Azure Synapse data explorer Hands on Lab

Azure Synapse data explorer (SDX) is a fast, fully managed data analytics service for real-time analysis on large volumes of data streaming from applications, websites, IoT devices, and more. Ask questions and iteratively explore data on the fly to improve products, enhance customer experiences, monitor devices, and boost operations. Quickly identify patterns, anomalies, and trends in your data. Explore new questions and get answers in minutes. Run as many queries as you need, thanks to the optimized cost structure.

This hands-on lab will bring you up to speed on the basics of SDX starting from deployment, ingestion and basic Kusto Query Language (KQL) queries.

# Prerequisites

Please make sure that you know the basics of KQL and attend one or more online courses

* [KQL from Scratch](https://app.pluralsight.com/library/courses/kusto-query-language-kql-from-scratch)
* [Azure data exploring](https://aka.ms/adx.pluralsight.azure-data-exploring)
* [How to start with Azure Data Explorer](https://aka.ms/adx.pluralsight.how-to-start-with-adx) ([blog](https://aka.ms/adx.pluralsight.how-to-start-with-adx.blog))
* [Advanced KQL](https://aka.ms/adx.pluralsight.advanced-kql) ([blog](https://aka.ms/adx.pluralsight.advanced-kql.blog))

This hands-on lab is driven by the Pass Camp team. To prepare everything, we need to get some information from you.

# Contact points

If you have any questions during the lab, feel free to contact one of the following ADX team members. All of us are willing to help you with any topic.

* Ralph Kemperdick [Ralph.Kemperdick@microsoft.com](mailto:Ralph.Kemperdick@microsoft.com)
* Christoph Seck [c.seck@kigroup.de](mailto:c.seck@kigroup.de)

# What’s in the lab?

The lab covers a pre-created setup and will allow you to start at a place where you feel comfortable.

As a participant of the lab, you are free to start by connecting to the <https://aka.ms/kustofree> infrastructure and create your database yourself.

# Tasks

The lab is intended to take 1 hour.

## Deploy an Azure Synapse data explorer pool

1. Create an SDX pool
2. Create a database

## Ingest Data

1. Ingest data (<https://pmlabstorage18008.blob.core.windows.net/lab/StormEvents.json>) into a table called “StormEvents” using OneClick ingestion
2. Ingest data (<https://pmlabstorage18008.blob.core.windows.net/lab/Population.json>) into a table called “PopulationData” using OneClick ingestion
3. Verify that the data has been ingested correctly

## Write queries

1. Understand when data has been ingested
2. Get an understanding of the data (filter, computed columns, lookups, etc.)
3. Render a pie chart on summarized data
4. Execute timeseries queries (Average value of a signal, make sure that the timeseries is complete)
5. Execute geospatial queries

## Visualize Data

Create a beautiful visualization / dashboard in the tool you like the most. Try to use either <http://dataexplorer.azure.com/> or PowerBI.

# Cheat sheet

Feel free to look at the solution and adapt it to your needs. You can skip the first two sections and work with our help cluster: <https://help.kusto.windows.net>

## Deploy an Azure Synapse data explorer pool

Create an SDX pool

* [Quickstart: Create a Data Explorer pool using the Azure portal (Preview) - Azure Synapse Analytics | Microsoft Docs](https://docs.microsoft.com/en-us/azure/synapse-analytics/data-explorer/data-explorer-create-pool-portal)
* [Quickstart: Create a Data Explorer pool using Synapse Studio (Preview) - Azure Synapse Analytics | Microsoft Docs](https://docs.microsoft.com/en-us/azure/synapse-analytics/data-explorer/data-explorer-create-pool-studio)

Create a database

* [Quickstart: Create an Azure Data Explorer cluster and database | Microsoft Docs](https://docs.microsoft.com/en-us/azure/data-explorer/create-cluster-database-portal#create-a-database)

## Ingest Data

Ingest data into StormEvents

* Ingest data (<https://pmlabstorage18008.blob.core.windows.net/lab/StormEvents.json>) into a table called “StormEvents” using OneClick ingestion
  1. [Use one-click ingestion to ingest data into Azure Data Explorer | Microsoft Docs](https://docs.microsoft.com/en-us/azure/data-explorer/ingest-data-one-click)
  2. Use the blob to generate the data model and ingest it

Ingest data into PopulationData

* Replicate the work you did for StormEvents. Use <https://pmlabstorage18008.blob.core.windows.net/lab/Population.json> and the table name “PopulationData”

Check for ingestion failures

* Connect to your cluster with <https://aka.ms/kwe>
* Add your cluster query URI or the one of the help cluster <https://help.kusto.windows.net>

1. .show ingestion failures
2. // Count records in the table
3. StormEvents
4. | count
5. // Peek at some arbitrary records
6. StormEvents
7. | take 10

## Write queries

KQL quick reference guide: [https://docs.microsoft.com/en-us/azure/data-explorer/kql-quick-reference](https://nam06.safelinks.protection.outlook.com/?url=https%3A%2F%2Fdocs.microsoft.com%2Fen-us%2Fazure%2Fdata-explorer%2Fkql-quick-reference&data=02%7C01%7Ctzgitlin%40microsoft.com%7C1c273450ecd649056bf208d7bdb9bf4b%7C72f988bf86f141af91ab2d7cd011db47%7C1%7C0%7C637186477782502685&sdata=rP9%2FVcXsb3ordj4XqUzIWl3R%2BNl8fdO71zdHcsaph9I%3D&reserved=0)

KQL cheat sheet: <https://github.com/marcusbakker/KQL/blob/master/kql_cheat_sheet_dark.pdf>

Get an understanding of the data (filter, computed columns, etc.)

1. // Find events that caused damage to crops
2. StormEvents
3. | where DamageCrops > 0
4. | take 50
5. // Find events that caused damage to property/crops in Florida
6. StormEvents
7. | where DamageProperty + DamageCrops > 0
8. | where State == "FLORIDA"
9. | order by DamageProperty desc
10. // Find events that caused damage to property in Florida in Q1 of 2007
11. StormEvents
12. | where StartTime between (datetime(2007-01-01) .. datetime(2007-03-31))
13. | where DamageProperty > 0
14. | where State == "FLORIDA"
15. | project StartTime, EventType, DamageProperty
16. | take 50
17. // Get statistics of storms per state
18. StormEvents
19. | summarize
20. count(),
21. EventsWithDamageToCrops = countif(DamageCrops > 0),
22. dcount(EventType),
23. min(StartTime),
24. max(EndTime) by State
25. | top 10 by count\_ desc

Render a pie chart on summarized data

1. // Find events invoving wind that caused most damage to property
2. StormEvents
3. | where EventType has "Wind"
4. | top 5 by DamageProperty desc
5. | summarize count() by State
6. | render piechart

Execute joins

1. // display the population per state
2. PopulationData
3. | where Population < 3000000
4. | join kind=inner (StormEvents) on State
5. | extend Damage = DamageProperty + DamageCrops
6. | top 5 by Damage desc
7. | project StartTime, State, EventType, Damage, Population

Execute timeseries queries

1. let MostFrequentEventType = toscalar(
2. StormEvents
3. | summarize count() by EventType
4. | top 1 by count\_ desc
5. | project EventType);
6. StormEvents
7. | where EventType == MostFrequentEventType
8. | summarize count() by bin(startofmonth(StartTime), 1d)
9. | render columnchart
10. StormEvents
11. | where State in ("CALIFIRNIA", "OKLAHOMA", "OHIO", "FLORIDA", "MISSISSIPPI", "WASHINGTON")
12. | extend month = startofmonth(StartTime)
13. | make-series SumDamage = sum(DamageProperty + DamageCrops) default = double(0.0) on month step 1d by State
14. | extend SumDamage = series\_fill\_linear(SumDamage, double(0.0), true)
15. | render timechart

Geospatial queries

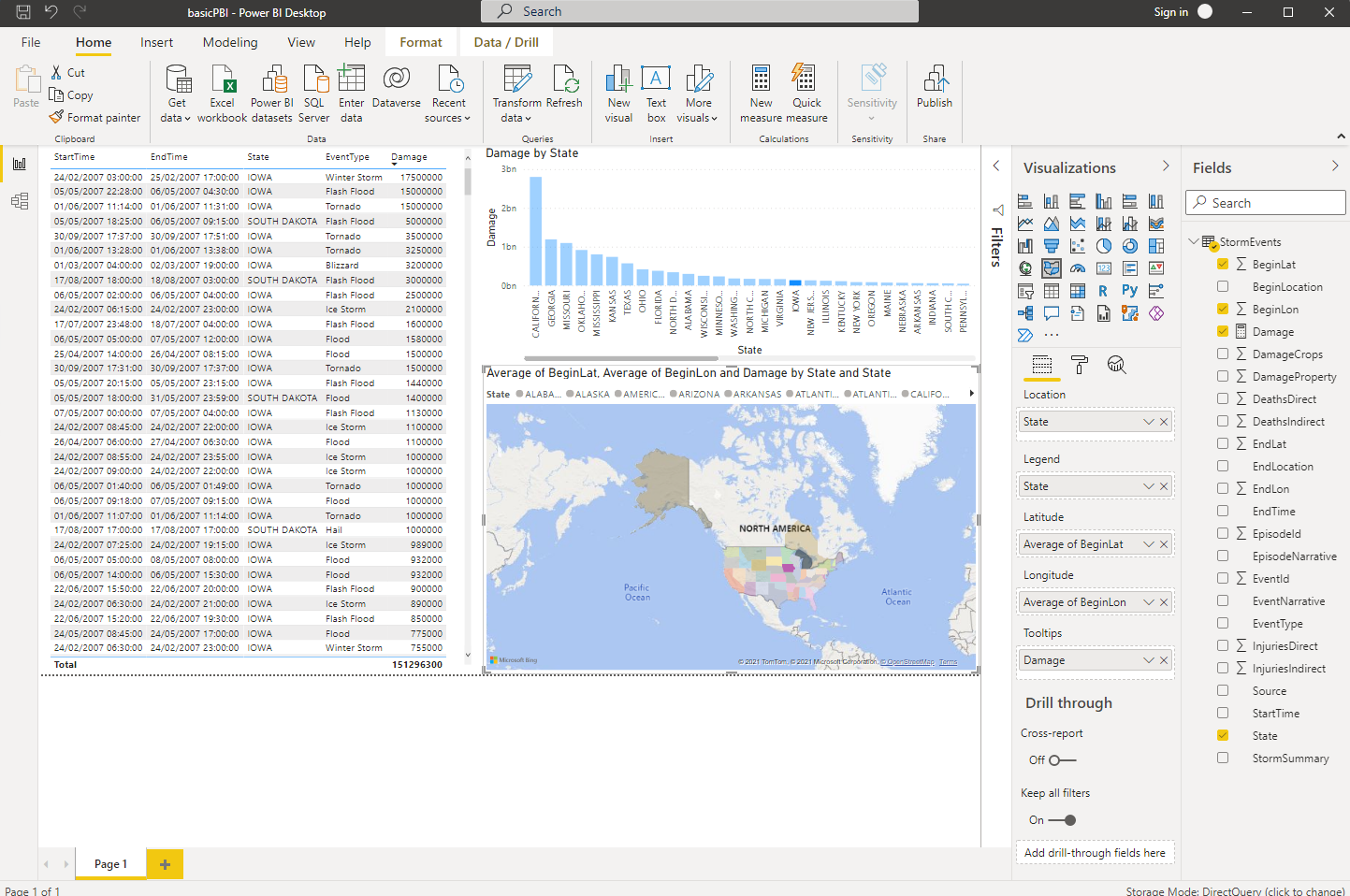
1. StormEvents
2. | where isnotempty( BeginLat) and isnotempty( BeginLon)
3. | project BeginLon, BeginLat, EventType
4. | where geo\_point\_in\_circle(BeginLon, BeginLat,-80.6048, 28.0393, 10000)
5. | render scatterchart with (kind = map)
6. //Show geofencing based on a polygon
7. StormEvents
8. | where isnotempty( BeginLat) and isnotempty( BeginLon)
9. | project BeginLon, BeginLat, EventType
10. | where geo\_point\_in\_polygon(BeginLon, BeginLat, dynamic({"type":"Polygon","coordinates":[[[-81.06880187988281,24.75306702526595],[-81.12510681152344,24.728122241065808],[-81.13609313964844,24.691319554166277],[-81.09901428222656,24.671978191593258],[-81.03858947753905,24.70005337937338],[-80.97129821777344,24.718766657061526],[-80.92391967773438,24.74558411549905],[-80.86624145507812,24.775513050757333],[-80.93215942382812,24.79483832122786],[-81.06880187988281,24.75306702526595]]]}))
11. | render scatterchart with (kind = map)
12. //Show clustering - based on S2 cells
13. // - Storm events in California. The events are filtered by a California state polygon and aggregated by event type and hash.
14. let california = dynamic({"type":"Polygon","coordinates":[[[-123.233256,42.006186],[-122.378853,42.011663],[-121.037003,41.995232],[-120.001861,41.995232],[-119.996384,40.264519],[-120.001861,38.999346],[-118.71478,38.101128],[-117.498899,37.21934],[-116.540435,36.501861],[-115.85034,35.970598],[-114.634459,35.00118],[-114.634459,34.87521],[-114.470151,34.710902],[-114.333228,34.448009],[-114.136058,34.305608],[-114.256551,34.174162],[-114.415382,34.108438],[-114.535874,33.933176],[-114.497536,33.697668],[-114.524921,33.54979],[-114.727567,33.40739],[-114.661844,33.034958],[-114.524921,33.029481],[-114.470151,32.843265],[-114.524921,32.755634],[-114.72209,32.717295],[-116.04751,32.624187],[-117.126467,32.536556],[-117.24696,32.668003],[-117.252437,32.876127],[-117.329114,33.122589],[-117.471515,33.297851],[-117.7837,33.538836],[-118.183517,33.763391],[-118.260194,33.703145],[-118.413548,33.741483],[-118.391641,33.840068],[-118.566903,34.042715],[-118.802411,33.998899],[-119.218659,34.146777],[-119.278905,34.26727],[-119.558229,34.415147],[-119.875891,34.40967],[-120.138784,34.475393],[-120.472878,34.448009],[-120.64814,34.579455],[-120.609801,34.858779],[-120.670048,34.902595],[-120.631709,35.099764],[-120.894602,35.247642],[-120.905556,35.450289],[-121.004141,35.461243],[-121.168449,35.636505],[-121.283465,35.674843],[-121.332757,35.784382],[-121.716143,36.195153],[-121.896882,36.315645],[-121.935221,36.638785],[-121.858544,36.6114],[-121.787344,36.803093],[-121.929744,36.978355],[-122.105006,36.956447],[-122.335038,37.115279],[-122.417192,37.241248],[-122.400761,37.361741],[-122.515777,37.520572],[-122.515777,37.783465],[-122.329561,37.783465],[-122.406238,38.15042],[-122.488392,38.112082],[-122.504823,37.931343],[-122.701993,37.893004],[-122.937501,38.029928],[-122.97584,38.265436],[-123.129194,38.451652],[-123.331841,38.566668],[-123.44138,38.698114],[-123.737134,38.95553],[-123.687842,39.032208],[-123.824765,39.366301],[-123.764519,39.552517],[-123.85215,39.831841],[-124.109566,40.105688],[-124.361506,40.259042],[-124.410798,40.439781],[-124.158859,40.877937],[-124.109566,41.025814],[-124.158859,41.14083],[-124.065751,41.442061],[-124.147905,41.715908],[-124.257444,41.781632],[-124.213628,42.000709],[-123.233256,42.006186]]]});
15. StormEvents
16. | project BeginLon, BeginLat, EventType
17. | where geo\_point\_in\_polygon(BeginLon, BeginLat, california)
18. | summarize count() by EventType, hash = geo\_point\_to\_s2cell(BeginLon, BeginLat, 7)
19. | project geo\_s2cell\_to\_central\_point(hash), EventType, count\_
20. | render piechart with (kind=map)

## Visualize Data

Create a beautiful visualization / dashboard in the tool you like the most. Try to use either <http://dataexplorer.azure.com/> or PowerBI ([Visualize data using the Azure Data Explorer connector for Power BI](https://docs.microsoft.com/azure/data-explorer/power-bi-connector#get-data-from-azure-data-explorer), [Using Azure Data Explorer timeseries capabilities in Power BI - Microsoft Tech Community](https://techcommunity.microsoft.com/t5/azure-data-explorer/using-azure-data-explorer-timeseries-capabilities-in-power-bi/ba-p/2727977))

PowerBI:





The SDX dashboard:



