Module 3

IoT Lab Exercises

*Review Appendix for Pre-Lab Setup
** You may want to print Labs to ensure you can follow along during immersion experience

Agenda

- Objectives
- Background
- Lab 1 - Raspberry Pi Basics
- Lab 2 - IoT Basics
- Lab 3 - Azure Basics
- Lab 4 - Data and Analytics Basics

Objectives

- Apply hands on learnings from Module 1 and Module 2 to real world IoT and Data & Analytics solutions
- Remove fear and uncertainty of IoT and Data & Analytics solutions
- Simplify jargon and theory to understandable terms and applications
- Deploy an end-to-end IoT real world solution
- Provide an understanding of all roles required to build and deploy an IoT and Data & Analytics solution

Background

- A custom image was created for Walsh STEM immersion experience
- The custom image was based off Raspbian Buster Full which is a Linux Distros
- In a real world scenario, you will use a different version of Raspbian Buster with less programs and packages installed. Additionally, you will not have a graphical user interface ("GUI"). This is often referred to as headless
- During the labs, your Raspberry Pi will serve two purposes
  - Development Machine
  - IoT Edge Device

Lab 1

Raspberry Pi Basics

Lab 1

Exercise 1 – Remote into Raspberry Pi

Background

- There are multiple ways to interact with the Raspberry Pi. The most common is Secure Shell (“SSH”)
- In this exercise we will use an SSH client from the terminal on your Raspberry Pi
- In real-world scenarios you will use PUTTy (primarily for Windows operating systems) or other SSH clients from a remote computer
- In this exercise you will learn to run Linux commands and can reference command in the cheat sheets provided in your toolkit

Lab Instructions

1. Click terminal icon >_ on Raspberry Pi
2. Type **ssh pi@walshu-raspberrypi-01** (change highlight based on Raspberry Pi name)
3. Enter password – **WalshPi**
4. Type **yes** and Press enter key
    a. Note: This step may not be required based on environment
5. Type **ls**
    a. Note: This is a Linux command to list files and folders
    b. Note: Repeat this step as often as you would like with other Linux commands (refer to appendix)
6. Type **ifconfig**
    a. Note: wlan0 is your network card with the ip address of your Raspberry Pi. Look for inet 192.x.x.x
7. Type **exit**
    a. Note: You should see logout
8. Type **exit** to close terminal

Lab 1

Exercise 2 – Sense HAT Emulator

Background

- This exercise is meant to exposure you to python code and simulate changing temperature on the Raspberry Pi
- Python is a programming language used for many applications

Lab Instructions

1. Click Raspberry Pi icon (bottom left of screen)
2. Click Programming > Sense HAT Emulator
3. Click File > Open example > Simple > temperature.py
    a. Note: Review the flow and logic of python code
4. Click Run icon ▶ Run
5. Click Minimize icon ⌄
6. Move temperature scroll bar up and down
    a. Note: You should notice the red and blue colors change based on temperature
7. Maximize Thonny application and click Stop icon ■ Stop
8. Close all windows by clicking X icon

Lab 1

Exercise 3 – Open walsh-stem-iot solution

Background

- This exercise will introduce you to the basics of Visual Studio Code and connect Visual Studio Code to Azure to simplify the development experience

Lab Instructions

1. Click Raspberry Pi icon (bottom left of screen)
2. Click Programming > VSCode
3. Click File > Open Folder
4. Click Desktop > Single Click walsh-stem-iot folder > Click OK
   a. Note: Close any open windows if they are open
5. Click Select IoT Hub
6. Click Sign In (bottom right of screen)
   a. Ensure you use your student username and password
   b. Sample: username - stem01@walsh.edu password – RaspberryPi01!
7. Close Browser by clicking X
8. Select Az Sub 1 in VSCode
9. Select walsh-stem-iot-iothub

Lab 2

IoT Basics

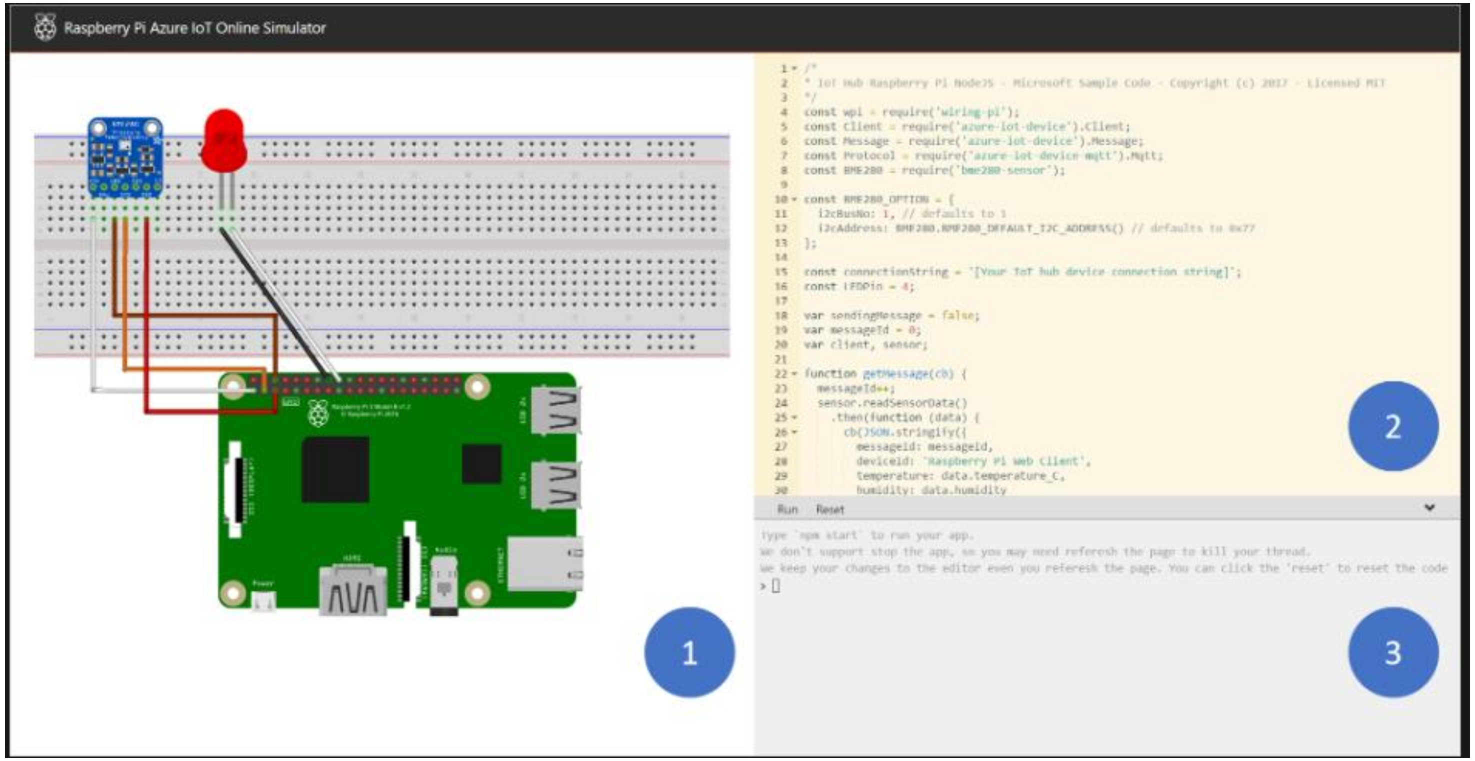
Lab 2

Exercise 1 – Raspberry Pi Simulator

Background

- We will use a Raspberry Pi simulator to show how to blink a led and send temperature data to Azure
- You will learn to navigate Azure and the Azure IoT Hub
- All code is written in Node.js and stored on GitHub
- The hardware in this lab could easily be added to your physical Raspberry Pi

Background



```
/*
 * IoT Hub Raspberry Pi NodeJS - Microsoft Sample Code - Copyright (c) 2017 - Licensed MIT
 */
const wpi = require('wiring-pi');
const Client = require('azure-iot-device').Client;
const Message = require('azure-iot-device').Message;
const Protocol = require('azure-iot-device-mqtt').Mqtt;
const BME280 = require('bme280-sensor');

const BME280_OPTION = {
  i2cBusNo: 1, // defaults to 1
  i2cAddress: BME280.BME280_DEFAULT_I2C_ADDRESS() // defaults to 0x77
};

const connectionString = '[Your IoT hub device connection string]';
const LEDPin = 4;

var sendingMessage = false;
var messageId = 0;
var client, sensor;

function getMessage(cb) {
  messageId++;
  sensor.readSensorData()
    .then(function (data) {
      cb(JSON.stringify({
        messageId: messageId,
        deviceId: 'Raspberry Pi Web Client',
        temperature: data.temperature_C,
        humidity: data.humidity
```

```
Type `npm start` to run your app.
We don't support stop the app, so you may need referesh the page to kill your thread.
We keep your changes to the editor even you referesh the page. You can click the 'reset' to reset the code
> ▯
```

1. Assembly area - The default circuit is that a Pi connects with a BME280 sensor and an LED. The area is locked in preview version so currently you cannot do customization.
2. Coding area - An online code editor for you to code with Raspberry Pi. The default sample application helps to collect sensor data from BME280 sensor and sends to your Azure IoT Hub. The application is fully compatible with real Pi devices.
3. Integrated console window - It shows the output of your code. At the top of this window, there are three buttons.

Lab Instructions

1. Navigate to Azure Raspberry Pi Simulator
   a. https://azure-samples.github.io/raspberry-pi-web-simulator/
2. Navigate to Azure Portal in new tab or additional browser window
   a. https://portal.azure.com
   b. Login with your Walsh provided student username and password
3. Get IoT device connection string from Azure Portal
   a. Click Resource Groups > walsh-stem-iot-rg > walsh-stem-iot-iothub
   b. Under Automatic Device Management, click IoT Edge
   c. Click your IoT device
   d. Click copy icon ⧉ next to Primary Connection String
4. Paste your connection between ' ' (highlighted below) on line 15
   a. const connectionString = '[Your IoT hub device connection string]';
5. Click Run

Lab 2

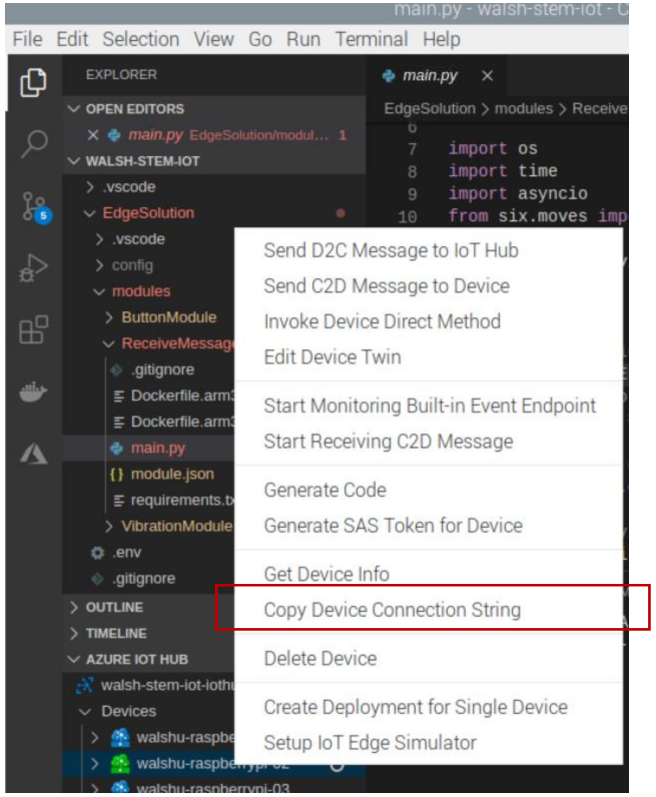
Exercise 2 – Raspberry Pi end-to-end solution

Background

- This exercise will show you how to update python code and deploy the updated code to your Raspberry Pi
- In real world scenarios, the device can exist anywhere in the world and you can follow a very similar process to deploy updates to specific devices

Lab Instructions

1. Follow Steps from Lab 1 – Exercise 3
2. Expand EdgeSolution > Modules > ReceiveMessageModule
3. Click main.py
4. Goto line 17 and paste your connectionstring
    a. To get your connection string, right click your device
    b. Click Copy Device Connection String
    c. Note: connectionstring goes between “paste here”
    d. Tip: delete between “ ” then right click and click paste
5. Click module.json
    a. Update line 7 to 1.[device number].1



```
import os
import time
import asyncio
from six.moves imp
```

Lab Instructions

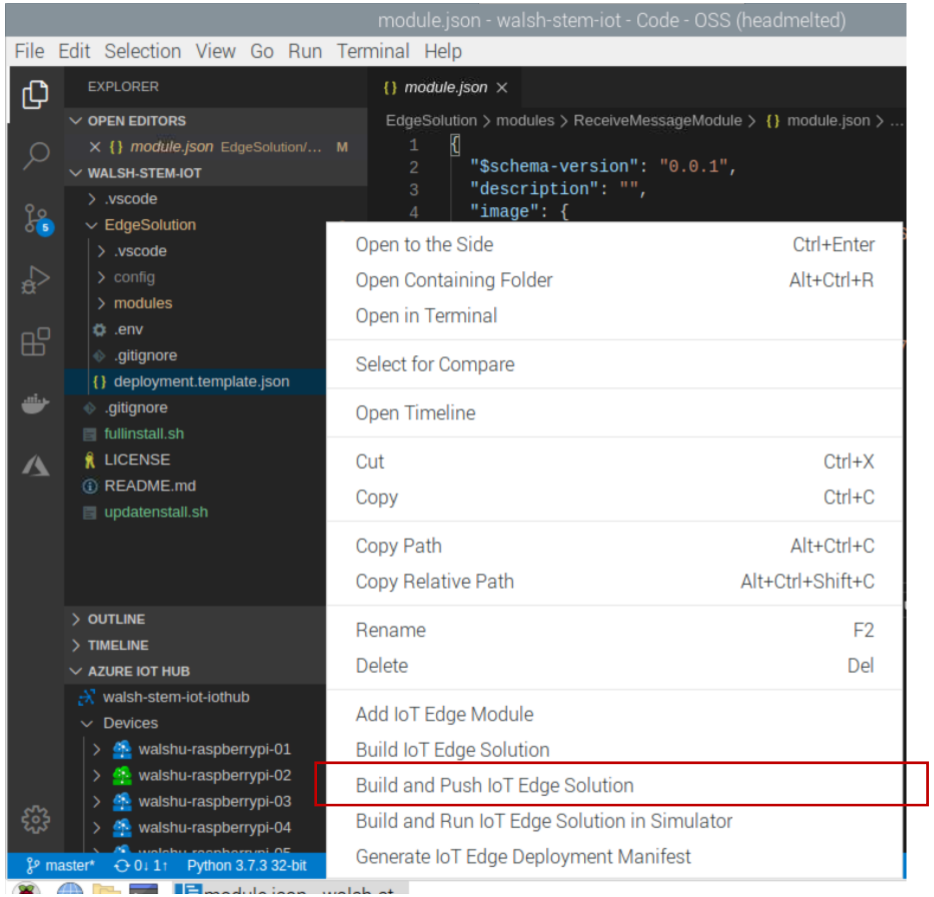
6. Expand EdgeSolution > Modules > ButtonModule
7. Click main.py
   a. Note: Review python code
8. Goto line 66
9. Change highlighted section below to your name
   a. message.custom_properties["studentName"] = "Student1“
10. Click module.json
    a. Update line 7 to 1.[device number].1

Lab Instructions

11. Expand EdgeSolution > Modules > VibrationModule
12. Click main.py
    a. Note: Review python code
13. Goto line 63
    a. Repeat for line 91
14. Enter your name between “ “ (highlighted below)
    a. message.custom_properties["studentName"] = "Student1“
15. Click module.json
    a. Update line 7 to 1.[device number].1

Lab Instructions

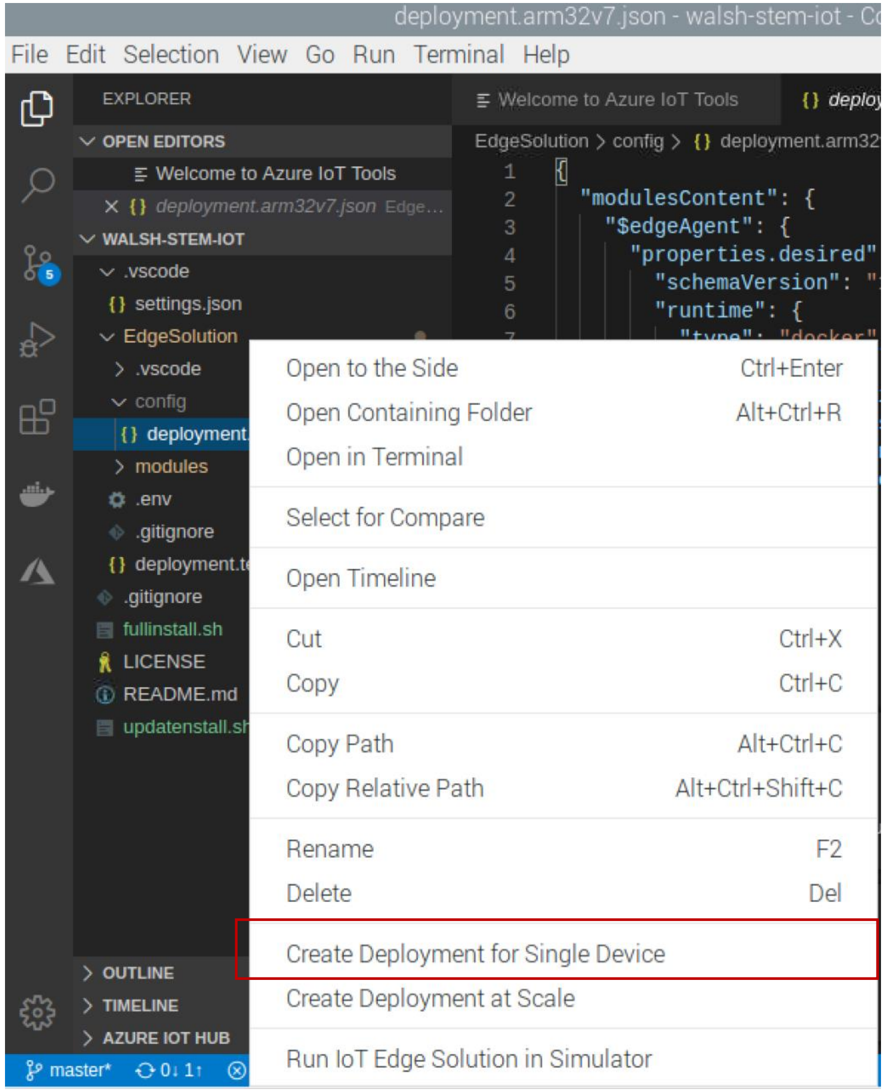
16. Click File Save All
17. Right click deployment.template.json
    a. Click Build and Push IoT Edge Solution



```
{
  "$schema-version": "0.0.1",
  "description": "",
  "image": {
```

Lab Instructions

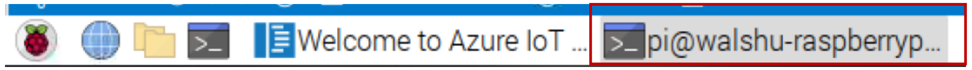
18. Expand Edge Solution > config
19. Right click deployment.arm32v7.json
    a. Click Deployment for Single Device
    b. Important: Select your Raspberry Pi
20. Click Terminal icon
21. Type **docker ps -a**
    a. Look for ReceiveMessageModule
    b. Status should be up X seconds or minutes
22. Type **docker logs ReceiveMessageModule -f**



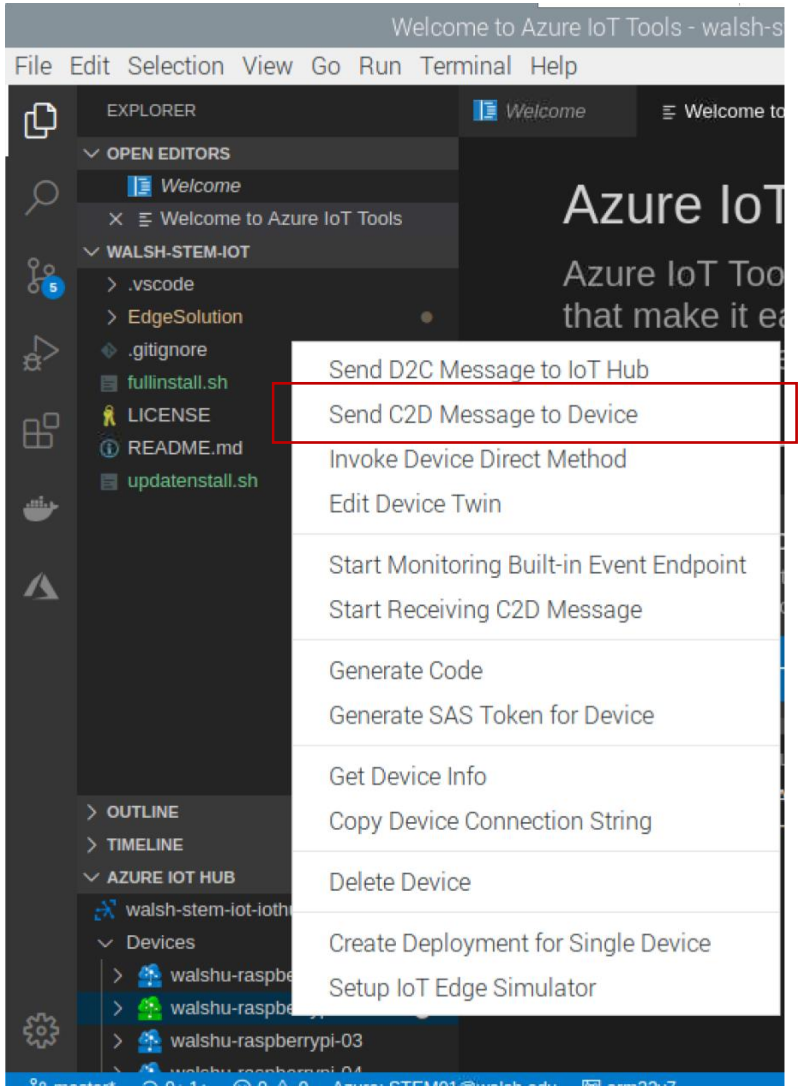
deployment.arm32v7.json - walsh-stem-iot - C
File Edit Selection View Go Run Terminal Help
EXPLORER
OPEN EDITORS
Welcome to Azure IoT Tools
deployment.arm32v7.json Edge...
WALSH-STEM-IOT
.vscode
settings.json
EdgeSolution
.vscode
config
deployment
modules
.env
.gitignore
deployment.t
.gitignore
fullinstall.sh
LICENSE
README.md
updatenstall.sh
OUTLINE
TIMELINE
AZURE IOT HUB
master* 0↓ 1↑
Welcome to Azure IoT Tools
deploy
EdgeSolution > config > deployment.arm32
Open to the Side Ctrl+Enter
Open Containing Folder Alt+Ctrl+R
Open in Terminal
Select for Compare
Open Timeline
Cut Ctrl+X
Copy Ctrl+C
Copy Path Alt+Ctrl+C
Copy Relative Path Alt+Ctrl+Shift+C
Rename F2
Delete Del
Create Deployment for Single Device
Create Deployment at Scale
Run IoT Edge Solution in Simulator

```
{
  "modulesContent": {
    "$edgeAgent": {
      "properties.desired"
        "schemaVersion": "
        "runtime": {
```

Lab Instructions

23. Click Minimize icon ⌄
24. Right Click your device
    a. Click Send C2D Message to Device
    b. Note: C2D = Cloud to Device
25. Type **hello from the cloud**
26. Click open Terminal from bottom taskbar
    a. 

    b. Note: you should see your message
27. Click X to close Terminal



Lab 2

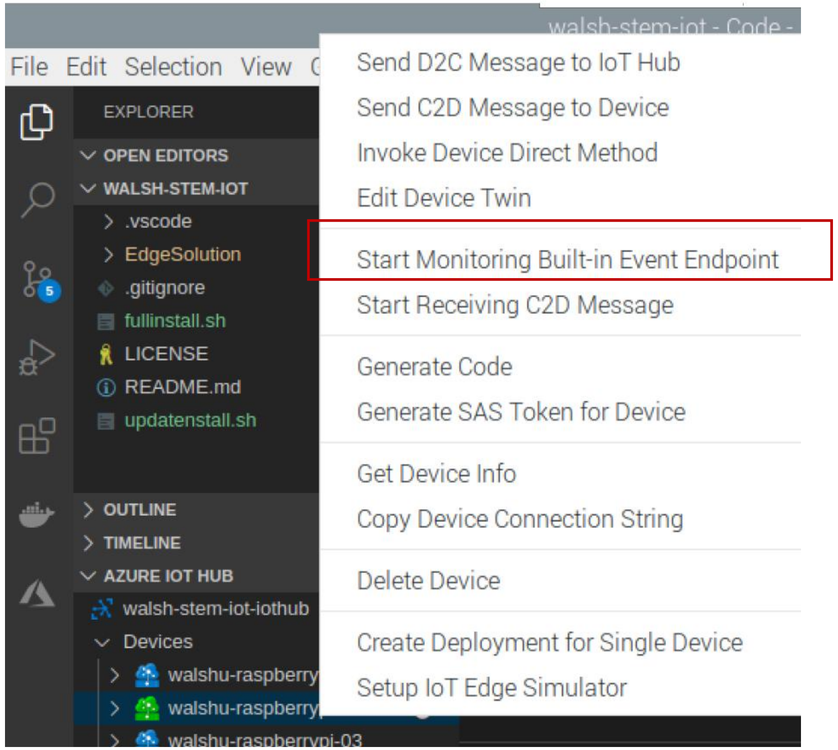
Exercise 3 – Send IoT Messages to IoT Hub

Background

- IoT devices can have multiple sensors for specific user cases
- In this exercise we will simulate a machine that requires maintenance after a certain number of presses
- There are multiple ways to monitor messages going to and from IoT devices
- In this exercise you will send messages to Azure and monitor Built-in Event endpoints

Lab Instructions

1. Follow Steps from Lab 1 – Exercise 3
   a. **Note**: Skip if you completed Lab 2 Exercise 2
2. Right click your device
3. Click Start Monitoring Built-in Event Endpoint
4. Push button or Use vibration sensor
   a. Note: you are simulating a machine stamping or a machine vibrating
   b. Note: push button multiple times
   c. Note: hold button down to see variations in data
   d. Note: you should see data in output window



Lab 2

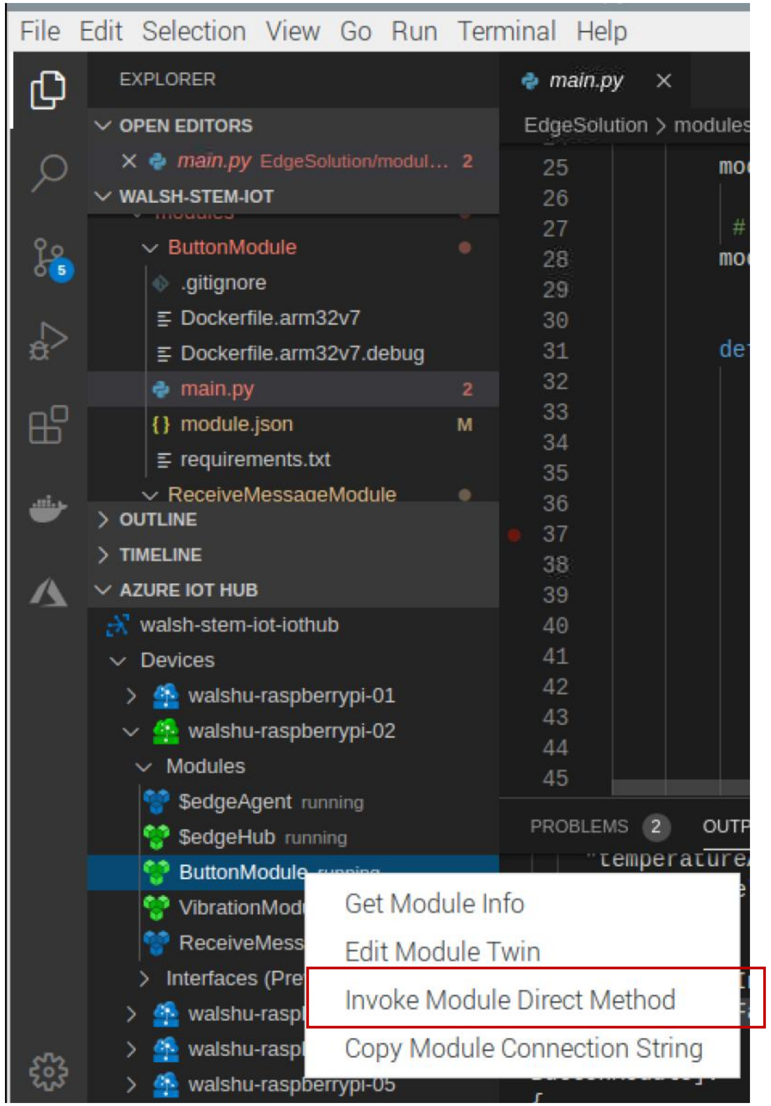
Exercise 4 – Invoke Module Direct Method

Background

- In this exercise you will set the value remotely when your device should alert you of a temperature issue

Lab Instructions

1. Follow Steps from Lab 1 – Exercise 3
   a. **Note**: Skip if you completed Lab 2 Exercise 2
2. Expand your device > Modules
3. Right click ButtonModule
   a. Click Invoke Module Direct Method
   b. Type **SetTemp**
   c. Press Enter
   d. Type **40**
   e. Press Enter



Lab Instructions

4. Press button sensor
5. Review output window
   a. You should notice the temperature you set
   b. Note: Based on timing, instructors will show PowerBi dashboard and your device should have an alert

Lab 3

Azure Basics

Lab 3

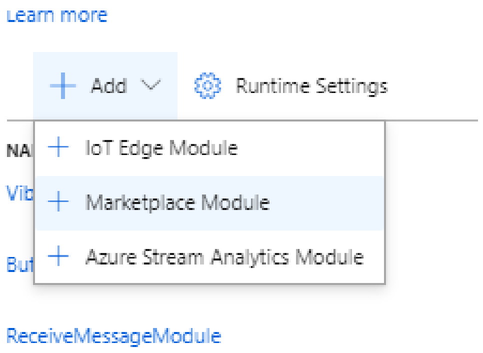
Exercise 1 – Add Temperature Module

Background

- The Azure marketplace has modules that can be deployed to your IoT device
- In this exercise, we will simulate a temperature sensor on your IoT device

Lab Instructions

1. Navigate to Azure Portal in browser window
   a. https://portal.azure.com
   b. Login with your Walsh provided student username and password
2. Deploy Module from Azure Portal
   a. Click Resource Groups > walsh-stem-iot-rg > walsh-stem-iot-iothub
   b. Under Automatic Device Management, click IoT Edge
   c. Click your IoT device
   d. Click Set Modules
   e. Click Add Marketplace Module
   f. Type **temperature** in search
   g. Click Simulated Temperature Sensor
   h. Click Review and Create > Create
3. Run docker ps -a on raspberry pi



Lab Instructions

4. Run docker logs SimulatedTemperatureSensor -f
5. Remove Simulated Temperature Sensor
   a. Click Set Modules
   b. Click trash icon 🗑 by Simulated Temperature Sensor
   c. Click Review and Create
   d. Click Create

Lab 3

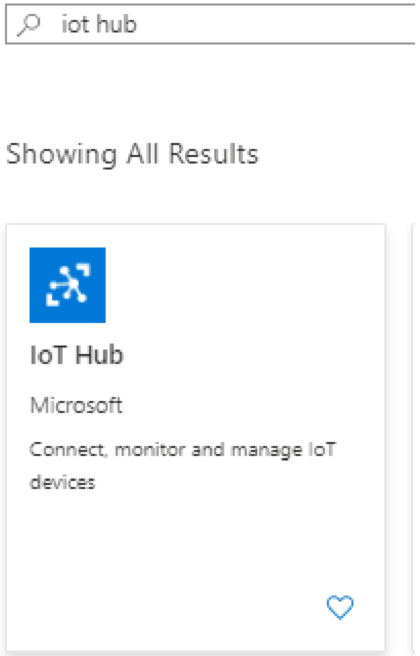
Exercise 2 – Create IoT Hub

Background

- Azure has over 200 services that be easily be deploy through the portal
- In this exercise you will learn how to create an IoT Hub very similar to what you been using for the labs
- In your free time ☺, you should sign-up for a free Azure account and connect your raspberry pi to your Azure IoT Hub
    - Additionally instructions and labs will be created in future to assist with end to end setup

Lab Instructions

1. Navigate to Azure Portal in browser window
   a. https://portal.azure.com
   b. Login with your Walsh provided student username and password
2. Deploy IoT Hub from Azure Portal
   a. Click Resource Groups > walsh-stem-iot-rg
   b. Click Add
   c. Type **IoT Hub**
   d. Select IoT Hub by Microsoft (image to right)
   e. Click Create
   f. Enter **walsh-stem-iot-iothub-student01**
   g. Click Size and scale
   h. Choose F1: Free tier for Pricing and scale tier
   i. Click Review + create
   j. Click Create



iot hub

Showing All Results

IoT Hub

Microsoft

Connect, monitor and manage IoT devices

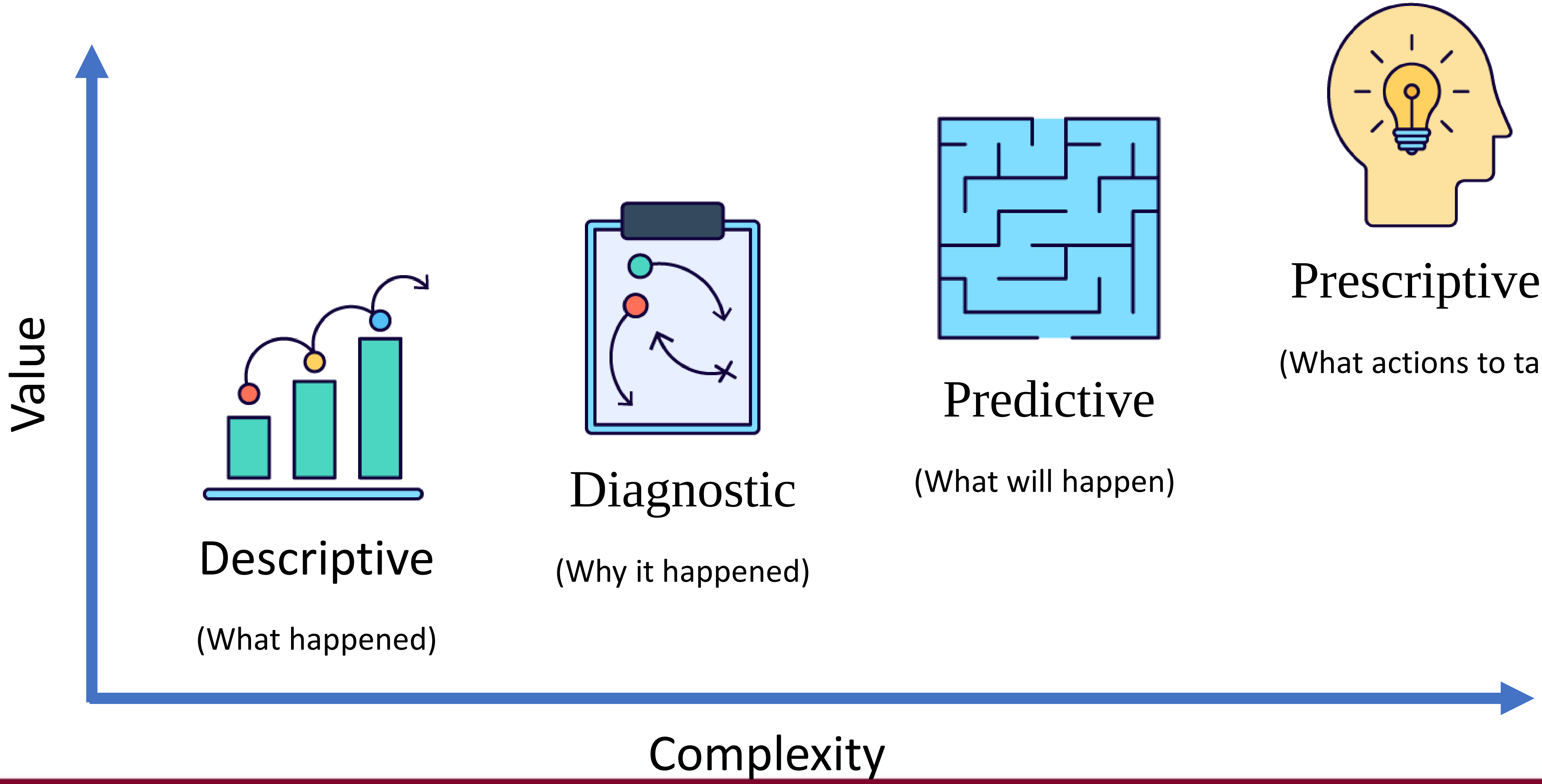
Lab 4

Data and Analytics Basics

Lab 4

Exercise 1 – Visualize IoT data with PowerBi Desktop

There are 4 main categories of analytics



Descriptive

(What happened)

Diagnostic

(Why it happened)

Predictive

(What will happen)

Prescriptive

(What actions to take)

Two fundamental terms

Measures

Dimensions

Convert the story to a data model

List your big questions:

1. What is my Total Sales for a Selected Year and Region?
2. How is my Total Sales doing Year Over Year?
3. How are my Units trending for various States in my region?
4. How is my Sales doing by Channel, Device, Category for selected Year?
5. Which categories are performing best to worst by Total Sales ?

What are you Measuring?

How are you describing or Slicing?

Units
Total Sales
Gross Profit

These are **Measures** which live in **Fact** tables

By Time (Year, Month)
By Geography (Region, State or City)
By Campaign (Channel or Device)

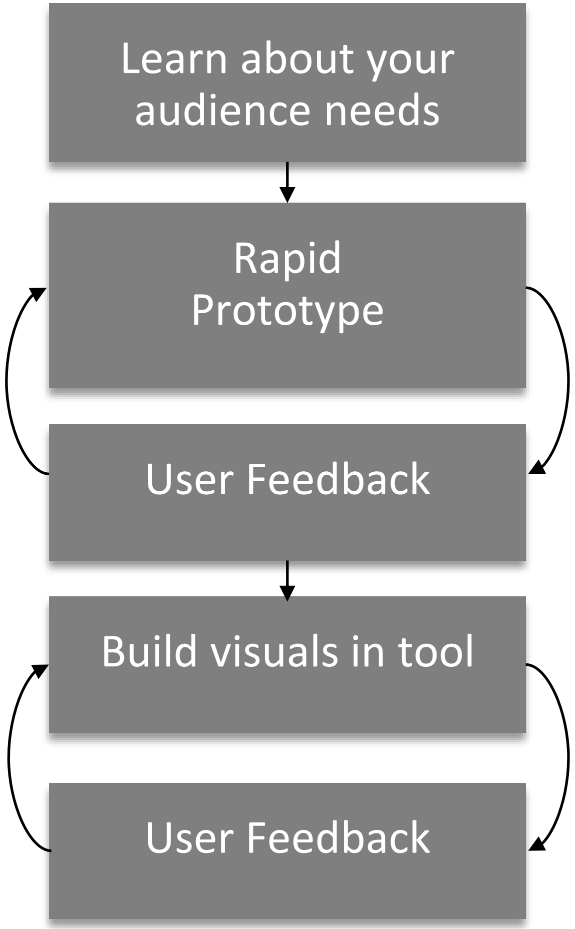
These are **Attributes** that live in **Dimension** tables

Data Visualization Process

Top 5 Questions

Who? Where?
What? How Many?
When? Why?

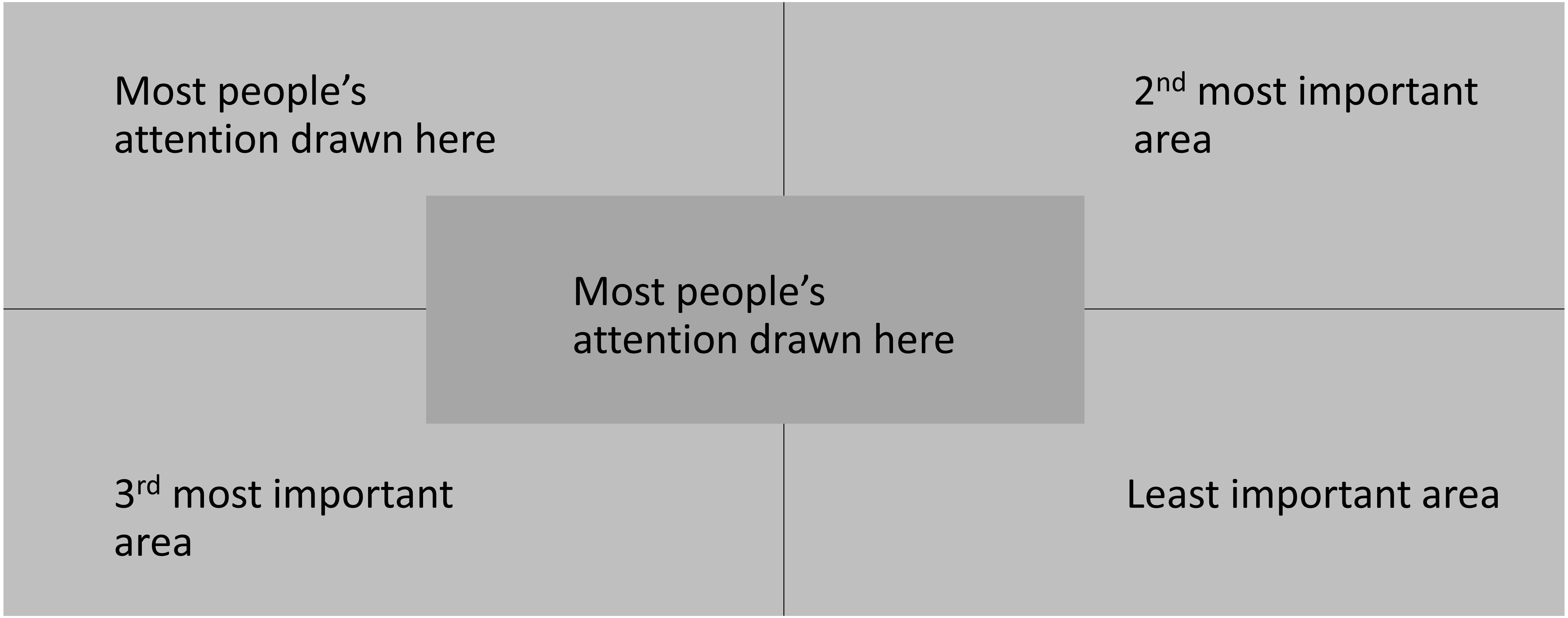
Start Visuals with Sample Data



Use Whiteboard or Storyboard to plan POC

Think Agile!!

Layouts



Most people's attention drawn here

2nd most important area

Most people's attention drawn here

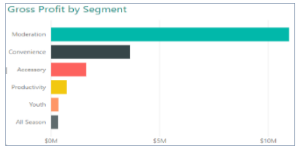
3rd most important area

Least important area

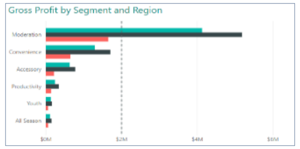
Choosing the right chart for the right purpose

WALSH UNIVERSITY
A Catholic University of Distinction

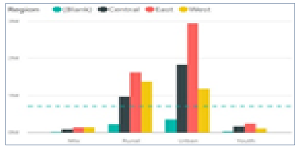
Comparison



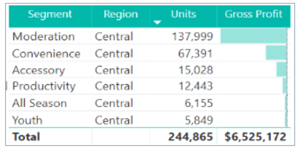
Stacked Bar



Clustered Bar

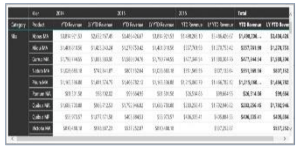


Clustered Column



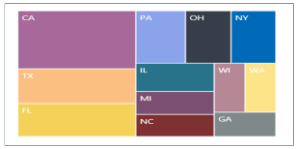
Segment	Region	Units	Gross Profit
Moderation	Central	137,999	
Convenience	Central	67,391	
Accessory	Central	15,028	
Productivity	Central	12,443	
All Season	Central	6,155	
Youth	Central	5,849	
Total		**244,865**	**$6,525,172**

Table

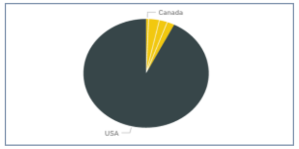


Matrix

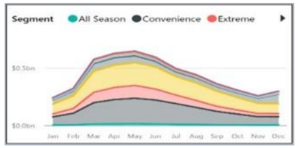
Composition



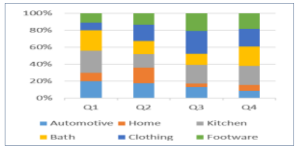
Tree



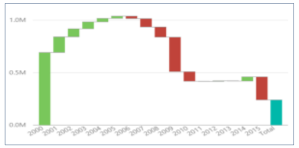
Pie



Stacked Area

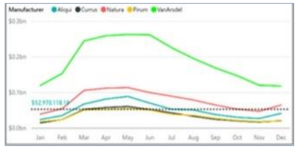


100% Stacked Column

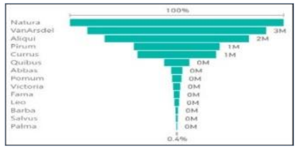


Water Fall

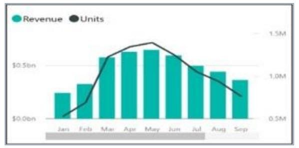
Trend



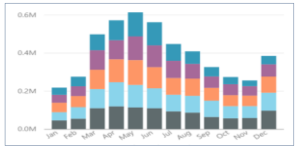
Line



Funnel

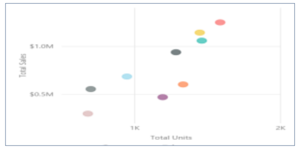


Line and Stacked Column

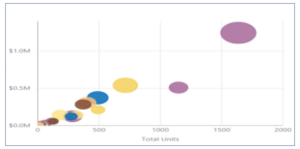


Stacked Column

Distribution



Scatter Plot

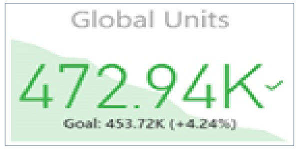


Bubble

NORTH
AMERICA
Atlantic
Ocean

Map

Status

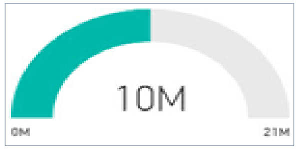


Global Units

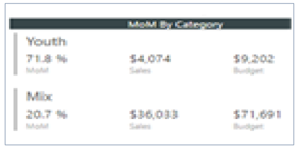
472.94K

Goal: 453.72K (+4.24%)

KPI



Gauge



MoM By Category

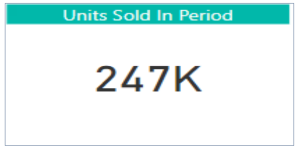
Youth

71.8 % MoM | $4,074 Sales | $9,202 Budget

Mix

20.7 % MoM | $36,033 Sales | $71,691 Budget

Multi-row Card



Units Sold In Period

247K

Card

Geographic

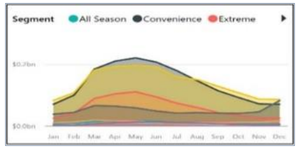
NORTH
AMERICA
Atlantic
Ocean

Map

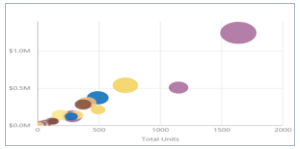
NORTH
AMERICA
EUROPE
Atlantic
Ocean
Pacific
bing

Filled Map

Relationships

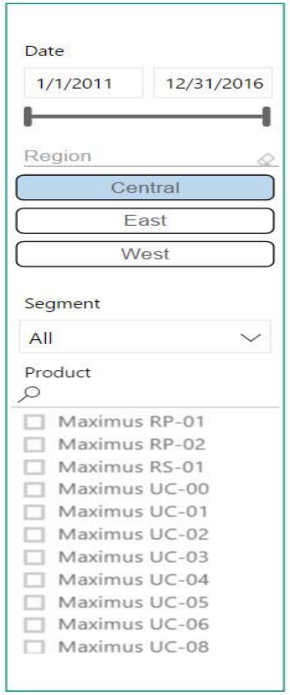


Area



Bubble

Slicing/Filtering



Date

1/1/2011 12/31/2016

Region

Central

East

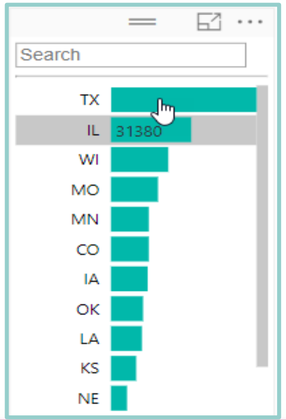
West

Segment

All

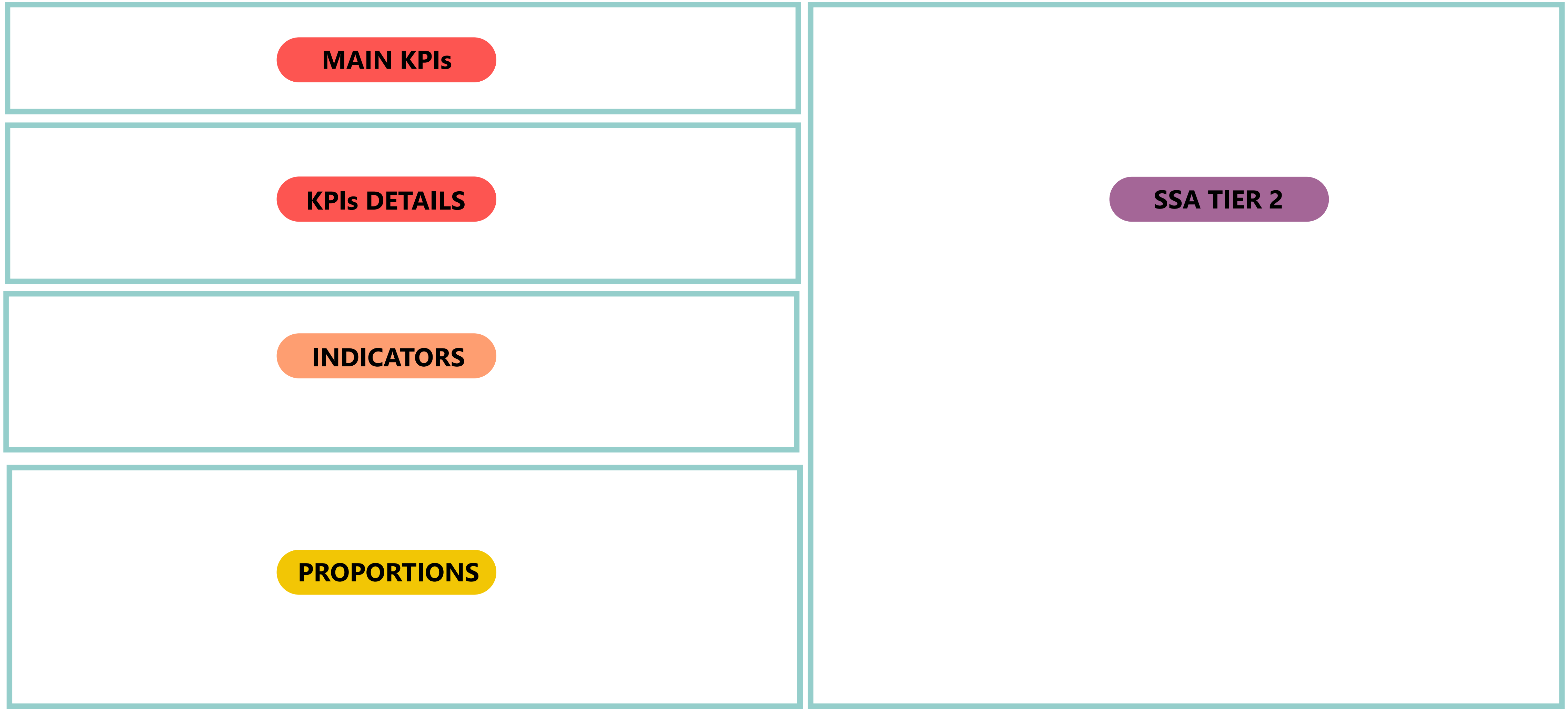
Product

□ Maximus RP-01
□ Maximus RP-02
□ Maximus RS-01
□ Maximus UC-00
□ Maximus UC-01
□ Maximus UC-02
□ Maximus UC-03
□ Maximus UC-04
□ Maximus UC-05
□ Maximus UC-06
□ Maximus UC-08



Search

Dashboard layout



MAIN KPIs

KPIs DETAILS

INDICATORS

PROPORTIONS

SSA TIER 2

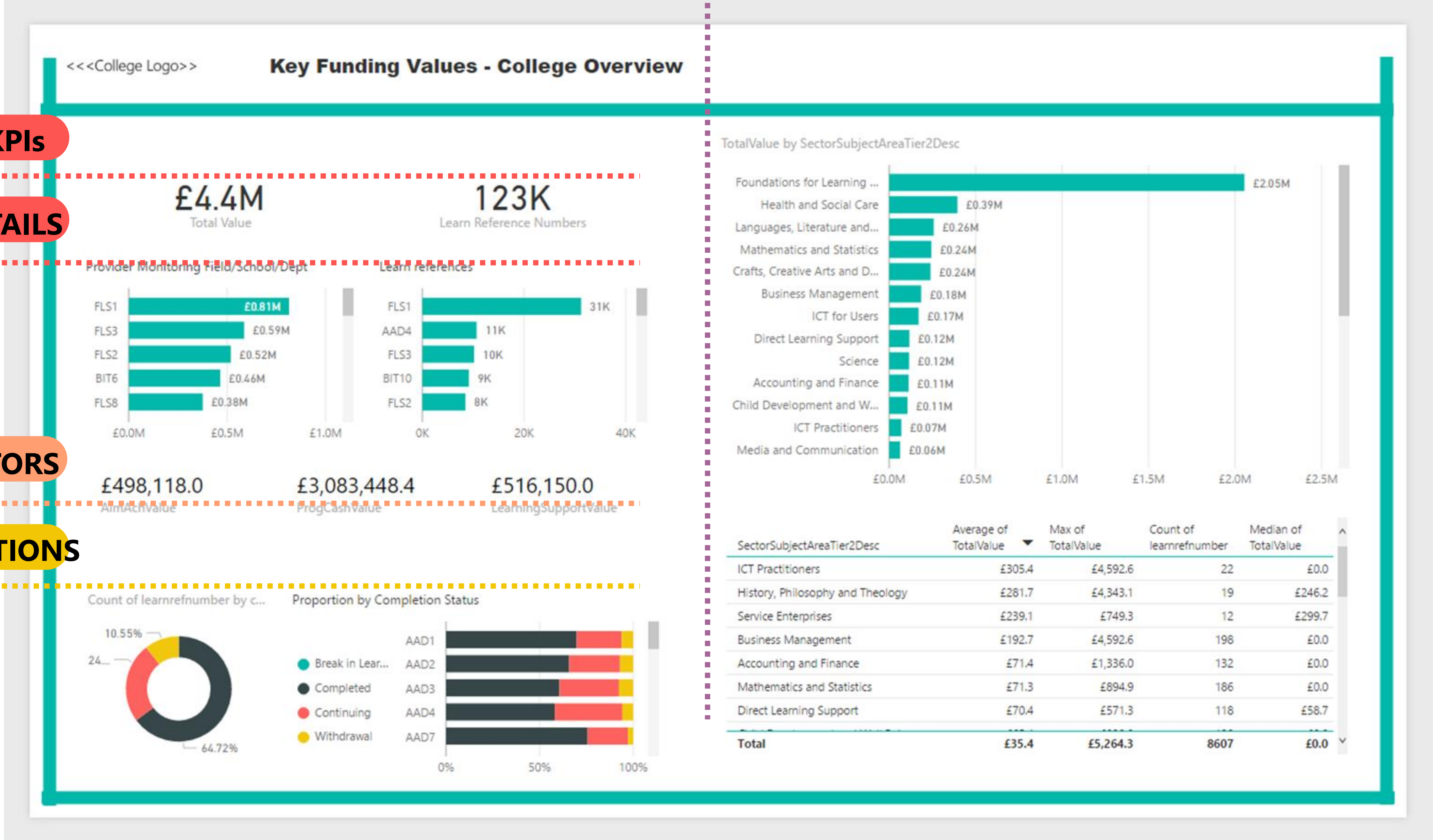
SSA TIER 2

MAIN KPIs

KPIs DETAILS

INDICATORS

PROPORTIONS



<<<College Logo>>>

Key Funding Values - College Overview

£4.4M
Total Value

123K
Learn Reference Numbers

£498,118.0
AimAchValue

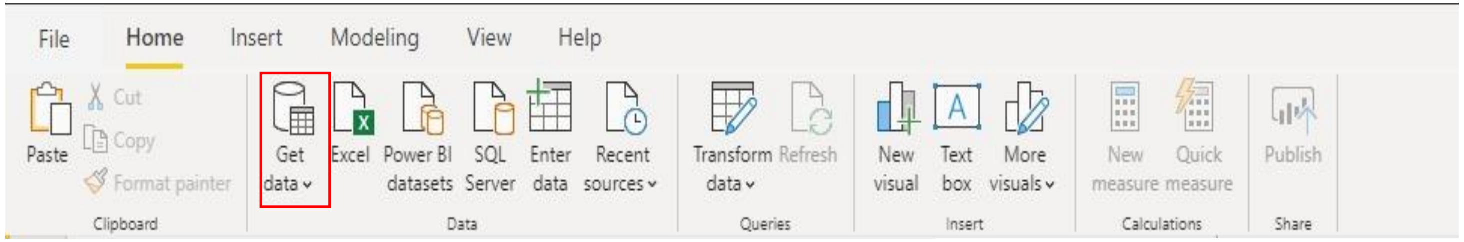
£3,083,448.4
ProgCashValue

£516,150.0
LearningSupportValue

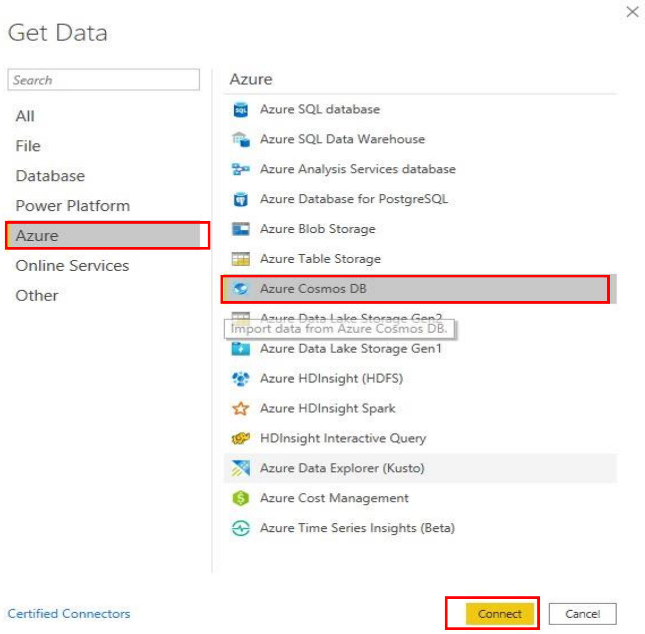
SectorSubjectAreaTier2Desc	Average of TotalValue	Max of TotalValue	Count of learnrefnumber	Median of TotalValue
ICT Practitioners	£305.4	£4,592.6	22	£0.0
History, Philosophy and Theology	£281.7	£4,343.1	19	£246.2
Service Enterprises	£239.1	£749.3	12	£299.7
Business Management	£192.7	£4,592.6	198	£0.0
Accounting and Finance	£71.4	£1,336.0	132	£0.0
Mathematics and Statistics	£71.3	£894.9	186	£0.0
Direct Learning Support	£70.4	£571.3	118	£58.7
Total	**£35.4**	**£5,264.3**	**8607**	**£0.0**

Lab Instructions

1. Click on Get Data in the Home tab



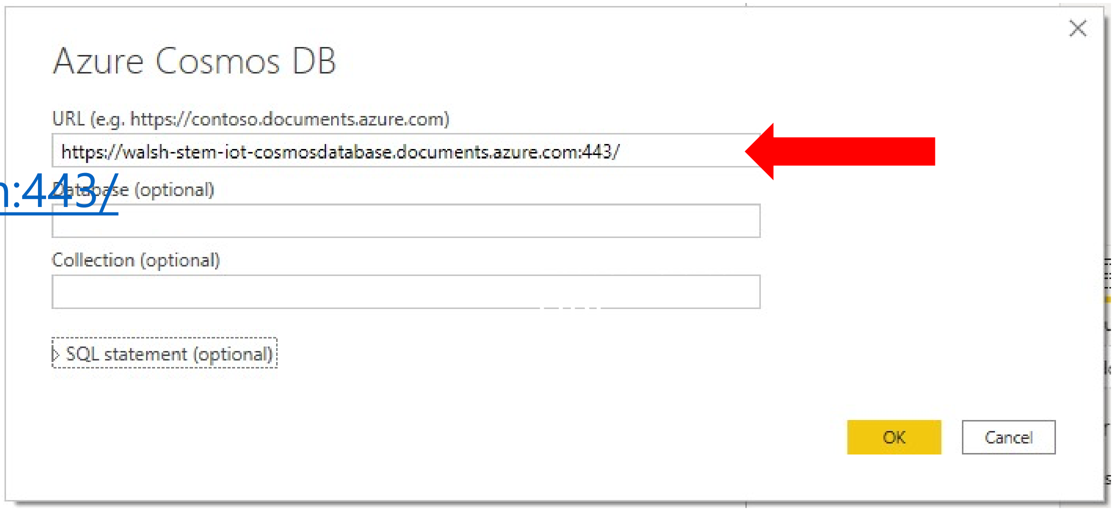
2. Select Azure
3. Select Azure Cosmos DB
4. Click Connect



Lab Instructions

5. Enter the URL for the database

https://walsh-stem-iot-cosmosdb.documents.azure.com:443/



Azure Cosmos DB

URL (e.g. https://contoso.documents.azure.com)

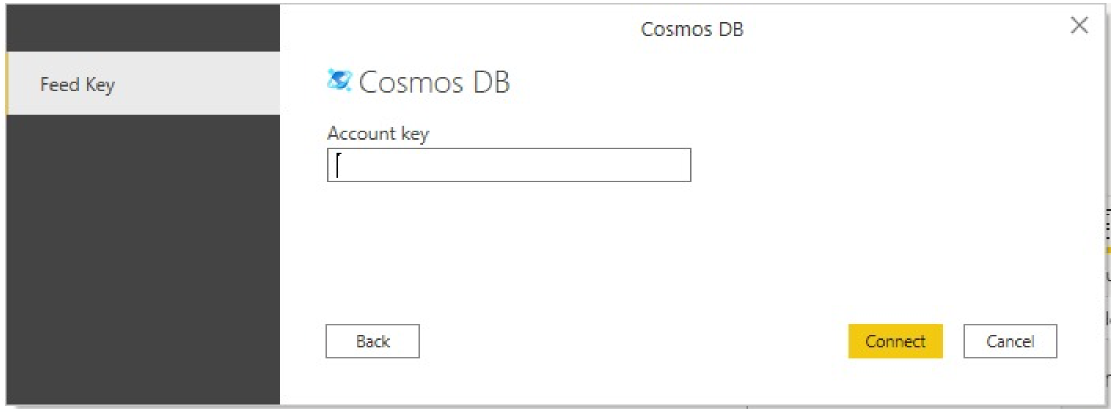
https://walsh-stem-iot-cosmosdatabase.documents.azure.com:443/

Database (optional)

Collection (optional)

6. Enter the Account key

coCbX3hLmzVwxv9ZbJ85Ay4aF8tSlaHOfHtTFHK3f8da9cOzZ5Opyjhw7COJMj55xEN0UauGD83vw3eyNIhOXA==



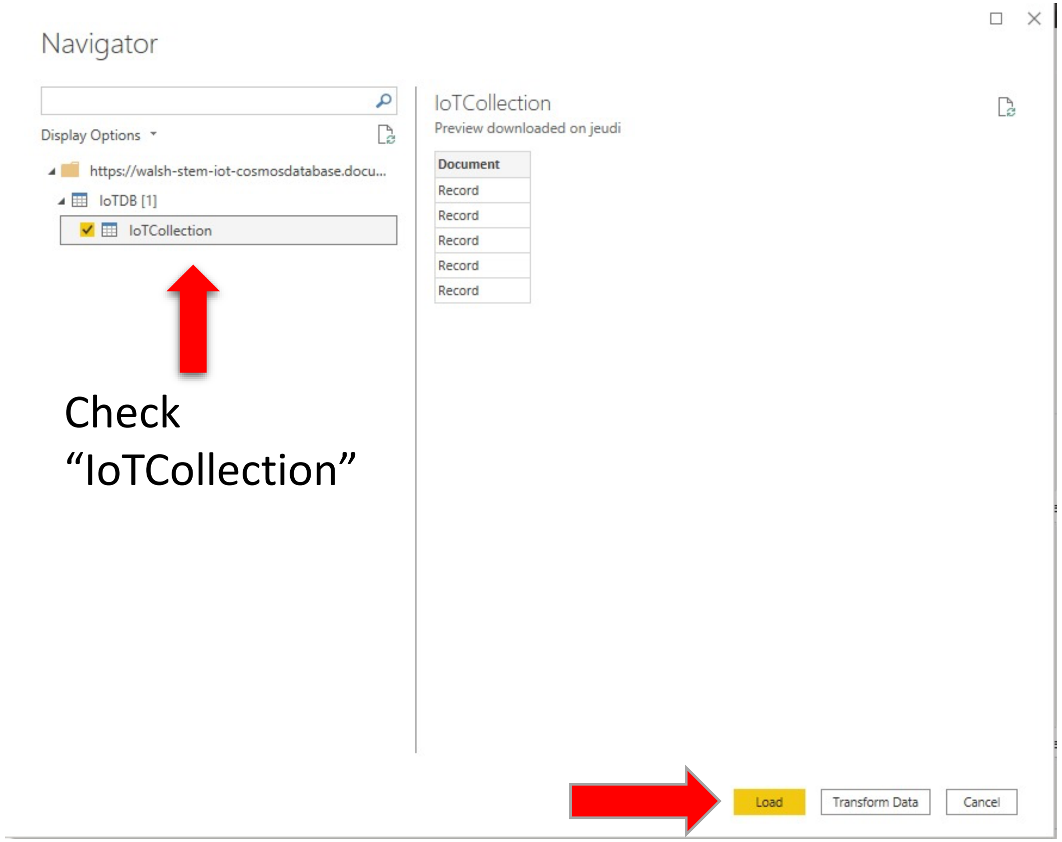
Cosmos DB

Cosmos DB

Account key

Lab Instructions

7. Check IoTCollection
8. Click Load



Navigator

Display Options

- https://walsh-stem-iot-cosmosdatabase.docu...
  - IoTDB [1]
    - ☑ IoTCollection

IoTCollection

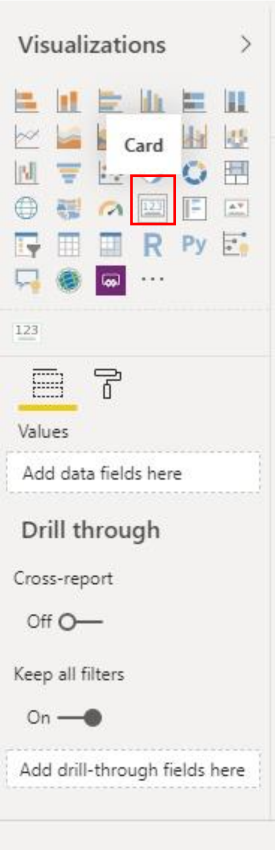
Preview downloaded on jeudi

Document
Record
Record
Record
Record
Record

Check “IoTCollection”

Lab Instructions

9. Click on card to show KPI



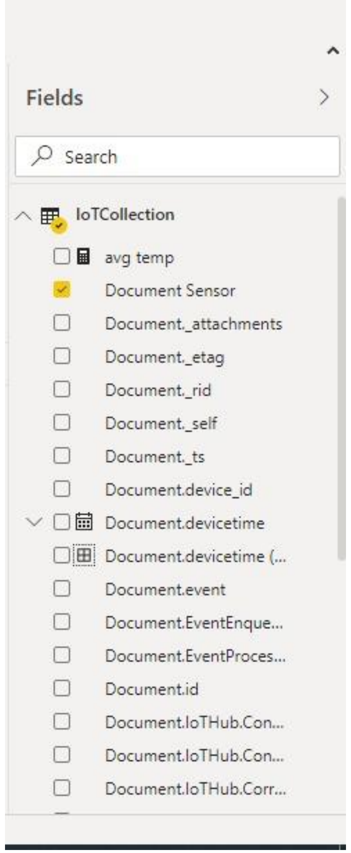
Cross-report

Off ○—

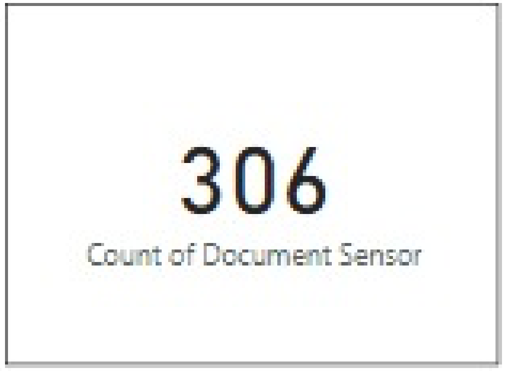
Keep all filters

On —●

10. Select the field to show the KPI



□ avg temp
☑ Document Sensor
□ Document._attachments
□ Document._etag
□ Document._rid
□ Document._self
□ Document._ts
□ Document.device_id
□ Document.devicetime
□ Document.devicetime (...
□ Document.event
□ Document.EventEnque...
□ Document.EventProces...
□ Document.id
□ Document.IoTHub.Con...
□ Document.IoTHub.Con...
□ Document.IoTHub.Corr...

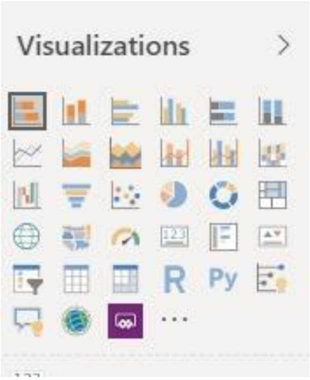


306

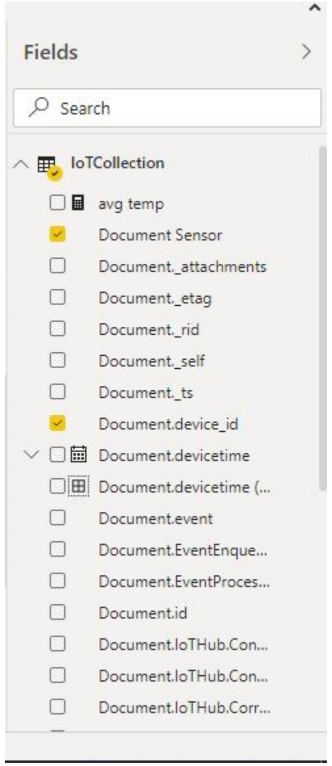
Count of Document Sensor

Lab Instructions

11. Click on “Stacked bar chart”



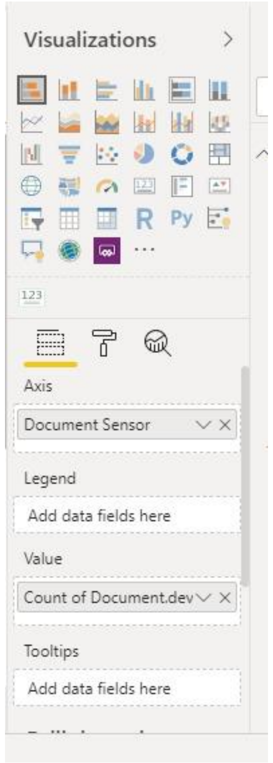
12. Select the fields ”Document Sensor” and “Document.device_id”


Fields

Search

IoTCollection

- ☐ avg temp
- ☑ Document Sensor
- ☐ Document._attachments
- ☐ Document._etag
- ☐ Document._rid
- ☐ Document._self
- ☐ Document._ts
- ☑ Document.device_id
- ☐ Document.devicetime
- ☐ Document.devicetime (...
- ☐ Document.event
- ☐ Document.EventEnque...
- ☐ Document.EventProces...
- ☐ Document.id
- ☐ Document.IoTHub.Con...
- ☐ Document.IoTHub.Con...
- ☐ Document.IoTHub.Corr...



Axis

Document Sensor

Legend

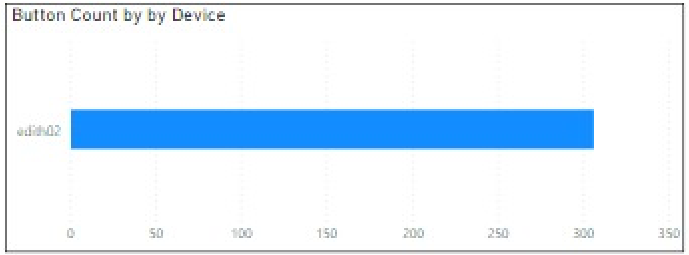
Add data fields here

Value

Count of Document.dev

Tooltips

Add data fields here



Power Bi Resource

Contact Support

Report Errors, Issues – Support.PowerBI.com

Resources

- ➢ *Community.PowerBI.com – Community Forum*
- ➢ *Report Theme Gallery – A showcase for stunning report themes*
- ➢ *Data Stories Gallery – Get inspired with Data Stories by other Power BI users*
- ➢ *R-Visuals Gallery – Get inspired by others use of R for analyzing their data*
- ➢ *Store.office.com – Custom PBI visuals and R visuals you can download and use in your story*

- ➢ *Power BI Blog - weekly updates*
- ➢ *User Voice for Power BI – Vote on (or submit) your favorite new ideas for Power BI*
- ➢ *Issues.PowerBI.Com – log issues with the community*
- ➢ *Whitepaper - Creating an Enterprise Class Dashboard Solution with Power BI*
- ➢ *Guided Learning Self Service Power BI training*

- ➢ *DAX Formula Language – syntax for DAX*
- ➢ *DAX Patterns – Great website to learn new patterns for the DAX Language*
- ➢ *Power Query Formula Language – syntax for the “Query” language*
- ➢ *Paletton.com – a color scheme generator*
- ➢ *https://unicode-table.com/en/ – Unicode Character Table*
- ➢ *Theme Generator*
- ➢ *Contrast Analyzer: a tool that creates a “lens” to show how people with different visual disabilities might see your reports*
- ➢ Charticulator: *a tool that helps to build custom visuals*

Appendix

Pre-Lab Setup

1. Plugin mouse into keyboard and keyboard into black usb on Raspberry Pi
2. Plugin micro-HDMI into Raspberry Pi and HDMI into monitor or tv
3. Plugin power into Raspberry Pi and into power outlet
4. Connect to home wifi (bottom right click on wifi icon 📶)
5. Prepare Lab Day via Option 1(preferred setup) or Option 2 (ensure you print labs)

*Walsh IT Department will be available for consultation prior to start of program with questions and to confirm system requirements and proper set-up. Call Help Desk at 330-244-4357 or helpdesk@walsh.edu

** Watch Video setup from the following url https://www.walsh.edu/skilled-tech-requirements.html

Lab Day – Setup Option 1

- Preferred Setup with external monitor or TV
- Raspberry Pi is connected to external monitor or TV
- Laptop will be connected to Zoom call

Lab Day – Setup Option 2

- Use this option only if you do NOT have an external monitor
- Laptop is connected to Zoom
- Laptop is connected Raspberry Pi via VNC viewer*
  - Enter walsh device id(label on side of Raspberry Pi) into VNC viewer
- Raspberry Pi is connected to power outlet

https://www.realvnc.com/en/connect/download/viewer/

Linux Commands

Command	Description
ls	The ls command lists the content of the current directory (or one that is specified). It can be used with the -l flag to display additional information (permissions, owner, group, size, date and timestamp of last edit) about each file and directory in a list format. The -a flag allows you to view files beginning with . (i.e. dotfiles).
cd	Using cd changes the current directory to the one specified. You can use relative (i.e. cd directoryA) or absolute (i.e. cd /home/pi/directoryA) paths.
pwd	The pwd command displays the name of the present working directory: on a Raspberry Pi, entering pwd will output something like /home/pi.
mkdir	You can use mkdir to create a new directory, e.g. mkdir newDir would create the directory newDir in the present working directory.
rmdir	To remove empty directories, use rmdir. So, for example, rmdir oldDir will remove the directory oldDir only if it is empty.
rm	The command rmremoves the specified file (or recursively from a directory when used with -r). Be careful with this command: files deleted in this way are mostly gone for good!
cp	Using cp makes a copy of a file and places it at the specified location (this is similar to copying and pasting). For example, cp ~/fileA /home/otherUser/ would copy the file fileA from your home directory to that of the user otherUser (assuming you have permission to copy it there). This command can either take FILE FILE (cp fileA fileB), FILE DIR (cp fileA /directoryB/) or -r DIR DIR (which recursively copies the contents of directories) as arguments.

Linux Commands

Command	Description
mv	The mv command moves a file and places it at the specified location (so where cp performs a 'copy-paste', mv performs a 'cut-paste'). The usage is similar to cp. So mv ~/fileA /home/otherUser/ would move the file fileA from your home directory to that of the user otherUser. This command can either take FILE FILE (mv fileA fileB), FILE DIR (mv fileA /directoryB/) or DIR DIR (mv /directoryB /directoryC) as arguments. This command is also useful as a method to rename files and directories after they've been created.
cat	You can use cat to list the contents of file(s), e.g. cat thisFile will display the contents of thisFile. Can be used to list the contents of multiple files, i.e. cat *.txt will list the contents of all .txt files in the current directory.
tail	The opposite of head, tail displays the end of a file. The starting point in the file can be specified either through -b for 512 byte blocks, -c for bytes, or -n for number of lines.
scp	The scp command copies a file from one computer to another using ssh. For more details see SCP (secure copy)
sudo	The sudo command enables you to run a command as a superuser, or another user. Use sudo -s for a superuser shell. For more details see Root user / sudo

Docker Commands

Command	Description
docker ps -a	Lists of all existing containers (not only running)
docker stop $(docker ps -a -q)	Stop all running containers
docker rm $(docker ps -a -q)	Delete all existing containers
docker rmi $(docker images -q -a)	Delete all existing images
sudo docker exec -i -t 878978547 bash	Attach to a running container
docker logs -f 878978547	Using Docker logs command

*change highlighted section based on your container. Use docker ps –a to see the ID