DAMG 7275
Advanced Database Management Systems

P4 submission

Project - Farm Environmental Monitoring System

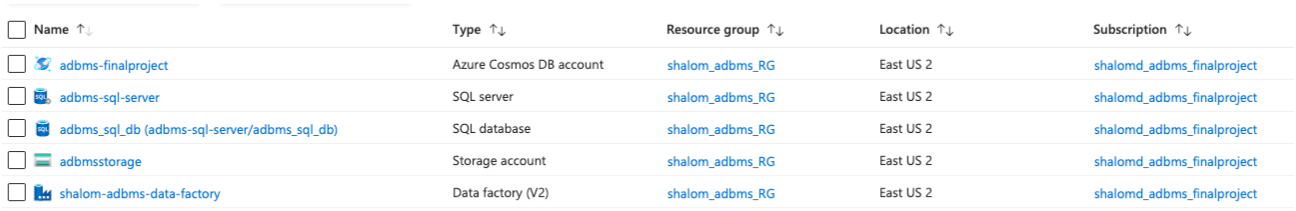
Project Group : 3

---

* Introduction :-

We have implemented our architecture diagram and pipeline on Azure cloud platform. Some of the services that we have used are :

- Azure Blob Storage
- Azure SQL Server
- Azure SQL database
- Azure Comsos DB
- Azure Data Factory

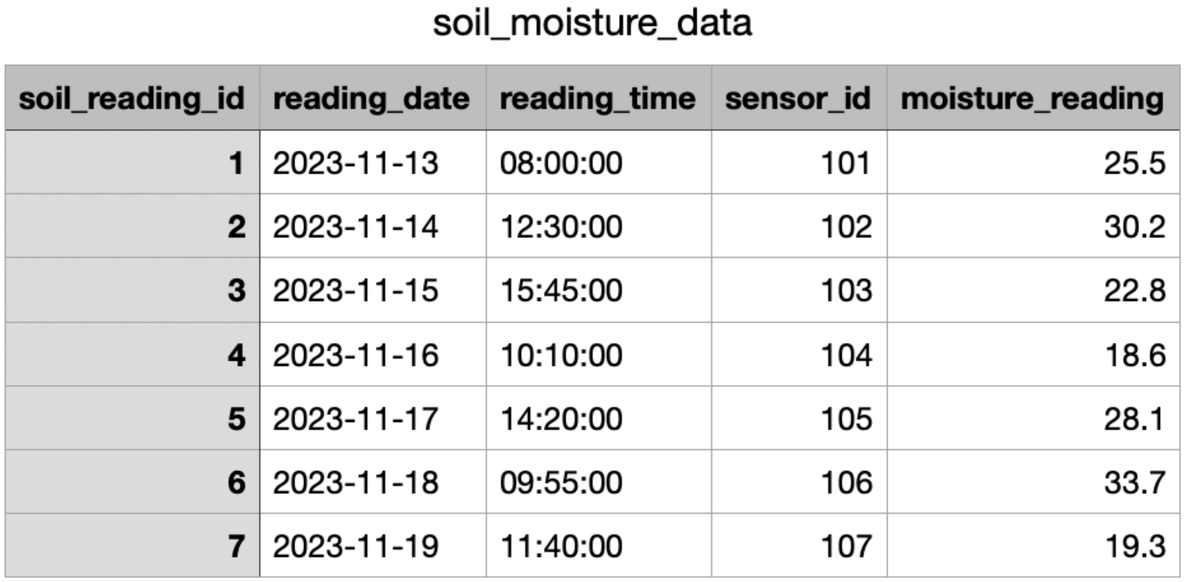


□ Name ↑↓	Type ↑↓	Resource group ↑↓	Location ↑↓	Subscription ↑↓
□ adbms-finalproject	Azure Cosmos DB account	shalom_adbms_RG	East US 2	shalomd_adbms_finalproject
□ adbms-sql-server	SQL server	shalom_adbms_RG	East US 2	shalomd_adbms_finalproject
□ adbms_sql_db (adbms-sql-server/adbms_sql_db)	SQL database	shalom_adbms_RG	East US 2	shalomd_adbms_finalproject
□ adbmsstorage	Storage account	shalom_adbms_RG	East US 2	shalomd_adbms_finalproject
□ shalom-adbms-data-factory	Data factory (V2)	shalom_adbms_RG	East US 2	shalomd_adbms_finalproject

* Dataset :-

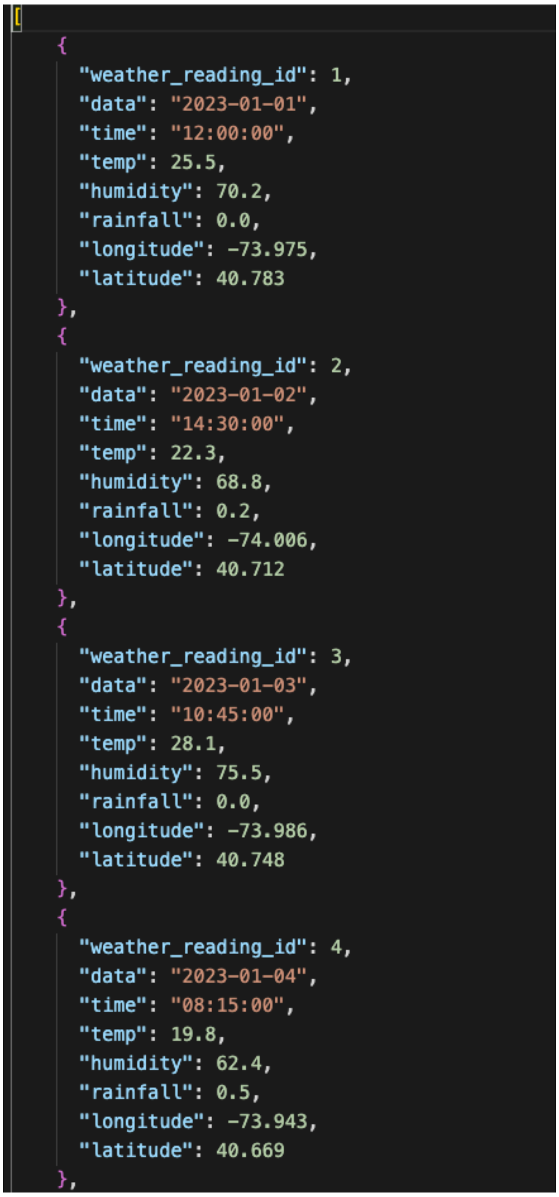
We have 3 primary sources of data for this project implementation :

- Weather data (JSON)
- Soil moisture data (CSV)
- Camera images (JPG)



soil_moisture_data

soil_reading_id	reading_date	reading_time	sensor_id	moisture_reading
1	2023-11-13	08:00:00	101	25.5
2	2023-11-14	12:30:00	102	30.2
3	2023-11-15	15:45:00	103	22.8
4	2023-11-16	10:10:00	104	18.6
5	2023-11-17	14:20:00	105	28.1
6	2023-11-18	09:55:00	106	33.7
7	2023-11-19	11:40:00	107	19.3



```json
[
    {
      "weather_reading_id": 1,
      "data": "2023-01-01",
      "time": "12:00:00",
      "temp": 25.5,
      "humidity": 70.2,
      "rainfall": 0.0,
      "longitude": -73.975,
      "latitude": 40.783
    },
    {
      "weather_reading_id": 2,
      "data": "2023-01-02",
      "time": "14:30:00",
      "temp": 22.3,
      "humidity": 68.8,
      "rainfall": 0.2,
      "longitude": -74.006,
      "latitude": 40.712
    },
    {
      "weather_reading_id": 3,
      "data": "2023-01-03",
      "time": "10:45:00",
      "temp": 28.1,
      "humidity": 75.5,
      "rainfall": 0.0,
      "longitude": -73.986,
      "latitude": 40.748
    },
    {
      "weather_reading_id": 4,
      "data": "2023-01-04",
      "time": "08:15:00",
      "temp": 19.8,
      "humidity": 62.4,
      "rainfall": 0.5,
      "longitude": -73.943,
      "latitude": 40.669
    },
```

* Data Auto-Refresh :-

For the P4 submission, we have implemented the automatic data refresh and update for the 3 data sources (mentioned above).

1) We are using the **‘triggers’** in Azure Data Factory to automatically run the pipeline to sync the pipeline jobs.

2) The CSV file (soil moisture data) & JSON file (weather data) will be manually updated for the sake of the project demo.

Note: *(In a real world scenario, these data sources would be populated by an API, or by IoT devices connected to Azure IoT hub, or with the help of real-time data streaming services)*

3) For the graph data model, we will manually update the data on our Cosmos db with Gremlin api account with the queries retrieved from the python script.

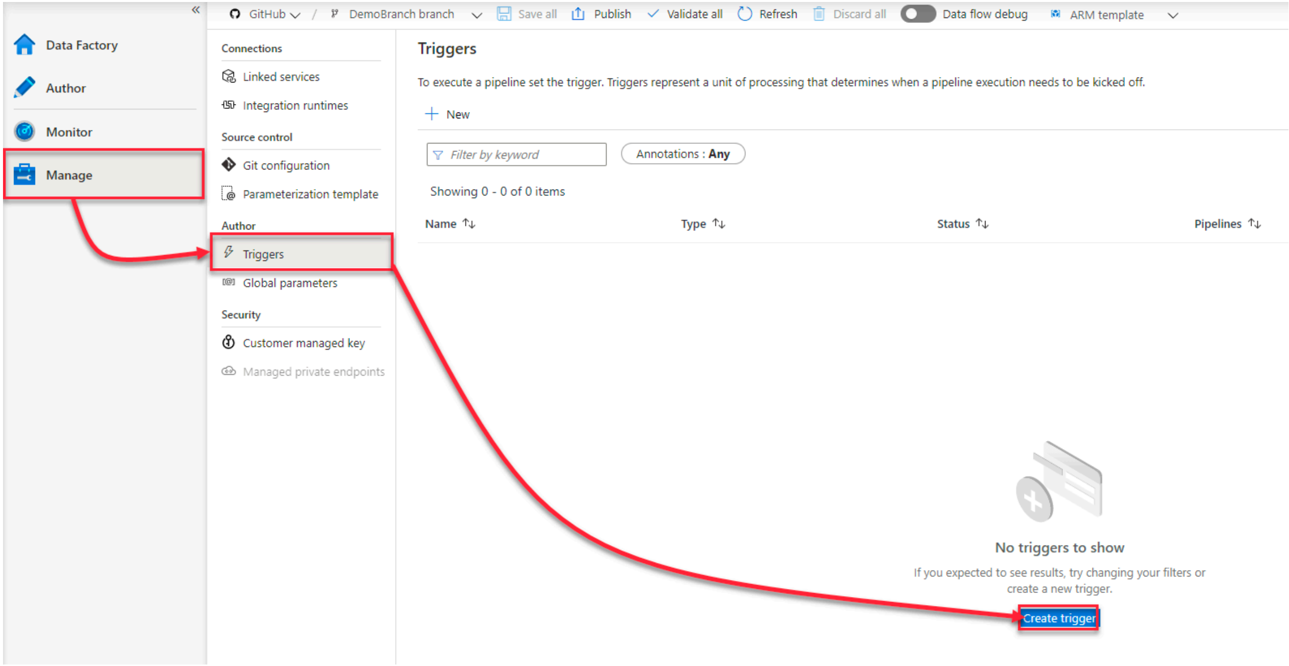
For (2) & (3), we can use ‘`az-cli`’ commands for azure blob storage to dynamically push latest files (cvs & json) to Azure on a periodic basis.

These commands can be run using **Azure Logic Apps** on periodically depending on updates to the files.

Note: *(We have not implemented Azure Logic Apps in this project demo to save costs. The files are pushed manually)*

* Implementation :-

Setup trigger on Azure Data Factory.



Triggers

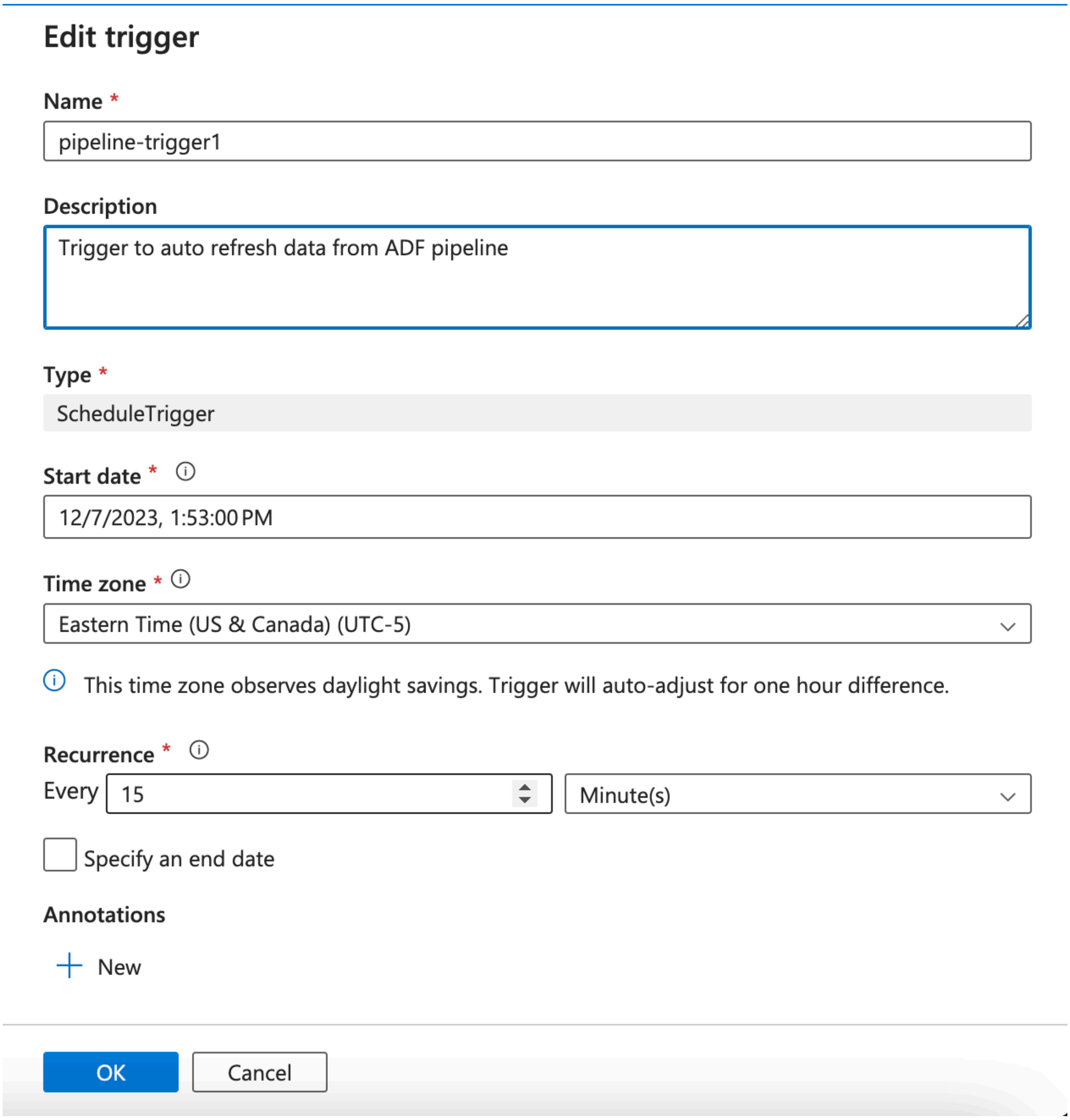
To execute a pipeline set the trigger. Triggers represent a unit of processing that determines when a pipeline execution needs to be kicked off.

Showing 0 - 0 of 0 items

No triggers to show

If you expected to see results, try changing your filters or create a new trigger.

Add trigger configuration and/or parameters for execution along with the recurrence.



Edit trigger

Name *

pipeline-trigger1

Description

Trigger to auto refresh data from ADF pipeline

Type *

ScheduleTrigger

Start date * ⓘ

12/7/2023, 1:53:00 PM

Time zone * ⓘ

Eastern Time (US & Canada) (UTC-5)

ⓘ This time zone observes daylight savings. Trigger will auto-adjust for one hour difference.

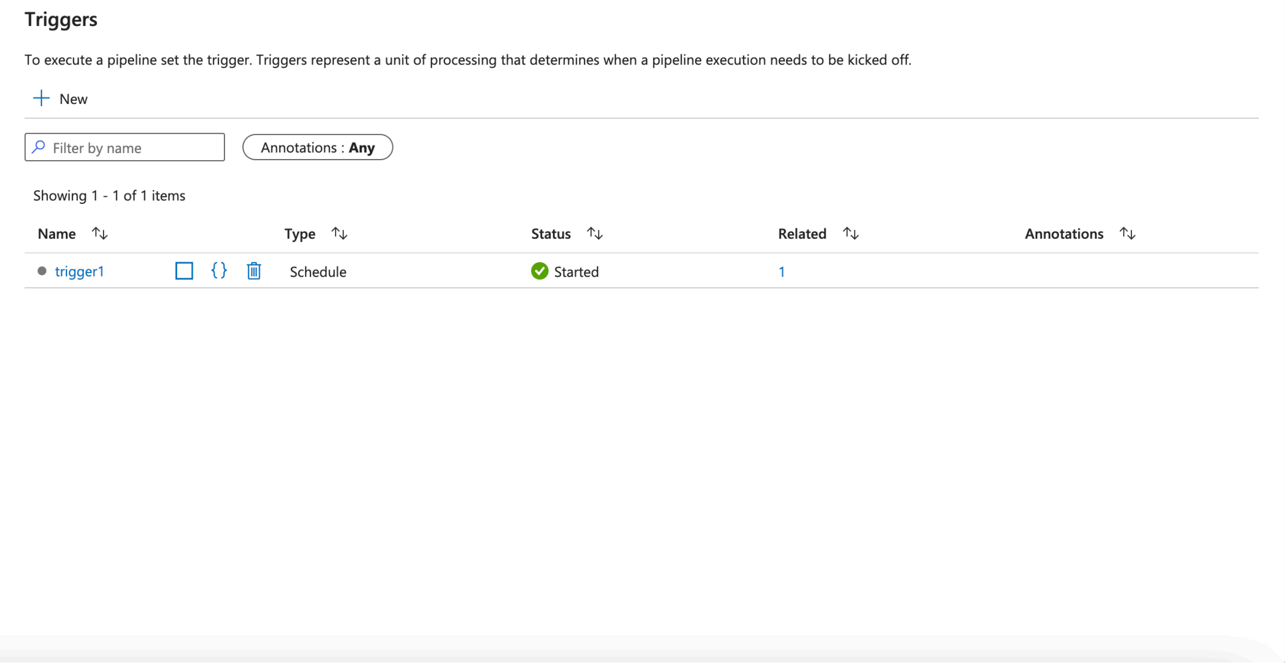
Recurrence * ⓘ

Every 15 Minute(s)

□ Specify an end date

Annotations

Once the trigger is created, you can check the trigger status based on the set schedule and monitor the logs.



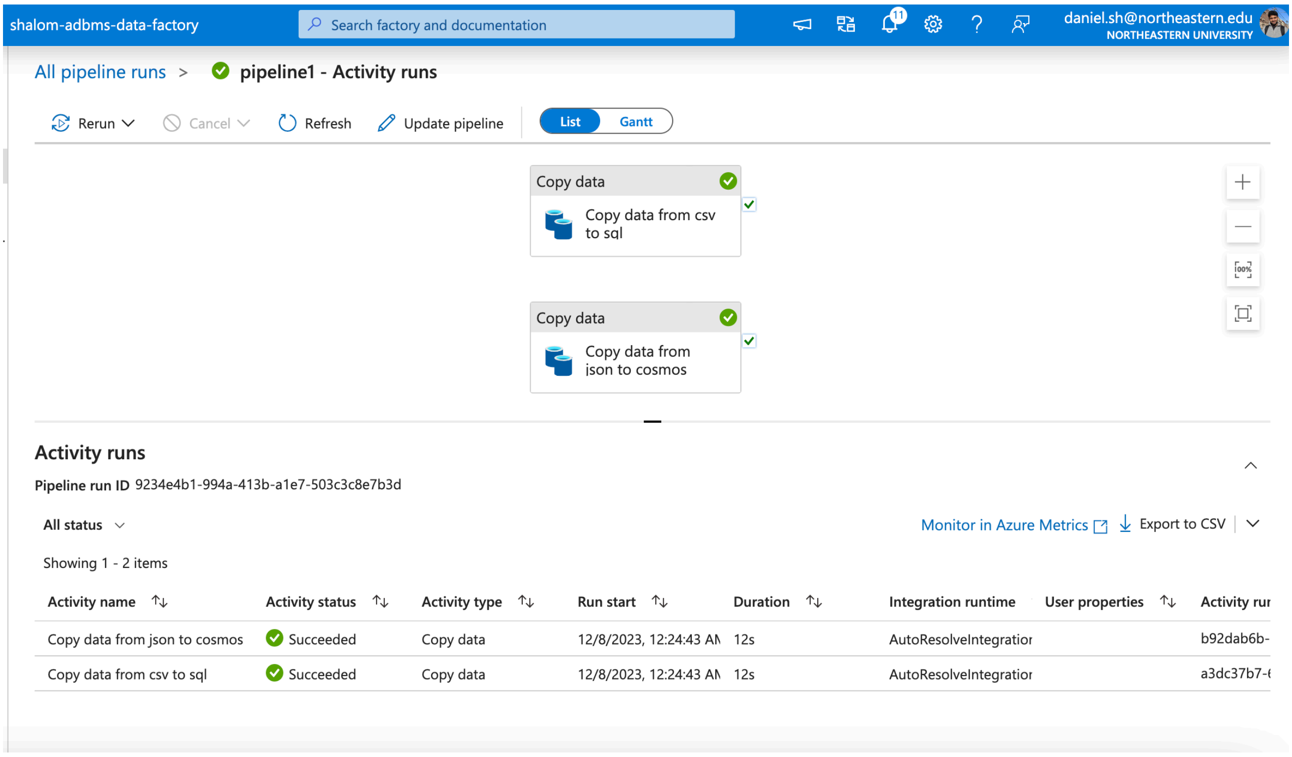
Triggers

To execute a pipeline set the trigger. Triggers represent a unit of processing that determines when a pipeline execution needs to be kicked off.

Showing 1 - 1 of 1 items

Name	Type	Status	Related	Annotations
trigger1	Schedule	✓ Started	1	

The trigger we set ran successfully and correctly imported the data as expected. Shown below is the activity run for the 2 pipeline jobs.



shalom-adbms-data-factory
Search factory and documentation
daniel.sh@northeastern.edu
NORTHEASTERN UNIVERSITY
All pipeline runs >
Rerun
Cancel
Refresh
Update pipeline
List
Gantt
Copy data
Copy data from csv to sql
Copy data
Copy data from json to cosmos

pipeline1 - Activity runs

Activity runs

Pipeline run ID 9234e4b1-994a-413b-a1e7-503c3c8e7b3d

Showing 1 - 2 items

Activity name	Activity status	Activity type	Run start	Duration	Integration runtime	User properties	Activity rur
Copy data from json to cosmos	✓ Succeeded	Copy data	12/8/2023, 12:24:43 AN	12s	AutoResolveIntegratior		b92dab6b-
Copy data from csv to sql	✓ Succeeded	Copy data	12/8/2023, 12:24:43 AN	12s	AutoResolveIntegratior		a3dc37b7-(

* Next steps & Future Scopes :-

1) Implement an application and dashboard to make sense of the data that we have used and for visualisations. (P5)

2) Implement az-cli & Azure logic apps to automate scripts based on CRON jobs instead of manual intervention.

3) Try and increase size of datasets for better visibility & analysis.

* Group members :-

1) Ankita Patil
2) Aditya Pande
3) Keshni Mulrajani
4) Sachit Wagle
5) Shalom Daniel

****