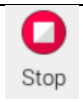
Walsh IoT Lab Cheat Sheet

Icons

						
Figure 1 – Terminal Icon	*Figure 2 - Raspberry Pi Icon*	*Figure 3 - Run Icon*	*Figure 4 - Minimize Icon*	*Figure 5 - Stop Icon*	*Figure 6 - Copy Icon*	*Figure 7 – Trash Icon*

Lab 1 - Raspberry Pi Basics

Exercise 1 - Remote into Raspberry Pi

1. Click terminal icon (Figure 1 – 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
    b. Note: The password will not show when you type into the terminal
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

Exercise 2 – Sense HAT Emulator

1. Click Raspberry Pi icon(Figure 2 - 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 (Figure 3 - Run Icon)

5. Click Minimize icon (Figure 4 - 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
8. Close all windows by clicking X icon

Exercise 3 – Open walsh-stem-iot solution

1. Click Raspberry Pi icon(Figure 2 - 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. 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

Exercise 1 – Raspberry Pi Simulator

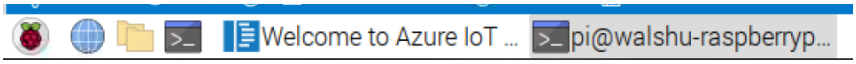
1. Navigate to Azure Raspberry Pi Simulator
    a. https://azure-samples.github.io/raspberry-pi-web-simulator/
2. Navigate to Azure Portal in new tab or additional browser window
    a. https://portal.azure.com
    b. username - stem01@walsh.edu password – RaspberryPi01!
3. Get IoT device connection 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(Figure 6 - 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

Exercise 2 – Raspberry Pi end-to-end solution

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
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
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

16. Click File Save All
17. Right click deployment.template.json
    a. Click Build and Push IoT Edge Solution
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 (Figure 1 – 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**
23. Click Minimize icon (Figure 4 - 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

Exercise 3 – Send IoT Messages to IoT Hub

1. Follow Steps from Lab 1 – Exercise 3
    a. **Note**: Skip if you completed Lab 2 Exercise 2
2. Click terminal icon (Figure 1 – Terminal Icon) on Raspberry Pi
3. Type **docker logs ButtonModule -f**
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 terminal window

Exercise 4 – Invoke Module Direct Method

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
4. Click terminal icon (Figure 1 – Terminal Icon) on Raspberry Pi
5. Type **docker logs ButtonModule -f**
6. Press button sensor
7. Review terminal window
    a. You should notice the temperature you set

Lab 3 - Azure Basics

Exercise 1 – Add Temperature Module

1. Navigate to Azure Portal in browser window
    a. https://portal.azure.com
    b. username - stem01@walsh.edu password – RaspberryPi01!
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. Click terminal icon (Figure 1 – Terminal Icon)
4. Type **docker ps -a**
5. Type **docker logs SimulatedTemperatureSensor -f**
6. Remove Simulated Temperature Sensor

a. Click Set Modules (switch back to browser window that has Azure Portal)

b. Click trash icon (Figure 7 – Trash Icon by Simulated Temperature Sensor)

c. Click Review and Create

d. Click Create

Exercise 2 – Create IoT Hub (instructor only)

1. Navigate to Azure Portal in browser window

a. https://portal.azure.com

b. username - stem01@walsh.edu password – RaspberryPi01!

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. Type **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

Lab 4 - Data and Analytics Basics (Instructor only)

Exercise 1 – Visualize IoT data with PowerBi Desktop

1. Open PowerBI
2. Click on Get Data in the Home tab

Walsh IoT Lab Cheat Sheet

3. Select Azure
4. Select Azure Cosmos DB
5. Click Connect
6. Enter the URL for the database
7. https://walsh-stem-iot-cosmosdatabase.documents.azure.com:443/
8. Enter the Account key
9. HPQH0DfohI4CznFasK8vsoC4jd5fFNzXgTPSplcX7ShHghQ8txjkP9jEHJ4BP7pRTY9gZekIRqHuJcwn0ZDzig==
10. Check IoTCollection
11. Click Load
12. Click on card to show KPI
13. Select the field to show the KPI
14. Click on “Stacked bar chart”
15. Select the fields ”Document Sensor” and “Document.device_id”