

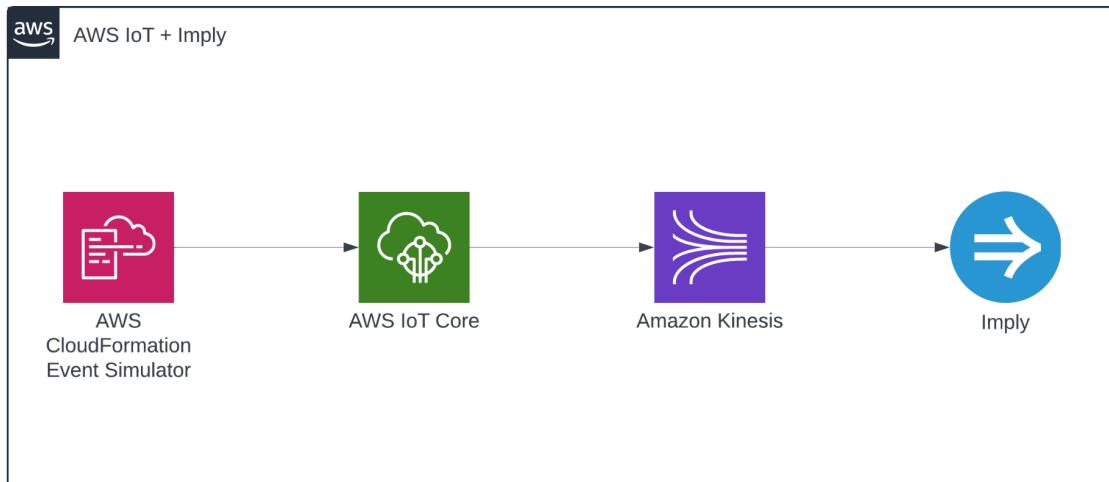
IoT Workshop for EV RT Analytics

Prerequisites

1. Make sure the conductors have your email addresses
2. Make sure that you are added to a Polaris project for the workshop. See inbox for the invite.
3. Make sure you have the AWS Kinesis IAM ARN from the workshop organizers.
4. Familiarize yourself with the architecture for the workshop and also read through the steps before attempting them.

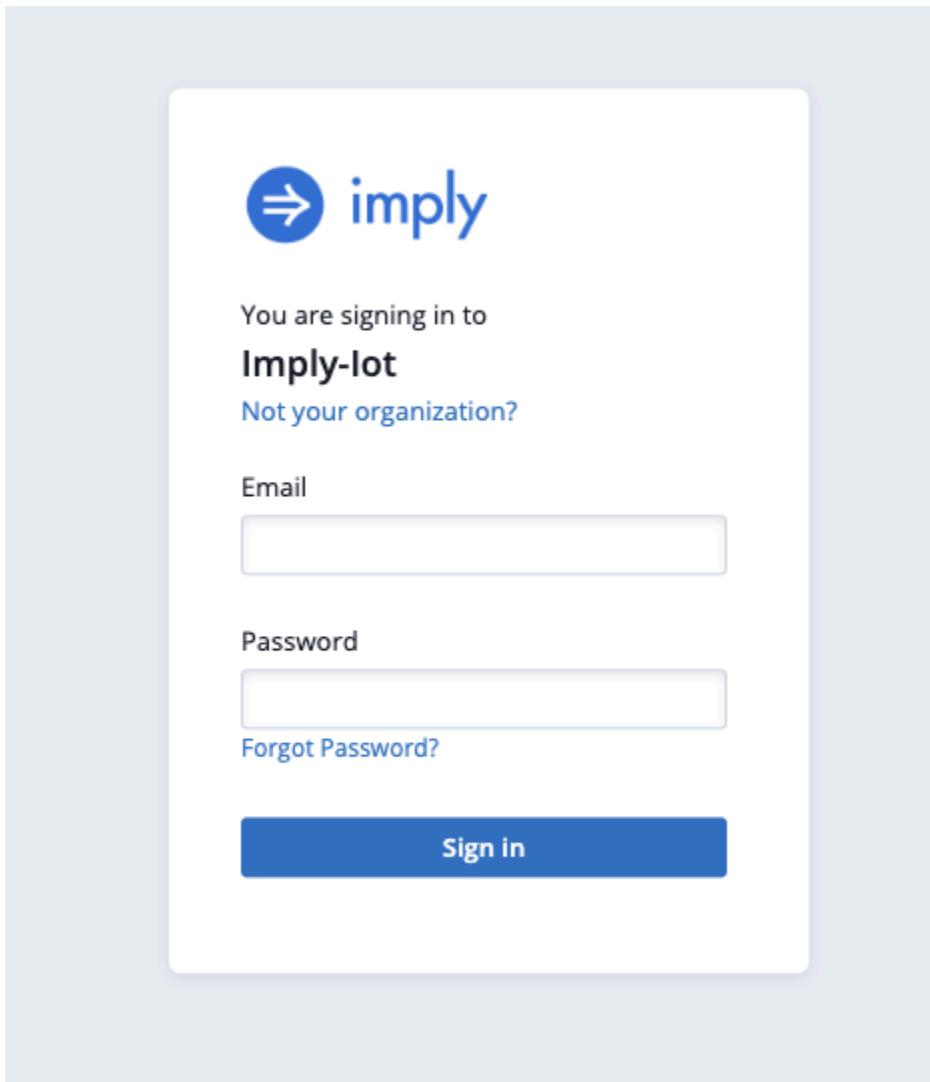
Architecture Walkthrough

The below picture provides an overview of the architecture that has been used to build this workshop. More information about the architecture can be found [here](#).



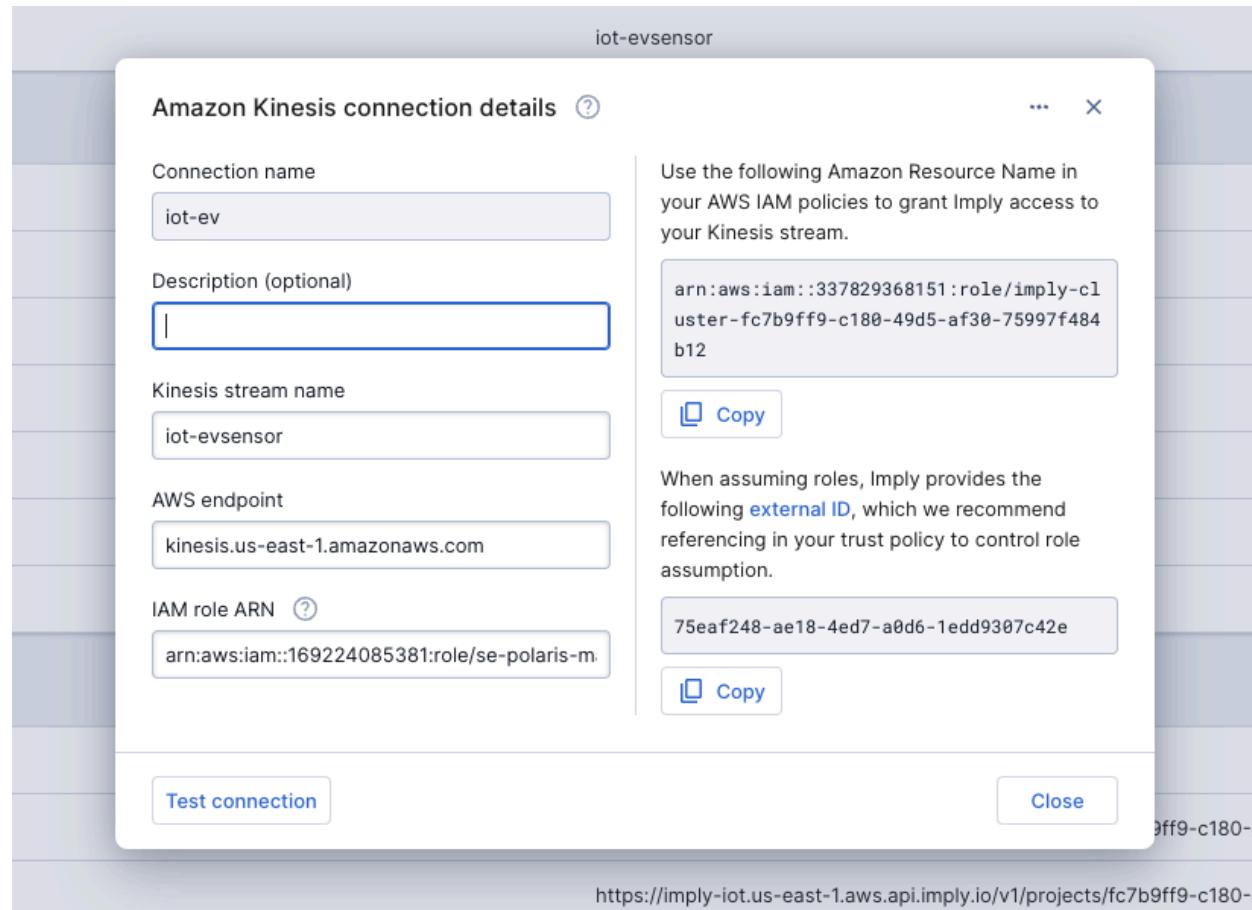
Connect Imply Polaris to Kinesis, and Create a Table

1. Sign into your Imply account



2. Click on “Sources” on the left nav, then at the top right click “Create source”, then “Amazon Kinesis”
Sources → Create Source → Amazon Kinesis

- For a comprehensive guide, here are the docs.
<https://docs.imply.io/polaris/ingestion-guide-kinesis>



Since the connection name has to be unique, please suffix your initials at the end of the “connection name” in the “new connection” pop-up window.

Use the details that were shared with you by the organizers for connection to AWS Kinesis. For example:

Unset

```
Kinesis stream name: iot-evsensor
AWS endpoint: kinesis.us-east-1.amazonaws.com
IAM Role ARN: arn:aws:iam::169224085381:role/se-polaris-marc
```

Click “Test connection” and if successful, click “Create connection”.

3. Create a table in Imply Polaris by clicking on the “Tables” tab and click “Create table” at the top right. Choose Detail and Flexible. (see tooltip and [docs](#) for more info). Since the table name has to be unique, please suffix your initials at the end of the “connection name” in the “new connection” pop-up window.

Create new table

i Name and table type cannot be changed once a table is created.

Name
iot-test

Table type *?*
Detail Aggregate

Schema mode *?*
Flexible Strict

Next

- Click the “Load data” button on the top right corner and then click “Insert data”. Select “Amazon Kinesis” on the left pane, and then select the source that you created in the previous step. Then click “Next”.

iot-test / Insert data

Step 1 Select source

Polaris only supports one input format per ingestion job. [Learn more](#) about supported data formats.

Switch to code editor *Next*

BATCH		STREAMING	
Files	Existing tables	Amazon Kinesis	Confluent Cloud
Kafka Connector	Kafka/MSK	Push	

+ New connection

Name	Stream name	Active jobs
auto-simulation	IOT-demo	1

Q Search

- Polaris will auto identify the schema, click “Continue”. If the schema is not identified, please explicitly choose JSON as the “input format”.

imply

auto-sim / Insert data

Step 2 Parse data

Verify the input format and fields below. You can correct data types and add any missing input fields.

Input format JSON

String timestamp	String trip_id	String VIN	Long brake	Long steeringWheelAngle	Double torqueAtTransmission	Double engineSpeed	Double vehicleSpeed	Double acceleration	String parkingBrakeStatus	String brakePedalStatus
2024-02-08 20:40:37.717...	mfSibJ4WYWyosLZ0l...	JNRZVHNP0D2XTX4Y...	0	0	345.7	3597.51	87.61	-0.5446	false	false
2024-02-08 20:40:38.033...	VSG9GDrRpBfrdayPf...	JUBUJMLCXATOJVVE	0	-39.1	2588.59	31.35	-0.544	false	false	
2024-02-08 20:40:38.052...	AuKIBgmaMW8w7ZWg...	VVQ4CZQHE8V35H8...	0	0	261.4	3161.95	77.24	0.0833	false	false
2024-02-08 20:40:38.548...	Pugbr9Hy05dDTQgQ...	KDWBC39EIOR2HQX...	0	-16.9	2323.89	28.37	-0.013	false	false	
2024-02-08 20:40:38.717...	mfSibJ4WYWyosLZ0l...	JNRZVHNP0D2XTX4Y...	0	0	346.3	3579.82	87.22	-0.4319	false	false
2024-02-08 20:40:39.052...	AuKIBgmaMW8w7ZWg...	VVQ4CZQHE8V35H8...	0	0	261.3	3164.56	77.3	0.0692	false	false
2024-02-08 20:40:39.548...	VSG9GDrRpBfrdayPf...	JUBUJMLCXATOJVVE	0	56.6	2634.62	32.97	-0.5049	false	false	
2024-02-08 20:40:39.548...	Pugbr9Hy05dDTQgQ...	KDWBC39EIOR2HQX...	0	-16.9	2322.89	28.35	-0.013	false	false	
2024-02-08 20:40:39.718...	mfSibJ4WYWyosLZ0l...	JNRZVHNP0D2XTX4Y...	0	0	346.8	3565.55	86.91	-0.3485	false	false
2024-02-08 20:40:40.034...	VSG9GDrRpBfrdayPf...	JUBUJMLCXATOJVVE	0	0	52.6	2764.41	34.51	1.6139	false	false
2024-02-08 20:40:40.068...	AuKIBgmaMW8w7ZWg...	VVQ4CZQHE8V35H8...	0	0	261.2	3166.76	77.35	0.0585	false	false
2024-02-08 20:40:40.549...	Pugbr9Hy05dDTQgQ...	KDWBC39EIOR2HQX...	0	-16.9	2321.93	28.34	-0.0182	false	false	
2024-02-08 20:40:40.718...	mfSibJ4WYWyosLZ0l...	JNRZVHNP0D2XTX4Y...	0	0	347.2	3554	86.66	-0.282	false	false
2024-02-08 20:40:41.034...	VSG9GDrRpBfrdayPf...	JUBUJMLCXATOJVVE	0	0	48.7	2886.37	35.95	1.5375	false	false
2024-02-08 20:40:41.069...	AuKIBgmaMW8w7ZWg...	VVQ4CZQHE8V35H8...	0	0	261.2	3168.62	77.39	0.0494	false	false
2024-02-08 20:40:41.549...	Pugbr9Hy05dDTQgQ...	KDWBC39EIOR2HQX...	0	-16.9	2321.03	28.33	-0.0113	false	false	
2024-02-08 20:40:41.719...	mfSibJ4WYWyosLZ0l...	JNRZVHNP0D2XTX4Y...	0	0	347.5	3544.63	86.45	-0.2286	false	false
2024-02-08 20:40:42.034...	VSG9GDrRpBfrdayPf...	JUBUJMLCXATOJVVE	0	0	85.6	3037.74	38.2	1.4439	false	false
2024-02-08 20:40:42.068...	AuKIBgmaMW8w7ZWg...	VVQ4CZQHE8V35H8...	0	0	292.6	3179.41	77.86	0.0417	false	false
2024-02-08 20:40:42.550...	Pugbr9Hy05dDTQgQ...	KDWBC39EIOR2HQX...	0	0	-43.8	2295.6	27.72	-0.0108	false	false
2024-02-08 20:40:42.719...	mfSibJ4WYWyosLZ0l...	JNRZVHNP0D2XTX4Y...	0	0	410.7	3555.5	87.14	0.2654	false	false
2024-02-08 20:40:43.033...	VSG9GDrRpBfrdayPf...	JUBUJMLCXATOJVVE	0	0	0	0	0	2.2461	false	false
2024-02-08 20:40:43.069...	AuKIBgmaMW8w7ZWa...	VVQ4CZQHE8V35H8...	0	0	0	0	0	0.469	false	false

Polaris accepts streaming data having post-transform event timestamps within 30 days of ingestion time.

6. You can customize the table further, but for now in this test click “Start Ingestion”

imply

auto-sim / Insert data

Step 3 Map source to table

Define the mapping between input fields and table columns.

Manage inputs Auto-discovery Deny list + Add column Filter Starting offset Beginning Partitioning Show column Search

Filter out late data after 30 days

Timestamp	Auto trip_id	Auto VIN	Auto brake "brake"	Auto steeringWheelAngle "steeringWheelAngle"	Auto torqueAtTransmission "torqueAtTransmission"	Auto engineSpeed "engineSpeed"	Auto vehicleSpeed "vehicleSpeed"	Auto acceleration "acceleration"	Auto parkingBrakeStatus "parkingBrakeStatus"	Auto brakePedalStatus "brakePedalStatus"	Auto transmissionGearPosition "transmissionGearPos..."	Auto gearLevel "gearLevel"
2024-02-08T20:40:37.717Z	mfSibJ4WYWyosLZ0l...	JNRZVHNP0D2XTX4Y...	0	0	345.7	3597.51	87.61	-0.5446	false	false	fourth	drive
2024-02-08T20:40:38.03...	VSG9GDrRpBfrdayPf...	JUBUJMLCXATOJVVE	0	-39.1	2588.59	31.35	-0.544	false	false	second	drive	
2024-02-08T20:40:38.05...	AuKIBgmaMW8w7ZWg...	VVQ4CZQHE8V35H8...	0	0	261.4	3161.95	77.24	0.0833	false	false	fourth	drive
2024-02-08T20:40:38.54...	Pugbr9Hy05dDTQgQ...	KDWBC39EIOR2HQX...	0	-16.9	2323.89	28.37	-0.013	false	false	second	drive	
2024-02-08T20:40:38.717Z	mfSibJ4WYWyosLZ0l...	JNRZVHNP0D2XTX4Y...	0	0	346.3	3579.82	87.22	-0.4319	false	false	fourth	drive
2024-02-08T20:40:39.05...	AuKIBgmaMW8w7ZWg...	VVQ4CZQHE8V35H8...	0	0	261.3	3164.56	77.3	0.0692	false	false	fourth	drive
2024-02-08T20:40:39.54...	VSG9GDrRpBfrdayPf...	JUBUJMLCXATOJVVE	0	56.6	2634.62	32.97	-0.5049	false	false	second	drive	
2024-02-08T20:40:39.54...	Pugbr9Hy05dDTQgQ...	KDWBC39EIOR2HQX...	0	-16.9	2322.89	28.35	-0.013	false	false	second	drive	
2024-02-08T20:40:39.717Z	mfSibJ4WYWyosLZ0l...	JNRZVHNP0D2XTX4Y...	0	0	346.8	3565.55	86.91	-0.3485	false	false	fourth	drive
2024-02-08T20:40:40.03...	VSG9GDrRpBfrdayPf...	JUBUJMLCXATOJVVE	0	0	52.6	2764.41	34.51	1.6139	false	false	second	drive
2024-02-08T20:40:40.06...	AuKIBgmaMW8w7ZWg...	VVQ4CZQHE8V35H8...	0	0	261.2	3166.76	77.35	0.0585	false	false	fourth	drive
2024-02-08T20:40:40.54...	Pugbr9Hy05dDTQgQ...	KDWBC39EIOR2HQX...	0	-16.9	2321.93	28.34	-0.0182	false	false	second	drive	
2024-02-08T20:40:40.717Z	mfSibJ4WYWyosLZ0l...	JNRZVHNP0D2XTX4Y...	0	0	347.2	3554	86.66	-0.282	false	false	fourth	drive
2024-02-08T20:40:41.03...	VSG9GDrRpBfrdayPf...	JUBUJMLCXATOJVVE	0	0	48.7	2886.37	35.95	1.5375	false	false	second	drive
2024-02-08T20:40:41.06...	AuKIBgmaMW8w7ZWg...	VVQ4CZQHE8V35H8...	0	0	261.2	3168.62	77.39	0.0494	false	false	fourth	drive
2024-02-08T20:40:41.54...	Pugbr9Hy05dDTQgQ...	KDWBC39EIOR2HQX...	0	-16.9	2321.03	28.33	-0.0113	false	false	second	drive	
2024-02-08T20:40:41.717Z	mfSibJ4WYWyosLZ0l...	JNRZVHNP0D2XTX4Y...	0	0	347.5	3544.63	86.45	-0.2286	false	false	fourth	drive
2024-02-08T20:40:43.03...	VSG9GDrRpBfrdayPf...	JUBUJMLCXATOJVVE	0	0	0	0	0	2.2461	false	false	second	drive
2024-02-08T20:40:43.06...	AuKIBgmaMW8w7ZWa...	VVQ4CZQHE8V35H8...	0	0	0	0	0	0.469	false	false	second	drive

Confirm primary timestamp
Polaris detected one time candidate and mapped it to __time

Go to column

Polaris accepts streaming data having post-transform event timestamps within 30 days of ingestion time.

7. Your table is now created, congratulations!!! You have created an end-to-end data application pipeline. You can now query your IoT event data from the UI or via API.

The screenshot shows the imply-iot interface with the 'auto-sim' table selected. The top navigation bar includes 'imply-iot', 'Home', 'Collections', 'DATA', 'Tables', 'Sources', 'Jobs', 'SQL', 'Analytics', 'Data cubes', 'Dashboards', 'Alerts', 'Reports', 'Embedding', 'Monitoring', 'User queries', 'Streaming', and 'Detailed metrics'. The 'Tables' section is active. The 'auto-sim' table details are shown: Query availability (Available), Type (Detail), Mode (Flexible), Date range (2024-02-08 to 2024-02-10), Size (2.09 MB), Number of rows (23,788), and Columns (19). A 'Visualize your data' button is present. The 'Jobs' section shows a single job named 'auto-simulation' in 'Running' status with 25 days duration, created on Feb 8, 2024, at 1:51 PM.

Timestamp	Auto.vehicleSpeed	Auto.trip_id	Auto.fuelLevel	Auto.brakePedalStatus	Auto.odometer	Auto._id_	Auto.fuelConsumedSinceRest	Auto.acceleration
2024-02-08T20:40:37.483Z	98.05	uTEsgaeqlqWtp1zM...	99.67	0	4.064	K9wdWy3	0.133222	-0.0
2024-02-08T20:40:37.717Z	87.61	ml5ibJ4WYWosLZ0L...	99.57	0	6.264	K9wdWy7	0.17345	-0.5
2024-02-08T20:40:37.949Z	0	20BQkS0SmGbg7pm...	99.88	0	1.624	K9wdWy9	0.049114	0
2024-02-08T20:40:38.03...	31.35	V99GkDrRphdayPf...	99.91	0	2.968	K9wdWy0	0.03767	-0.5
2024-02-08T20:40:38.05...	77.24	AufBgamW8w7ZWg...	99.84	0	4.024	K9wdWy1	0.063843	0.08
2024-02-08T20:40:38.43...	0	EzJepqzUFTzolykL...	99.78	0	2.816	K9wdWy2	0.086061	0
2024-02-08T20:40:38.54...	28.37	PugbrhNy0sdDTQgQ...	99.91	0	2.7	K9wdWy4	0.037449	-0.0
2024-02-08T20:40:38.65...	8.59	AwLW4w6Sz6-PTB_T...	99.81	0	4.542	K9wdWy5	0.076158	-0.7
2024-02-08T20:40:38.66...	62.96	HHUJ-pCmOyOjPP...	99.72	0	5.218	K9wdWy6	0.112801	0.22
2024-02-08T20:40:38.717Z	87.22	ml5ibJ4WYWosLZ0L...	99.57	0	6.288	K9wdWy7	0.174132	-0.4
2024-02-08T20:40:38.813Z	90.5	sn7iuTMn1EOHfBaQ...	99.54	0	6.428	K9wdWy8	0.182817	0.18
2024-02-08T20:40:38.95...	0	20BQkS0SmGbg7pm...	99.88	0	1.624	K9wdWy9	0.049114	0

8. Lets query the table by following the below steps:

- From the left pane, click on “SQL”. Then click the newly created table name and choose the first query option from the pop up.
 - i. SELECT ... columns ... FROM iot-ev
- Click on “Run”.

The screenshot shows the imply-iot interface with the 'SQL' section selected. The left sidebar includes 'Home', 'Collections', 'DATA', 'Tables', 'Sources', 'Jobs', 'SQL', 'Analytics', 'Data cubes', 'Dashboards', 'Alerts', 'Reports', 'Embedding', 'Monitoring', 'User queries', 'Streaming', and 'Detailed metrics'. The 'SQL' section is active. In the center, there's a search bar with 'druid' and a dropdown menu listing various tables: auto-sim, cfilt-demo-dubai-dff, distributech-2024-demo, dtest, du_agg, emirates_demo, flightdata, flightdata-1, imply-iot, IImply-iot-ewp, iot-agg-cfilt, and iot-ev. Below the search bar is a 'Tab 1' tab with the following query:

```

1  SELECT
2    "__time",
3    "fault_properties",
4    "charging_properties",
5    "torque_achieved",
6    "fleet",
7    "_subtype",
8    "event_timing",
9    "_type",
10   "_id_",
11   "car_ownership_id",
12   "_sed",
13   "temperature_IGBT_A",
14   "motor_temperature",
15   "torque_available_motoring",
16   "temperature_IGBT_C",
17   "temperature_IGBT_B",
18   "VIN",
19   "updated",
20   "tenant".

```

Below the query are several suggestions:

- ↳ SELECT ...columns... FROM iot-ev
- ↳ SELECT * FROM iot-ev
- ↳ SELECT COUNT(*) AS "Count" FROM iot-ev
- ↳ SELECT MIN(__time), MAX(__time) FROM iot-ev
- ↳ Copy: "iot-ev"

- Click on ‘+’ above the query editor pane to get a new tab.
- Below are some example queries you can test in the SQL UI:

Unset

```
### In English, this query answers "For each hour, for each tenant, how many events showed the motor temperature outside of normal operating range?"
```

```
SELECT
    TIME_FLOOR(CAST(t."__time" AS TIMESTAMP), 'PT1H', NULL, 'Etc/UTC') AS
    "__time",
    "tenant",
    (COUNT(*)) AS "count"
FROM "iot-ev" AS t
WHERE ((0 <= CAST(t."motor_temperature" AS DOUBLE) AND
CAST(t."motor_temperature" AS DOUBLE) < 50)) IS NOT TRUE
GROUP BY 1,2
```

```
## How many events came in from different fleet categories each hour?
```

```
SELECT
    TIME_FLOOR(CAST(t."__time" AS TIMESTAMP), 'PT1H', NULL, 'Etc/UTC') AS
    "__time",
    CAST(t."fleet" AS VARCHAR) AS "fleet",
    (COUNT(*)) AS "count"
FROM "iot-ev" AS t
WHERE CAST(t."fleet" AS VARCHAR) IN ('Fleet Van', 'Passenger Car', 'Fleet
Truck', 'Passenger Truck', 'Fleet Car')
GROUP BY 1, 2
```

The screenshot shows the Imply Data Platform interface. On the left, there is a navigation sidebar with various sections like Home, Collections, DATA, Tables, Sources, Jobs, SQL, ANALYTICS, Data cubes, Dashboards, Alerts, Reports, Embedding, MONITORING, User queries, Streaming, Detailed metrics, and Server. The SQL section is currently selected.

In the main area, there are two tabs: Tab 1 and Tab 2. Tab 1 contains the following SQL query:

```

1 SELECT
2   TIME_FLOOR(CAST(`__time` AS TIMESTAMP), 'PT1H'), NULL, 'Etc/UTC') AS "__time",
3   "tenant" AS "tenant"
4   COUNT(*) AS "count"
5 FROM `iot-ev` AS t
6 WHERE ((0 <= CAST(t.`motor_temperature` AS DOUBLE) AND CAST(t.`motor_temperature` AS DOUBLE) < 50)) IS NOT TRUE
7 GROUP BY 1,2

```

Tab 2 displays the results of the query:

__time	tenant	count
2024-03-27T15:00:00.000Z	rivian	17,224
2024-03-27T16:00:00.000Z	rivian	24,017
2024-03-27T17:00:00.000Z	rivian	23,901
2024-03-27T18:00:00.000Z	rivian	23,990
2024-03-27T19:00:00.000Z	rivian	23,950
2024-03-27T20:00:00.000Z	rivian	24,025
2024-03-27T21:00:00.000Z	rivian	23,940

At the bottom right of the results table, it says "9 results in 0.22s".

Real time Analytics & Dashboarding

1. Creating a data cube can be done in just a couple short steps:
 - a. On the left menu, click “Data Cubes”, then at the top right “Create data cube”

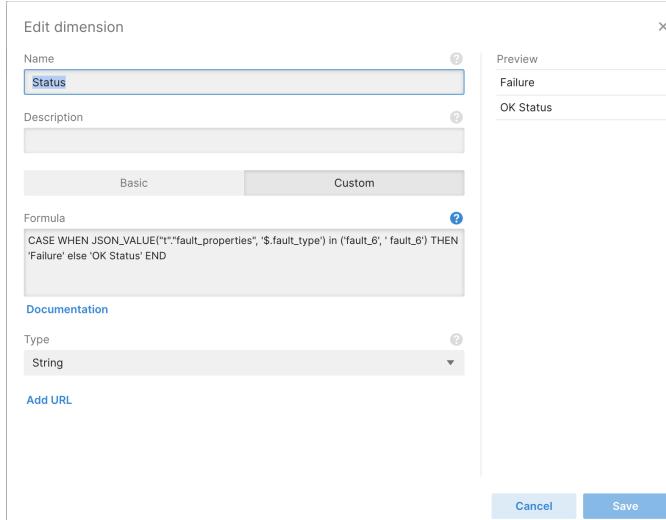
The screenshot shows the Imply Data Platform interface with the Data cubes section selected in the left sidebar. The main area displays a list of existing data cubes:

Name	Table	Data assets	Edited
Iot Ev - Old	iot-ev	25 dimensions, 11 measures	Mar 27, 2024, 4:51 PM
Iot Ev	iot-ev	24 dimensions, 12 measures	Mar 27, 2024, 4:50 PM
Iot Webinar	iot-webinar	27 dimensions, 13 measures	Mar 27, 2024, 3:55 AM
Auto Sim	auto-sim	26 dimensions, 14 measures	Mar 25, 2024, 3:50 AM
Vehicle Data	vehicle-data	26 dimensions, 14 measures	Mar 20, 2024, 10:49 AM
Tillmanfiber Demo	tillmanfiber	26 dimensions, 14 measures	Mar 1, 2024, 9:10 AM
Distribitech 2024 Demo	distribitech-2024	26 dimensions, 14 measures	Feb 25, 2024, 4:55 PM
Tillmanfiber	tillmanfiber	26 dimensions, 14 measures	Feb 9, 2024, 8:56 AM
Imply IoT EON	imply-iot-eon	26 dimensions, 14 measures	Jan 19, 2024, 4:09 AM
Iot Table 01	iot-table-01	27 dimensions, 13 measures	Dec 8, 2023, 6:40 AM
Emirates Demo	emirates-demo	26 dimensions, 14 measures	Nov 2, 2023, 3:54 AM
Test	dtest	9 dimensions, 6 measures	Oct 27, 2023, 12:52 PM
Noah Test	NoahTest	7 dimensions, 1 measure	Oct 26, 2023, 4:13 PM
Imply IoT	imply-iot	27 dimensions, 14 measures	Oct 25, 2023, 4:46 AM
Imply IoT	imply-iot	27 dimensions, 14 measures	Oct 25, 2023, 3:17 AM
Plants Streams	plants-streams	19 dimensions, 18 measures	Oct 17, 2023, 8:20 AM
Imply IoT App	imply-iot-app	14 dimensions, 15 measures	Oct 6, 2023, 10:18 AM

A modal window titled "Create new data cube" is open, showing a dropdown menu with "From table" and "From SQL query" options. A search bar contains the text "iot-ev", and a dropdown list shows "iot-ev" as a suggestion. At the bottom of the modal are "Cancel" and "Next: Create data cube" buttons.

- b. From the drop down select your new table we created “iot-test”, and click “Next: Create data cube”. Leave the “Auto-fill dimensions and measures” checked
- c. Now let’s modify the **status** dimension and replace the default expression with this new one, it will help to simulate a failure status only when **fault_type** = ‘**fault_6**’:

```
CASE WHEN JSON_VALUE("t"."fault_properties", '$.fault_type') in ('fault_6', 'fault_6') THEN 'Failure' else 'OK Status' END
```



- d. Now let’s modify the **fault_type** dimension and replace the default expression with this new one in order to decode the fault type descriptions :

```
CASE WHEN JSON_VALUE("t"."fault_properties", '$.fault_type') = 'fault_1' THEN 'charging level 1' WHEN JSON_VALUE("t"."fault_properties", '$.fault_type') = 'fault_2' THEN 'charging level 2' WHEN JSON_VALUE("t"."fault_properties", '$.fault_type') = 'fault_3' THEN 'fast charging' WHEN JSON_VALUE("t"."fault_properties", '$.fault_type') = 'fault_4' THEN 'low battery' WHEN JSON_VALUE("t"."fault_properties", '$.fault_type') = 'fault_5' THEN 'charging issue' WHEN JSON_VALUE("t"."fault_properties", '$.fault_type') = 'fault_6' THEN 'engine overheating' END
```

Edit dimension

X

Name

Fault Type



Description



Basic

Custom

Formula



```
CASE WHEN JSON_VALUE("t"."fault_properties", '$.fault_type') = 'fault_1' THEN 'charging level 1' WHEN JSON_VALUE("t"."fault_properties", '$.fault_type') = 'fault_2' THEN 'charging level 2' WHEN JSON_VALUE("t"."fault_properties", '$.fault_type') = 'fault_3' THEN 'fast charging' WHEN JSON_VALUE("t"."fault_properties", '$.fault_type') = 'fault_4' THEN 'low
```

Documentation

Type



String



Add URL

Preview

charging issue

charging level 1

charging level 2

engine overheating

fast charging

low battery

Cancel

Save

e. Now, let's rename the **subtype** dimension into **Firmware version** :

Edit dimension

Name	<input type="text" value="Firmware version"/>	Preview
Description	<input type="text"/>	v2023.11 v2023.12 v2024.02 v2024.04 v2024.06 v2024.08
Column	<input type="button" value="Basic"/> <input type="button" value="Custom"/>	
_subtype	<input type="button"/>	
Type	<input type="button" value="String"/>	
Add URL		
		<input type="button" value="Cancel"/> <input type="button" value="Save"/>

f. Now let's add/modify some custom measures to the datacube.

← IoT Ev Final ⚙

Edit data cube

<input checked="" type="checkbox"/> General	Measures	<input type="button"/> Suggestions (3) <input type="button"/> New measure
<input type="checkbox"/> Dimensions		
<input type="checkbox"/> Measures		
<input type="checkbox"/> Access		
<input type="checkbox"/> Access filters		
... Advanced		
		<input type="button"/> Cancel <input type="button"/> Save
	<ul style="list-style-type: none"> <input type="checkbox"/> Number of Events COUNT(*) <input type="checkbox"/> Event Timing SUM(L."Event_timing") <input type="checkbox"/> Defect Ratio $100.0 * \text{COUNT}(\text{FILTER (WHERE CASE WHEN JSON_VALUE("fault_properties", "\$fault_type") in ("fault_B", "fault_B') THEN '1' else '0' END)}) / \text{COUNT}(\text{*})$ <input type="checkbox"/> Torque <ul style="list-style-type: none"> <input type="checkbox"/> Torque Achieved (Avg) AVG(L."torque_achieved") <input type="checkbox"/> Torque Achieved (Max) MAX(L."torque_achieved") <input type="checkbox"/> Torque Available Regen (Avg) AVG(L."torque_available_regen") <input type="checkbox"/> Torque Available Regen (Max) MAX(L."torque_available_regen") <input type="checkbox"/> Torque Available Motoring (Avg) AVG(L."torque_available_motoring") <input type="checkbox"/> Torque Available Motoring (Max) MAX(L."torque_available_motoring") <input type="checkbox"/> Temperature <ul style="list-style-type: none"> <input type="checkbox"/> Motor Temperature (Avg) AVG(L."motor_temperature") <input type="checkbox"/> Motor Temperature (Max) MAX(L."motor_temperature") 	

- for each **Torque** and **Temperature** measure, only keep/add an **AVG** and a **MAX** expression :

		Motor Temperature (Avg) AVG(t."motor_temperature")
↳	Torque Achieved (Avg) AVG(t."torque_achieved")	Motor Temperature (Max) MAX(t."motor_temperature"-100)
↳	Torque Achieved (Max) MAX(t."torque_achieved")	Temperature IGBT A (Avg) AVG(t."temperature_IGBT_A")
↳	Torque Available Regen (Avg) AVG(t."torque_available_regen")	Temperature IGBT A (Max) MAX(t."temperature_IGBT_A"-100)
↳	Torque Available Regen (Max) MAX(t."torque_available_regen")	Temperature IGBT B (Avg) AVG(t."temperature_IGBT_B")
↳	Torque Available Motoring (Avg) AVG(t."torque_available_motoring")	Temperature IGBT B (Max) MAX(t."temperature_IGBT_B"-100)
↳	Torque Available Motoring (Max) MAX(t."torque_available_motoring")	Temperature IGBT C (Avg) AVG(t."temperature_IGBT_C")
		Temperature IGBT C (Max) MAX(t."temperature_IGBT_C"-100)

- g. finally , add a last measure named “**Total Energy added**” with following SQL expression:

```
SUM(JSON_VALUE("t"."charging_properties",'$.total_charge_energy_added'))
```

Edit measure

Time Period	Value
Last minute	20.94 k
Last hour	1.41 m
Last day	33.45 m

General

Format

Advanced

Basic **Custom**

Formula

```
SUM(JSON_VALUE("t"."charging_properties",'$.total_charge_energy_added'))
```

[Documentation](#)

Cancel **Save**

- h. Give your datacube a name. And click “Save” :

Cancel **Save**

General

Dimensions

Measures

Access

Access filters

Advanced

Name: iot Ev

Description

Source: From table

Default timezone: Etc/UTC

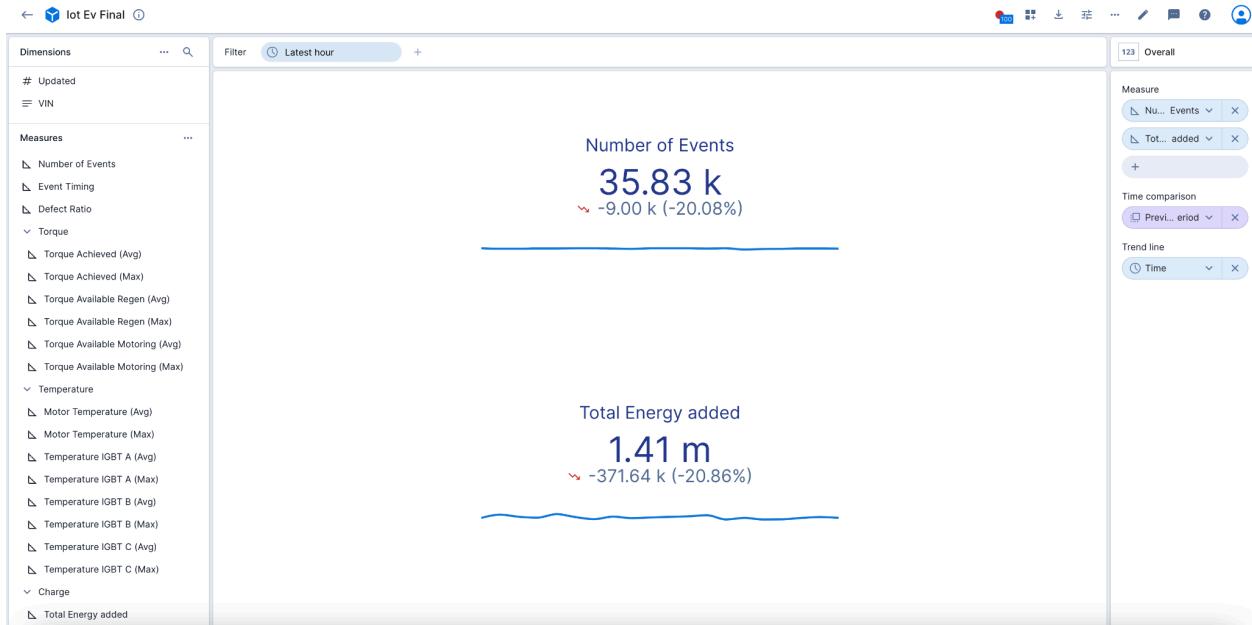
Minimum auto-refresh rate: None

Primary time dimension: Time

Query timeout override

Customize data cube UI colors

- i. Now you can use the UI to drag and drop and analyze the data with existing dimensions and measures, for example, drag “Total Energy added” onto the measure lists. You can do this with multiple measures and change the measures according to your need. From here you can click  at the top right, to add this to a dashboard, but we’ll get into that further in a second. Go ahead and try it. We’ll get into creating dashboards next.

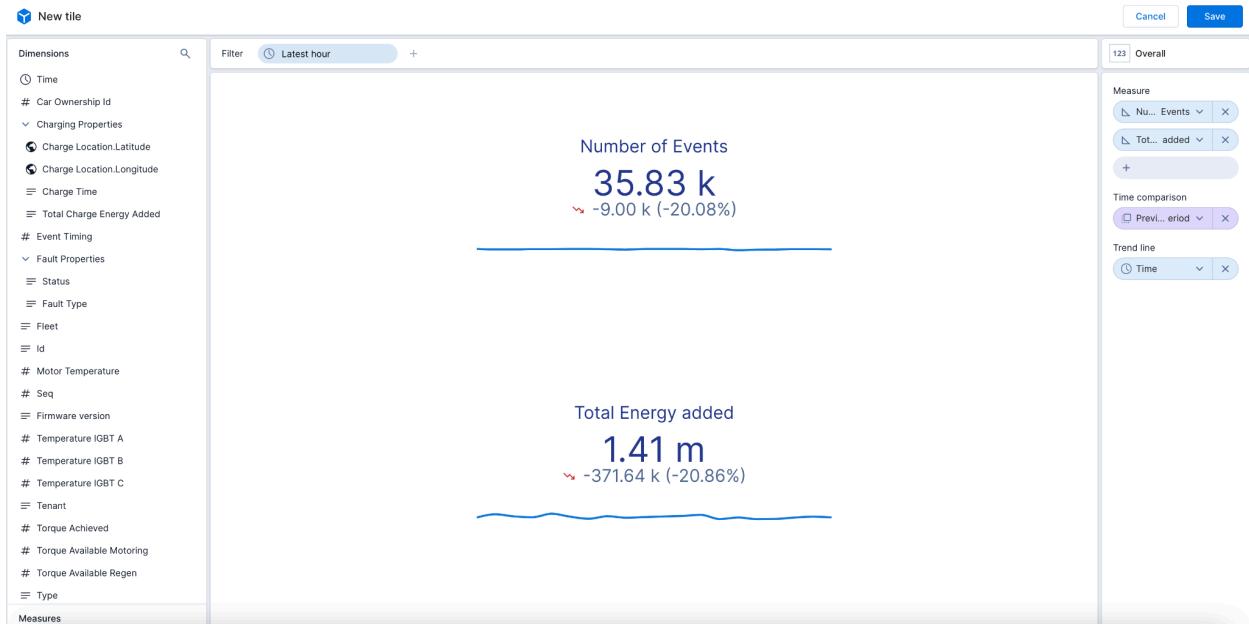


2. Creating a dashboard from a datacube is easy:

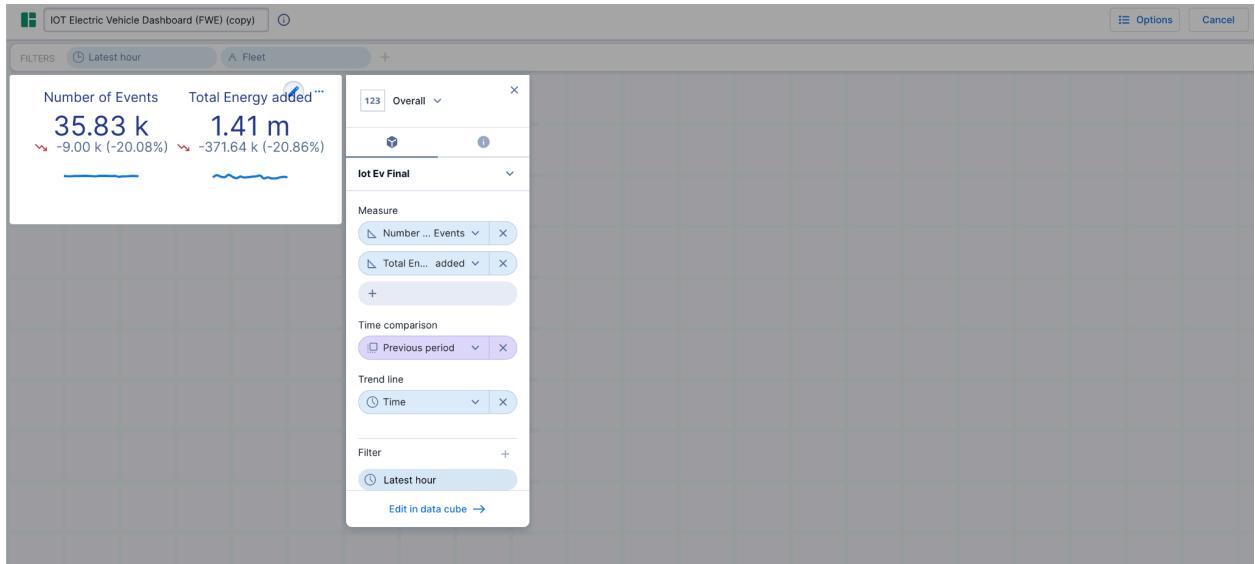
- Navigate back to the home screen and click “Dashboards” and “Create dashboard” at the top. Name the dashboard as “IOT Dashboard”



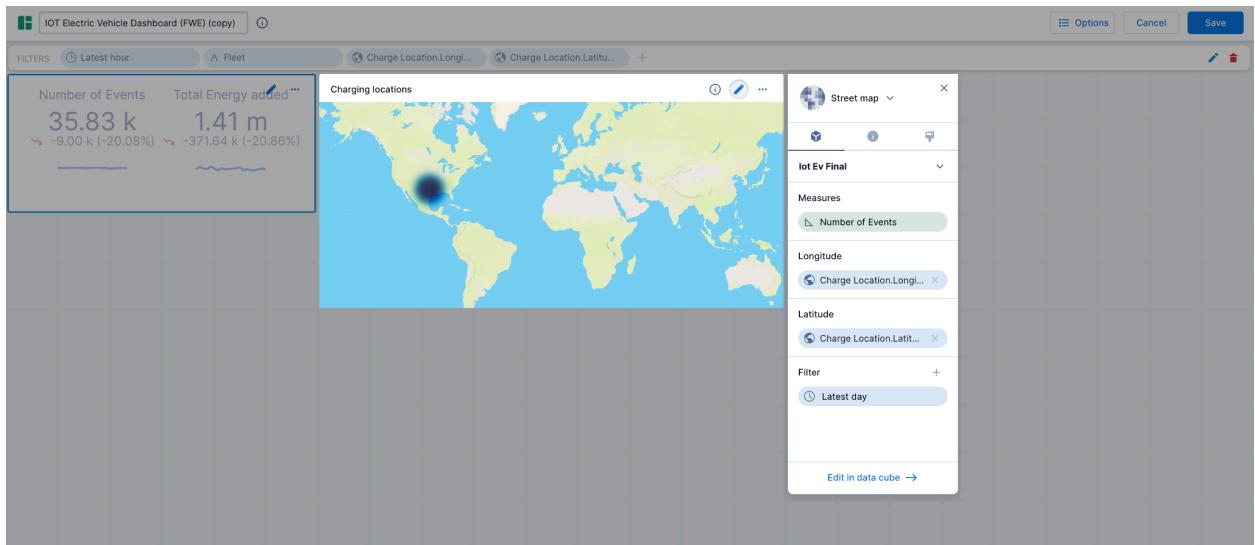
- Click and select an area on the dashboard, in the drop down select the cube you just created, it may be selected by default if you only have one cube in your polaris project. For chart type, select “Overall”, and click “Edit in data cube”, just create the same Overall visualization as previous step, then save :



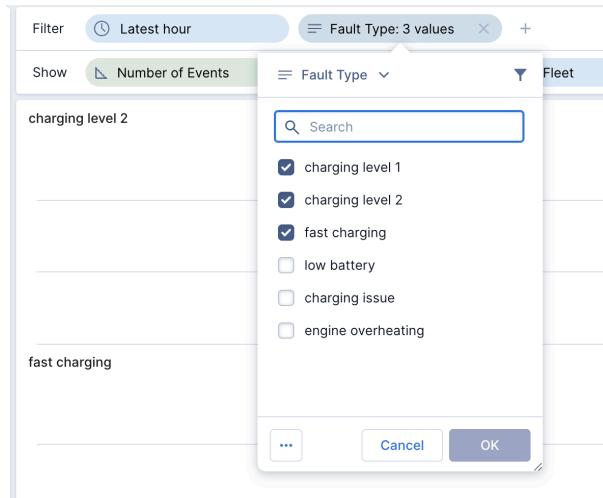
Visualization is added to the dashboard :



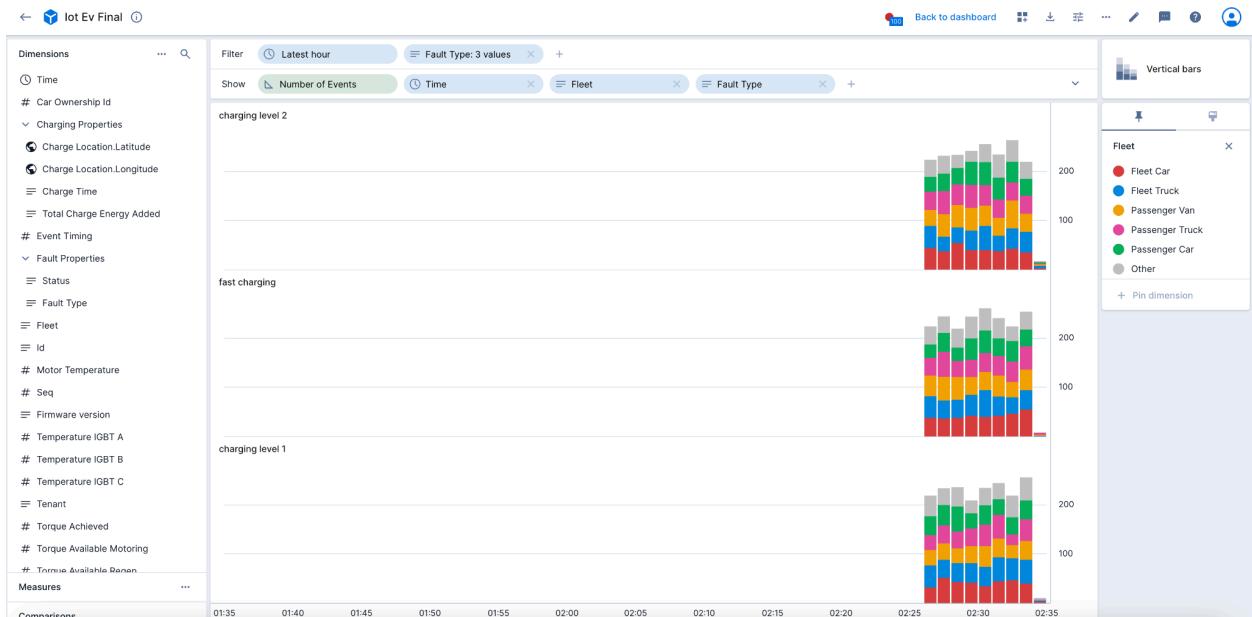
- c. Now, let's create a new chart against the same datacube, select the **street map** visualization, it will automatically propose to use the 2 existing geo dimensions **LAT** and **LON** :



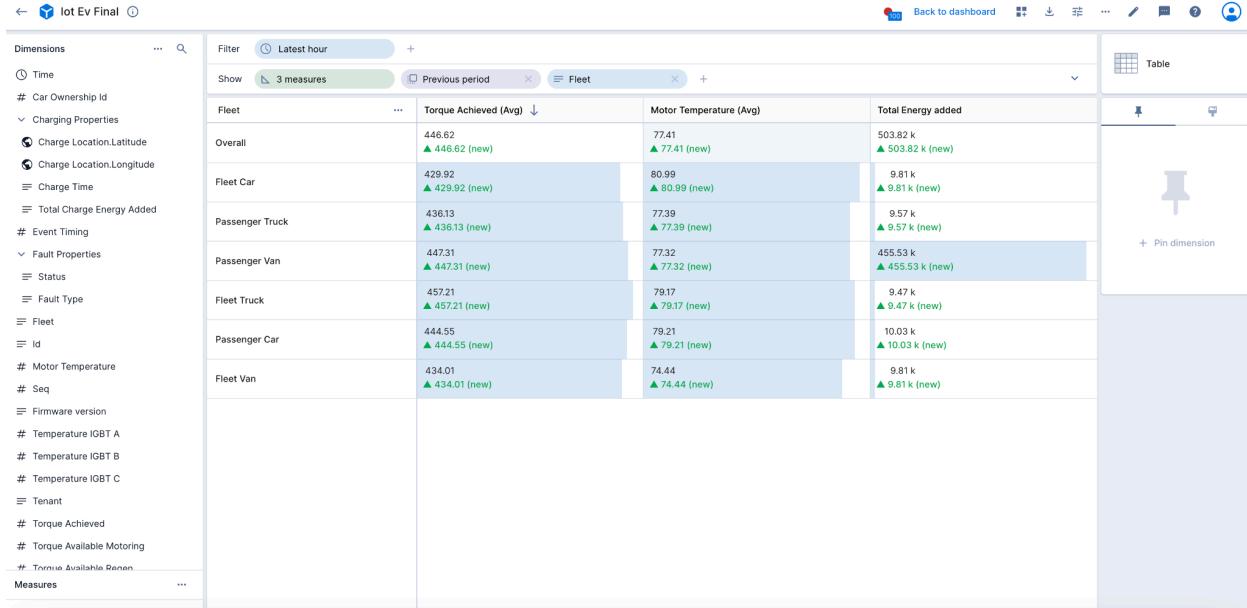
- d. Now, let's create a new chart on our dashboard against the same datacube to monitor the charging status per fleet type, select vertical bars chart type, then add **time**, **fleet** and **fault type**, also add a filter on **fault type** restricted to the 3 charging types below :



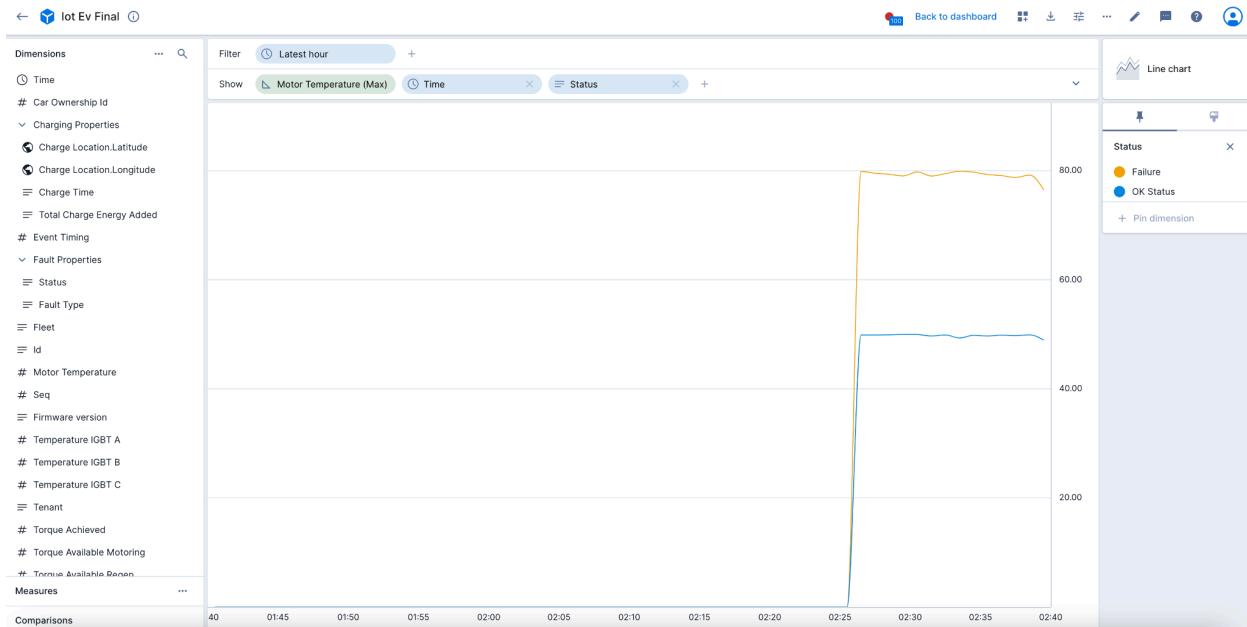
Your report should look like this before you add it to the dashboard :



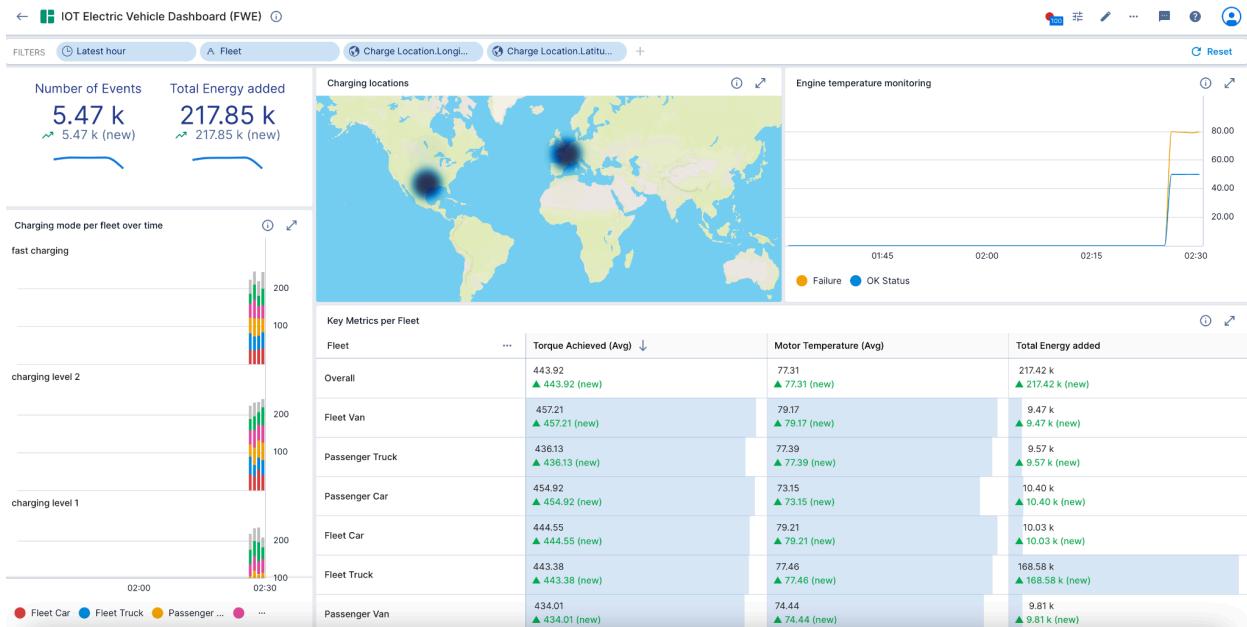
- e. Now let's add a new chart to monitor Avg Torque, Avg Motor Temperature and Total energy added by selecting a Table visualization, adding these 3 measures and adding the previous period for comparison, report should look like this before you add it to the dashboard :



- f. Finally, let's add one final chart to the dashboard to measure the Max Motor temperature per Status by selecting a Line chart visualization, and adding **Motor Temperature (Max)**, **Time** and **Status** to the show bar, your report should look like this before you add it to the dashboard :



- g. Finally rearrange your tiles to fit the screen, edit predefined filters (time to filter on latest hour only, fleet) and save the dashboard. It should look like the example below. You can now click and interact with the dashboard, and filter it. You can also share the dashboard URL with other users, as you filter it the URL preserves the state of the filter. You can click the “Reset” button at the top right and revert to the unfiltered version. For more tips and trick see <https://docs.imply.io/polaris/create-a-dashboard>



3. For instructions on how to create alerts see: <https://docs.imply.io/polaris/set-up-alerts/>
4. For instructions on how to create embedded visualizations see: <https://docs.imply.io/polaris/embed-visualizations>

General documentation around querying polaris via the API and all the building blocks is all available here.

<https://docs.imply.io/polaris/query>

<https://docs.imply.io/polaris/api-query>

<https://docs.imply.io/api/polaris/api-reference>