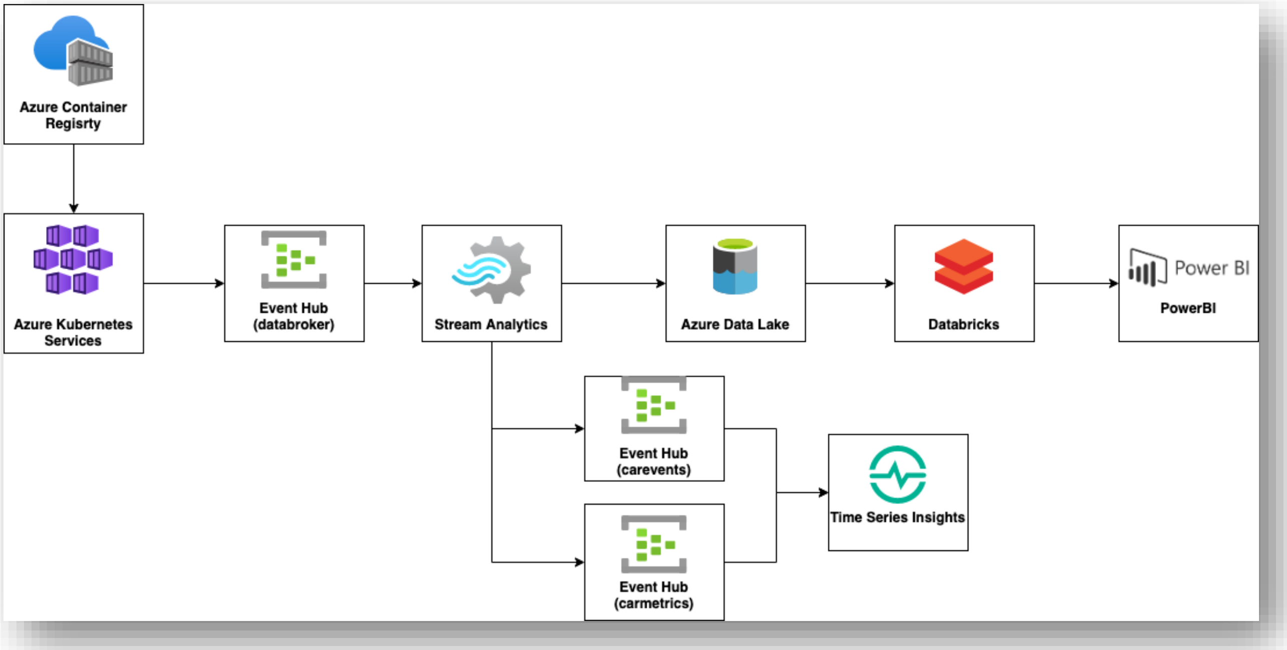
PwC Hackathon

LAB Overview

This lab will show you how to build solution that generates data and using Azure services process data and visualize them.



Task 1: Prepare Docker Development Environment

In this section you will learn how to prepare environment for a Docker.

1. On the left **Hub** menu click **Virtual Machines.**
2. On the **Virtual Machines** blade click the **Create** button marked with a plus sign and choose **Virtual Machine**.
3. On the **Configuration basic settings** blade provide the following configuration:

    - **Subscription:** XXXXX
    - **Resource Group:**
        - **<**Name of your resource group**>**
    - **Virtual machine name:** vm-hackathonX
    - **Location:** West Europe
    - **Availability options:** No infrastructure redundancy required
    - **Image:** Windows Server 2019 Datacenter – Gen2
    - **Size:** D4s_v3
    - **Username:** hackathonX
    - **Password:** type and remember your password-secret
    - **Confirm password:** confirm your password-secret above
    - **Public inbound ports:** Allow selected ports
    - **Select public inbound port**:
        - **HTTP (80)**
        - **RDP (3389)**

In **Disks** section:

- **OS disk type:** Premium SSD

In **Networking** section:

- **Virtual network**: (new) hackathon0X-vnet
- **Subnet**: (new) default (10.0.0.0/24)
- **Public IP address**: (new) vm-hackathonX-ip
- **NIC Network Security Group**: basic
- **Authentication type:** Password
- **VM disk type:** SSD

Other options leave as it was a default.

In **Management** section:

- **Boot diagnostics**: Disable
- **Enable auto-shutdown**: Unmarked
- **Backup**: Unmarked

Other options leave as it was a default.

Next click on **Review + Create** and then **Create.**

4. Next step the Azure Virtual Machine will be deployed.
5. After successful deployment, connect to VM using RDP.
6. https://docs.docker.com/desktop/windows/install/
7. Click Get Docker for Windows (Stable) and download the installer. There is an option that installs Docker Desktop for Windows, azure CLI, Visual Studio Code, and Git for Windows using this PowerShell script: https://github.com/djkormo/k8s-AKS-primer/blob/master/install/VM/install-all.ps1. In case you do not want to use the script, the next steps will provide you with how to install these applications manually.
8. Run the installer and procced with instructions showed on your screen.
   Warning: Your VM will be during the installation process.
9. Go to page: https://nodejs.org/en/ and download LTS version. This software will be required to run our code.
10. Run installer.

Task 2: Create an Event Hub namespace

In this section you will learn how to create an Event Hub namespace from Azure Portal.

1. On the left **Hub** menu click on **Create a resource.**
2. On the **New** blade click on **Internet of Things** and select **Event Hubs.**
3. On the **Create namespace** blade provide the following configurations:

   - **Subscription:** XXXXXX
   - **Resource group:** <Name of your resource group**>**
   - **Namespace name:** event-hub-hackathon0X
   - **Location:** West Europe
   - **Pricing Tier:** Standard
   - **Throughput Units:** 1
   - **Enable Auto-inflate:** None

When, you finish click on button **Create.**

4. Next step the Azure Event Hub namespace will be deployed.

Task 3: Add Event Hubs to namespace

In this section you will learn how to add a new instance to namespace of created Azure Event Hub from previous task.

1. After successful deployment of Azure Event Hub, go to created service **event-hub-student0X**.
2. On the **Event Hubs Namespace** page click on the **Event Hub** button marked with a plus sign.
3. On the **Create Event Hub** blade provide the following configurations:

    - **Name**: databroker
    - **Partition Count**: 2
    - **Message Retention**: 1
    - **Capture**: Off

When you finish, click on button **Create.**

4. Next step the Event Hub will be added to namespace.
5. **Now add two more EventHubs:**
    - **carMetrics**
    - **carEvents**

Task 4: Add SAS Policy key to Event Hub

In this section you will learn how to add SAS Policy key to instance of Event Hub from Azure Portal.

1. On the main **Overview** page of **Event Hubs Namespace**, click on **Event Hubs** from the left menu.
2. On the **Event Hubs** page click on created instance of Event Hub - **dataBroker**.
3. On the instance of Event Hub – **dataBroker** page click on **Shared access policies** from the left menu.
4. On the **Shared access policies** page click on **Add** button.
5. On the **Add SAS Policy** blade provide the following configurations:

    - **Policy name:** sender-application
    - **Mange:** <leave-empty>
    - **Send:** Select
    - **Listen:** <leave-empty>

    At the end click on **Create.**
6. After creating SAS Policy key click on new created policy **sender-application** and copy to the clipboard value from **Connection string-primary key** or **Connection string-secondary key.** You will need this key at a later Task.

Task 5: Add consumer groups to Event Hub

In this section you will learn how to add consumer group to instance of Event Hub from Azure Portal.

7. On the main **Overview** page of **Event Hubs Namespace**, click on **Event Hubs** from the left menu.
8. On the **Event Hubs** page click on created instance of Event Hub - **dataBroker**.
9. On the instance of Event Hub – **dataBroker** page click on **Consumer groups** from the left menu.
10. On the **Consumer Group** page click on button **Consumer Group** marked with plus assign and add two groups named:

    - eventhub
    - timewindows

Task 6: Create Docker Image

In this section you will learn how to create Docker Image based on NodeJS application which is simulated simple metrics to Azure Event Hub.

1. Go to https://github.com/cloudstateu/hackathon/blob/main/Zadanie/ContainerApp.zip and download this file by clicking **Download**.
2. Unzip this file on your VM.
3. You will see folder with 3 files.
4. Open **index2.js** in some editor and replace in code **EVENTHUB_CONNECTION_STRING** value with value from **TASK 4 point 6.**

You should have this effect (it is an example):

```
const client = EventHubClient.createFromConnectionString("Endpoint=sb://event-hub-namespace-
chmuromaniak2.servicebus.windows.net/;SharedAccessKeyName=sender-
application;SharedAccessKey=xlz02dy7wu6UAUkAGNr0wXJ4ZzTakGmaZwhGK9pcmUo=;EntityPath=databroker",
"databroker");
```

5. Open console in unzipped folder.
6. Build image running in command line: **docker build –t containerapp .**

Task 7: Create image repository service: ACR and push image.

In this section you will learn how to create Azure Container Registry and how to push image to it.

1. Open Azure Portal.
2. Click **Create Resource** on left menu.
3. Click **Containers** -> **Azure Container Registry.**
4. On the Create container registry page provide the following configurations:
   - **Registry name:** hackathonXregistry
   - **Resource Group:**
     - **Use existing:** <Your Resource Group>
   - **Location:** West Europe
   - **SKU:** Standard
   - Next click on **Create.**
5. Go to deployed resource and on left menu click **Access keys.**
6. Make option "Admin user" enable. Copy username, password, and login server.

   Make sure you open a Console not a PowerShell ☺ Otherwise it will not work!
7. Open console and type **docker login –p passwordCopiedFromPoint6 –u userNameCopiedFromPoint6 loginServerFromPoint6**
8. Tag image with repository by typing in console: **docker tag containerapp loginServerFromPoint6/containerapp**
9. Push image to repository by typing in console: **docker push loginServerFromPoint6/containerapp**
10. After push process is finished, go to portal to Azure Container Registry.
11. Open your ACR.
12. On left menu, in Services section click Repositories.
13. Check if your image is on the list.

Task 8: Create Azure Kubernetes Services.

In this section, you will learn how to create Azure Kubernetes Service using Azure CLI and how to install Kubernetes CLI (kubectl).

1. Log on your Azure Virtual Machine.
2. Open web browser and go to: https://docs.microsoft.com/en-us/cli/azure/install-azure-cli-windows?view=azure-cli-latest

3. Install Azure CLI.
4. Open **PowerShell** and type: **az login**
5. Go to your Resource Group page and copy Subscription ID. In PowerShell type: **az account set --subscription <Copied Subscription ID>**
6. Procced with instruction in console window.
7. After login process type in console: **az aks create --resource-group <name of your resource group> --name hackathonXCluster --node-vm-size Standard_Ds2_v2 --node-count 1 --enable-addons monitoring --generate-ssh-keys**
8. After cluster is deployed (it make takes up to 20 minutes), install Kubernetes CLI by typing in console: **az aks install-cli**
9. Next execute the next command:

   $env:path += 'C:\Users\hackatchonX\.azure-kubectl'

10. Get credentials to cluster: **az aks get-credentials --resource-group hackathonX --name hackathonXCluster**
11. Verify connection using command: **kubectl get nodes**

Task 9: Deploy image to Kubernetes cluster.

In this section, you will learn how to create deployment file and how to deploy it.

1. Configure ACR (Azure Container Registry) integration for existing AKS (Azure Kubernetes Service) cluster**:**

az aks update -n <Name of AKS Cluster> -g <Name of Resource Group> --attach-acr <acr-name>

2. Ensure you have the proper AKS credentials:
   az aks get-credentials -g <Name of Resource Group -n <Name of AKS Cluster>

3. Create deployment file (deployment.yml):

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: containerapp
  labels:
    app: containerapp
spec:
  replicas: 1
  selector:
    matchLabels:
      app: containerapp
  template:
    metadata:
      labels:
        app: containerapp
    spec:
      containers:
      - name: containerapp
        image: loginServerFromPoint6Task7/containerapp
```

4. Open console and type: **kubectl create –f pathToDeploymentFile**
5. Verify status using command: **kubectl get pods**

Task 10: Scale POD

In this section, you will learn how to scale POD.

1. Open deployment.yaml
2. Change value in property **replicas** from **1** to **3**:

```
replicas: 3
```

3. Open console and type: **kubectl apply –f pathToDeploymentFile**
4. Verify status using command: **kubectl get pods**
5. Then you should see 3 instances of POD.

Task 11: Create Storage Account and Data Lake Storage Gen2

In this section you will learn how to create Data Lake Store from Azure Portal.

1. On the left **Hub** menu click on **Create a resource.**
2. On the **Search resources, services, and docs** window type **Storage accounts.**
3. On the **Storage accounts** page click on button **Create** marked with plus assign**.**
4. On the **Create a storage account** blade provide the following configurations**:**

In **Basics** section:

- **Subscription:** XXXXXX
- **Resource group:** <Name of your resource group**>**
- **Name**: hackathonXstorage
- **Region:** West Europe
- **Performance:** Standard
- **Redundancy:** Locally-redundant storage (LRS)

In **Advanced** section:

- **Enable hierarchical namespace:** marked
  Other options leave as it was a default.

  Next click on **Review + Create** and then **Create.**

5. Next step the Storage Account will be deployed.
6. Go to deployed resource and click on **Containers** from the left menu.
7. On the **New container** window type name of your container (hackathonXcontainer) and choose **Private.** Next click on **Create.**

Task 12: Create Stream Analytics

In this section you will learn how to create Azure Stream Analytics.

1. On the left **Hub** menu click on **Create a resource.**
2. On the **Search resources, services, and docs** window type **Stream Analytics job.**
3. On the **New Stream Analytics job** blade provide the following configurations**:**

- **Job name**: sa-hackathon0X
- **Subscription: XXXXX**
- **Resource Group:** <Name of your resource group**>**
- **Location:** West Europe
- **Hosting Environment:** Cloud
  Other options leave as it was a default.

When, you finish click on button **Create.**

4. Next step the Azure Stream Analytics will be deployed.

Task 13: Grant the Stream Analytics job permissions to access the Event Hub

Because for the Stream Analytics job to access your Event Hub using managed identity, the service principal you created must have special permissions to the Event Hub.
This section will learn grant this permission.

1. Go to **Access control (IAM)** in your **Event Hub (databroker)**
2. Select **+Add** and **Add role assignment.**
3. On the **Add role assignment** page, enter the following options:
    - **Role:** Azure Event Hubs Data Owner
    - **Assign access to**: User, group, or service principal
    - **Select**: The name of your Stream Analytics job created in previous Task.

Task 14: Add Event Hub input to Stream Analytics

In this section you will learn how to add input source of Event Hub in Stream Analytics.

1. On the main **Overview** page of **Stream Analytics**, click on **Inputs**.
2. On the **Inputs** page click on button **Add Stream inputs** and select **Event Hub**.
3. On the **Event Hub new input** provide the following configurations:

    - **Input alias:** EventHub
    - **Select Event Hub from your subscription**
    - **Event Hub namespace:** event-hub-hackathonX
    - **Event Hub name**
        - **Use existing:** databroker
    - **Event Hub consumer group:** eventhub
    - **Authentication mode:** Managed Identity
    - **Event serialization format:** JSON
    - **Encoding:** UTF-8
    - **Event compression type:** None

And click on **Save.**

Task 15: Give the Stream Analytics job access to the Storage Account

In this section you will learn how to give the job access to a specific existing container.

1. Go to **Access control (IAM)** in your **Storage Account -> Containers (name of your container)**
2. Select **+Add** and **Add role assignment.**
3. On the **Add role assignment** page, enter the following options:
    - **Role:** Storage Blob Data Contributor
    - **Assign access to**: User, group, or service principal
    - **Select**: The name of your Stream Analytics job.

Task 16: Add Data Lake Storage output to Stream Analytics

In this section you will learn how to add output source of Azure Data Lake in Stream Analytics.

1. On the main **Overview** page of **Stream Analytics**, click on **Outputs**.
2. On the **Outputs** page click on button **Add** and select **Blob storage/ADLS Gen2.**
3. On the **Data Lake Store New output** provide the following configurations:

    - **Output alias:** DataLakeStore
    - **Select Data Lake Store from your subscriptions**
    - **Subscription:** XXXXX
    - **Storage account:** hackathonXstorage
    - **Container:** (Existing) and choose: hackathonXcontainer
    - **Path prefix pattern:** data/{date}/{time}
    - **Date format:** YYYY/MM/DD
    - **Time format:** HH
    - **Event serialization number:** CSV (comma (,))
    - **Encoding:** UTF-8

And click on **Save.**

Task 17: Add Data Lake query to Stream Analytics

In this section you will learn how to add SQL query in Azure Stream Analytics which processing all data from Event Hub to Data Lake Store.

1. On the main **Overview** page of **Stream Analytics**, add below lines of SQL code clicking on **Edit Query** in **Query** section:

```
SELECT *
INTO DataLakeStore
```

FROM EventHub

And click on **Save.**

2. On the main **Overview** page of **Stream Analytics** click on **Start** to run Stream Analytics job.
3. On the main **Overview** page of **Storage Account** click on **your Container** and check if new data files appear.

Task 18: Create Time Series Insight

In this section you will learn how to create Azure Time Series Insight

1. On the left **Hub** menu click on **Create a resource.**
2. On the **New** blade click on **Internet of Things** and select **Time Series Insight**
3. On the **Time Series Insight environment** blade provide the following configurations**:**

- **Subscription:** XXXXX
- **Resource Group:** Your Resource Group
- **Environment name:** timeinsighthackathonX
- **Location:** West Europe
- **Tier:** Gen2
- **Capacity:** Leave default

Time Series ID:

- **Property Name:** carID

Cold Store:

- **Storage Account Name:** coldstorehackathonX
- **Storage Account kind:** StorageV2
- **Storage Account replication:** LRS
- **Hierarchical Namespace:** Disabled

Warm Store:

- **Enable worm store:** No

In Event Source section:

- **Create an event source:** No

In Networking section:

- **Connectivity method:** Public endpoint

When, you finish click on button **Create.**

Task 19: Add TimeWindows input to Stream Analytics

In this section you will learn how to add input source of Event Hub in Stream Analytics.

1. On the main **Overview** page of **Stream Analytics**, click on **Stop** to stop the job of Azure Stream Analytics.
2. On the main **Overview** page of **Stream Analytics**, click on **Inputs**.
3. On the **Inputs** page click on button **Add Stream inputs** and select **Event Hub**.
4. On the **Event Hub new input** provide the following configurations:

    - **Input alias:** TimeWindows
    - **Select Event Hub from your subscription**
    - **Event Hub namespace:** event-hub-hackathonX
    - **Event Hub name**
        - **Use existing:** databroker
    - **Event Hub consumer group:** timewindows
    - **Event serialization format:** JSON
    - **Encoding: UTF-8**
    - **Event compression type:** None

And click on **OK.**

Task 20: Add metrics and events output to Stream Analytics

In this section you will learn how to add output source of events and metrics Event Hub in Stream Analytics.

1. On the main **Overview** page of **Stream Analytics**, click on **Outputs**.
2. On the **Outputs** page click on button **Add** and select **Event Hub**.
3. On the **Event Hub New output** provide the following configurations:

    - **Output alias:** CarMetrics
    - **Select Event Hub from your subscription**
    - **Event Hub namespace:** event-hub-hackathonX
    - **Event Hub name**
        - **Use existing:** carMetrics
    - **Authentication Mode:** Managed Identity
    - **Partition key column:** <leave empty>
    - **Event serialization format:** JSON
    - **Encoding:** UTF-8
    - **Format:** Line separeted

4. Also add another Event Hub output:

    - **Output alias:** carEvents
    - **Select Event Hub from your subscription**
    - **Subscription: XXXXX**
    - **Event Hub namespace:** event-hub-hackathonX
    - **Event Hub name**
        - **Use existing:** carEvents
    - **Authentication Mode:** Managed Identity
    - **Partition key column:** <leave empty>
    - **Event serialization format:** JSON
    - **Encoding:** UTF-8
    - **Format:** Line separeted

And click on **Save.**

Task 21: Events and Metrics query of Event Hub to Stream Analytics

In this section you will learn how to add SQL query in Azure Stream Analytics which processing all data from Event Hub to Data Lake Store.

1. On the main **Overview** page of **Stream Analytics** click on **Stop** to stopping Stream Analytics job.
2. On the main **Overview** page of **Stream Analytics**, add below lines of SQL code clicking on **Edit Query** in **Query** section:

    ```
    SELECT carID, speed, fuelLevel, batteryStatus, temperature, CAST(Time as datetime) as Time
    INTO carMetrics
    FROM EventHub

    SELECT carID, avg(fuelLevel) AS avgfuelevel
    INTO carEvents
    FROM EventHub TIMESTAMP BY Time
    GROUP BY carID, TumblingWindow(second,30)
    HAVING avgfuelevel < 20
    ```

And click on **Save.**

3. On the main **Overview** page of **Stream Analytics** click on **Start** to run Stream Analytics job.

Task 22: Add source to Time Series Insight

In this section you will learn how to add source event from Event Hub to Time Series Insight.

1. On the main **Overview** page of **Time Series Insight Environment**, click on **Event Sources** from the left menu.
2. On the **Event Sources** page click on button **Add** with marked plus assign.
3. On the **New event source** blade provide the following configurations:

   - **Event source name:** carMetricsEventHub
   - **Source:** EventHub
   - **Import option:** Use Event Hub from available subscriptions.
   - **Subscription Id: XXXXX**
   - **Event bus namespace:** event-hub-hackathonX
   - **Event hub name:** carMetrics
   - **Event hub policy value:** RootMangeSharedAccessKey
   - **Event hub consumer group:** $Default
   - **Start time:** Beginningnow
   - **Event serialization format:** JSON
   - **Timestamp property name:** Time

When you finish, click on button **Save.**

4. In this same way add another source from **events** Event Hub:

   - **Event source name:** eventsEventHub
   - **Source:** EventHub
   - **Import option:** Use Event Hub from available subscriptions.
   - **Subscription Id: XXXXX**
   - **Service bus namespace:** event-hub-hackathonX
   - **Event hub name:** carEvents
   - **Event hub policy name:** RootMangeSharedAccessKey
   - **Event hub consumer group:** $Default
   - **Start time:** Beginning now
   - **Event serialization format:** JSON
   - **Timestamp property name:** Time

Task 23: Explore data in TSI explorer

In this section you will learn how to visualize metrics in TSI Explorer

1. On the main **Overview** page of **Time Series Insight Environment**, click on **Go to TSI Explorer**.
2. Try to plot some graphs using the explorer

Task 24: Create Databricks workspace

In this section, you will learn how to create Azure Databricks workspace

1. Click on left menu Create Resource button.
2. Type Databricks in search field.
3. Choose Azure Databricks from list.
4. Click Create.
5. Enter configuration:

- **Subscription:** XXXXX
- **Resource Group:** hackathonX
- **Workspace name:** adb-hackathonX
- **Location:** West Europe
- **Pricing Tier:** Trial

6. Leave all other fields by default and click **Create**

Task 25: Create DataBricks compute cluster

In this section, you will learn how to create cluster for Azure Databricks.

1. In the Azure portal, go to the Databricks workspace that you created, and then click **Launch Workspace**.
2. You are redirected to the Azure Databricks portal. From the portal, click **New Cluster**.
3. In the New cluster page, provide the values to create a cluster.
    - **Cluster Name**: hackatchonXCluster
    - **Cluster Mode**: Standard
    - **Databricks Runtime Version: 8.3**
    - **Enable autoscaling:** off

- **Terminate After:** 120 minutes
- **Worker type:** Standard_DS3_V2
- **Workers:** 1
- **Drive Type:** Same as worker

Task: 26 Create databricks notebook

In this section, you will learn how to create Databricks notebook.

1. From left menu click **Workspace**.
2. Click on drop-down arrow and select **Create > Notebook**

   - **Name**: hackacthonXnotebook
   - **Default** language: Python
   - **Cluster**: hackatchonXCluster

3. Click **Create** button

Task 27: Mount datalake storage into dbfs

1. From the Azure portal, select the Azure Storage Account you created (hackatonXstorage), and then from left menu select **Access Keys.**
2. Copy one of the keys.
3. Go back to databricks notebook and paste the code below replacing proper values:

```
dbutils.fs.mount(
   source="wasbs://CONTAINER_NAME@STORAGE_NAME.blob.core.windows.net",
   mount_point="/mnt/hackathon",
   extra_configs = {"fs.azure.account.key.STORAGE_NAME.blob.core.windows.net": "ACCESS_KEY"}
)
```

This piece of code will connect from databricks into Azure Storage Account

4. Add new notebook cell and paste the code**:**

```
from pyspark.sql.types import StructType, DoubleType, IntegerType, TimestampType

schema = StructType() \
    .add("Index", IntegerType(), True) \
    .add("carID",IntegerType(),True) \
    .add("latitude",DoubleType(),True) \
    .add("longitude",DoubleType(),True) \
    .add("speed",IntegerType(),True) \
    .add("batteryStatus",IntegerType(),True) \
    .add("fuelLevel",IntegerType(),True) \
    .add("temperature",IntegerType(),True) \
    .add("MOPdistance",IntegerType(),True) \
    .add("numberOfPolice5km",IntegerType(),True) \
    .add("numberOfCostsOnWay",IntegerType(),True) \
    .add("roadTolls",IntegerType(),True) \
    .add("Time",TimestampType(),True) \
    .add("PartitionId",IntegerType(),True) \
    .add("EventEnqueuedUtcTime",TimestampType(),True)

data = spark.read.format("csv")\
  .option("header", "true") \
  .schema(schema) \
  .option("delimiter", ",") \
  .load("/mnt/hackathon/data/*/*/*/*")
```

This piece of code reads all .csv files from datalake into one Spark dataframe.

5. Display the dataframe using this command:

```
display(data)
```

Task 28: Plot some data in Azure Databricks

1. Add new cell in your databricks notebook
2. From the data table select one of carID (whichever from 1 to 1000)
3. Display data just for this car:

```
display(data.filter("carID = XXX"))
```

4. Change display type from table to line chart
5. In **Plot Options** give **Time** as a Key and **speed** as a Value

And click on **Apply.**

Task 29: Set-up Power BI environment

In this section, you will learn how to activate 60-days Free Trial of PowerBI Pro. PowerBI Pro is an on-line version of Power BI reporting service.

1. Go to app.powerbi.com
2. Login in using your credentials (the same as to portal.azure.com)]
3. In the top bar click “Free trial” and activate it.
4. Now you can use Power BI online for free for 60 days!
5. Log in to your virtual machine using RDP protocol
6. Install PowerBI desktop software from: https://www.microsoft.com/en-us/download/details.aspx?id=58494
7. Launch PowerBI desktop, close the welcome screen.

Task 30: Load batch data into Power BI

In this section, you will learn how to load data from Azure Databricks into PowerBI desktop. First, we need to create a table in Databricks, then make a connection between Databricks and Power BI and load the data at the end.

1. Got to your databricks workspace and open your notebook.
2. Add new cell at the end.
3. Paste the following code:

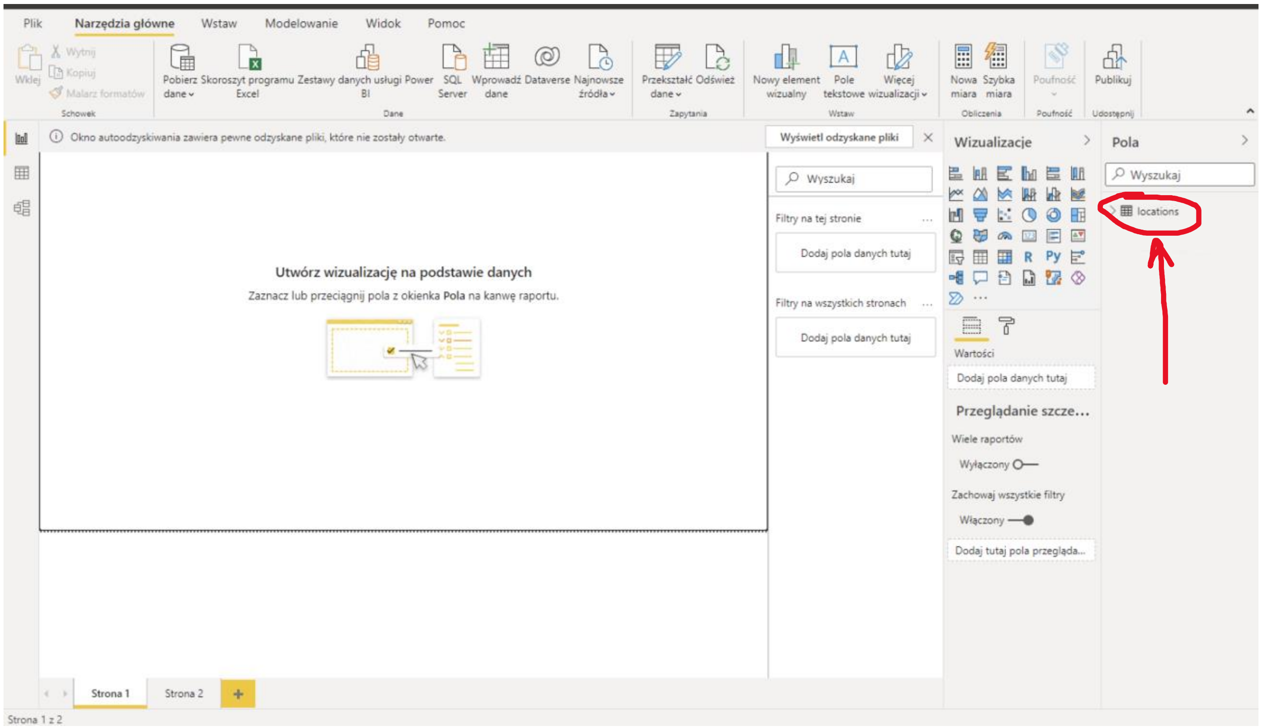
```
from  pyspark.sql.functions import window, avg

windowed_data = data.groupBy("carID",window("Time", "30 seconds")).agg(avg("latitude").alias("latitude"),
avg("longitude").alias("longitude"))
windowed_data.write.saveAsTable("locations")
```

4. This code snippet does two thigs:
    a. It groups data over time window. We do so in order to reduce granularity and reduce amount of data.
    b. It save the result DataFrame into “locations” table
5. In order to connect from PowereBI to Databricks we need a few variables. Here is how to retrieve them:
    a. Server Hostname – on the left menu click **Compute** button and click on your cluster. Expand **Advanced Options** menu and select **JDBC/ODBC**. Note down the **Server Hostname** parameter
    b. HTTP Path – From the same view note down **HTTP Path** parameter
6. Open Power BI desktop on your VM

7. Click **Get Data** button and **More…**
8. In the search bar enter “databricks” and select **Azure Databricks.** Click **Connect**
9. Fill in the form:

- **Server hostname**: paste the value retrieved in point 5.a
- **HTTP Path**: paste the value retrieved in point 5.b

10. Click **OK**
11. On the next windows you need to authenticate to Databricks. Select **Azure Active Directory** option and log in with your credentials.
12. When you log in successfully, you will see a screen when you can select data that you want to import. Check **locations** table in **default** database.
13. Click **Load** button
14. You should see a dataset created here:



Okno autoodzyskiwania zawiera pewne odzyskane pliki, które nie zostały otwarte.

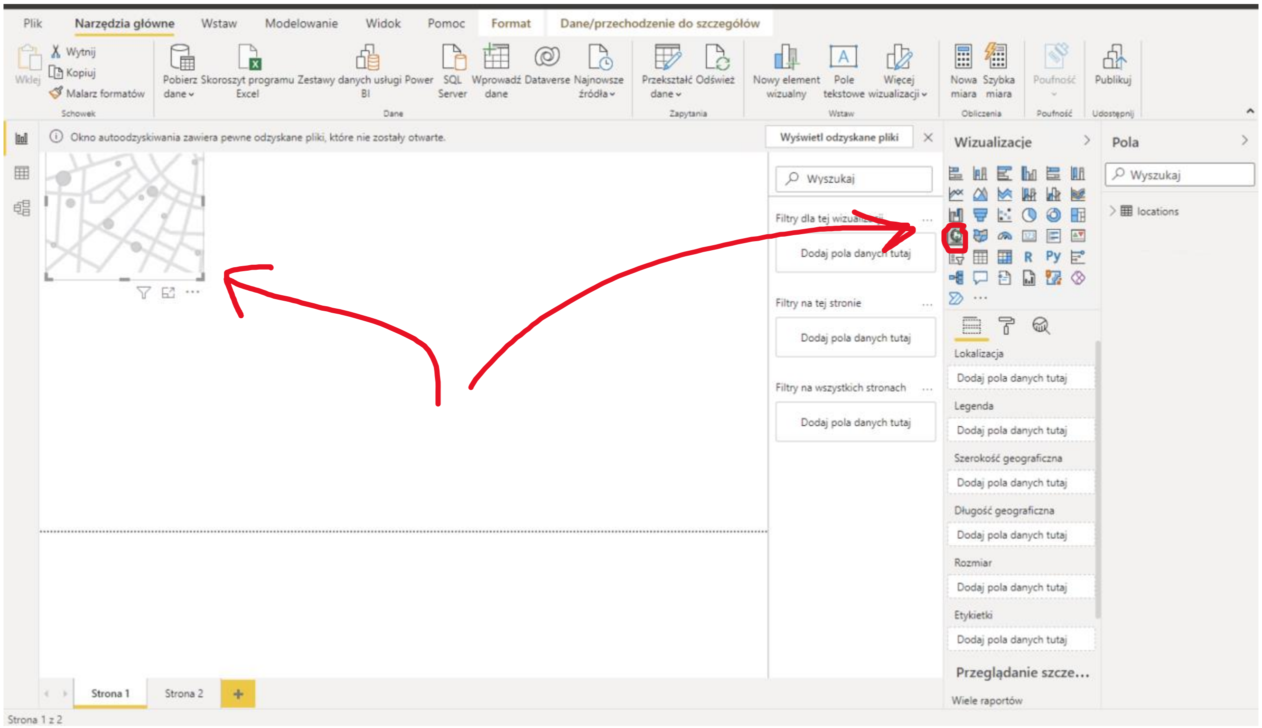
Utwórz wizualizację na podstawie danych

Zaznacz lub przeciągnij pola z okienka Pola na kanwę raportu.

Task 31: Visualize GPS data in Power BI

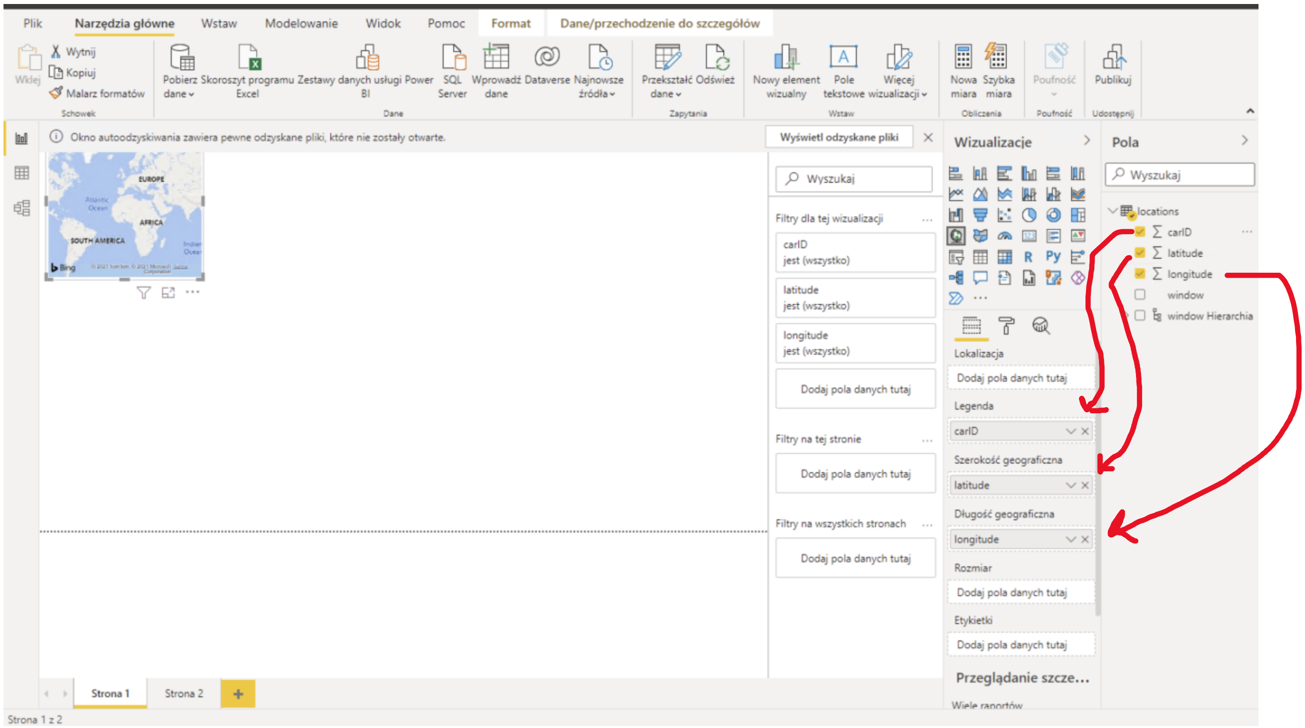
In this section, you will learn how to create simple report in Power BI that will allow you to track car’s locations.

1. Add map visualization into your report:



Okno autoodzyskiwania zawiera pewne odzyskane pliki, które nie zostały otwarte.

2. Expand location dataset and put:
    - Latitude to latitude
    - Longitude to longitude
    - CarID to legend



Okno autoodzyskiwania zawiera pewne odzyskane pliki, które nie zostały otwarte.

carID
jest (wszystko)

latitude
jest (wszystko)

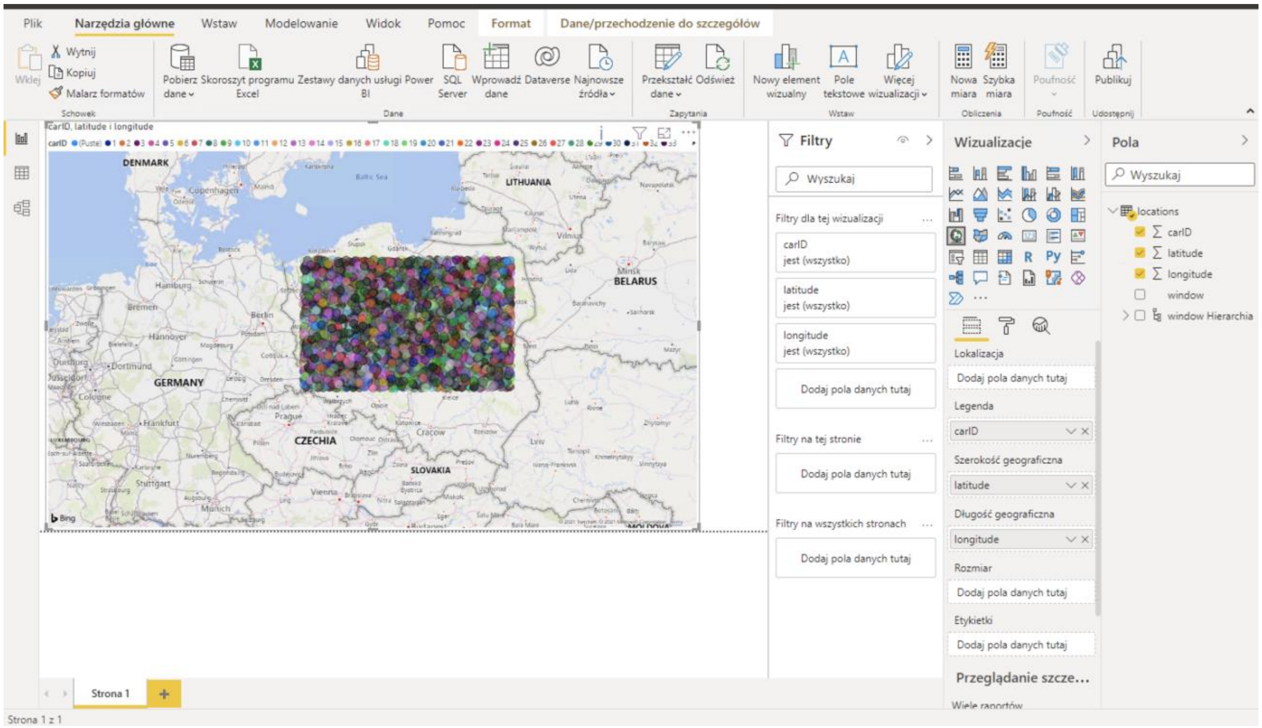
longitude
jest (wszystko)

Legenda: carID

Szerokość geograficzna: latitude

Długość geograficzna: longitude

3. See a location history of every car.



carID
jest (wszystko)

latitude
jest (wszystko)

longitude
jest (wszystko)

Legenda: carID

Szerokość geograficzna: latitude

Długość geograficzna: longitude

☑ carID
☑ latitude
☑ longitude

4. Try playing with different filters and optons in Power BI

Task 32: Publish your PowerBI report

In this section, you will learn how to publish your report into PowerBI service

1. From top menu click Publish.
2. Follow the publish wizard. If prompted for logging in, use your student credentials. Publish your report to **MyWorkspace** workspace.
3. After publishing succeed go to app.powerbi.com
4. From select MyWorkspace and open your report.
5. Use filters on the right. Display just first 6 cars on the map.