로 이동 : <https://github.com/azure-datasolution/CloudScaleAnalytics>)
- 저장소 페이지에서 **복제 또는 다운로드**를 선택한 다음 **ZIP 다운로드**를 선택하십시오.

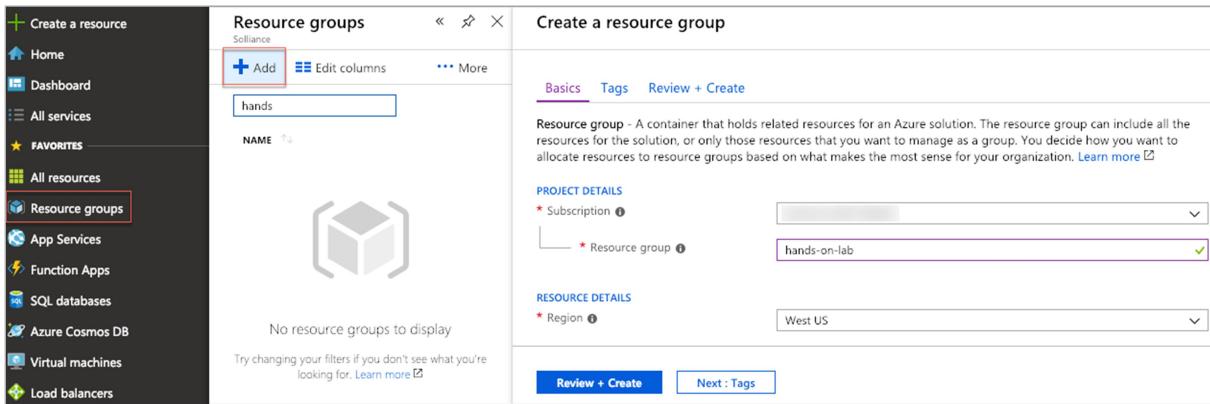


- 폴더의 내용을 다음 경로에 압축 해제하십시오 **C:\CloudScaleAnalytics**.

작업 2 : 리소스 그룹 만들기 (선택)

이 작업에서는 이 랩 전체에서 사용되는 리소스에 대한 Azure 리소스 그룹을 만듭니다.

1. [애저 포털](#)에서, 선택 자원 그룹, 왼쪽 탐색 메뉴에서를 선택 + 추가, 다음은 Azure Resource Group (자원 그룹)을 만듭니다 블레이드에 다음을 입력 :
 - **구독** :이 실습에 사용중인 구독을 선택하십시오.
 - **자원 그룹 이름** : hands-on-lab-SUFFIX(**실습에서는 mobile 뒷 자리 4자리 사용 권장**)를 입력하십시오.
 - **지역** :이 실습에서 리소스에 사용하려는 지역을 선택합니다. 이 실습 전체에서 프로비저닝 할 다른 리소스에 사용할 수 있도록 이 위치를 기억하십시오.

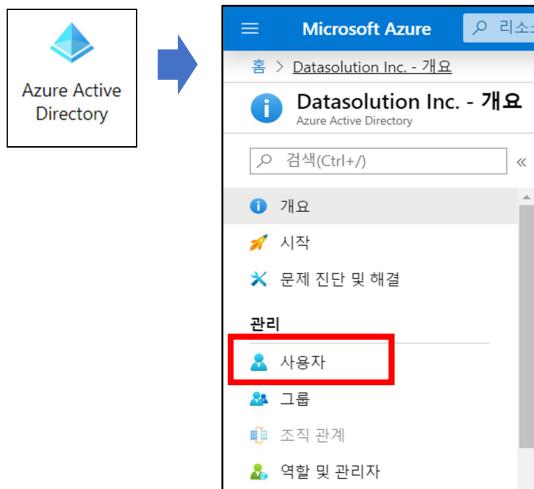


2. 검토 + 작성을 선택하십시오 .

3. 요약 블레이드에서 생성을 선택하여 리소스 그룹을 프로비저닝 합니다.

작업 3 : 환경 배포

1. Azure Portal에서 Azure Active Directory로 이동한 다음 관리에서 사용자를 선택합니다.



2. 현재 Azure Portal 에 로그인 한 사용자를 선택하십시오.

(본인의 계정을 검색으로 찾을 수 있습니다.)

The screenshot shows the Azure Active Directory 'All Users' page. On the left, there's a sidebar with options like 'All users', 'Deleted users', 'Change password', etc. The main area has a search bar with 'yh201' and a results table. One row is selected, showing '이영' (Lee Young) and '@spssk' (spssk). A red box highlights the 'Object ID' column, which contains 'fd24637a-3aee-40ab-b8bf-2c5ad4...'.

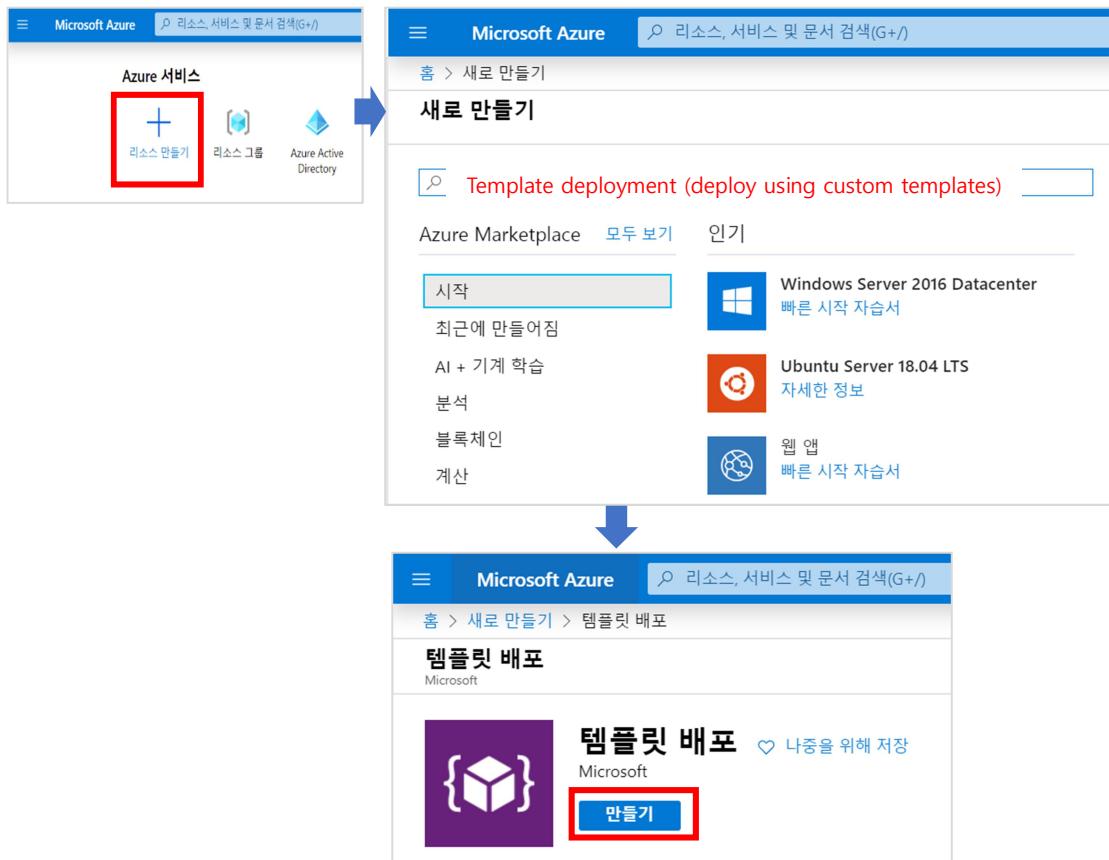
3. 사용자 블레이드에서, 개체 ID 를(Object ID)를 복사해 놓으세요.

(※ 이번 실습에서 텍스트를 복사해서 붙여넣기를 하는 작업이 많기 때문에 메모장이나 Notepad에 붙여넣고 사용하는 것이 용이합니다.)

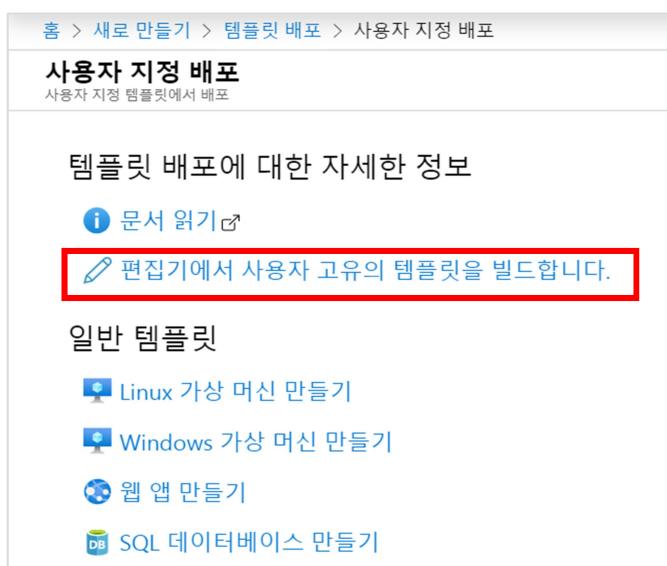
The screenshot shows the Azure Active Directory 'Profile' page for a user named '이영'. The sidebar includes 'All users', 'Deleted users', 'Change password', and 'Log in'. The main area shows a profile picture, login statistics (40 logins), and an 'ID' section. In the 'ID' section, the 'Object ID' field is highlighted with a red box and contains 'fd24637a-3aee-40ab-b8bf-2c5ad4...'. Below it, the 'Source' is listed as 'Azure Active Directory'.

4. 리소스를 생성에서 '템플릿 배포(Template Deployment)'을 검색창에 입력하세요.

템플릿 배포를 선택하고 만들기를 누르세요.



5. 사용자 지정 배포(Custom Deployment)에서, '편집기에서 사용자 고유의 템플릿을 빌드(Build your own template in the editor)' 링크를 선택합니다.



6. 템플릿 편집에서, Azure Resource Manager 템플릿을 선택하기 위해 파일로드(Load File) 버튼을 누릅니다.

홈 > 새로 만들기 > 템플릿 배포 > 사용자 지정 배포 > 템플릿 편집

템플릿 편집

Azure Resource Manager 템플릿 편집

+ 리소스 추가 ↑ 빠른 시작 템플릿 ↓ 파일 로드 다운로드

매개 변수(0)
변수(0)
리소스(0)

```

1 {
2   "$schema": "https://schema.mar
3   "contentVersion": "1.0.0.0",
4   "parameters": {},
5   "resources": []
6 }

```

7. 'C:\CloudScaleAnalytics\Hands on Lab\Day2\Deployment\environment-template.json' 파일을 선택하십시오.

8. json 파일이 로드가 되면 저장을 클릭하십시오.

홈 > 새로 만들기 > 템플릿 배포 > 사용자 지정 배포 > 템플릿 편집

템플릿 편집

Azure Resource Manager 템플릿 편집

+ 리소스 추가 ↑ 빠른 시작 템플릿 ↓ 파일 로드 다운로드

매개 변수(1)
변수(15)
리소스(6)

```

1 {
2   "$schema": "https://schema.management.azure.com/schemas/2015-01-
3   "contentVersion": "1.0.0.0",
4   "parameters": {
5     "KeyVaultAccessPolicyUserObjectId": {
6       "type": "string",
7       "defaultValue": "",
8       "metadata": {
9         "description": "ObjectId for the Azure AD User runni
10      }
11    },
12    "variables": {
13      "name_suffix": "[uniqueString(resourceGroup().id)]",
14      "primary_region": "Korea Central",
15      "secondary_region": "Korea South",
16      "cosmosdb_databaseAccount_name": "[concat('woodgrove-db-', v
17      "cosmosdb_region": "[variables('primary_region')]",
18      "cosmosdb_region_secondary": "[variables('secondary_region')",
19      "cosmosdb_databaseName": "Woodgrove",
20      "cosmosdb_databaseId": "[resourceId('Microsoft.DocumentDB/
21      "cosmosdb_databaseId": "[resourceId('Microsoft.DocumentDB/

```

저장 취소

9. 다음 값을 입력하십시오.

- 리소스 그룹 : 랩에 대해 이전에 작성된 각자의 리소스 그룹을 선택하십시오.
- Key Vault Access Policy User Object Id (키 저장소 액세스 정책 사용자 개체 ID) : 이전에 Azure Active Directory에서 복사한 사용자 개체 ID에 붙여 넣기
- 약관 동의에 체크함
- 구매 버튼을 누름

홈 > 새로 만들기 > 템플릿 배포 > 사용자 지정 배포

사용자 지정 배포

사용자 지정 템플릿에서 배포

사용자 지정된 템플릿
6개 리소스
템플릿 편집
매개 변수 편...
자세한 정보

기본 사항

구독 *	Microsoft Azure
리소스 그룹 *	리소스 그룹 선택 새로 만들기
위치 *	(아시아 태평양) 대한민국 중부

설정

Key Vault Access Policy User Object Id

사용 약관

Azure Marketplace 약관 | Azure Marketplace

[구독]를 클릭함으로써 본인은 (a) 제품과 관련된 해당 약관에 동의하고, (b) 제품 사용을 중단할 때까지 Azure 구독과 동일한 청구 주기를 사용하여 Microsoft가 현재 결제 방법으로 관련 세금을 비롯해 제품과 연관된 요금을 청구하도록 권한을 부여하는 데 동의합니다. 또한 (c) 배포에 타사 제품이 포함되는 경우 Microsoft가 내 연락처 정보 및 이러한 배포의 기타 세부 정보를 해당 제품의 게시자와 공유할 수 있다는 데 동의합니다.

위에 명시된 사용 약관에 동의함

구매

10. 배포에는 5-10 분이 소요될 수 있습니다.

The screenshot shows the 'Microsoft.Template - Overview' page in the Azure portal. The main heading says '... Your deployment is underway'. Below it, there's a note: 'Check the status of your deployment, manage resources, or troubleshoot deployment issues. Pin this page to your dashboard to easily find it next time.' A purple icon with a cube symbol is shown, followed by deployment details: Deployment name: Microsoft.Template, Subscription: [REDACTED], Resource group: hands-on-lab. Under 'DEPLOYMENT DETAILS (Download)', it shows Start time: 5/12/2019, 12:57:19 PM, Duration: 41 seconds, and Correlation ID: [REDACTED]. A table lists the deployed resources:

RESOURCE	TYPE	STATUS	OPERATION DETAILS
woodgrove-db-wsz7okrr...	Microsoft.DocumentDB/...	OK	Operation details
woodgrove-wsz7okrr7u7sc	Microsoft.KeyVault/vaults	OK	Operation details
cosmosdb-mcw-wsz7okr...	Microsoft.Databricks/wo...	Created	Operation details
woodgrove-wsz7okrr7u...	Microsoft.EventHub/na...	OK	Operation details
adlsgen2stgwsz7okrr7u...	Microsoft.Storage/stora...	OK	Operation details

- 실습을 수행하기 전에 제공된 모든 단계를 따라야 합니다. -

실습 랩 가이드

Cosmos DB 실시간 고급 분석 - 단계별 목차

- 학습 목표 및 요약
- 개요
- 솔루션 아키텍처
- 요구 사항
- 연습 1 : 스트리밍 트랜잭션 데이터 수집
 - 작업 1 : Cosmos DB로 스트리밍 데이터 수집
- 연습 2 : 대규모 거래 데이터 이해 및 준비
 - 작업 1 : ADLS Gen2 파일 시스템에 대한 OAuth 액세스를 위한 서비스 주체 만들기
 - 작업 2 : Azure Key Vault에 서비스 사용자 자격 증명 및 테넌트 ID 추가
 - 작업 3 : 키 저장소에서 ADLS Gen2 저장소 계정 구성
 - 작업 4 : Key Vault에서 Cosmos DB 키 구성
 - 작업 5 : Azure Databricks 클러스터 만들기
 - 작업 6 : Azure Databricks 열기 및 랩 노트북
 - 작업 7 : Azure Databricks 주요 자격 증명 지원 비밀 구성
 - 작업 8 : Databricks에 Azure Cosmos DB Spark 커넥터 및 scikit-learn 라이브러리 설치
 - 작업 9 : Azure Databricks 및 Spark를 사용하여 기록 트랜잭션 데이터 탐색
 - 작업 10 : Azure Databricks에서 Cosmos DB 변경 피드 및 Spark 구조적 스트리밍을 사용하여 스트리밍 트랜잭션에 응답
- 연습 3 : 사기 모델 작성 및 평가
 - 작업 1 : Databricks에 AzureML 및 Scikit-Learn 라이브러리 설치
 - 작업 2 : 스코어링 웹 서비스 준비 및 배포
 - 작업 3 : 배치 스코어링 모델 준비
- 연습 4 : 배치 점수 데이터 배포
 - 작업 1 : Cosmos DB를 사용하여 배치 점수 데이터 배포
 - 작업 2 : Azure Databricks 작업을 사용하여 일정에 따라 트랜잭션 점수를 일괄 처리
- 연습 5 : BI 보고서
 - 작업 1 : 사기 추세를 요약하고 시각화하기 위해 Power BI 활용
 - 작업 2 : Azure Databricks에서 대시 보드 만들기
- 연습 후
 - 작업 1 : 리소스 그룹 삭제

Cosmos DB 실시간 고급 분석 연습 단계별 학습목표 및 요약

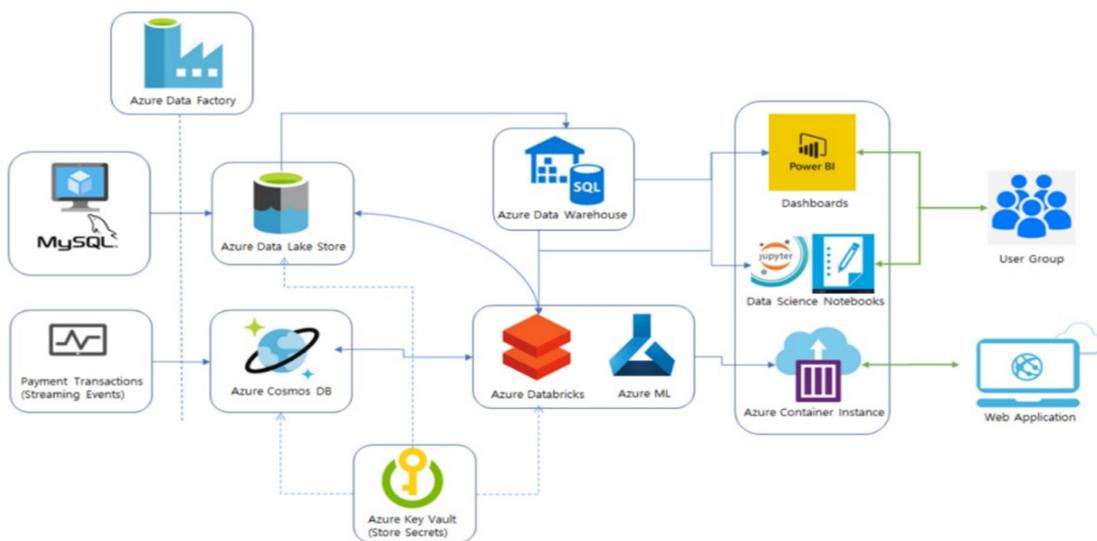
이 실습 랩 세션에서는 Woodgrove 은행의 요구를 지원할 수 있는 데이터 파이프 라인의 PoC 를 구현합니다. 이 워크샵이 끝나면 확장 가능한 머신 러닝, 빅 데이터 및 데이터 처리와 함께 높은 처리량 수집, 낮은 대기 시간 제공 및 글로벌 규모가 필요한 고급 분석 솔루션을 지원하여 Cosmos DB 의 장점을 활용하는 솔루션을 보다 효과적으로 구현할 수 있습니다.

○ 개요

상거래를 위한 지불 처리 서비스를 제공하는 Woodgrove 은행은 혁신적인 사기 탐지 솔루션의 개념 증명 (PoC)을 설계하고 구현하려고 합니다. 이들은 가맹점 고객에게 새로운 서비스를 제공하여 기계 학습 및 고급 분석을 적용하여 사기 거래를 감지함으로써 비용을 절감 할 수 있도록 돋고자 합니다. 고객은 전 세계에 있으며 올바른 솔루션은 가능한 한 많은 솔루션을 고객이 서비스를 사용하는 지역에 최대한 가깝게 배포함으로써 서비스 사용 경험이 지연되는 것을 최소화합니다.

○ 솔루션 아키텍처

아래는 이 실습에서 구축할 솔루션 아키텍처의 다이어그램입니다.



솔루션은 Azure Cosmos DB 에 트랜잭션을 쓰는 결제 트랜잭션 시스템으로 시작합니다. Cosmos DB 의 기본 제공 변경 피드 기능을 사용하여 트랜잭션을 사용하여 Azure Databricks 노트북 내에서 들어오는 데이터 스트림을 읽을 수 있습니다. azure-cosmosdb-spark 커넥터에 연결하고 Azure Data Lake Storage 가 지원하는 Azure Databricks Delta

테이블 내에 장기적으로 저장됩니다. 델타 테이블은 트랜잭션 데이터에 대한 삽입 및 업데이트 (예 : 업 서트)를 효율적으로 관리합니다.

비즈니스 분석가는 Power BI 의 Spark 커넥터를 사용하여 Power BI 의 대시 보드 및 보고서를 사용하여 이 데이터에 대해 Databricks 에서 작성된 테이블에 액세스 할 수 있습니다. 또는 의미 분석 모델을 Azure Analysis Service 에 저장하여 데이터를 Power BI 에 제공 할 수 있으므로 보고를 위해 전용 Databricks 클러스터를 계속 실행할 필요가 없습니다.

데이터 과학자와 엔지니어는 Azure Databricks 노트북을 사용하여 Databricks 테이블에 대한 자체 보고서를 만들 수 있습니다. Azure Databricks 는 Azure Data Lake Storage 에 저장된 기록 데이터를 사용하여 기계 학습 모델의 교육 및 유효성 검사를 지원합니다. 델타 테이블 또는 다른 히스토리 테이블에 저장된 데이터를 사용하여 모델을 주기적으로 재 훈련 할 수 있습니다. Azure Machine Learning 서비스는 학습 모델을 고가용성 Azure Kubernetes Service 클러스터 (AKS 클러스터)에서 실행되는 실시간스코어링 웹서비스로 배포하는데 사용됩니다.

훈련된 모델은 또한 Databricks 작업을 통한 예약된 오프라인 스코어링에 사용되며 "의심스러운 활동" 출력은 Azure Cosmos DB 에 저장되므로 웹 응용 프로그램을 통해 Woodgrove 은행 고객과 가장 가까운 지역에서 전 세계적으로 사용할 수 있습니다.

마지막으로 Azure Key Vault 는 계정 키 및 연결 문자열과 같은 비밀을 안전하게 저장하는데 사용되며 Azure Databricks 비밀 범위의 백업 역할을 합니다.

○ 요구 사항

1. Microsoft Azure 구독 (Microsoft 이외의 구독은 유료 구독이어야 합니다.)
2. Databricks Runtime 5.1 이상을 실행하는 Azure Databricks 클러스터
Azure Data Lake Storage Gen2 와의 Azure Databricks 통합은 **Databricks Runtime 5.1** 에서 완전히 지원됩니다.

- **중요** :이 실습의 OAuth 2.0 액세스 구성 요소를 완료하려면 다음을 수행해야 합니다.
 - Databricks Runtime 5.1 이상을 실행하는 클러스터가 있어야 합니다.
 - Azure 구독 내에서 Azure Active Directory 내에 앱 등록 및 서비스 주체를 만들 수 있는 권한이 있습니다

연습 1 : 스트리밍 트랜잭션 데이터 수집

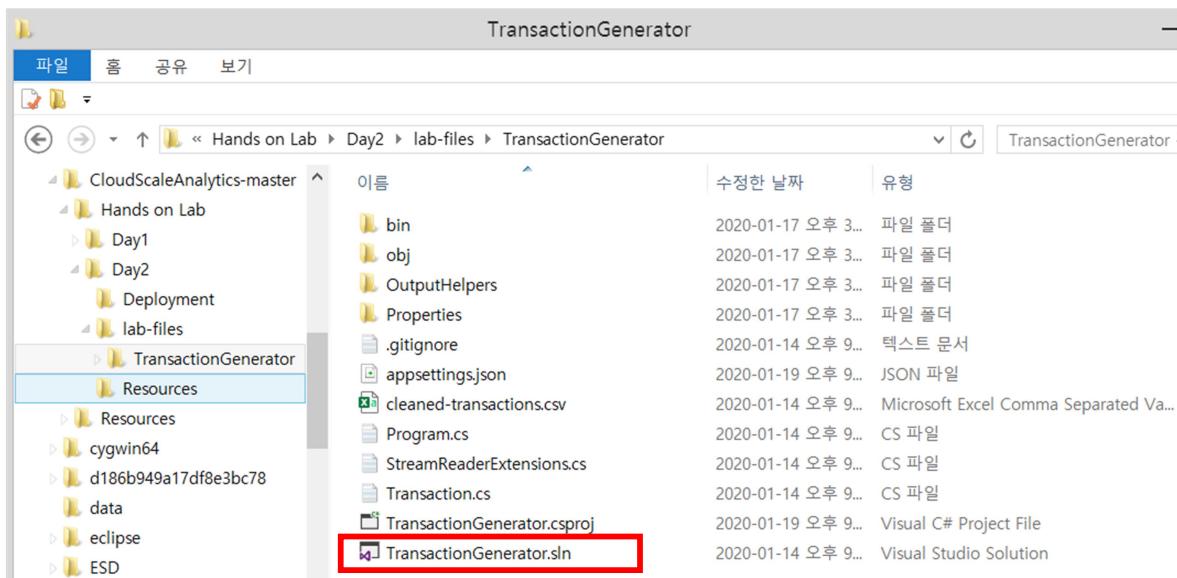
[소요 시간 : 30 분]

이 연습에서는 실시간 스트리밍 온라인 결제를 Azure Cosmos DB에 쓰도록 결제 트랜잭션 생성기를 구성합니다. 결국 생성된 데이터를 처리할 다음 실습을 계속하기 전에 최상의 수집 옵션을 선택하게 됩니다.

태스크 1 : 트랜잭션 생성기 구성

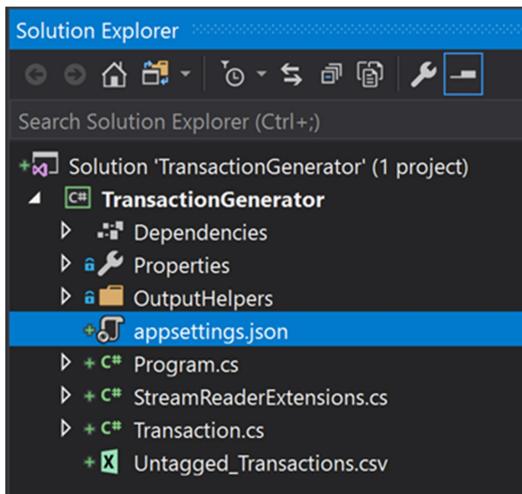
이 태스크에서는 소스 코드에서 연결 정보를 추가하여 지불 트랜잭션 데이터 생성기 프로젝트를 구성합니다.

- 컴퓨터 또는 VM에서 파일 탐색기를 열고 CloudScaleAnalytics-master.zip 파일을 추출한 위치 (C:\CloudScaleAnalytics)로 이동합니다.
 - ⇒ Day1을 실습하신 분들은 이미 해당 경로에 파일이 존재합니다.
 - ⇒ Day2만 실습하시는 분들은 앞선 단계에서 GitHub에서 다운로드 받은 파일의 압축을 해제한 경로입니다.
- C:\CloudScaleAnalytics\Hands on Lab\Day2\lab-files\TransactionGenerator 디렉토리에서 **TransactionGenerator.sln**을 더블클릭하면 Visual Studio에서 솔루션이 열립니다.



- 솔루션 탐색기에서 '**appsettings.json**' 더블클릭합니다. (기존에 파일이 열려 있을 수도 있습니다.) 이 파일에는 콘솔 앱에서 Azure 서비스에 연결하고 응용 프로그램 동작 설정을 구성하는 데 사용되는 설정이 포함되어 있습니다. 콘솔 앱은 이 파일에 저장된

값을 사용하거나 시스템의 환경 변수 내에 사용되도록 프로그래밍되어 있습니다. 이를 통해 실행 파일을 배포하거나 컨테이너화하고 명령 줄을 통해 환경 변수를 전달할 수 있습니다.



`appsettings.json` 파일은 다음 내용이 포함되어 있습니다.

```
{
    "COSMOS_DB_ENDPOINT": " ",
    "COSMOS_DB_AUTH_KEY": "",
    "SECOND_TO_LEAD": "0",
    "SECOND_TO_RUN": "600",
    "ONLY_WRITE_TO_COSMOS_DB": "true"
}
```

`SECONDS_TO_LEAD` 결제 거래 데이터를 보내기 전에 대기하는 시간입니다. 기본값은 0입니다. `SECONDS_TO_RUN` 데이터 전송을 중지하기 전에 생성기가 실행될 수 있는 최대 시간입니다. 기본값은 600 (10 분)입니다. 포함 된 `Untagged_Transactions.csv` 파일의 데이터가 전송 된 후에도 데이터 전송이 중지됩니다.

`ONLY_WRITE_TO_COSMOS_DB` 속성을 `true`로 설정합니다. 기본값은 `false`입니다.

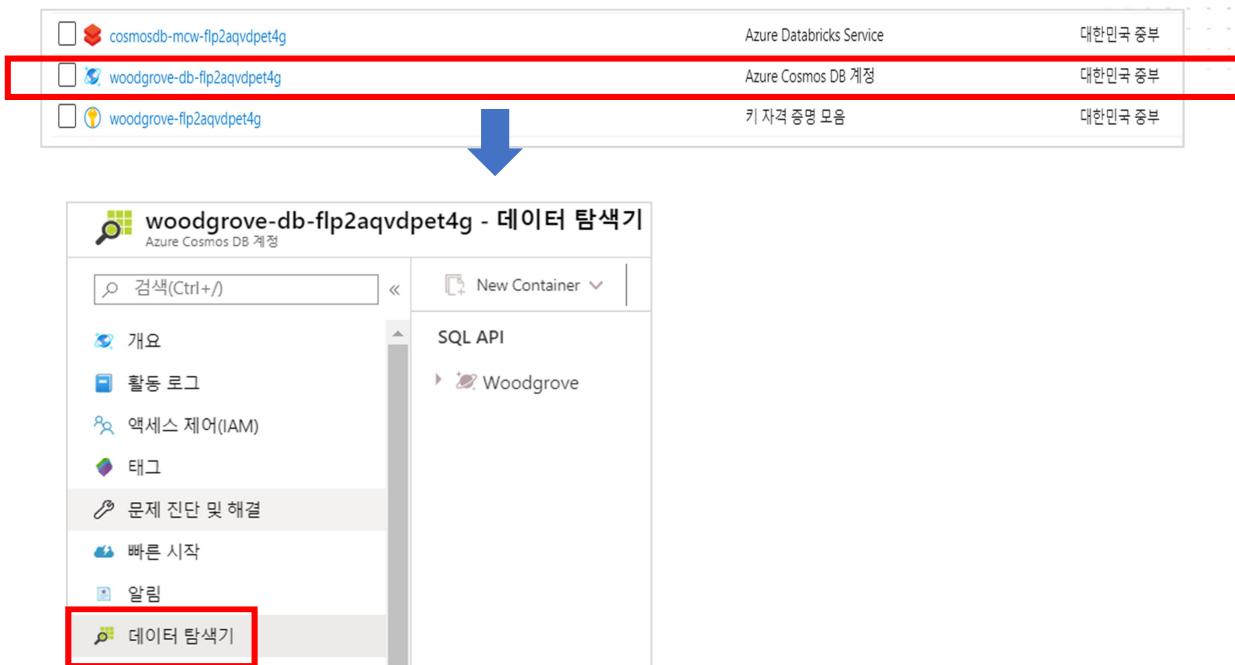
`COSMOS_DB_ENDPOINT`, `COSMOS_DB_AUTH_KEY` 설정값은 **작업 1**에서 확인합니다. 그리고 이 파일을 저장하십시오.

작업 1 : Cosmos DB 로 스트리밍 데이터 수집

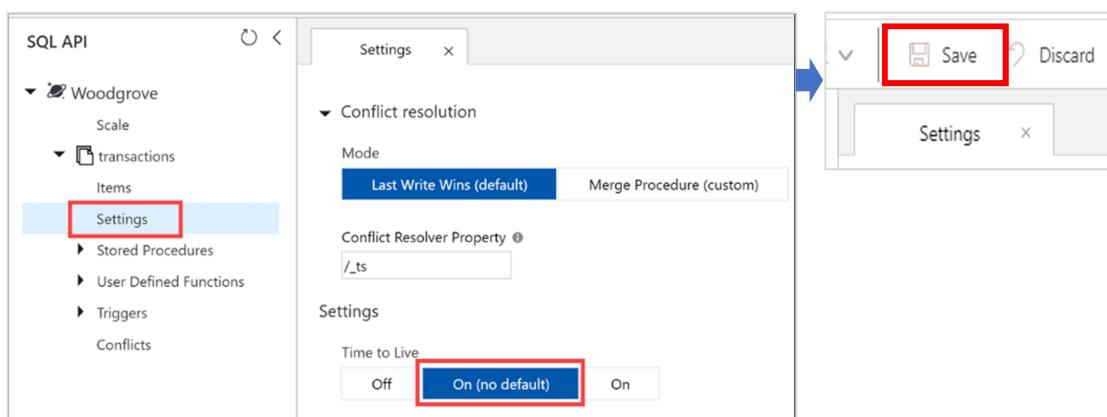
이 작업에서는 Cosmos DB 의 TTL (Time-To-Live) 설정을 기본값 없이 On 으로 구성합니다. 이를 통해 데이터 생성기 TTL 는 개별 메시지가 전송 될 때 TTL 값 (개체 속성)을 설정하여 원하는 시간이 지나면 수집 된 메시지를 만료하거나 삭제할 수 있습니다.

다음으로 Azure Cosmos DB URI 및 키 값을 데이터 생성기로 전달하여 컬렉션에 연결하고 컬렉션에 이벤트를 보낼 수 있습니다.

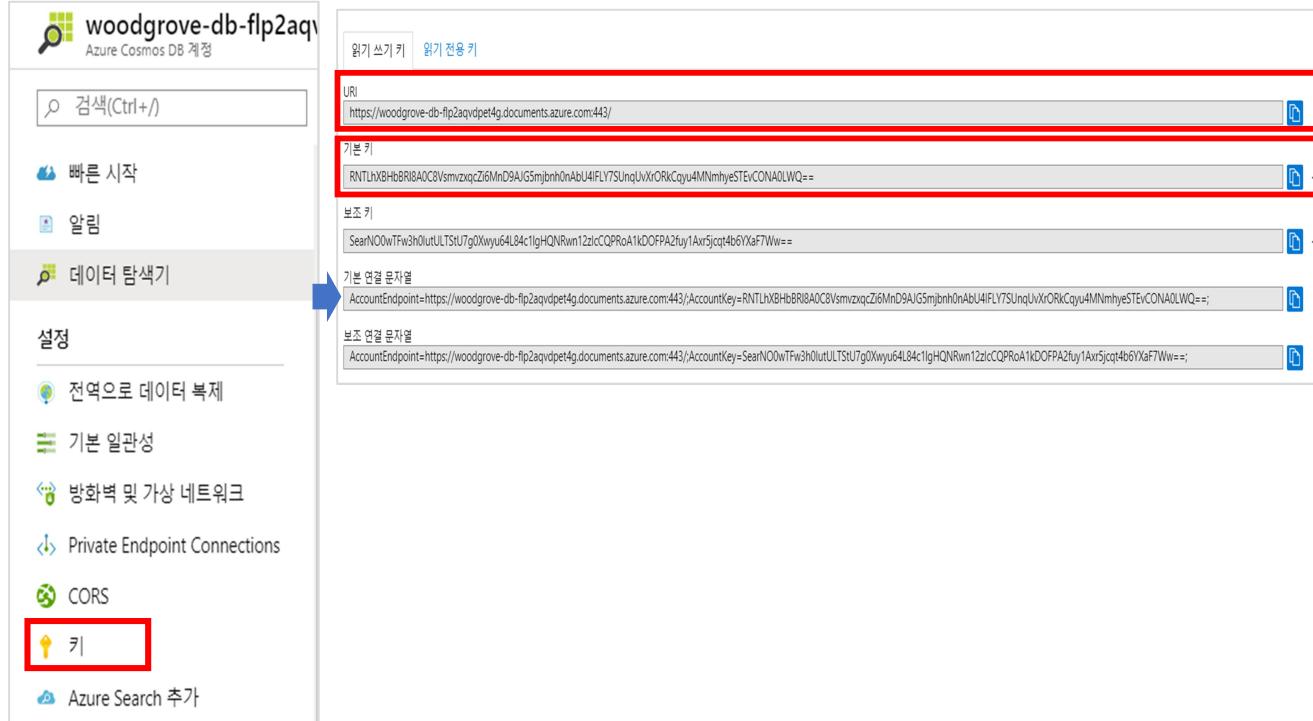
- Azure Portal에서 Azure Cosmos DB 계정으로 이동한 다음 왼쪽 메뉴에서 데이터 탐색기를 선택합니다.



- Woodgrove 데이터베이스와 transactions 컨테이너를 확장한 다음 Settings(설정)을 선택하십시오.
- Settings 내의 Settings에서 TTL (Time to Live)에 대해 On(no default) 옵션을 선택합니다. 컨테이너에 추가된 documents를 자체 TTL 값으로 구성하려면 이 설정이 필요합니다.
- 저장(Save)을 선택하여 설정을 적용하십시오.



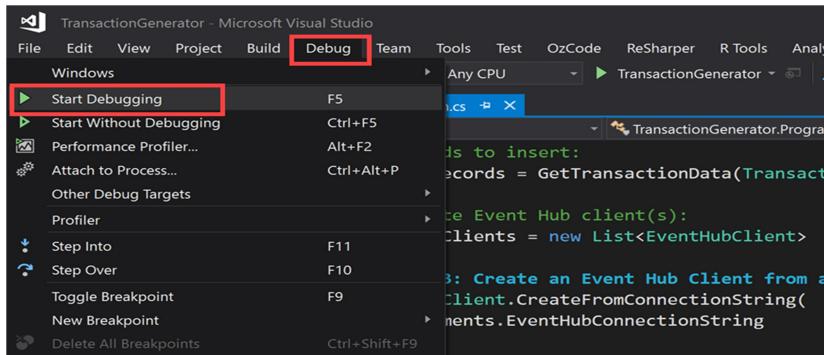
5. Azure Cosmos DB 계정(Account) 의 왼쪽 메뉴에서 키(Keys)를 선택하고 설정 .
6. Cosmos DB 의 엔드 포인트 URI 및 기본 키를 복사하십시오 . 나중에 사용하기 위해 이 값을 메모장 또는 이와 유사한 곳에 저장하십시오.



7. Visual Studio 를 열어 TransactionGenerator 프로젝트로 돌아갑니다.
8. **appsettings.json** 을 열어 **COSMOS_DB_ENDPOINT** 와 **COSMOS_DB_AUTH_KEY** 에 각각 복사한 **URI**, **기본키**를 붙여 넣습니다. 예를 들면 다음과 같습니다. .

```
"COSMOS_DB_ENDPOINT": "https://woodgrove-db.documents.azure.com:443/",  
"COSMOS_DB_AUTH_KEY": "X40BHKQtn0kryyA3LayCqe8TmGok63SzS07eYtn5UCihxPWVY801FdM6"
```

9. 변경 사항을 저장하십시오.
10. Visual Studio 의 최상위 메뉴에서 디버그 를 클릭 한 다음 디버깅 시작 을 클릭하여 콘솔 앱을 실행하거나 키보드에서 F-5 를 누르십시오.



11. PaymentGenerator 콘솔 창이 열리고 몇 초 후에 데이터 전송이 시작되는 것을 볼 수 있습니다. 당신은 창을 누르거나 닫을 수 있습니다 **Ctrl+C** 또는 **Ctrl+Break** 를 이용해 코스모스 DB 에 데이터 전송을 언제든지 중지할 수 있습니다.

```

Payment Generator
Press Ctrl+C or Ctrl+Break to cancel.
Statistics for generated payment data will be updated for every 1000 sent.

Retrieving sample transaction data...
Sample transaction data retrieved. 198999 records found.
Found collection 'transactions' with 15000 RU/s (15000 reads/second; 3000 writes/second @ 1KB doc size)
The collection will cost an estimated $1.25 per hour ($980.00 per month (per write region))
Total requests: requested 1000

Event Hub: inserted 993 docs @ 92.51 writes/s
Event Hub: processing time 10734 ms (slower)
Event Hub: total elapsed time 10.73 seconds (slower)
Event Hub: total succeeded 993
Event Hub: total pending 0
Event Hub: total failed 0

Cosmos DB: inserted 995 docs @ 166.83 writes/s, 7463.79 RU/s (19.35B max monthly 1KB writes)
Cosmos DB: processing time 5964 ms (faster)
Cosmos DB: total elapsed time 5.96 seconds (faster)
Cosmos DB: total succeeded 995
Cosmos DB: total pending 0
Cosmos DB: total failed 0

Total requests: requested 2000

Event Hub: inserted 1002 docs @ 211.13 writes/s
Event Hub: processing time 4746 ms (faster)
Event Hub: total elapsed time 15.48 seconds (slower)
Event Hub: total succeeded 1995
Event Hub: total pending 0
Event Hub: total failed 0

Cosmos DB: inserted 1000 docs @ 201.61 writes/s, 9075.14 RU/s (23.52B max monthly 1KB writes)
Cosmos DB: processing time 4050 ms (slower)
Cosmos DB: total elapsed time 10.92 seconds (faster)
Cosmos DB: total succeeded 1995
Cosmos DB: total pending 0
Cosmos DB: total failed 0

Total requests: requested 3000

Event Hub: inserted 1001 docs @ 239.42 writes/s
Event Hub: processing time 4181 ms (faster)
Event Hub: total elapsed time 19.66 seconds (slower)
Event Hub: total succeeded 2996
Event Hub: total pending 0
Event Hub: total failed 0

Cosmos DB: inserted 1000 docs @ 217.91 writes/s, 9838.46 RU/s (25.50B max monthly 1KB writes)
Cosmos DB: processing time 4589 ms (slower)
Cosmos DB: total elapsed time 15.51 seconds (faster)
Cosmos DB: total succeeded 2995
Cosmos DB: total pending 0

```

출력 맨 위에는 작성된 Cosmos DB 컨테이너 (트랜잭션), 요청 된 RU / s 및 예상 시간별 및 월별 비용에 대한 정보가 표시됩니다. 1,000 개의 레코드를 보내도록 요청한 후 출력 통계가 표시됩니다.

- 삽입 된 줄은 이 배치의 성공적인 삽입 및 RU / s 사용량 및 Cosmos DB 통계에 추가 된 예상 월간 수집 속도로 초당 쓰기 처리량을 보여줍니다.
- 처리 시간 : 지난 1,000 개의 요청 된 삽입에 대한 처리 시간이 다른 서비스보다 빠르거나 느린 지 여부를 표시합니다.
- 총 경과 시간 : 모든 문서를 처리하는 데 걸린 총 시간입니다.

- Cosmos DB 에서 이 값이 계속 높아지면 Cosmos DB 요청이 조절되고 있음을 나타내는 좋은 지표입니다. 컨테이너의 RU 를 늘리십시오.
- 성공은 서비스에 누적 된 성공적인 삽입 수를 표시합니다.
- 보류는 벌크 헤드 대기열에있는 항목입니다. 서비스가 수요를 충족시킬 수없는 경우이 금액은 계속 증가합니다.
- 예외가 발생한 누적 실패한 요청입니다.

12. 실험적으로 Cosmos DB 컨테이너에 대해 요청 된 RU/s 수를 700 으로 줄여보세요.

이렇게하면 Cosmos DB 로의 데이터 전송 속도가 점점 느려질 것입니다. 또한 보류중인 큐가 더 높은 속도로 증가하는 것을 볼 수 있습니다. 그 이유는 쓰기 횟수 (쓰기가 일반적으로 5 RU/s 를 사용하고 1KB 크기의 문서를 읽는 데 1RU/s 만 사용함)가 할당 된 RU/s 를 초과하면 Cosmos DB 가 소비자에게 리소스가 제한되어 있음을 알리기 위해 *retry_after* 헤더 값이 있는 429 응답 . SDK는 지정된 시간 동안 기다렸다가 다시 시도하여 이를 자동으로 처리합니다. 실험이 끝나면 RU 를 15,000 으로 다시 설정하십시오.

연습 2 : 대규모 거래 데이터 이해 및 준비

[소요 시간 : 45 분]

이 연습에서는 Databricks 작업 영역에서 ADLS Gen2 및 Cosmos DB 로의 연결을 작성합니다. 그런 다음 Azure Databricks 를 사용하면 Woodgrove 에서 제공 한 일부 기존 원시 트랜잭션 데이터를 가져 와서 탐색하여 데이터를 사용하여 기계 학습 모델을 구축 및 교육하기 전에 수행해야하는 준비 사항을 더 잘 이해할 수 있습니다. 그런 다음 Databricks 에서 Cosmos DB 에 대한 연결을 사용하여 Cosmos DB 변경 피드에서 직접 스트리밍 트랜잭션을 읽습니다. 마지막으로 들어오는 스트리밍 트랜잭션 데이터를 데이터 레이크에 저장된 Azure Databricks Delta 테이블에 씁니다.

작업 1 : ADLS Gen2 파일 시스템에 대한 OAuth 액세스를 위한 서비스 주체 만들기

Databricks 를 사용하여 ADLS Gen2 파일 시스템에 액세스 할 때 추가 보안 계층으로 OAuth 2.0 을 사용하여 인증 할 수 있습니다. 이 작업에서는 Azure CLI 를 사용하여 서비스 주체라고하는 Azure AD (Azure Active Directory)에서 ID 를 만들어 OAuth 인증을 쉽게 사용할 수 있습니다.

중요 :이 작업을 완료하려면 Azure 구독 내에 App 등록 및 서비스 주체를 만들려면 Azure 구독 내 권한이 있어야합니다.

1. [애저 포털](#) 에서, 상단 도구모음에서 클라우드 웰 아이콘을 선택합니다.



2. Cloud Shell 창에서 **PowerShell** 이 선택되어 있는지 확인하십시오 .

```

PowerShell < ? ? {} > 
Requesting a Cloud Shell...Succeeded.
Connecting terminal...

Welcome to Azure Cloud Shell

Type "az" to use Azure CLI
Type "help" to learn about Cloud Shell

Your Cloud Shell session will be ephemeral so no files or system changes will persist beyond your current session.

MOTD: Discover installed Azure modules: Get-Module Az* -ListAvailable

VERBOSE: Authenticating to Azure ...
VERBOSE: Building your Azure drive ...
Azure:/ PS Azure:> 

```

3. 다음으로 **woodgrove-sp** 라는 서비스 주체를 만들고 **ADLS Gen2 저장소 계정** 의 *Storage Blob Data Contributor* 역할에 할당 하는 명령을 실행 합니다 .
명령 형식은 다음과 같습니다.

```
az ad sp create-for-rbac -n "woodgrove-sp####" --role "Storage Blob Data Contributor"
--scopes {adls-gen2-storage-account-resource-id}
```

중요 : {adls-gen2-storage-account-resource-id} 값을 ADLS Gen2 저장소 계정의 리소스 ID 로 바꿔야 합니다. #####는 개인 고유번호로 수정해야 합니다.

4. 위에서 교체해야하는 ADLS Gen2 저장소 계정 리소스 ID 를 검색하려면 Azure 탐색 메뉴에서 **리소스 그룹** 으로 이동 하고 필터 상자에 "Hands-on-Lab-SUFFIX"를 입력 한 다음 Hands-on-Lab-SUFFIX 를 선택하십시오.
5. Hands-on-Lab 자원 그룹에서 ADLS Gen2 Storage account 를 선택하고 들어가면 왼쪽 메뉴의 **설정(Properties)**을 눌러 오른쪽에 나오는 **저장소 계정 리소스 ID (storage account resource id)** 값을 복사합니다.

aldsgen2store - Properties	
Settings	Storage account resource ID /subscriptions/30fc406c-c745-44f0-be2d-63b1c860cde0/resourceGroups/hands-on-lab/providers/Mi... Copy to clipboard
	Blob service resource ID /subscriptions/30fc406c-c745-44f0-be2d-63b1c860cde0/resourceGroups/hands-on-lab/providers/Mi... Copy
	Primary Blob Service Endpoint https://aldsgen2store.blob.core.windows.net/ Copy
	File service resource ID /subscriptions/30fc406c-c745-44f0-be2d-63b1c860cde0/resourceGroups/hands-on-lab/providers/Mi... Copy
	Primary File Service Endpoint https://aldsgen2store.file.core.windows.net/ Copy
Properties	
Locks	

6. 스토리지 계정 리소스 ID 를 위의 명령에 붙여 넣은 다음 업데이트 된 `az ad sp create-for-rbac` 명령을 복사 하여 Cloud Shell 프롬프트에 붙여 넣고 **Enter** 을 누릅니다 . 이 명령은 구독 ID 및 자원 그룹 이름을 가진 다음과 유사해야합니다.

```
az ad sp create-for-rbac -n "woodgrove-sp####" --role "Storage Blob Data Contributor"
--scope /subscriptions/XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXX/resourceGroups/hands-on-lab
/providers/Microsoft.Storage/storageAccounts/aldsgen2store
```

```
PS Azure:> az ad sp create-for-rbac -n "woodgrove-sp" --role contributor --scopes /subscriptions/
/resourceGroups/hands-on-lab/providers/Microsoft.Storage/storageAccounts/aldsgen2store
Changing "woodgrove-sp" to a valid URI of "http://woodgrove-sp", which is the required format used for service principal names
{
  "appId": "68b7968d-d0e6-4dbf-83f9-3da68cba4719",
  "displayName": "woodgrove-sp",
  "name": "http://woodgrove-sp",
  "password": "2c2d601b-5b44-4ae9-8cb4-8ad90e533658",
  "tenant": "9c37bff0-cd20-XXXX-XXXX-XXXXXXXXXXXX"
}
Azure:>
PS Azure:>
```

7. 다음 단계에서 필요하므로 명령을 실행해서 나온 **출력결과를 텍스트 편집기로 복사하십시오.** 출력은 다음과 유사해야합니다.

```
{
  "appId": "68b7968d-d0e6-4dbf-83f9-3da68cba4719",
  "displayName": "woodgrove-sp####",
  "name": "http://woodgrove-sp####",
  "password": "2c2d601b-5b44-4ae9-8cb4-8ad90e533658",
  "tenant": "9c37bff0-cd20-XXXX-XXXX-XXXXXXXXXXXX"
}
```

8. 역할 할당을 확인하려면 ADLS Gen2 저장소 계정의 왼쪽 메뉴에서 **IAM (액세스 제어)** 을 선택한 다음 **역할 할당** 탭을 선택하고 *Storage Blob Data Contributor* 역할에서 **woodgrove-sp** 를 찾습니다.

NAME	TYPE	ROLE
woodgrove-sp	App	Storage Blob Data Contributor

작업 2 : Azure Key Vault에 서비스 사용자 자격 증명 및 Tenant ID 추가

1. Azure Databricks에서 ADLS Gen2 계정에 대한 액세스를 제공하려면 Azure Key Vault 계정에 저장된 비밀을 사용하여 Databricks 내에서 새로 만든 서비스 주체의 자격 증명을 제공합니다. Azure Portal에서 Azure Key Vault 계정으로 이동한 다음 **액세스 정책**을 선택하고 **+ 새 추가** 버튼을 선택합니다.
2. 현재 포털에 로그인한 계정을 **선택하고 key permissions, secret permissions, certificate permissions 모두 선택**을 선택한 다음 확인을 클릭하고 **저장**을 클릭합니다.
3. 이제 왼쪽 메뉴의 설정에서 **비밀(Secrets)**을 선택하십시오. 오른쪽 상단 도구 모음에서 **+ 생성/가져 오기**를 선택합니다.

woodgrove-vault - Secrets

Key vault

Search (Ctrl+ /) « + Generate/Import ⌂ Refresh ⌄ Restore

Diagnose and solve problems

Settings

Keys

Secrets (highlighted with a red box)

Certificates

Access policies

NAME

There are no secrets available.

4. Secrets(비밀) 작성 화면에서 각각의 입력화면에 다음 표의 값을 입력하십시오.

업로드 옵션	수동(Manual)
이름	Woodgrove-SP-Client-ID
값	이전 단계에서 복사한 Azure CLI 출력의 appId

Create a secret

Upload options: Manual

Name: Woodgrove-SP-Client-ID

Value: (red box)

Content type (optional):

Set activation date?

Set expiration date?

Enabled? Yes No

Create

5. 작성을 선택 하십시오

위와 같은 방법으로 Secrets 놀러 + 생성/가져 오기 놀러 새로 작성되는 입력 화면에 아래의 표에 있는 이름(Name)과 값(Value)을 입력하고 생성(create)을 합니다.

업로드옵션	이름	값 (전 단계에서 Azure CLI 출력값 붙여넣기)
수동	Woodgrove-SP-Client-Key	Password : (Azure CLI 의 출력값)
수동	Azure-Tenant-ID	Tenant : (Azure CLI 의 출력값)

작업 3 : Azure Key Vault에서 ADLS Gen2 저장소 계정 구성

이 작업에서는 Key Vault 내에서 ADLS Gen2 스토리지 계정의 키를 구성합니다.

1. Azure Portal에서 ADLS Gen2 저장소 계정으로 이동한 다음 왼쪽 메뉴의 설정에서 액세스 키를 선택합니다. 저장소 계정 이름과 키 값을 복사하여 Key Vault 계정에 Secrets(비밀)에 추가합니다.

The screenshot shows the 'adlsgen2storage - Access keys' page. On the left, there's a sidebar with 'Overview', 'Activity log', 'Access control (IAM)', 'Tags', 'Diagnose and solve problems', 'Settings' (which is selected), and 'CORS'. Under 'Settings', the 'Access keys' tab is highlighted with a red box. The main area shows a key named 'key1' with its value: 'CLOEQtpZ5nib6icmYGcQtU9QwKrij4rKeeJGWcl1mM6myw+AR6paA9kYtsB2u0DNgHkulvlu...'. A red box highlights the key value field.

2. 새 브라우저 탭 또는 창을 열고 Azure Portal에서 Azure Key Vault 계정으로 이동한 다음 왼쪽 메뉴의 설정에서 비밀을 선택합니다. 비밀 블레이드의 상단 도구 모음에서 + 생성 / 가져 오기 를 선택합니다.

The screenshot shows the 'woodgrove-vault - Secrets' page. On the left, there's a sidebar with 'Diagnose and solve problems', 'Settings' (which is selected), and 'Keys', 'Certificates', 'Access policies'. Under 'Settings', the 'Secrets' tab is highlighted with a red box. At the top right, there are three buttons: '+ Generate/Import' (highlighted with a red box), 'Refresh', and 'Restore'. Below the buttons, it says 'NAME' and 'T'. At the bottom, it says 'There are no secrets available.'

이번 단계도 작업 2 와 같이 Key Vault Secrets 에 아래의 표에 있는 이름(Name)과 값(Value)을 입력하고 생성(create)을 합니다.

업로드옵션	이름	값 (전 단계에서 복사한 저장소 계정 (storage account) 과 키(Key1) 값을 붙여넣기)
수동	ADLS-Gen2-Account-Name	Storage account name 값 붙여넣기
수동	ADLS-Gen2-Account-Key	storage account key 값 붙여넣기

작업 4 : Key Vault에서 Cosmos DB 키 구성

- 새 브라우저 탭 또는 창을 열고 Azure Portal에서 Azure Key Vault 계정으로 이동한 다음 왼쪽 메뉴의 설정에서 **비밀** 을 선택합니다. 비밀 블레이드의 상단 도구 모음에서 **+ 생성 / 가져 오기** 를 선택합니다.

이번 단계도 작업 3 와 같이 Key Vault Secrets 에 아래의 표에 있는 이름(Name)과 값(Value)을 입력하고 생성(create)을 합니다.

업로드옵션	이름	값 (앞에서 편집기에 복사한 값 붙여넣기)
수동	Cosmos-DB-URI	이전 단계에서 복사 한 Azure Cosmos DB URI 값
수동	Cosmos-DB-Key	이전 단계에서 복사 한 Azure Cosmos DB Primary Key 값

▶ [Key Vault Secrets] 전체 입력 정보 확인하기

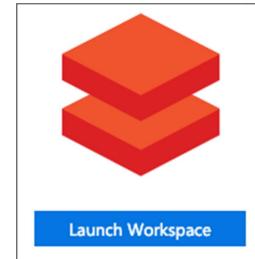
[작업 2] 부터 [작업 4] 단계에서 입력한 Key vault secrets 의 전체 정보가 정상적으로 생성이 되었는지 확인을 합니다.

이름	형식	상태	만료 날짜
ADLS-Gen2-Account-Key		✓ 사용	
ADLS-Gen2-Account-Name		✓ 사용	
Azure-Tenant-ID		✓ 사용	
Cosmos-DB-Key		✓ 사용	
Cosmos-DB-URI		✓ 사용	
Woodgrove-SP-Client-ID		✓ 사용	
Woodgrove-SP-Client-Key		✓ 사용	

작업 5 : Azure Databricks 클러스터 만들기

이 작업에서는 Azure Databricks 작업 영역에 연결하고 이 실습에 사용할 클러스터를 만듭니다.

1. [Azure Portal](#)로 돌아가서 위에서 프로비저닝 한 Azure Databricks 작업 영역으로 이동하고 개요 블레이드에서 **작업 영역 시작을 선택**하고 필요한 경우 Azure 자격 증명으로 작업 영역에 로그인합니다.



2. 선택 **클러스터** 왼쪽 탐색 메뉴에서를 누른 다음 선택 + **클러스터를 만듭니다**.

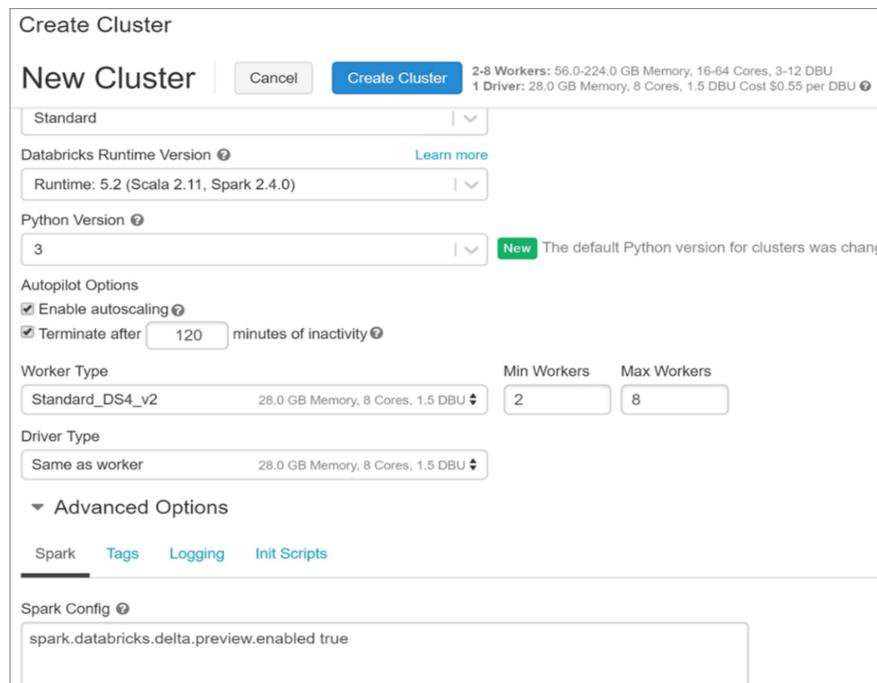
A screenshot of the Microsoft Azure Clusters blade. On the left is a vertical navigation menu with icons for Azure Databricks, Home, Workspace, Recents, Data, Clusters (which is highlighted with a red border), and Jobs. The main area is titled "Clusters" and contains a "Create Cluster" button with a plus sign, which is also highlighted with a red border. Below this are sections for "Interactive Clusters" and "Job Clusters", each showing an empty list.

3. 클러스터 생성 화면에서 다음을 입력하십시오.

- **클러스터 이름** : lab-cluster와 같은 클러스터 이름을 입력하십시오.
- **클러스터 모드** : 표준을 선택합니다.
- **Databricks 런타임 버전** : 런타임 : 5.5LTS (Scala 2.11, Spark 2.4.3)를 선택하십시오.
- **파이썬 버전** : 선택 3.
- **자동 크기 조정 사용** : 이것이 선택되어 있는지 확인하십시오.

- **XX 분 동안 활동이 없으면 종료** : 이 확인란을 선택된 상태로 두고 분 수를 120 으로 설정하십시오.
- **작업자 유형** : Standard_DS3_v2 를 선택하십시오.
 - **최소 Worker** : 1 로 설정합니다.
 - **최대 Worker** : 2 로 설정하십시오.
- **Driver 유형** : Worker 와 동일(Same as worker)하게 설정합니다.
- Advanced Options 을 펼치고 Spark 구성 상자에 다음을 입력하십시오.

```
spark.databricks.delta.preview.enabled true
```

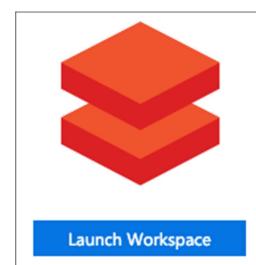


4. 클러스터 작성을 선택 하십시오 . 클러스터를 생성하고 시작하는 데 3-5 분이 걸립니다.

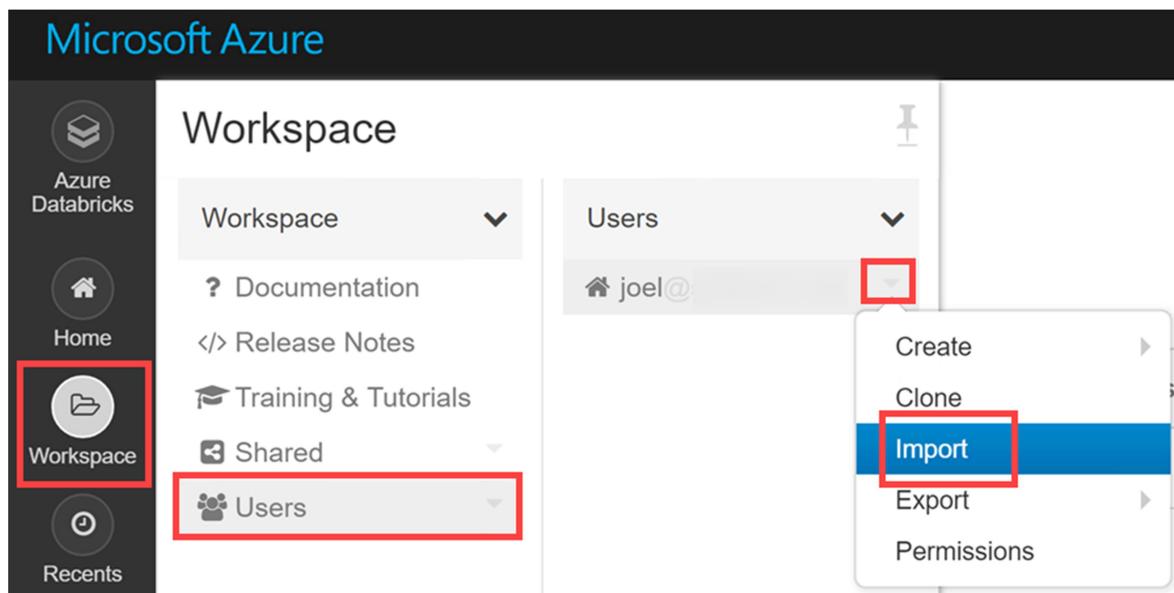
작업 6 : Azure Databricks 열기 및 랩 노트북

이 작업에서는 [Cosmos DB 실시간 고급 분석 GitHub](#) 리포지토리에 포함된 노트북을 Azure Databricks 작업 영역으로 가져옵니다.

1. Azure Portal에서 Azure Databricks 작업 영역으로 이동하고
개요 블레이드에서 **작업 영역 시작**을 선택하고 필요한 경우
Azure 자격 증명으로 작업 영역에 로그인합니다.



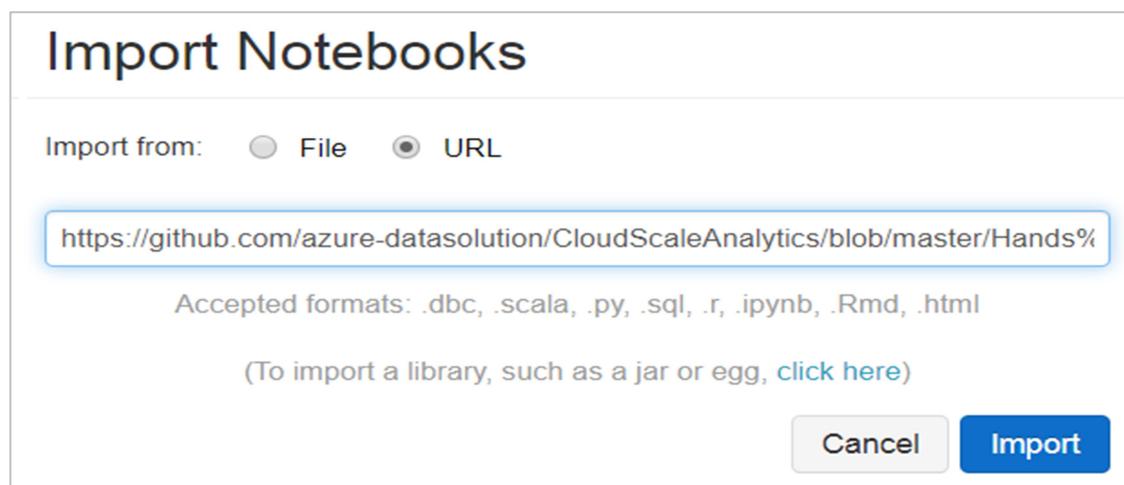
2. 왼쪽 메뉴에서 **작업 영역(Workspace)**을 선택한 다음 **사용자/Users** 및 사용자 계정(이메일 주소)를 선택한 다음 사용자 작업 공간의 상단에 있는 아래쪽 화살표를 선택하고 **Import(가져오기)**를 선택합니다.



3. Notebooks Import 대화 상자에서 가져올 대상 URL 을 선택한 다음 다음을 붙여 넣습니다.

<https://github.com/azure-datasolution/CloudScaleAnalytics/tree/master/Hands%20on%20Lab/Day2/lab-files/ScaleAnalyticsdbc>

※ Github URL로 연결해서 dbc 파일을 받을 수도 있지만 File을 선택해서 다운받은 Github 파일 (ex. C:\CloudScaleAnalytics-master\Hands on Lab\Day2\lab-files\ScaleAnalyticsdbc)을 불러올 수 있습니다.



4. Import(가져 오기)를 선택하십시오 .
5. 이제 사용자 작업 공간에 **ScaleAnalytics**라는 폴더가 나타납니다 . 이 폴더에는이 실습에서 사용할 모든 노트북이 들어 있습니다.

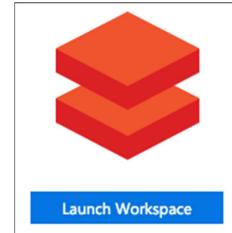
작업 7 : Azure Databricks 주요 자격 증명 지원 비밀 구성

이 작업에서는 Azure Databricks 작업 영역에 연결하고 Azure Key Vault 계정을 백업 저장소로 사용하도록 Azure Databricks 비밀을 구성합니다.

1. [Azure Portal](#)로 돌아가서 새로 프로비저닝 된 Key Vault 계정으로 이동 한 다음 왼쪽 메뉴에서 **속성** 을 선택 합니다.
2. **DNS 이름 및 Resource ID(자원 ID)** 값을 메모장에 복사하여 붙여 넣습니다. 이 값들은 다음 섹션에서 사용됩니다.

The screenshot shows the 'woodgrove-vault - Properties' blade in the Azure portal. On the left, there's a sidebar with 'Properties' selected. The main area displays the Key Vault settings. Two specific fields are highlighted with a red box: 'DNS NAME' containing 'https://woodgrove-vault.vault.azure.net/' and 'RESOURCE ID' containing '/subscriptions/.../resourcegroups/hands-on-lab/pr...'. These are the values to be copied for the next step.

3. 위에서 프로비저닝 한 Azure Databricks 작업 영역으로 이동 하고 개요 블레이드에서 **작업 영역 시작**을 선택 하고 필요한 경우 Azure 자격 증명으로 작업 영역에 로그인합니다.



4. 브라우저의 URL 표시 줄에서 # secrets / createScope 를 Azure Databricks 기본 URL에 추가하십시오.
(예 : <https://koreacentral.azuredatabricks.net#secrets/createScope>)
5. Scope Name 에 **key-vault-secrets** 입력하십시오 .
6. Manage Principal 에서 **Creator** 선택 지정합니다.

관리 권한을 통해 사용자는이 비밀 범위를 읽고 쓸 수 있으며 Azure Databricks Premium Plan 의 계정 인 경우 범위에 대한 권한을 변경할 수 있습니다.

Creator 를 선택할 수 있으려면 계정에 Azure Databricks Premium Plan 이 있어야합니다. 이 방법은 권장되는 방법입니다. Secret Scope(비밀 범위)를 만들 때 생성자에게 MANAGE 권한을 부여한 다음 범위를 테스트 한 후 더 세분화 된 액세스 권한을 할당하십시오.

7. Key Vault 생성 단계에서 이전에 복사 한 DNS 이름과 Resource ID 를 입력하십시오.

DNS Name	https://woodgrove-vault.vault.azure.net/
Resource ID	/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx/resourcegroups/hands-on-lab/providers/Microsoft.KeyVault/vaults/woodgrove-vault

8. 작성을 선택 하십시오 .

HomePage / Create Secret Scope

Create Secret Scope

A store for secrets that is identified by a name and backed by a specific store type. [Learn more](#)

Scope Name ?

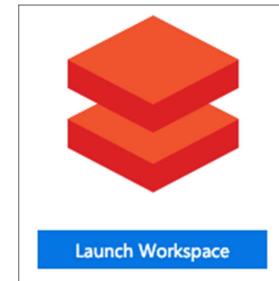
Manage Principal ?

잠시 후 비밀 범위가 만들어 졌음을 확인하는 대화 상자가 나타납니다.

작업 8 : Databricks 에 Azure Cosmos DB Spark 커넥터 및 scikit-learn 라이브러리 설치

이 작업에서는 [Azure Cosmos DB Spark 커넥터](#) 및 scikit-learn 라이브러리를 Databricks 클러스터에 설치합니다. Cosmos DB 커넥터를 사용하면 Apache Spark DataFrames 를 통해 Azure Cosmos DB 에서 쉽게 읽고 쓸 수 있습니다.

1. [Azure Portal](#) 에서 Azure Databricks 작업 영역으로 이동하고
개요 블레이드에서 **작업 영역 시작**을 선택하고 필요한 경우
Azure 자격 증명으로 작업 영역에 로그인합니다.



2. **Workspace** 를 선택, 왼쪽 메뉴에서 **shared(공유)** 옆에 있는 drop-down 선택하고
create(만들기) 선택하고 다음은 **Library** 선택합니다..

The screenshot shows the Microsoft Azure Databricks workspace interface. On the left sidebar, the 'Workspace' icon is highlighted with a red box. In the main workspace area, there is a dropdown menu under 'Shared' that is also highlighted with a red box. A sub-menu is open, showing options: 'Create' (highlighted with a red box), 'Clone', 'Import', and 'Export'. The 'Library' option is selected and highlighted with a blue box. The background shows other workspace details like documentation, release notes, training & tutorials, and users.

3. 라이브러리 생성 페이지의 라이브러리 소스에서 **Maven** 을 선택한 다음 좌표 텍스트 상자
옆의 **패키지 검색** 을 선택하십시오 .

Create Library

Library Source

- [Upload](#)
- [DBFS](#)
- [PyPI](#)
- [Maven](#)
- [CRAN](#)

Repository ?

Optional

Coordinates

Maven Coordinates (com.databricks:spark-csv_2.10:1.0.0)

[Search Packages](#)

Exclusions

Dependencies to exclude (log4j:log4j,junit:junit)

[Create](#) [Cancel](#)

4. 검색 패키지 대화 상자 의 소스 드롭 다운에서 **Maven Central** 을 선택 하고 검색 상자에 **azure-cosmosdb-spark** 를 입력 한 다음 **azure-cosmosdb-spark_2.4.0_2.11 release 1.5.0 선택을 클릭 합니다.**

Search Packages

Maven Central	Q azure-cosmosdb-spark		
Group Id	Artifact Id	Releases	Options
com.microsoft.azure	azure-cosmosdb-spark_2.4.0_2.11	1.3.5	Select
com.microsoft.azure	azure-cosmosdb-spark_2.1.0_2.11	1.2.6	Select
com.microsoft.azure	azure-cosmosdb-spark_2.2.0_2.11	1.1.1	Select
com.microsoft.azure	azure-cosmosdb-spark_2.2.0_2.10	1.0.0	Select

5. 라이브러리 설치를 마치 려면 **작성을 선택 하십시오 .**

Create Library

Library Source

- [Upload](#)
- [DBFS](#)
- [PyPI](#)
- [Maven](#)
- [CRAN](#)

Repository ?

Optional

Coordinates

com.microsoft.azure:azure-cosmosdb-spark_2.4.0_2.11:1.3.5

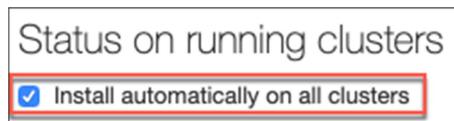
[Search Packages](#)

Exclusions

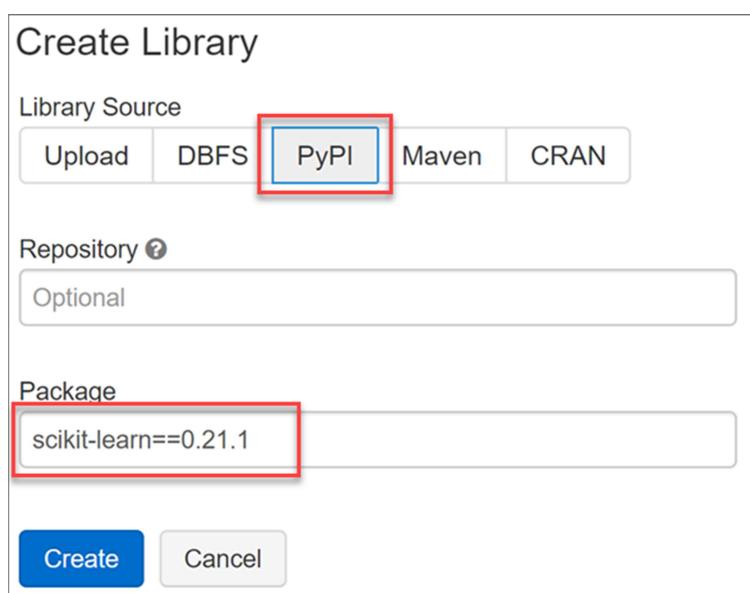
Dependencies to exclude (log4j:log4j,junit:junit)

[Create](#) [Cancel](#)

6. 다음 화면에서 모든 클러스터에 자동 설치 상자를 선택하고 프롬프트가 표시되면 확인을 선택하십시오.



7. 작업 공간에서 공유 폴더를 다시 선택하고 컨텍스트 메뉴에서 작성 및 라이브러리를 선택하십시오.
8. 라이브러리 작성 대화 상자에서 PyPI 를 라이브러리 소스로 선택하고 패키지 상자에 scikit-learn == 0.21.1 을 입력 한 후 작성 을 선택 하십시오.

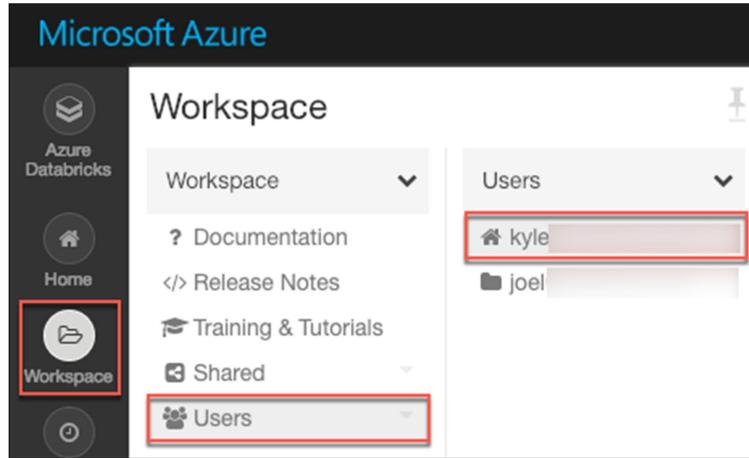


9. 다음 화면에서 모든 클러스터에 자동 설치 확인란을 선택하지 말고 프롬프트가 표시되면 확인을 선택하십시오. 이 라이브러리는 작업 클러스터에 대한 참조로만 필요합니다. 이 scikit-learn 을 Exercise 3 의 실습 클러스터에 직접 추가합니다.

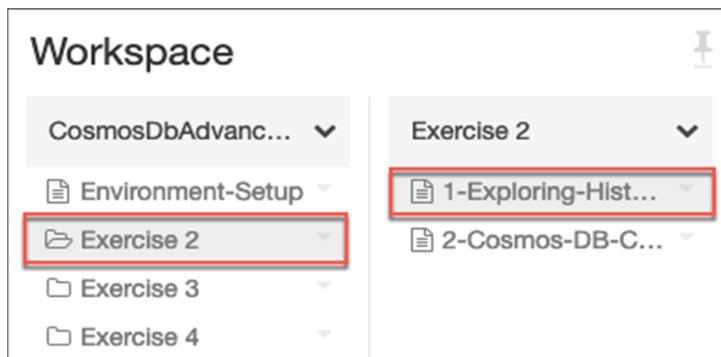
작업 9 : Azure Databricks 및 Spark 를 사용하여 기록 트랜잭션 데이터 탐색

이 작업에서는 Azure Databricks 노트북을 사용하여 Transaction 이력 데이터를 다운로드하고 탐색합니다.

1. 당신의 Databricks 작업 공간에서, 선택 작업을 한 후, 왼쪽 메뉴에서 선택 사용자 및 사용자 계정을 선택하세요



2. 사용자 작업 공간에서 **ScaleAnalytics** 폴더를 선택한 후 **Exercise2** 폴더를 선택하고 이름이 **1-Exploring-Historical-Transactions** 노트북을 선택하십시오



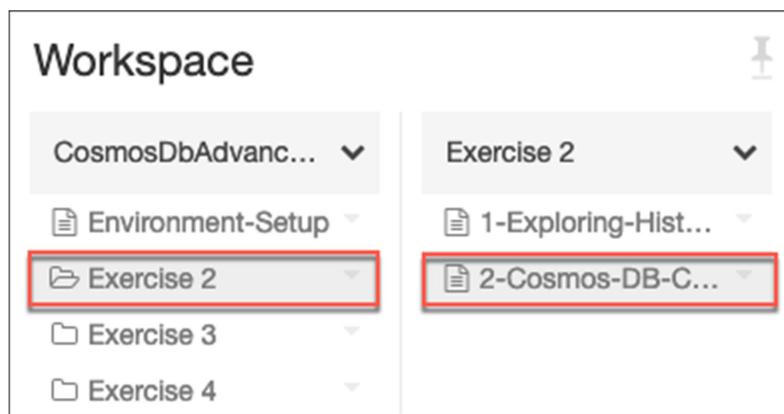
3. [Exercise2] **1-Exploring-Historical-Transactions** 노트북이 작업의 나머지 단계를 완료하기 위한 지시 사항을 따르십시오.

참고 :이 연습에서는 각 노트북의 맨 아래에 다음 작업을 위해 노트북으로 이동하는 링크가 있으므로이 연습을 위해 문서와 Databricks 노트북간에 앞뒤로 이동할 필요가 없습니다.

작업 10 : Azure Databricks에서 Cosmos DB 변경 피드 및 Spark 구조적 스트리밍을 사용하여 스트리밍 트랜잭션에 응답

이 작업에서는 Azure Databricks 노트북을 사용하여 Azure Databricks 노트북에서 Cosmos DB 인스턴스에 대한 연결을 만들고 Cosmos DB 변경 피드에서 스트리밍 데이터를 쿼리합니다.

- 당신의 Databricks 작업 공간에서, 선택 **작업**을 한 후, 왼쪽 메뉴에서 선택 **사용자 및 사용자 계정**을 선택하세요.
- 사용자 작업 공간에서 **ScaleAnalytics** 폴더를 선택한 후 Exercise 2 폴더를 선택하고 이름이 **2-Cosmos-DB-Change-Feed** 노트북을 선택하십시오 .



- 2-Cosmos-DB-Change-Feed 노트북에서 ,이 작업의 나머지 단계를 완료하는 지침을 따르십시오.

연습 3 : 사기 모델 작성 및 평가

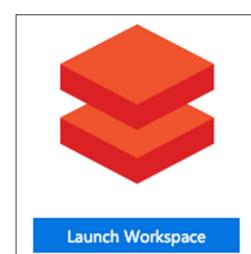
[소요 시간 : 45 분]

이 연습에서는 웹 프런트 엔드에서 발생하는 트랜잭션의 실시간 스코어링에 사용되는 사기 모델을 만들고 평가합니다. 목표는 사기 거래가 처리되기 전에 차단하는 것입니다. 그런 다음 실습 4에서 발생하는 일괄 처리 중에 실행되는 의심스러운 트랜잭션을 탐지하기 위한 모델을 만듭니다. 마지막으로, 사기성 트랜잭션 모델을 배포하고 Databricks 노트북 내에서 HTTP REST 호출을 통해 테스트합니다.

작업 1 : Databricks에 AzureML 및 Scikit-Learn 라이브러리 설치

이 작업에서는 Databricks 클러스터에 필수 **AzureML** 및 **Scikit-Learn** 라이브러리를 설치합니다. 이 라이브러리는 기계 학습 모델을 교육하고 배포 할 때 사용됩니다. 표시된 순서대로 설치하는 것이 중요합니다.

- [Azure Portal](#) 에서 Azure Databricks 작업 영역으로 이동 하고 개요 블레이드에서 **작업 영역 시작**을 선택하고 필요한 경우 Azure 자격 증명으로 작업 영역에 로그인합니다.



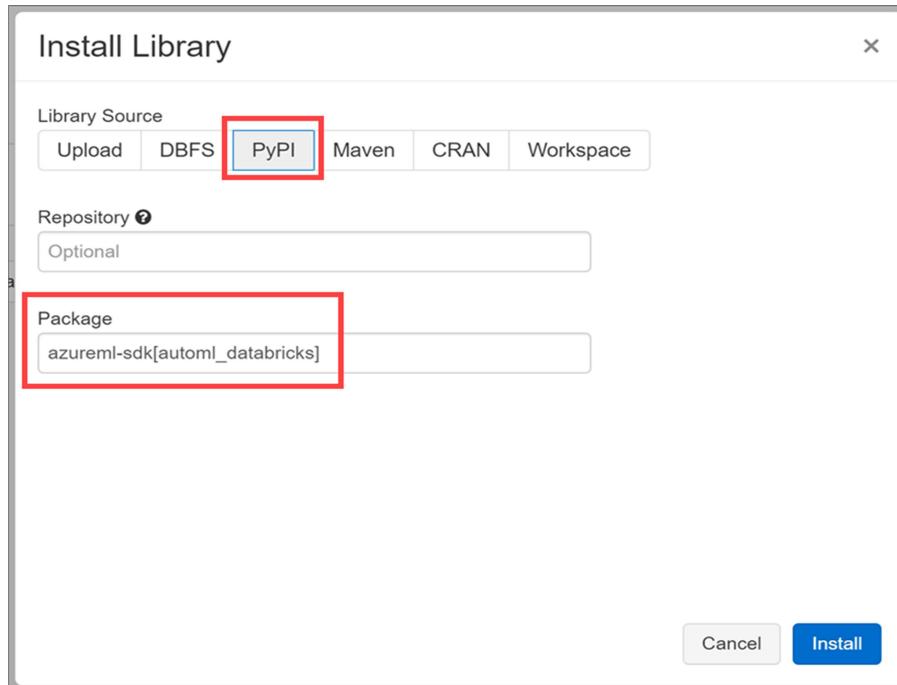
2. 왼쪽 목록에서 클러스터를 선택합니다.

The screenshot shows the Microsoft Azure Clusters page. On the left, there's a sidebar with icons for Azure Databricks, Home, Workspace, Recents, and Data. The 'Clusters' icon is highlighted with a red box. The main area is titled 'Clusters' and shows a table for 'Interactive Clusters'. A blue button at the top right says '+ Create Cluster'. Under 'Interactive Clusters', there's a table with columns 'Name', 'State', and 'Nodes'. One row is selected, showing 'lab-cluster' with 'Running' status and '3' nodes. A red box highlights the 'lab-cluster' entry. Below it, there's a section for 'Job Clusters' which is currently empty.

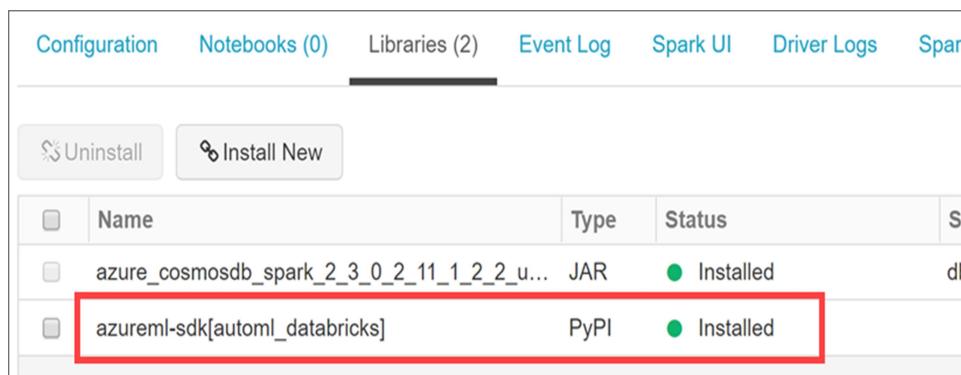
3. 라이브러리에 설치된 라이브러리 목록을 표시 하는 라이브러리 탭을 선택하십시오 . Azure Cosmos DB Spark 커넥터가 설치되어 있어야합니다. 라이브러리 목록 위에서 새로 설치를 선택하십시오 .

The screenshot shows the 'Clusters / lab-cluster' details page. At the top, there's a cluster icon labeled 'lab-cluster' with a green dot, followed by three buttons: 'Edit', 'Clone', and 'Restart'. Below that, there are tabs: 'Configuration', 'Notebooks (0)', 'Libraries (1)', 'Event Log', and 'Spark UI'. The 'Libraries (1)' tab is selected and highlighted with a red box. At the bottom, there's a table with columns 'Name', 'Type', and 'Status'. A button 'Install New Library' is visible above the table, also highlighted with a red box. The table shows one entry: 'azure_cosmosdb_spark_2_3_0_2_11_1_2_2_u...' with 'JAR' type and 'Insta' status.

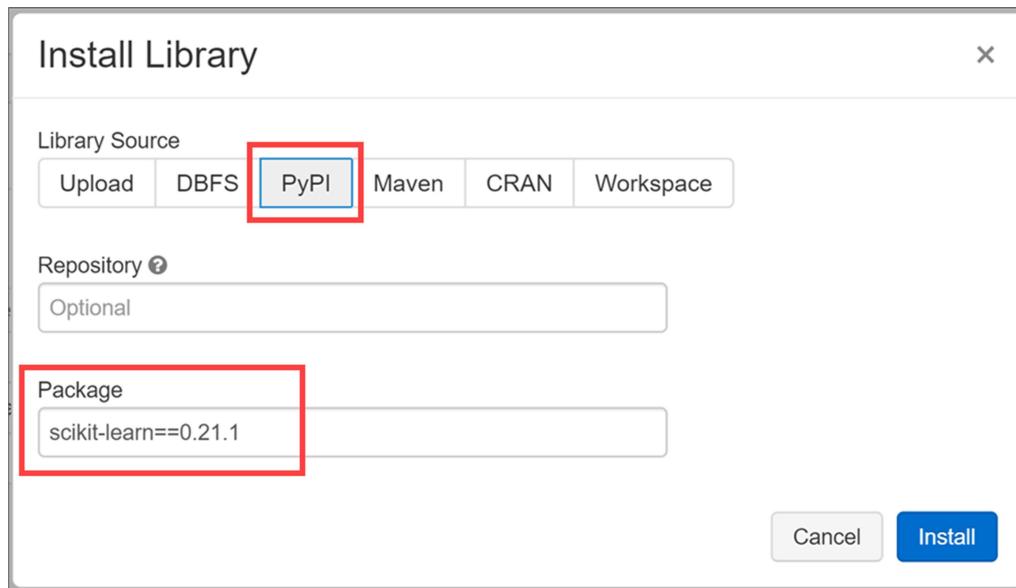
4. 대화 상자가 나타나면 라이브러리 소스로 PyPI를 선택하십시오. 입력에서 패키지를 선택한 다음 **azureml-sdk[automl_databricks]** 설치합니다.



5. 중요 : AzureML 라이브러리의 상태가 **Installed**로 표시될 때까지 기다리십시오. Scikit-Learn을 설치하기 전에 완료해야합니다.



6. 새로 설치를 다시 선택 하십시오.
7. 대화 상자가 나타나면 라이브러리 소스에서 PyPI를 선택하십시오. 입력에서 패키지를 선택한 다음 **scikit-learn==0.21.1**를 설치합니다.



- Scikit-Learn 라이브러리가 설치되면 라이브러리 목록은 다음과 같아야합니다.

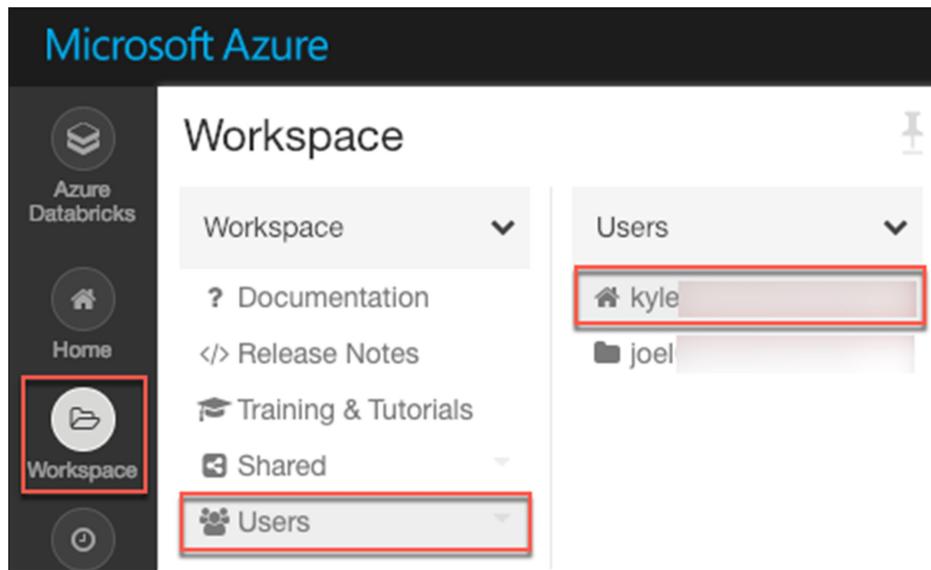
The screenshot shows the Databricks Clusters interface for the 'lab-cluster'. At the top, there's a green circle icon for the cluster, the name 'lab-cluster', and action buttons: Edit, Clone, Restart, and Terminate. Below that, tabs are shown: Configuration, Notebooks (1), Libraries (which is selected and highlighted in blue), Event Log, Spark UI, Driver Logs, and SQL. Under the Libraries tab, there are two buttons: 'Uninstall' and 'Install New'. A table lists the installed libraries:

	Name	Type	Status
<input type="checkbox"/>	azureml-sdk[automl_databricks]	PyPI	Installed
<input type="checkbox"/>	com.microsoft.azure:azure-cosmosdb-spark_2....	Maven	Installed
<input type="checkbox"/>	scikit-learn==0.21.1	PyPI	Installed

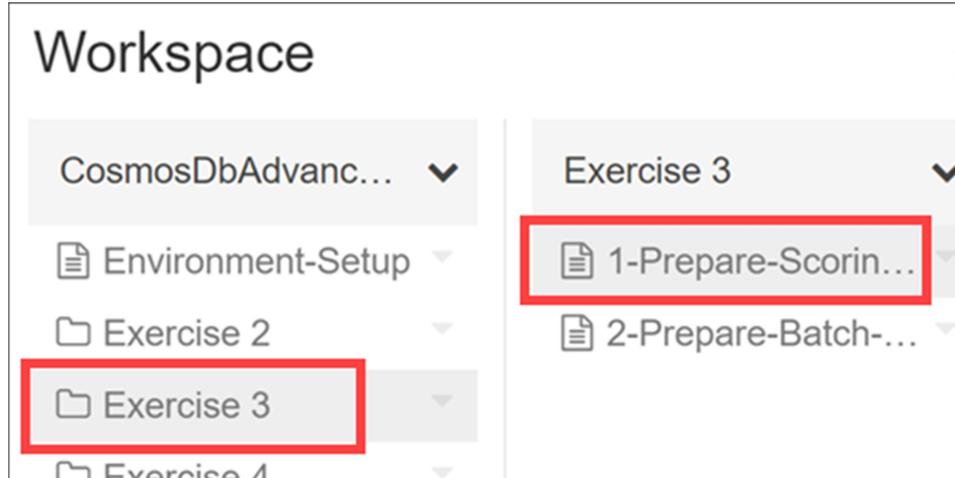
작업 2 : 스코어링 웹 서비스 준비 및 배포

이 작업에서는 Azure Databricks 노트북을 사용하여 트랜잭션 및 계정 데이터를 탐색합니다. 또한 데이터 정리를 수행하고 데이터가 스코어링을 위해 모델로 전달 될 때마다 이러한 변환을 적용하는 기능 엔지니어링 파이프 라인을 작성합니다. 마지막으로 사기 거래를 탐지하는 기계 학습 모델을 교육하고 배포합니다.

- 당신의 Databricks 작업 공간에서, 왼쪽 메뉴에서 **Workspace** 를 선택 한 후, 사용자 선택 및 사용자 계정을 선택합니다.



- 사용자 작업 공간에서 **ScaleAnalytics** 폴더를 선택한 후 Exercise 3 폴더를 선택하고 **1-Prepare-Scoring-Web-Service** 라는 노트북을 선택하십시오 .



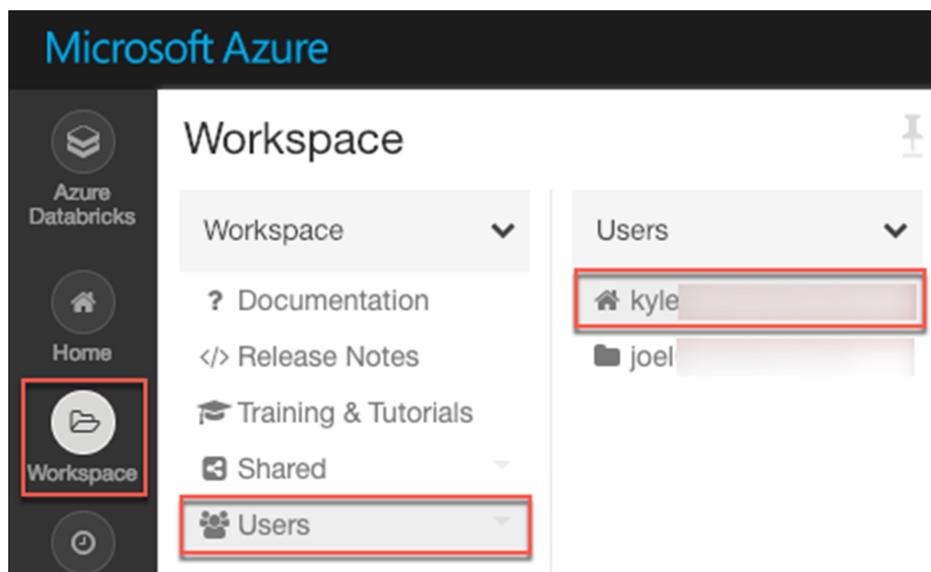
- 1-Prepare-Scoring-Web-Service** 노트북 에서 ,이 작업의 나머지 단계를 완료하기위한 지시 사항을 따르십시오.

참고 : 이 연습에서는 각 노트북의 맨 아래에 다음 작업을 위해 노트북으로 이동하는 링크가 있으므로이 연습을 위해이 문서와 Databricks 노트북간에 앞뒤로 이동할 필요가 없습니다.

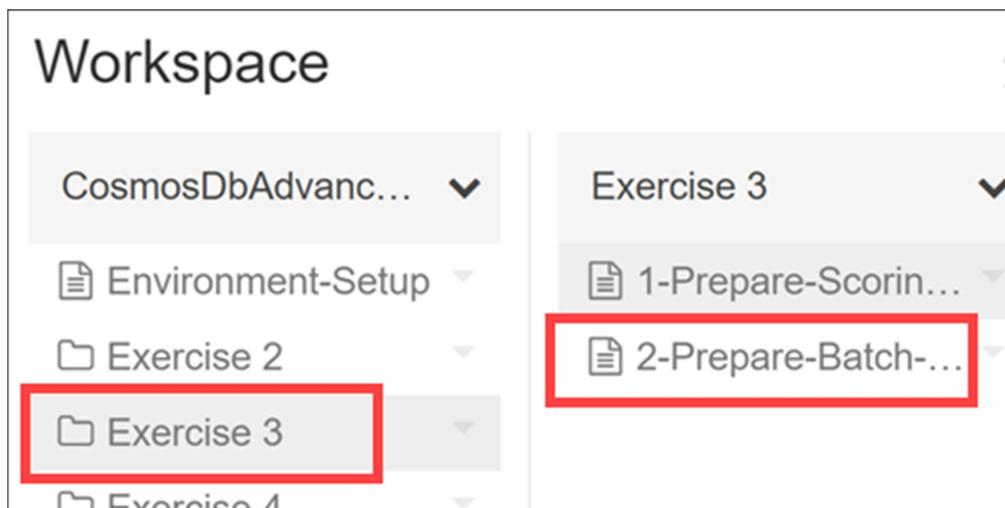
작업 3 : 배치 스코어링 모델 준비

이 작업에서는 Azure Databricks 노트북을 사용하여 배치 스코어링에 사용될 의심스러운 활동을 감지하는 데 사용되는 모델을 준비합니다.

- 당신의 Databricks 작업 공간에서, 왼쪽 메뉴에서 **Workspace** 를 선택 한 후, **사용자 선택** 및 사용자 계정을 선택합니다



- 사용자 작업 공간에서 **ScaleAnalytics** 폴더를 선택한 후 Exercise 3 폴더를 선택하고 이름이 **2-Prepare-Batch-Scoring-Model** 인 노트북을 선택하십시오 .



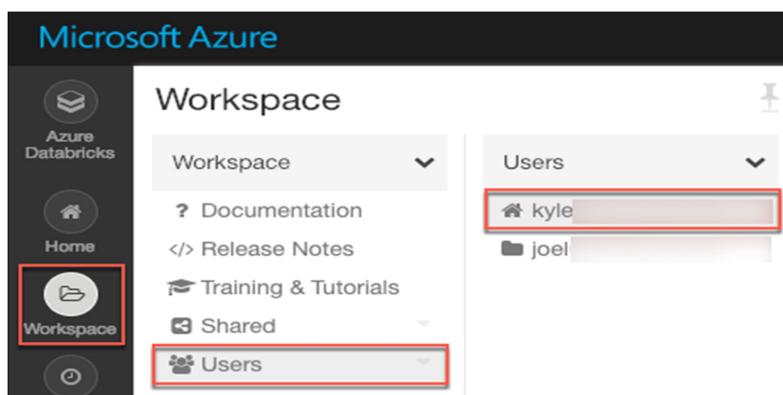
- 2-Prepare-Batch-Scoring-Model** 노트북에서 이 작업의 나머지 단계를 완료하기 위한 지시 사항을 따르십시오.

연습 4 : 배치점수 데이터 배포

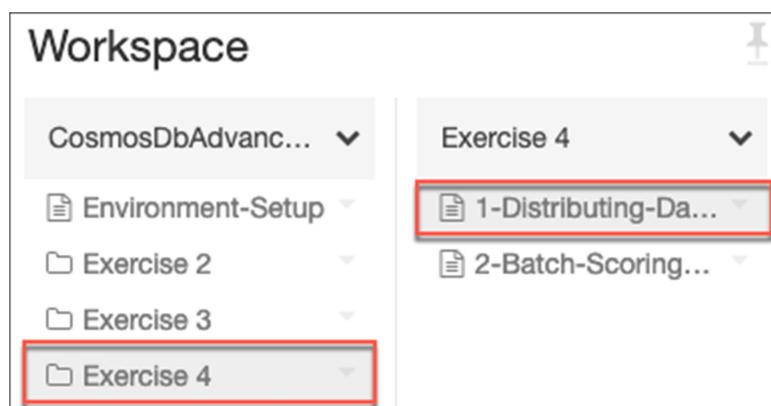
작업 1 : Cosmos DB 를 사용하여 배치 점수 데이터 배포

이 작업에서는 Azure Databricks 노트북을 **transactions** 사용하여 머신 러닝 모델로 Databricks 델타 테이블에 저장된 데이터를 일괄 처리 합니다. 스코어링 결과는 새로운 **scored_transactions** 델타 테이블에 기록되며 의심스러운 거래는 Cosmos DB 에도 다시 기록됩니다.

- 당신의 Databricks 작업 공간에서, 왼쪽 메뉴에서 **Workspace** 를 선택 한 후, 사용자 선택 및 사용자 계정을 선택합니다.



- 사용자 작업 공간에서 **ScaleAnalytics** 폴더를 선택한 다음 Exercise 4 폴더를 선택하고 **1-Distributing-Data-Globally** 라는 노트북을 선택하십시오

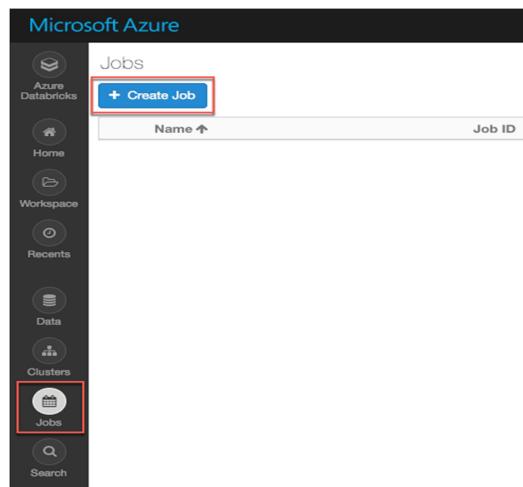


- 1-Distributing-Data-Globally** 노트북에서 이 작업의 나머지 단계를 완료하기 위한 지시 사항을 따르십시오.

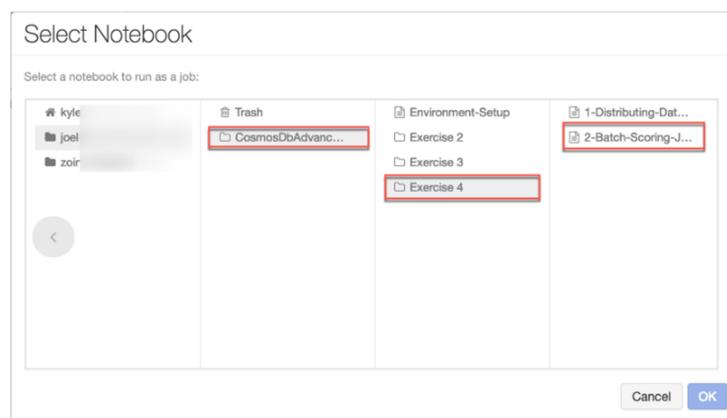
작업 2 : Azure Databricks 작업을 사용하여 일정에 따라 트랜잭션 점수를 일괄 처리

이 작업에서는 Azure Databricks 작업을 생성합니다. 이 작업은 시간별 일정에 따라 트랜잭션에 대해 일괄 평가를 수행하는 노트북을 실행합니다.

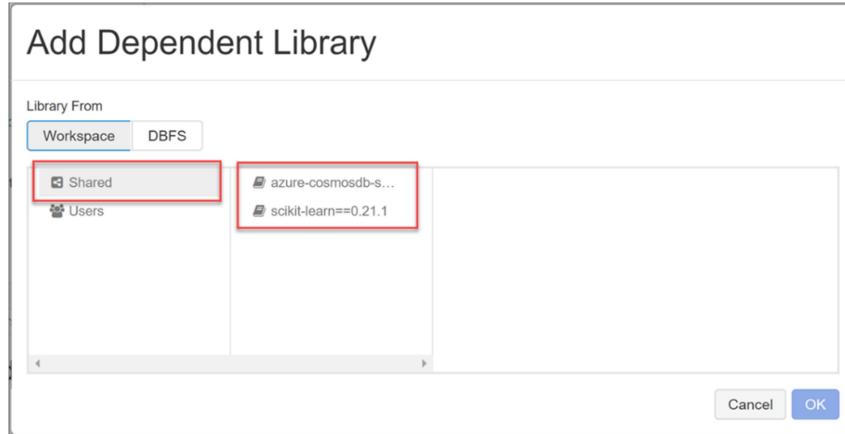
- 당신의 Databricks 작업 공간으로 이동 후 왼쪽 메뉴에 **Jobs** 선택한 다음 **+선택** 하여 작업을 만듭니다 .



- 제목없는 작업 화면에서 다음 단계를 완료하십시오.
 - Batch-Scoring-Job** 과 같은 제목을 입력하십시오 .
 - 작업 옆의 **노트북 선택** 링크를 선택하고 **노트북** 선택 대화 상자에서 사용자 -> 사용자 계정-> Scale Analytics-> Exercise 4 를 선택한 다음 **2-Batch-Scoring-Job** 노트북을 선택하고 **확인을 선택하십시오 .**

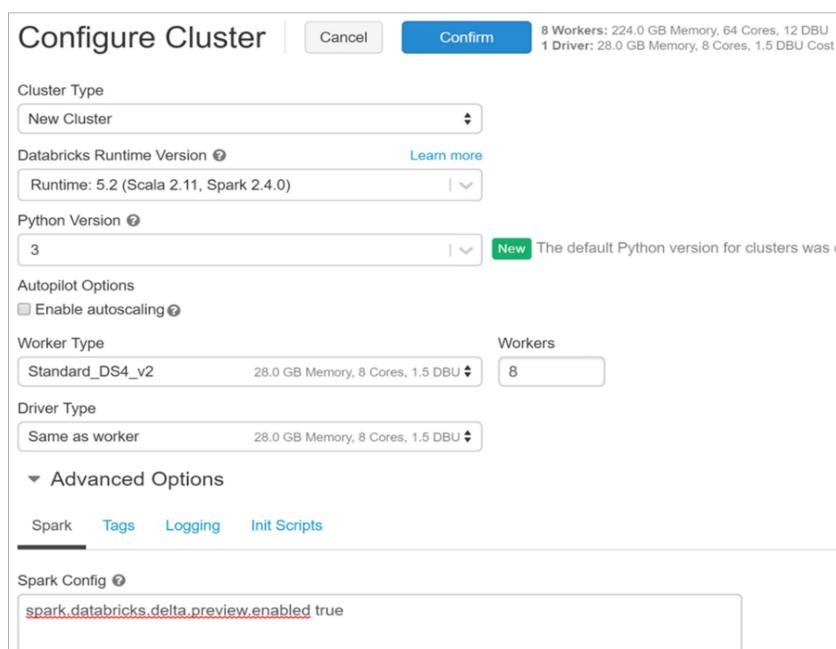


- 종속 라이브러리 옆에있는 **추가를** 선택하고 공유 폴더로 이동한 다음 **azure-cosmosdb-spark** 라이브러리를 선택하고 **확인을** 선택하십시오 .
- **scikit-learn == 0.21.1** 라이브러리도 추가하려면 위 단계를 반복하십시오 .

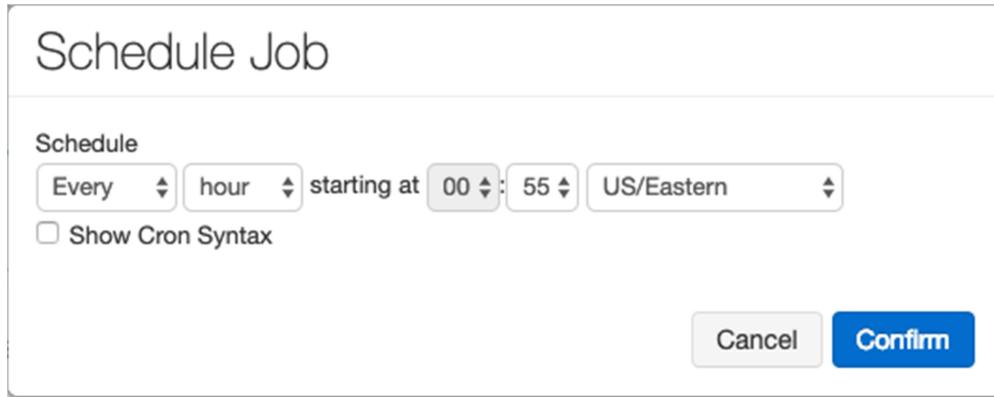


- 클러스터 편집을 선택합니다.
 - **Databricks 런타임 버전** : 런타임 5.5LTS (Scala 2.11, Spark 2.4.3)
 - **파이썬 버전** : 3
 - **Worker Type** : Standard_DS3_v2
 - **Workers:** 1
 - **Driver Type** : Worker 와 동일
 - Advanced options 을 열어 Spark Config 에
`spark.databricks.delta.preview.enabled true` 입력되어 있는지 확인하십시오 .

○



- 클러스터 구성을 저장하려면 **확인을 선택하십시오** .
- 일정 옆에있는 **편집을 선택** 하고 작업 예약 대화 상자에서 일정을 현재 시간에 가까운 값에서 시작하여 매시간으로 설정하면 적절한 시간에 트리거 된 것을 볼 수 있습니다. 시간대를 선택하고 **확인을 선택하십시오** .



3. 최종 작업 화면은 다음과 같아야합니다.

The screenshot shows the 'Batch-Scoring-Job' details page. It includes:

- Job ID: 1
- Task: Notebook at /Users/joel@solliance.net/CosmosDbAdvancedAnalytics/Exercise 4/2-Batch-Scoring-Job - [Edit](#) / [Remove](#)
- Parameters: [Edit](#)
- Dependent Libraries:
 - com.microsoft.azure:azure-cosmosdb-spark_2.4.0_2.11:1.4.0 - (Maven) [Remove](#)
 - scikit-learn==0.21.1 - (Pypi) [Remove](#)
- Cluster: Driver: Standard_DS4_v2, Workers: Standard_DS4_v2, 8 workers, 5.2 (includes Apache Spark 2.4.0, Scala 2.11) [Edit](#)
- Schedule: Every hour starting at 12:05am (US/Pacific) [Edit](#) / [Remove](#)
- Advanced ▾

4. 완료되면 작업 목록으로 돌아가 려면 <모든 작업 을 선택하십시오 .
5. 작업 시작을 기다리는 동안 왼쪽 메뉴에서 **작업 공간** 을 선택 **2-Batch-Scoring-Job** 하고 Exercise 4 폴더 아래의 노트북으로 이동하십시오 .
6. 노트북을 열고 몇 분 동안 배치 스코어링 프로세스를 수행하는 데 사용되는 단계를 이해하십시오. 보시다시피, 이전 작업에서 트랜잭션 데이터를 준비하고 변환하는 과정과 거의 동일합니다.
7. Databricks 의 왼쪽 메뉴에서 **클러스터** 를 선택한 후 작업 클러스터에 대한 **작업 실행** 을 선택 하여 작업 진행 상황을 모니터 할 수 있습니다 . 노트북이 표시되고 노트북 내에서 실행 시간과 결과를 볼 수 있습니다.

Name	State	Nodes	Driver	Worker	Runtime	Creator	Actions
lab-cluster	Running	3	Standard_DS4_v2	Standard_DS4_v2	5.1 (includes Apa...)		Job Run / Spark UI / Logs
job-1-run-1	Terminating	9	Standard_DS4_v2	Standard_DS4_v2	5.1 (includes Apa...)		Job Run / Spark UI / Logs

연습 5 :보고서

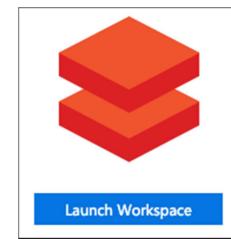
[소요 시간 : 30 분]

이 연습에서는 비즈니스 분석가가 사용할 수 있도록 Power BI에서 대시 보드 및 보고서를 만들고 데이터 과학자와 분석가가 데이터를 대화 형으로 쿼리하고 시각화 할 수 있도록 Azure Databricks 내에서 생성합니다.

작업 1 : 전 세계 사기 추세를 요약하고 시각화하기 위해 Power BI 활용

이 작업에서는 Azure Databricks 클러스터의 JDBC URL을 사용하여 Power BI 데스크톱에서 연결합니다. 그런 다음 보고서를 작성하고 대시 보드에 추가하여 글로벌 사기 경향을 요약하고 시각화하여 데이터에 대한 통찰력을 얻습니다.

1. [Azure Portal](#)에서 Azure Databricks 작업 영역으로 이동하고 개요 블레이드에서 **작업 영역 시작**을 선택하고 필요한 경우 Azure 자격 증명으로 작업 영역에 로그인합니다.
2. 왼쪽 메뉴에서 **클러스터**를 선택합니다



Name	State	Nodes
lab-cluster	Running	3

3. 아래로 스크롤하여 고급 옵션 섹션을 펼친 다음 JDBC / ODBC 탭을 선택 하십시오.
첫 번째 JDBC URL 값을 복사하십시오 .

The screenshot shows the Databricks Cluster configuration interface for a cluster named 'lab-cluster'. The 'JDBC/ODBC' tab is active. In the 'JDBC URL' section, two URLs are listed. The top URL is highlighted with a red box: `jdbc:spark://eastus.azuredatabricks.net:443/default;transportMode=http;ssl=1;httpPath=/sql/protocolv1/o/8433778235244215/0210-035431-quad242;AuthMech=3;UID=token;PWD=<personal-access-token>`. Below it is another URL: `jdbc:spark://eastus.azuredatabricks.net:443/default;transportMode=http;ssl=1;httpPath=/sql/protocolv1/o/8433778235244215/lab-cluster;AuthMech=3;UID=token;PWD=<personal-access-token>`.

4. 이제 Power BI Desktop에서 Spark 클러스터 연결을 설정하는 데 사용할 JDBC 서버 주소를 구성하도록 JDBC URL 을 수정해야합니다.

- JDBC URL 에서 :
 - https 를 jdbc:spark 로 교체 합니다.
 - 포트 번호와 /sql 사이의 경로에 모든 것을 제거하십시오
 - URL 끝에서 ;AuthMech=3;UID=token;PWD=<personal-access-token> 문자열을 제거하십시오.

아래 이미지에서 강조 표시로 표시된 구성 요소를 유지하십시오

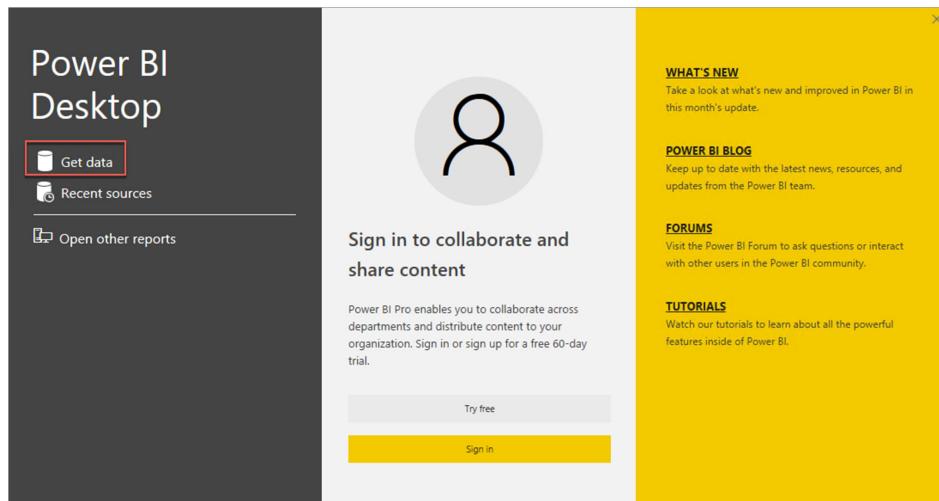
```
JDBC URL ?  
jdbc:spark://eastus.azuredatabricks.net:443/default;transportMode=http;ssl=1;http  
Path$sql/protocolv1/o/8433778235244215/0210-035431-  
quad242;AuthMech=3;UID=token;PWD=<personal-access-token>
```

- 이 예에서 서버 주소는 다음과 같습니다.

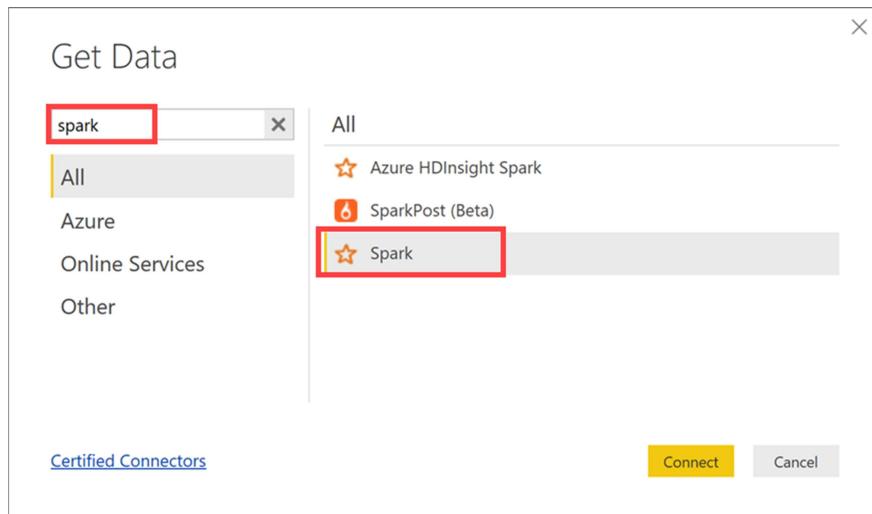
<https://eastus.azuredatabricks.net:443/sql/protocolv1/o/8433778235244215/0210-035431-quad242>

- 서버 주소를 복사하십시오.

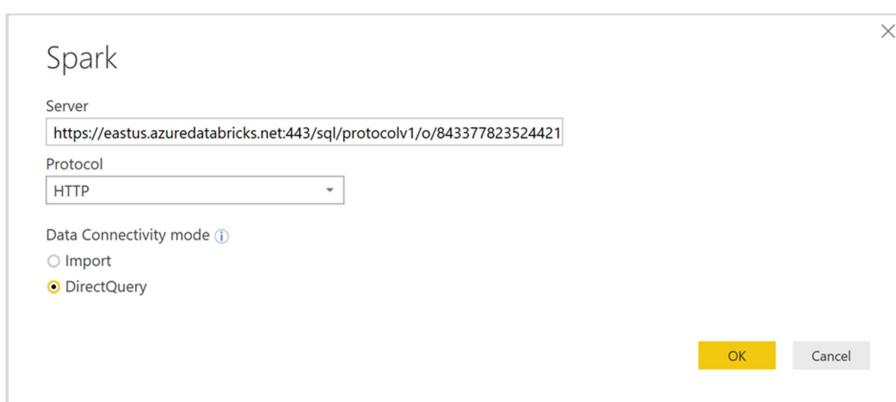
5. Power BI Desktop 을 연 다음 **데이터 가져 오기** 를 선택 합니다 .



6. 대화상자에서 데이터 가져 오기를 선택 후에 spark 를 넣은 결과 목록에서 **Spark** 를 선택하십시오 .

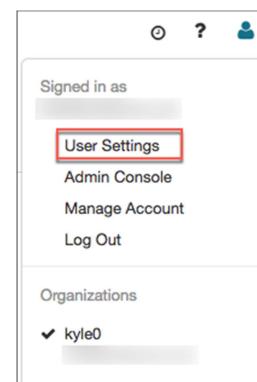


7. 위에서 작성한 Databricks 서버 주소를 서버 필드에 입력하십시오.
8. 프로토콜을 HTTP 로 설정하십시오.
9. 데이터 연결 모드로 DirectQuery 를 선택한 다음 확인을 선택하십시오.

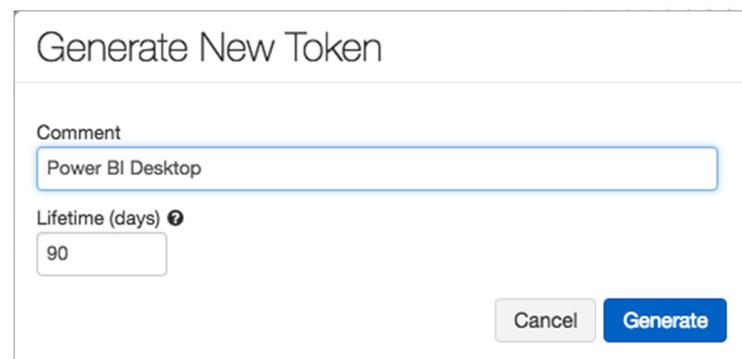


데이터 연결 모드를 DirectQuery 로 설정하면 처리를 Spark 로 오프로드 할 수 있습니다. 이는 대량의 데이터가 있거나 거의 실시간 분석을 원할 때 이상적입니다.

10. 다음 화면에서 자격 증명을 입력하기 전에 Azure Databricks에서 액세스 토큰을 만들어야합니다. Databricks 작업 공간의 오른쪽 상단에서 계정 아이콘을 선택한 다음 메뉴에서 사용자 설정을 선택하십시오.



11. 사용자 설정 페이지에서 새 토큰 생성을 선택하고 주석에 "Power BI Desktop"을 입력 한 다음 생성을 선택하십시오.

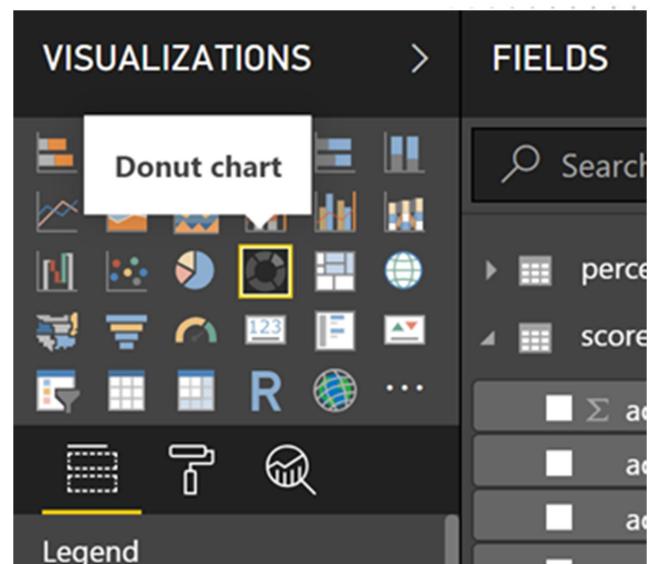


12. 생성 된 토큰을 복사하여 아래에 두 번 이상 필요할 수 있으므로 저장하십시오. 참고 : 토큰 생성 대화 상자를 닫으면 Databricks에서 토큰에 액세스 할 수 없으므로이 값을 텍스트 랩 또는이 실습 중에 액세스 할 수있는 다른 위치에 저장하십시오.

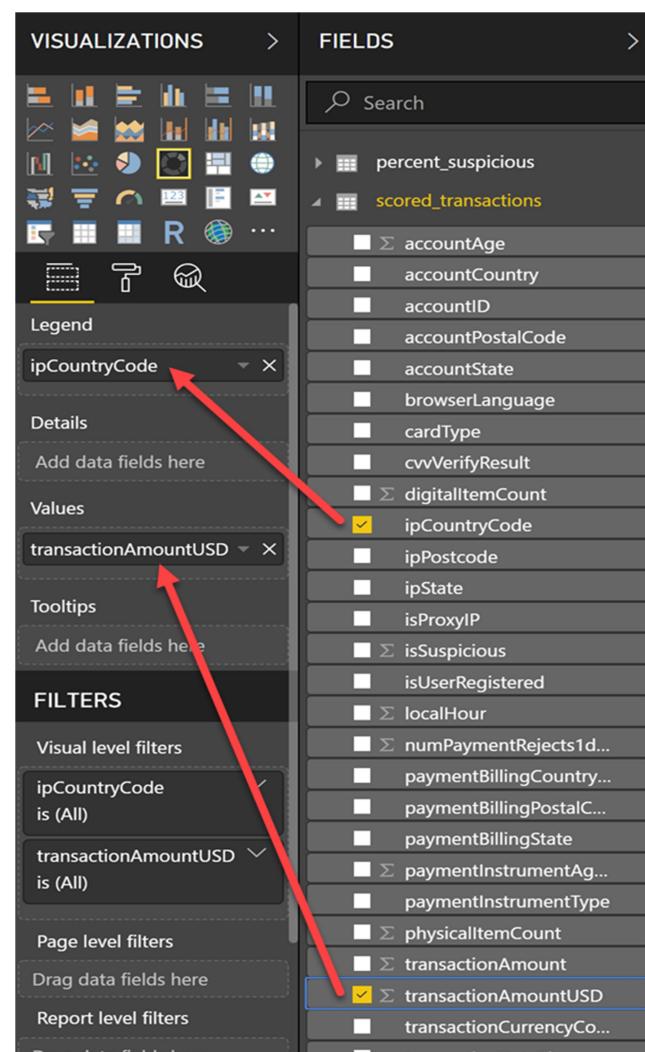
13. Power BI Desktop으로 돌아가서 사용자 이름으로 "토큰"을 입력하고 Databricks에서 복사한 액세스 토큰을 암호 필드에 붙여 넣습니다.

14. `scored_transactions` 및 `percent_suspicious` 을 선택하신 후 **로드** 를 클릭하세요.

15. 잠시 후 표가 오른쪽에 나열된 빈 보고서 화면으로 리디렉션됩니다. 오른쪽 메뉴에서 시각화 아래 및 테이블 및 필드 목록 옆의 **도넛형 차트** 시각화를 선택 합니다.

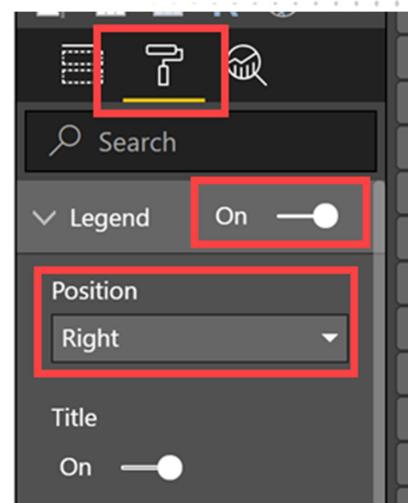


16. 확장 `scored_transactions` 테이블 아래에서, `ipCountryCode` 를 범주(Legend)로 드래그 하고, 그리고 `transactionAmountUSD` 를 마찬가지 값(Values)으로 드래그합니다.

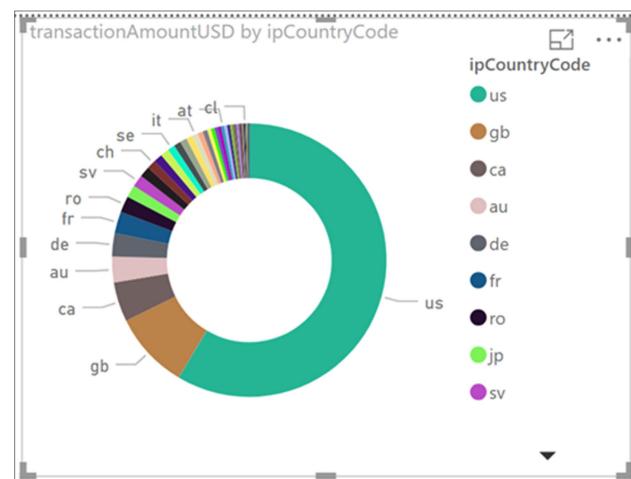


17. 이제 도넛 형

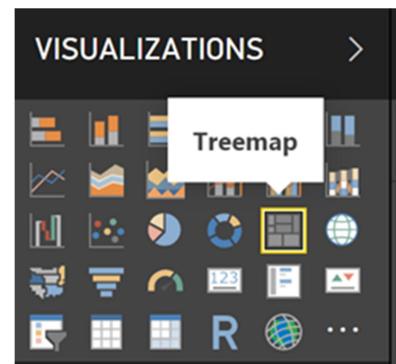
차트의 **형식** 탭을 선택하고 아래의 **범례** 섹션을 확장하십시오. 범례를 켜 려면 **켜기**를 선택하고 **위치** 아래에서 **오른쪽**을 선택하십시오. 차트 범례가 켜지고 오른쪽에 표시됩니다.



18. 도넛 형 차트는 다음과 유사하며 국가 코드별로 미국 달러 거래량을 표시합니다.



19. 보고서에서 빈 영역을 선택하여 도넛 형 차트를 선택 해제하십시오. 이제 트리 맵 시각화를 선택하십시오.



20. `scored_transactions` 테이블에서 `transactionAmountUSD` 필드를 그룹으로, `isSuspicious` 를 값으로 드래그 합니다 .

The screenshot shows the Power BI Fields pane on the right and the Visualizations pane on the left. In the Fields pane, under the 'scored_transactions' table, 'transactionAmountUSD' and 'isSuspicious' are both selected, indicated by yellow checkmarks. A large red 'X' is drawn over the entire visualization pane, indicating that the current configuration is incorrect or needs to be corrected.

VISUALIZATIONS

FIELDS

Search

- percent_suspicious
- scored_transactions
 - transactionAmountUSD
 - isSuspicious
- accountAge
- accountCountry
- accountID
- accountPostalCode
- accountState
- browserLanguage
- cardType
- cvvVerifyResult
- digitalItemCount
- ipCountryCode
- ipPostcode
- ipState
- isProxyIP
- isUserRegistered
- localHour
- numPaymentRejects1d...
- paymentBillingCountry...
- paymentBillingPostalC...
- paymentBillingState
- paymentInstrumentAg...
- paymentInstrumentType
- physicalItemCount
- transactionAmount
- transactionAmountUSD
- transactionCurrencyCo...

21. 보고서에서 빈 영역을 선택하여 트리 맵 차트를 선택
해제하십시오. 이제 맵 시각화를 선택하십시오. `scored_transactions` 테이블에서 `ipCountryCode`의 필드를 위치(Location)로, `isSuspicious`를 크기로 드래그 합니다.

VISUALIZATIONS > FIELDS

Search

scored_transactions

- accountAge
- accountCountry
- accountID
- accountPostalCode
- accountState
- browserLanguage
- cardType
- cvvVerifyResult
- digitalItemCount
- ipCountryCode
- ipPostcode
- ipState
- isProxyIP
- isSuspicious
- isUserRegistered
- localHour
- numPaymentRejects1d...
- paymentBillingCountry...
- paymentBillingPostalC...

ipCountryCode

Legend

Add data fields here

Latitude

Add data fields here

Longitude

Add data fields here

Size

isSuspicious

Tooltips

22. 보고서에서 빈 영역을 선택한 다음 ArcGIS Maps for Power BI 시각화를 선택하십시오. 이 시각화 사용에 대한 조건을 수락하라는 메시지가 표시되면 지금 하십시오. `percent_suspicious` 테이블에서 `ipCountryCode` 필드를 위치(Location)로 드래그 한 다음 `SuspiciousTransactionCount`를 색상 아래로 드래그 합니다.

VISUALIZATIONS > FIELDS

Search

percent_suspicious

- ipCountryCode
- PercentSuspicious
- SuspiciousTransactionCount
- TotalTransactionCount

scored_transactions

- accountAge
- accountCountry
- accountID
- accountPostalCode
- accountState
- browserLanguage
- cardType
- cvvVerifyResult
- digitalItemCount
- ipCountryCode
- ipPostcode
- ipState
- isProxyIP

Location
ipCountryCode

Latitude
Add data fields here

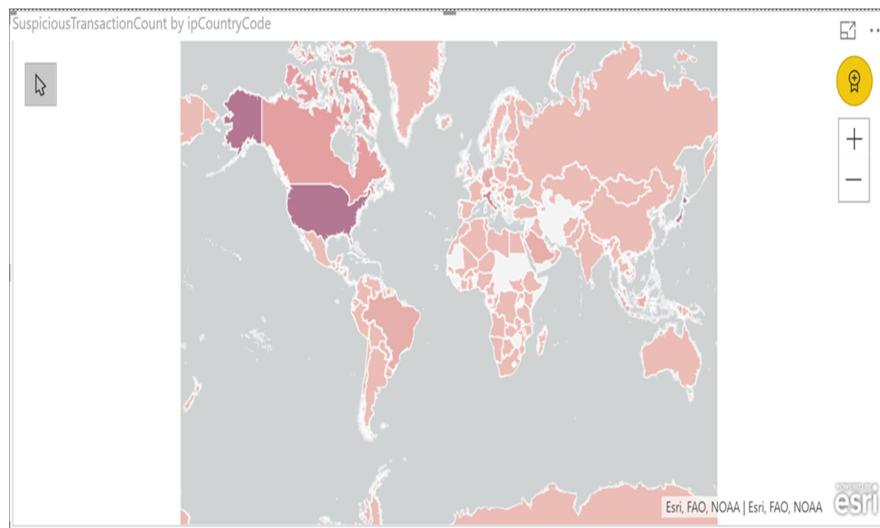
Longitude
Add data fields here

Size
Add data fields here

Color
SuspiciousTransactionCount

Time

23. ArcGIS Map 은 다음과 유사하게 표시되며 금액이 적은 국가보다 어두운 색상으로 의심스러운 거래가 많은 국가를 표시합니다.

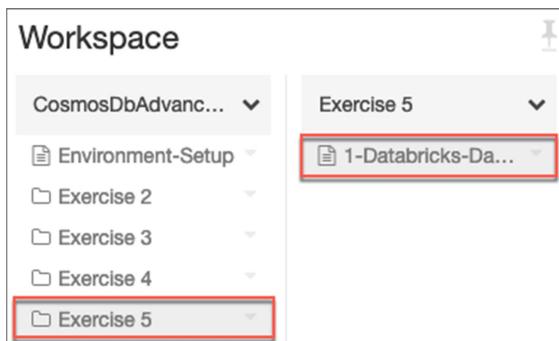


차트를 로컬 디스크에 저장할 수 있습니다. 저장된 후에는 차트를 Power BI 웹 사이트에 업로드하여 모든 차트 및 데이터 연결을 그대로 유지하면서 온라인으로 사용할 수 있습니다.

작업 2 : Azure Databricks에서 대시 보드 만들기

이 작업에서는 Azure Databricks 노트북을 사용하여 Azure Databricks에 시각화를 표시하기 위한 대시 보드를 작성합니다.

- 당신의 Databricks 작업 공간에서, 왼쪽 메뉴에서 **Workspace**를 선택한 후, **사용자 선택 및 사용자 계정**을 선택합니다
- 사용자 작업 공간에서 **Scale analytics** 폴더를 선택한 다음 **Exercise 5** 폴더를 선택하고 **1-Databricks-Dashboards** 노트북을 선택하십시오.



- 1-Databricks - Dashboard** 노트북에서, 이 작업의 나머지 단계를 완료하기 위한 지시 사항을 따르십시오.

실습 후

[소요 시간 : 10 분]

이 연습에서는 랩을 지원하기 위해 만든 모든 Azure 리소스를 삭제합니다. 실습 랩에 참석 한 후 제공되는 모든 단계를 수행하여 랩 리소스에 대한 계정 요금이 계속 청구되지 않도록해야합니다.

작업 1 : 리소스 그룹 삭제

- [애저 포털](#) 왼쪽 메뉴에서 자원 그룹을 선택하여 사용했던 자원 그룹으로 이동합니다.
- 연구 그룹의 이름을 검색하고 목록에서 선택하십시오.
- 명령 모음에서 삭제를 선택하고 자원 그룹 이름을 다시 입력하고 삭제를 선택하여 삭제를 확인하십시오.

참조

이 내용은 아래에 게시 되어있습니다 :

<https://github.com/azure-datasolution/CloudScaleAnalytics>