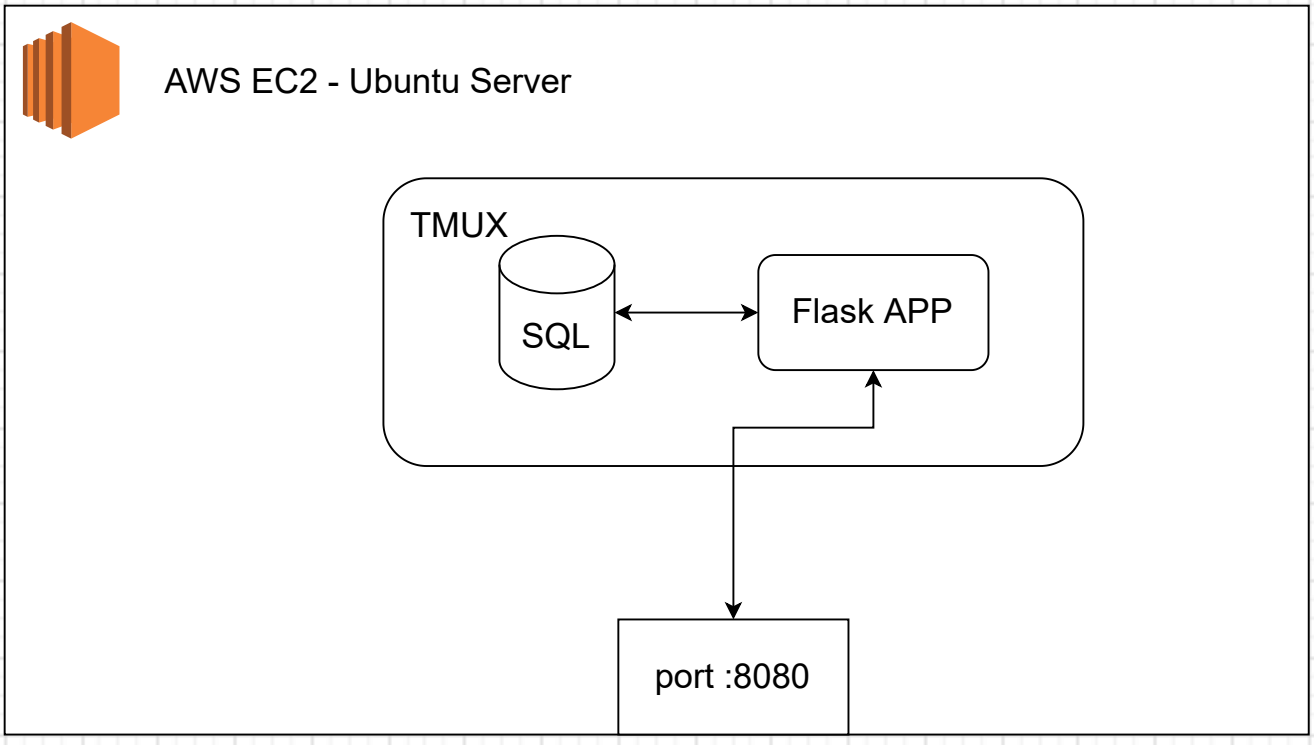
1. API Documentation - Summary of Work

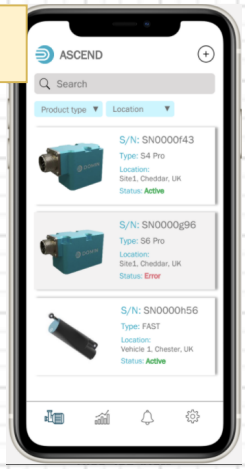
Task	Status	Required?	Description	Page
Summarise Problem & outline data	Done	Core	Map out data needed for the app screens and the endpoints that will deliver this	2 - 4
EP1 - Get Owner Basic	Done	Core	Get basic data on all devices owned by a user	5
EP2 - Get Device Full	Done	Core	Get all data on a single device (excluding data for graphs)	6
EP3 - Patch Control Parameters	Done	Core	Write Data to the database	7
EP4 - List Historical Data	Done	Core	Get historical data filtered by parameter and time period	8
Setup Tests	Done	Additional	Standard tests to verify functionality. Improvement recommended here	9
Host / Deploy	Done	Additional	Host API so it can be accessed on web	10
Put	Future	Additional	Create Record	NA
Del	Future	Additional	Delete Record	NA
OAuth	Future	Additional	Only return data if user is authorised	NA
Better UNIT Tests	Future	Additional	Use pytest or similar to get better test coverage	NA

System Diagram

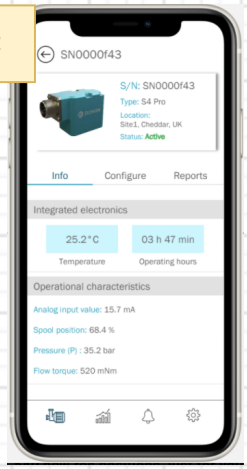


API Documentation - Problem Overview

1

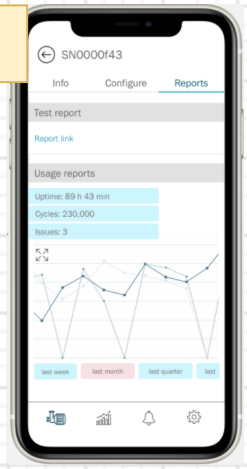


2



3

4



Parameters
owner_ref
serial_number
device_type
location
status
temp
operating_hours
analogue_in
spool_position
pressure
flow_torque
signal_type
signal_range_low
signal_range_high
flow_limit
pressure_controller
uptime
cycles
issues

Get
Get
Get
Get

Note

GET must return multiple products if the user has more than one device

Get
Get
Get
Get
Get
Get
Get
Get
Get
Get
Get

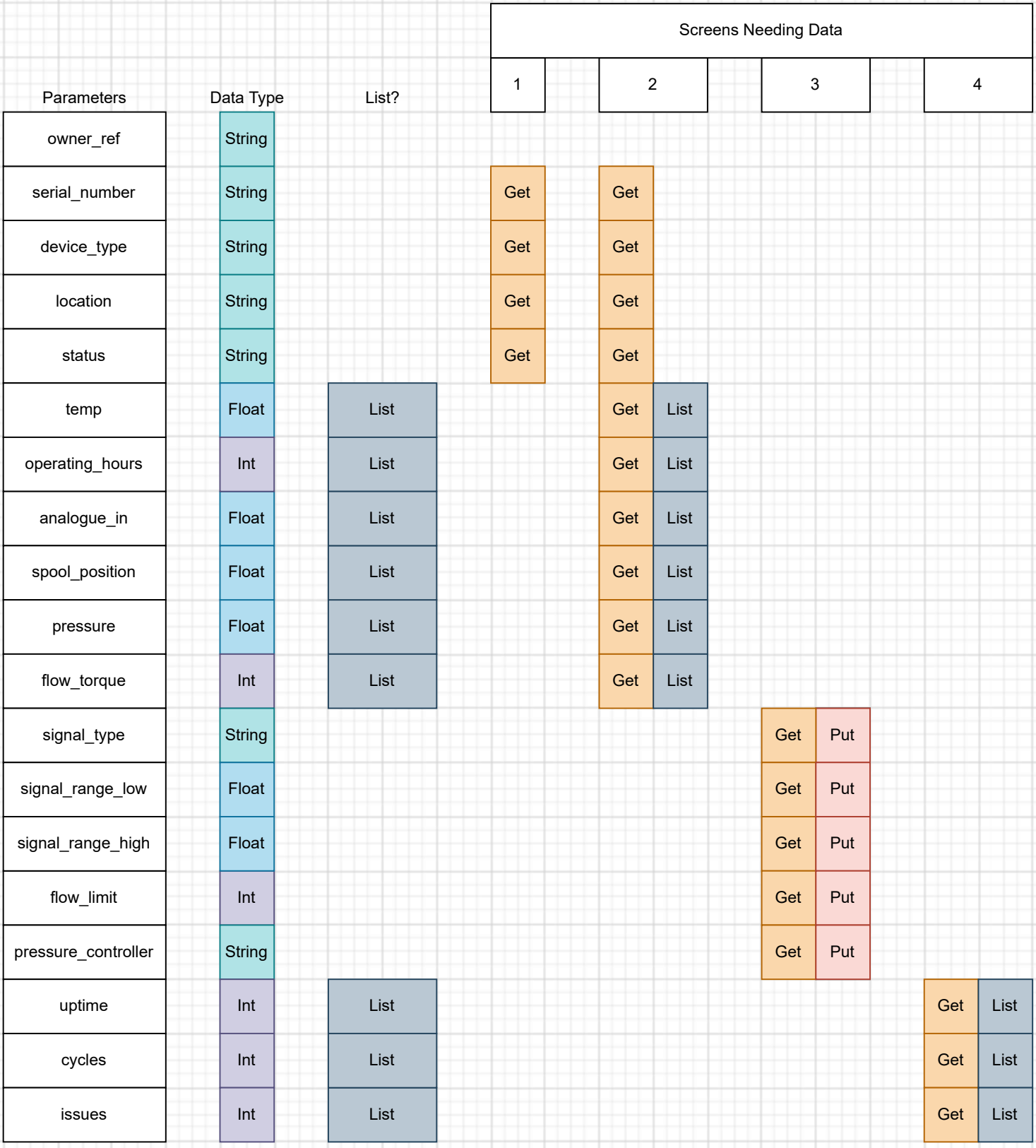
Note

This data can be returned in 1 request

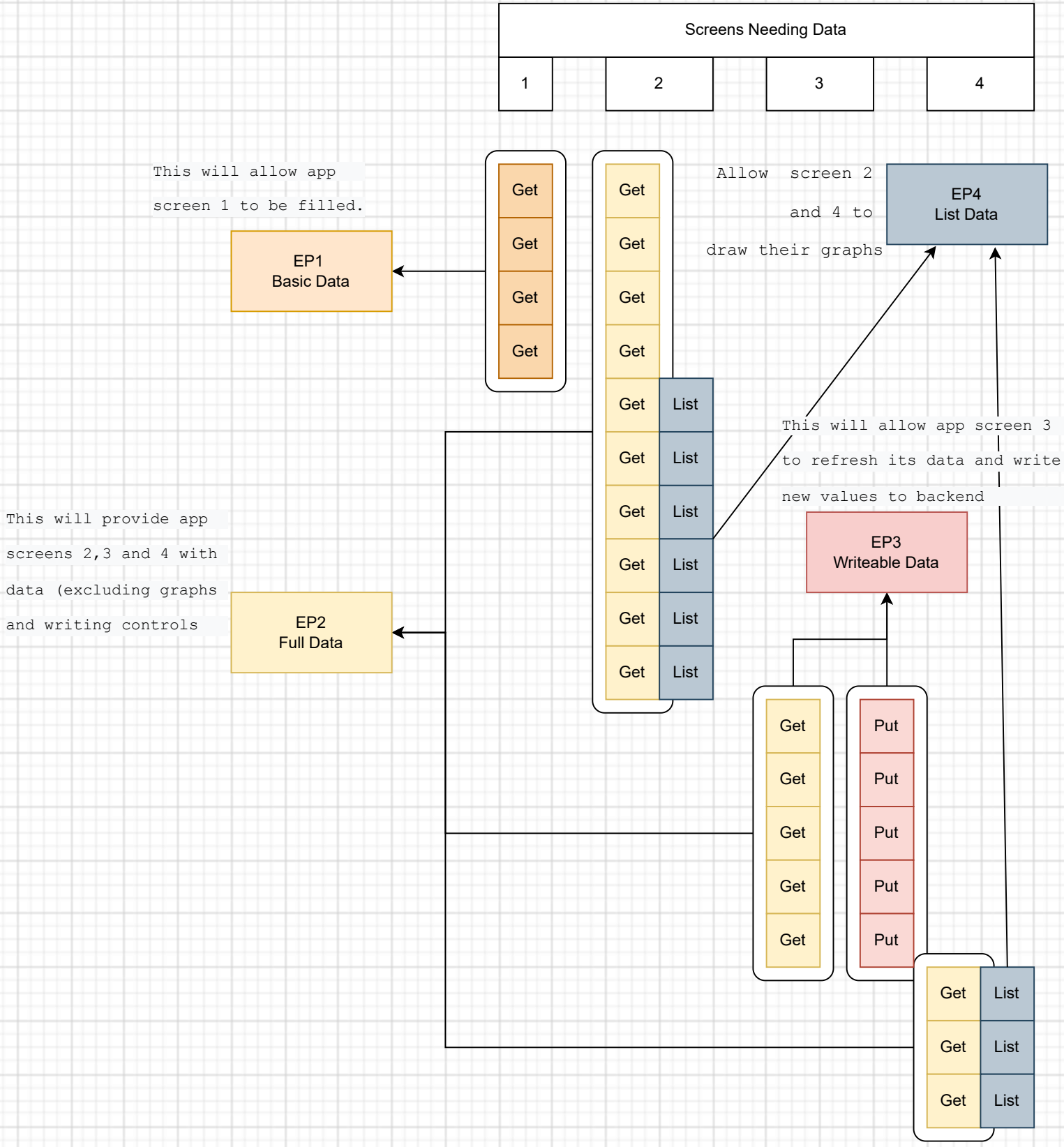
Get	Put
Get	Put
Get	Put
Get	Put
Get	Put

Get
Get
Get

API Documentation - Problem Overview



API Documentation - End Point Overview



API Documentation - End Point Overview

EP1
Basic Data

End Point 1 - /device/<user>

Supports GET requests. GET reads data from SQL database. Returns full list of attribute data for mathing id

See "full_resource_fields" for attributes

Example

http://54.172.7.149:8080/device/full/1

returns


```
{"id": 1,
"owner_ref": "andrew_merrin",
"serial_number": "SN0000f43",
"device_type": "S4 Pro",
"location": "Site1, Cheddar, UK",
"status": "Active",
"img_url": "https://cdn.shopify.com/s/files/1/0502/7817/0780/files/S4_Pro_Homepage.png?v=1608112774"
},
{"id": 3,
"owner_ref": "andrew_merrin",
"serial_number": "4",
"device_type": "S6 Pro X",
"location": "Vehicle 1, Cheddar, UK",
"status": "Active",
"img_url": "https://cdn.shopify.com/s/files/1/0502/7817/0780/files/S6_Pro_X.png?v=164563
},
```


.... for the number of devices the user owns. The above query returns 4.

This will allow app screen 1 to be filled.

API Documentation - End Point Overview

EP2
Full Data

End Point 2 - /device/full/<id>

Supports GET requests. GET reads data from SQL database. It returns basic information for devices with the mathing ow

Example

http://54.172.7.149/:8080/**device**/andrew_merrin

returns


```
{"id": 1,
"owner_ref": "andrew_merrin",
"serial_number": "SN0000f43",
"device_type": "S4 Pro",
"location": "Site1, Cheddar, UK",
"status": "Active",
"img_url": "https://cdn.shopify.com/s/files/1/0502/7817/0780/files/S4_Pro_Homepage.png?v=
"temp": 25.5,
"operating_hours": 216,
"analogue_in": 15.7,
"spool_position": 68.4,
"pressure": 32.2,
"flow_torque": 520,
"signal_type": "4mA \u2013 20mA",
"signal_range_low": 0.5,
"signal_range_high": 10.0,
"flow_limit": 15,
"pressure_controller": "moderate",
"uptime": 5330,
"cycles": 230000,
"issues": 3}
```


For the device matching the id requested. .

This will provide app screens 2,3 and 4 with data (excluding graphs and writing controls

API Documentation - End Point Overview

EP3
Writeable Data

End Point 3 - /device/control/<id>

Supports GET and PUT requests.

GET reads data from SQL database. Returns control data for mathing id

PUT updates values in SQL database. Returns updated control data for mathing id

see "control_resource_fields" for attributes

Example

http://54.172.7.149:8080/device/control/2

returns


```
{"id": 2,
"flow_torque": 111,
"signal_type": "6mA \u2013 30mA",
"signal_range_low": 0.5,
"signal_range_high": 10.0,
"flow_limit": 20,
"pressure_controller": "balanced"}
```


in python data is written as follows

```
data1 = {"flow_limit": 30}
response = requests.patch(BASE + "device/control/1", data1)
```

This will update flow_limit to 30.

As many key value pairs as are required can be passed.

All attributes above are supported (excluding id)

This will allow app screen 3 to refresh its data and write new values to backend

API Documentation - End Point Overview

EP4
List Data

End Point 4 - /device/history/<id>

Supports GET request. Generates random data for testing.

Requires: device_id, timeframe, attribute, and data_points

Returns key value pair Attribute:[list of random ints, length =data_points]

Example

http://54.172.7.149:8080/device/history/1,month,temp,15

returns

```
{"temp": [25, 15, 24, 14, 6, 19, 12, 17, 13, 22, 29, 10, 21, 23, 8]}
```

Your data will be random.

Supported attributes:

"temp",
"operating_hours",
"analogue_in",
"spool_position",
"pressure",
"flow_torque",
"uptime",
"cycles",
"issues",

Supported timeframe:

"month"

"week"

"day"

"hour"

datapoints has a limit of 31

This will allow app screen 2 and 4 to draw their graphs

API Documentation - Tests

TEST

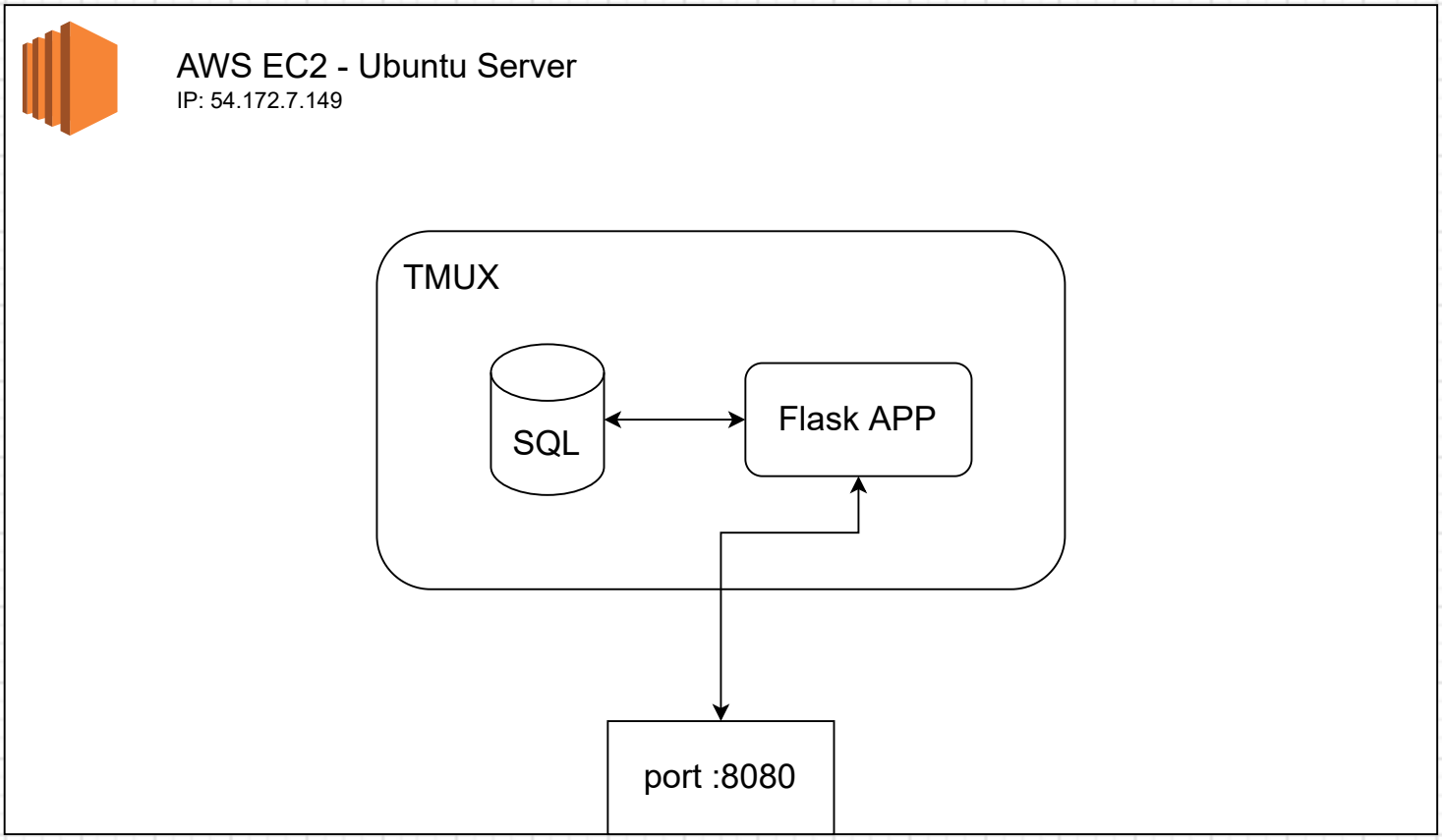
Please see test.py

There are a number of basic tests to verify that the api is returning correct values

Before deployment more work here is recommended.

http://54.172.7.149:8080/device/history/1,month,temp,15
http://54.172.7.149:8080/device/control/2
http://54.172.7.149:8080/device/history/1,month,temp,100
http://54.172.7.149:8080/device/full/1
http://54.172.7.149:8080/device/andrew_merrin

API Documentation - Deployment



Example Calls - try in your browser

http://54.172.7.149:8080/device/history/1,month,temp,15
http://54.172.7.149:8080/device/control/2
http://54.172.7.149:8080/device/history/1,month,temp,100
http://54.172.7.149:8080/device/full/1
http://54.172.7.149:8080/device/andrew_merrin