

# Enterprise Application Development in the Cloud

## Workshop Training

v0.1



---

GRAND CANYON  
UNIVERSITY™

# Introduction



# Expectations

- We will meet 1-2 times a week for an hour.
- All meetings will be held remotely via Zoom Conferencing.
- We will meet for 4-5 weeks.
- We will review the SDK along with the architecture, technologies, and walk through the code for the IoT Reference Applications that have already been implemented.
  - You will then be given resources (tutorials, web sites, and sample code) so that you can then build your own IoT apps in the Cloud.
  - You are expected review the workshop materials and build your apps outside of the workshop.
  - You are expected to setup your own tools and cloud accounts
  - I have setup a Padlet for Questions and Collaboration.
  - We will use the workshop to review questions and answers.

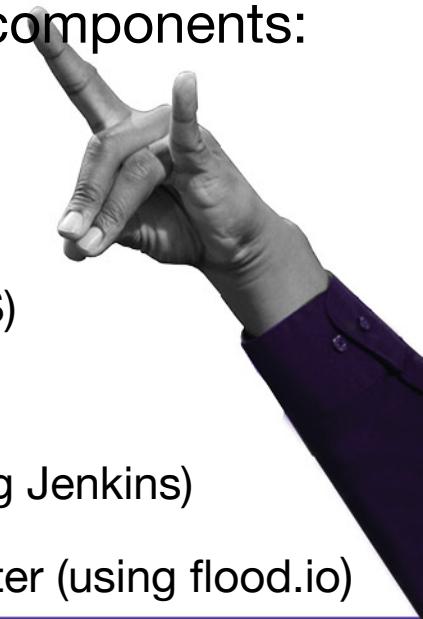
# What New Stuff Will You Learn?

- Spring Framework (Core, Security, JDBC) - alt. to Java EE.
- Maven - Build and Dependency Mgmt. Tool.
- GitHub - Version Control System.
- JAX-RS API - To build a REST based Server App in Java.
- Apache Tomcat Java Server - alt. Java Application Server.
- LavaChart and Guzzle - To build the Reporting App.
- Codenvy - Cloud based IDE (or you can use Eclipse).
- OpenShift - PaaS based Cloud.
- Foundation - To build other IoT and/or Cloud based apps.



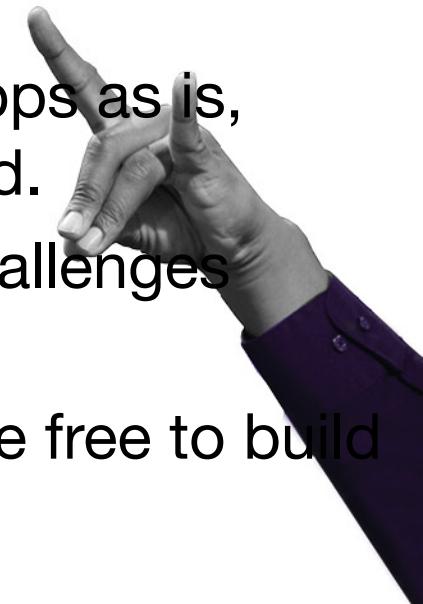
# Workshop Introduction

- The next generation Development Platform for developing Enterprise Applications will likely be browser and most certainly cloud based.
- This Workshop will demonstrate what this Development Platform will look like and give you a hands on opportunity to experience this platform.
- The Development Platform will consist of the following components:
  - ★ Cloud based IDE - Codenvy
  - ★ Cloud based Development Runtimes - Codenvy
  - ★ Cloud based Production Runtimes - Redhat OpenShift (on PaaS)
  - ★ Cloud based Source Control System - Github
  - ★ Cloud based Automated Build System - Redhat Openshift (using Jenkins)
  - ★ Cloud based DevOps Automation - Jenkins, JUnit, Maven, JMeter (using flood.io)



# More Expectations

- You are free to leverage my Raspberry Pi as I can post IoT data to your IoT Services App or you can just use your own Pi.
  - My Raspberry Pi IoT app is designed to connect to a list of backend Cloud Servers however you will need to build your IoT Services app to the specifications outlined in the SDK.
- You are free to build and deploy the Reference apps as is, study the code, as well as modify them as desired.
- You all know Java so you should not have any challenges learning how to build the IoT Services app.
- If you do not know the Laravel Framework you are free to build the IoT Reporting app in plain PHP.



# Planned Workshop Sessions

- #1a - Overview of SDK and Reference Architecture
- #1b - IoT Device App. (Python)
- #2 - IoT Services App. (JAX-RS, Spring, Maven)
- #3 - Build and deploy the IoT Services App.
- #4 - IoT Reporting App. (PHP, LavaCharts, Guzzle)
- #5 - Build and deploy the IoT Reporting App.
- #6 - Next Steps Etc.



**Questions?  
Are You Ready?**





# Workshop Session 1

- SDK and Reference Architecture:
  - Overview of the Cloud Architecture
  - Overview of the IoT Reference Apps
  - Overview of the Development Tools
  - Overview of SDK structure
- IoT Device Reference Application:
  - Overview of Application Architecture
  - Overview of Application Logic
  - Code Walk Through
- Action Items/Homework



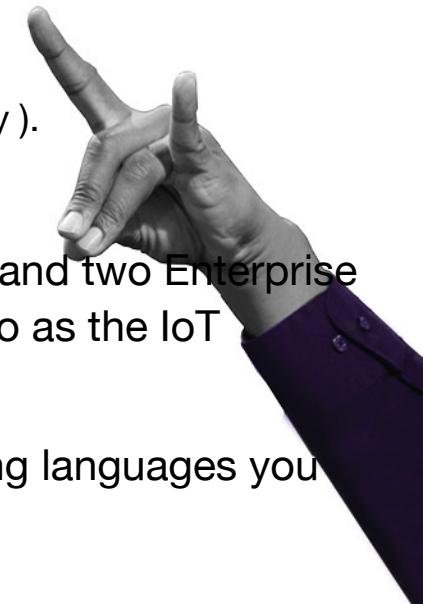
# SDK and Reference Architecture



# Reference SDK

The Workshop will leverage a SDK (Software Development Kit) as a teaching tool:

- ★ The SDK will be used as guide and a reference for the class materials.
- ★ Everything is documented in the SDK using README Markdown files.
- ★ There is JavaDoc (with UML diagrams) and PhpDoc that can be referenced as needed.
- ★ The SDK also provides you with all the scaffolding to get your Cloud development started:
  - ✓ Spring Frame application template (can be cloned from a GIT repository).
  - ✓ Laravel Application application template (can be cloned from a GIT repository ).
  - ✓ Step by step instructions for how to get started.
- ★ A fully functioning end to end system using one IoT Device application and two Enterprise applications are available (in code and operational). These are referred to as the IoT Reference Applications.
- ★ The Reference Applications were written in Python, Java, and PHP using languages you already know and can be used as is or used as example code.



# IoT Reference Applications

The following applications will be built in the Workshop using the Cloud based Development and Deployment Platform:

- ★ IoT Services App - Using Tomcat Application Server, written in Java and leveraging the JAX-RS API, JSON, and the Spring Framework
- ★ IoT Reporting App - Responsive browser based app. written in PHP leveraging Bootstrap (for responsive support), Guzzle (for HTTP Client to REST API), LavaCharts (for charting), and the Laravel Framework
  - \* An IoT Device Reference App. using a Raspberry Pi 3 and Sense HAT, written in Python, will be available for you to use in the Workshop.
  - \* If you build your own IoT Device App. using your own Raspberry Pi then I am assuming you will build this outside of the Workshop.



# IoT Reference Architecture

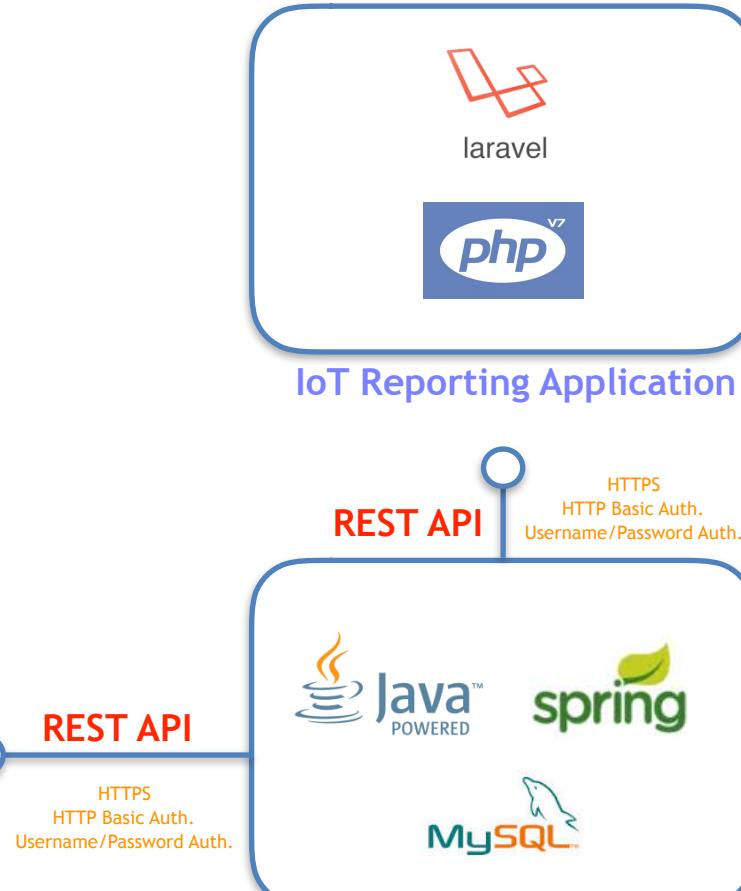
IoT Device Application  
(Sense HAT)



IoT Device Application  
(Raspberry Pi)



IoT Reference Applications



SDK

- Developer Templ.
- Development Docs
- Tools

# Getting the Reference Applications Code

- **IoT Services App:**
  - Go to <https://github.com/markreha/cloudservices>.
    - Clone or download the repository from here.
    - See the README.md Markdown file for an overview.
- **IoT Reporting App:**
  - Go to <https://github.com/markreha/cloudapp>.
    - Clone or download the repository from here.
    - See the README.md Markdown file for an overview.
- **IoT Device App:**
  - Go to <https://github.com/markreha/cloudpi>.
    - Clone or download the repository from here.
    - See the README.md Markdown file for an overview.



# SDK

- The SDK contains all the documentation, tools, etc. to get started building our Cloud based IoT applications.
- Go to <https://github.com/markreha/cloudworkshop>.
  - Clone or download the repository from here.
  - See the README.md markdown file for how to get started using.

Folders of interest:

sdk

└ developer

└ docs

└ tools

Root - See README.md  
App Templates, etc.  
All docs, see README.md  
Misc. doc tools gen. tools



Let's Walk Through the SDK!

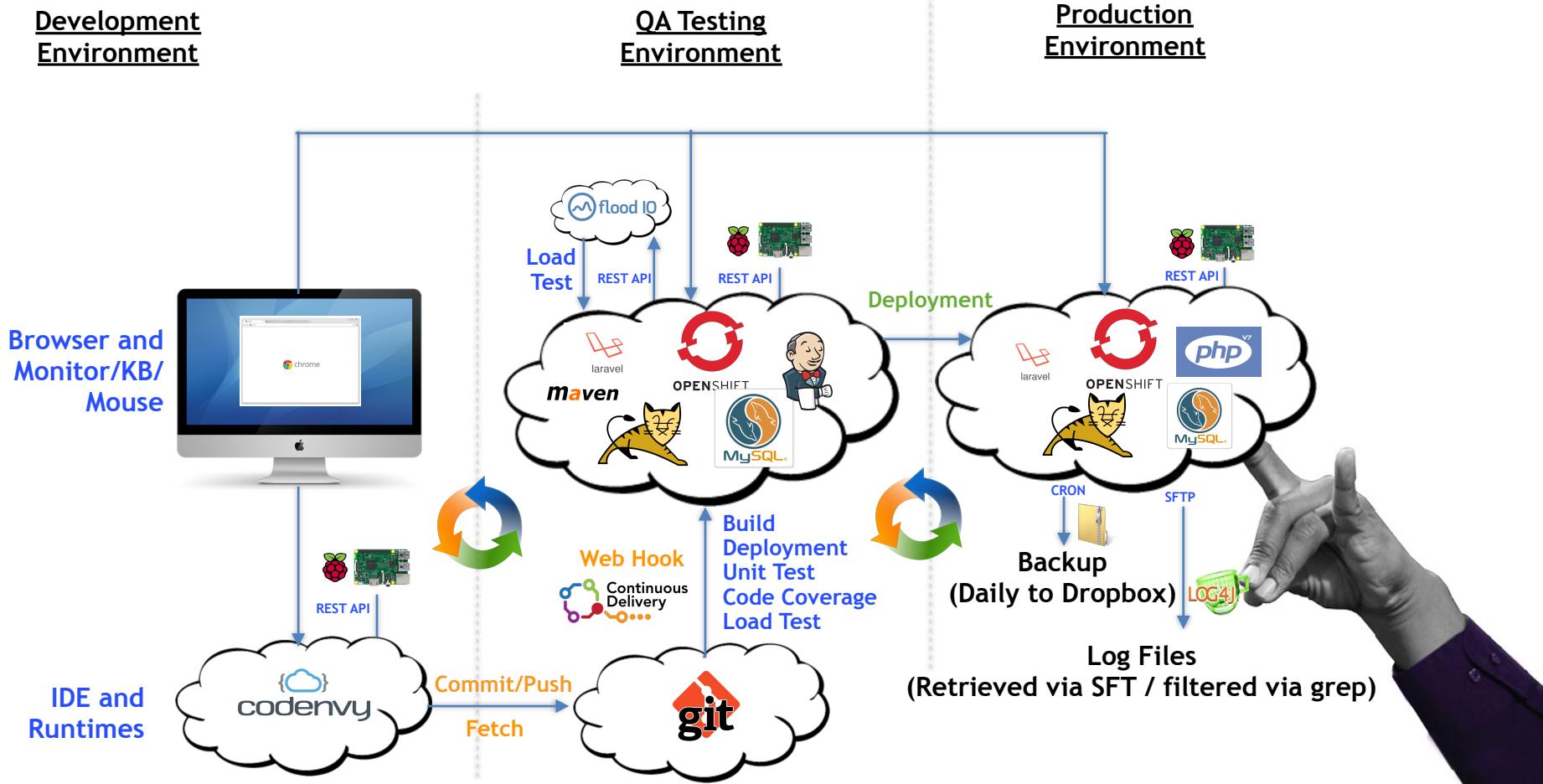
# Development Tools

- \* For the Desktop you need to install and setup a **few tools**.....
  - ✓ Eclipse Neon for EE and Tomcat.
  - ✓ Eclipse Neon for PHP.
  - ✓ MAMP Stack.
- \* For the Cloud you don't need to install **anything**.....
  - ✓ I would encourage you to develop and build some or all of your code using the Cloud based Codenvy IDE.
  - ✓ The beauty of using Codenvy is that you don't need to install anything!
  - ✓ And you can always snapshot your environment if it gets messed up!
- \* You will also need to use a GIT client, such as SourceTree or you can use the browser or you can use the GIT clients built into Eclipse or Codenvy.

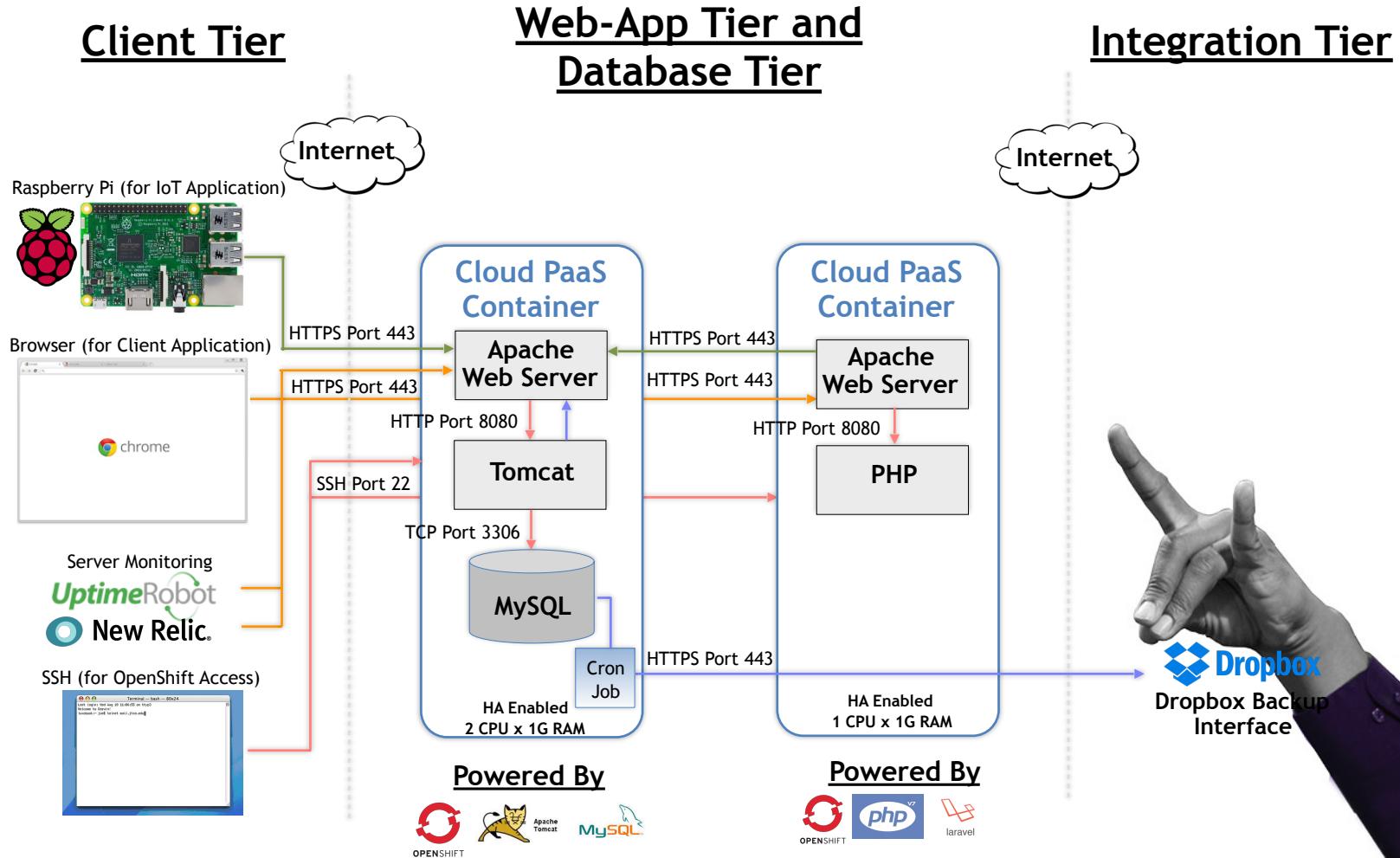
# Deployment Environment

- \* You (or IT) don't need to install anything.....
  - ✓ We will use a Cloud PaaS.
  - ✓ We will use Redhat's Public OpenShift Cloud.
  - ✓ Once you have “procured” your OpenShift Containers, setup your database, and completed some minimal configuration you will be ready to build and deploy your applications in the Cloud.
  - ✓ OpenShift uses Github as a source for all its builds.
    - Uses Maven if you are building a Java app.
    - Uses Composer if you are building a PHP app.
  - ✓ The beauty of the PaaS Cloud is that you (or IT) don't need to install anything to support application development!

# Cloud Based Platform



# Cloud Based Architecture



# IoT Device Reference Application



# IoT Device

- IoT Device is a Raspberry Pi 3 with a Sense HAT.
- IoT Sense HAT features:
  - Temperature, Humidity, and Barometric pressure
  - Gyroscope, Accelerometer, and Magnetometer
  - LED Display
  - Joystick

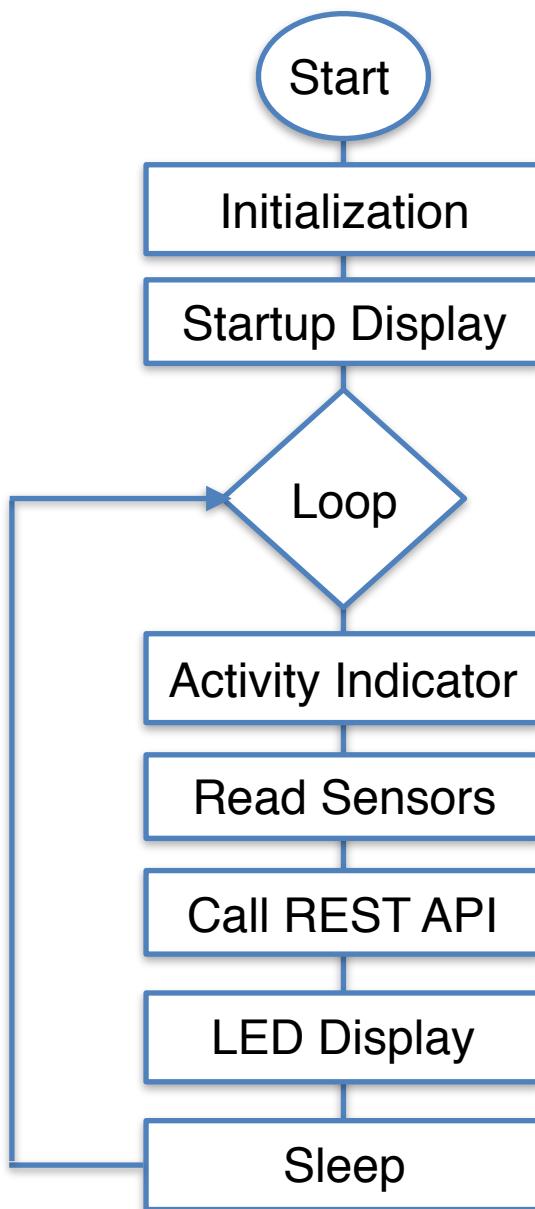


# IoT Device Reference Application 2

- IoT Device Reference Application is written in Python.
  - Polls for Temperature, Humidity, and Barometric pressure IoT Data every 30 minutes
  - Then calls the IoT Services REST API's to save the data.
- IoT Device Reference Application is actually pretty simple because all of the Sense HAT API's are built into the Raspberry Pi OS (Raspbian Linux).
- I did most of the development by connecting to the Raspberry Pi using a Display via HDMI cable and a USB based keyboard and mouse.
- However, now I just connect to the Raspberry Pi remotely using VNC.



# IoT Device Application Logic



Initialize logging, read config file, local variables.

Displays G, C, U on Sense HAT LED Display.  
Display App Startup Message on Console.

Loop Forever.

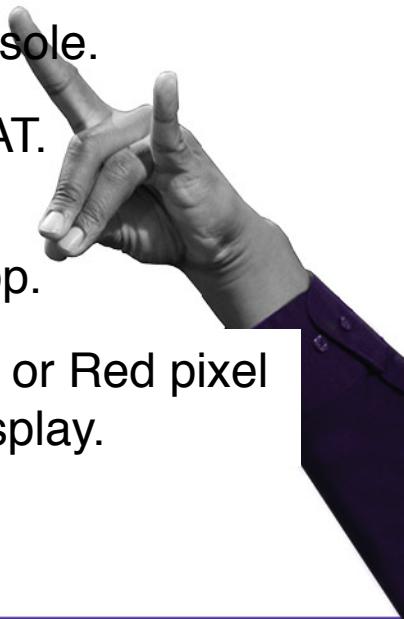
Display Activity Dot (.) on Console.

Read IoT Data from Sense HAT.  
Round and do conversions.

POST data to IoT Services App.

Display Purple pixel (no error) or Red pixel (error) on Sense HAT LED Display.

Sleep for specified time.

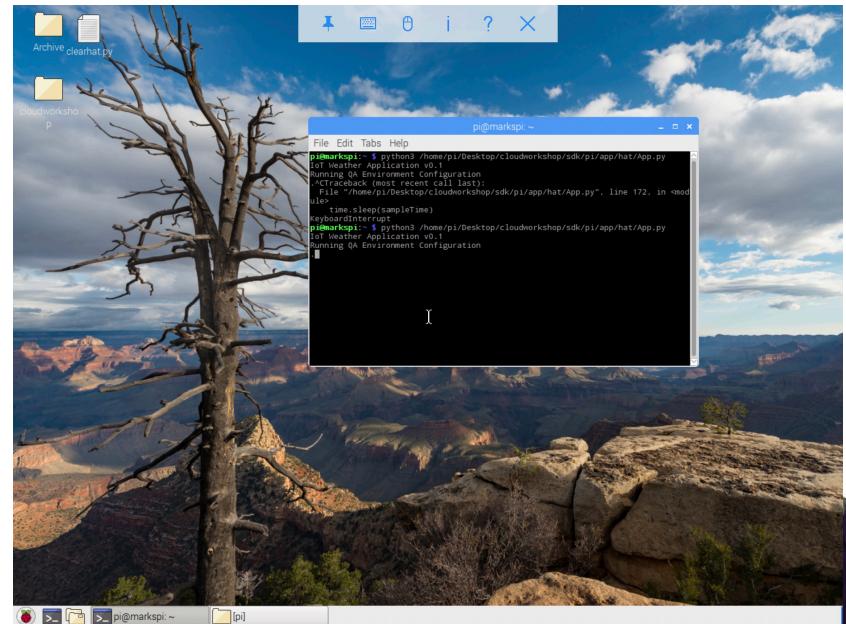


# IoT Device Application Code Walkthrough



Sense HAT

Raspberry Pi 3



Connected via VNC

Let's Walk Through the Code!

# Action Items / Next Steps



# Action Items / Homework

1. Setup the following Cloud accounts:

- Github account (go to <https://github.com>).
- Codenvy account (go to <https://codenvy.com>).
- OpenShift account (go to <https://www.openshift.com>).
  - Be prepared to setup 2 accounts so you can deploy Java/MySQL and PHP apps.

2. Install the following tools if you plan on using the desktop for development (optional if using Codenvy IDE):

- Eclipse EE edition and Tomcat 8.5.
- Eclipse PHP edition.
- MAMP and MySQL Workbench.
- Postman.

3. Start using the SDK:

- Clone the SDK from the ‘cloudworkshop’ Github repository (see Clone Repo slide).
- Review the README.md in the *root* directory of the SDK or the ‘cloudworkshop’ Github repository.
- Review the README.md in the *docs/development* directory in the SDK or the Github repository.

4. Get ready for Workshop Session #2:

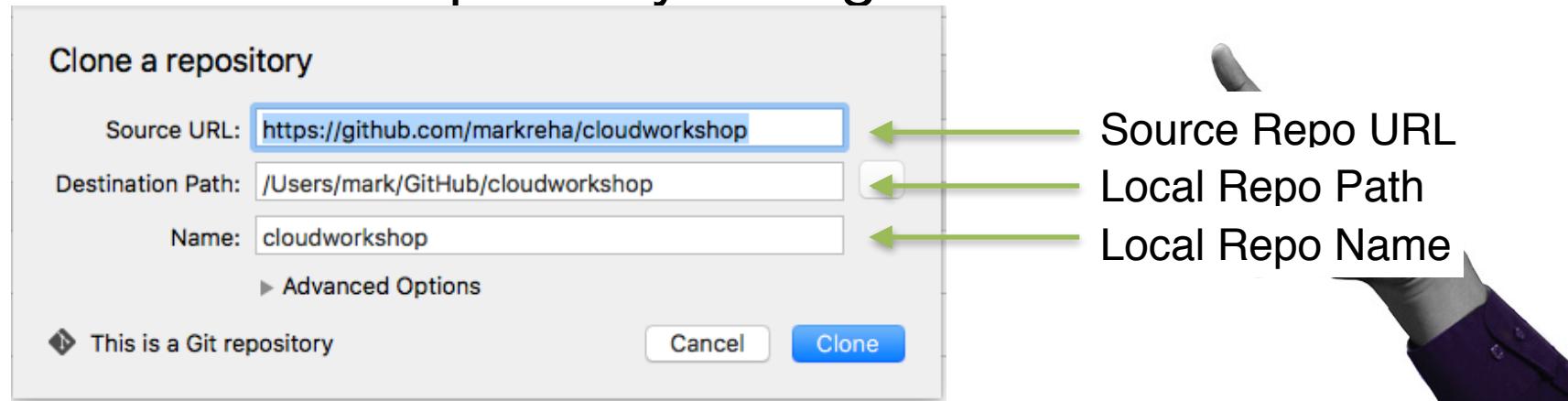
- Go thru Maven Tutorials (see Reference Resources slide).
- Go thru Spring Tutorials (see Reference Resources slide).
- Go thru REST Tutorials. (see Reference Resources slide)
- Clone the IoT Services Reference Application from the ‘cloudservices’ Github repository.
- Review the IoT Services Reference App. code from the ‘cloudservices’ Github repository.

# How Do I Clone the Workshop Repositories?

1. Click the Remote Tab -> +New -> Clone from URL.



2. Fill in the Clone Repository dialog as follows.



3. Click the Clone button.

4. Click the Local Tab. Commit and Push to YOUR Repo's.

# How Do View Local Markdown Files?

1. All of the README files in Github are Markdown files.
2. They are rendered by default if you go to the web version of Github.
3. If you clone the SDK to your local file system you can still view the README Markdown files.
  - You must use the Chrome Browser.
  - Install the Markdown Preview Chrome Plugin from the Chrome Web Store.
  - Use the File->Open File menu in Chrome to navigate and view your Markdown (md) files.



# Reference Materials

- Maven Tutorials:
  - [Tutorial from TutorialsPoint](#)
  - [What Is Maven?](#)
  - [Maven In 5 Minutes](#)
  - [Maven Home Page](#)
- Spring Tutorials:
  - [Tutorial on Spring Core from TutorialsPoint](#)
  - [Tutorial on Spring JDBC from TutorialsPoint](#)
  - [Spring Framework Documentation](#)
  - [Spring Framework Home Page](#)
- REST Tutorials:
  - [Tutorial from TutorialsPoint](#)
  - [Tutorial from Oracle](#)
  - [JAX-RS Specification](#)



# Next Up

- Workshop Session 2 will be scheduled for next week.
- Agenda:
  - Q/A.
  - Review the IoT Services App. Architecture.
  - Review the IoT Services App. code.
  - Start building the IoT Services App. (time permitting).
- We will be monitoring our Padlet for questions.
  - Post your tool/account questions in the Padlet.
  - Post your technical questions in the Padlet.





# Workshop Session 2

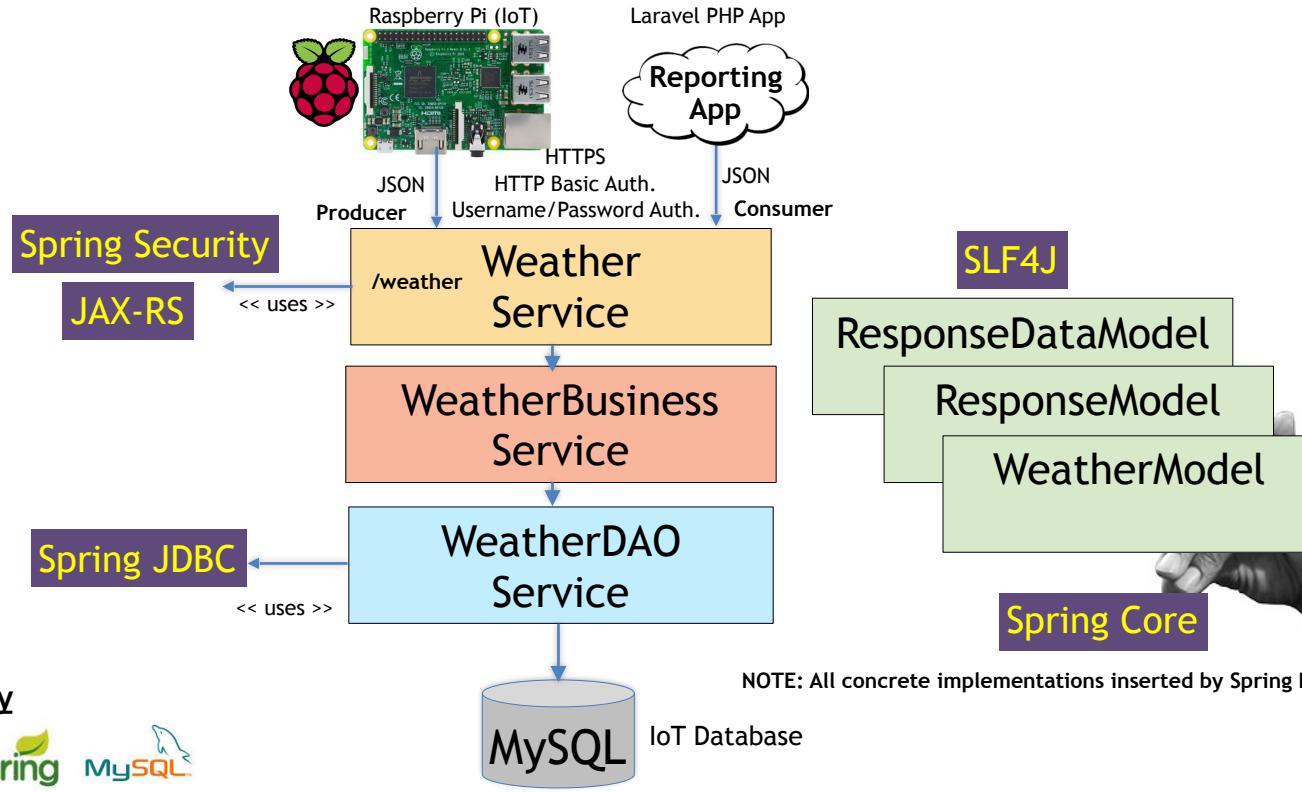
- IoT Device Reference Application:
  - Overview of the Application Architecture
  - Overview of Maven
  - Overview of REST API's and JAX-RS
  - Code Walk Through
  - Development Tooling Setup and Build
  - Cloud Deployment
- Action Items/Homework



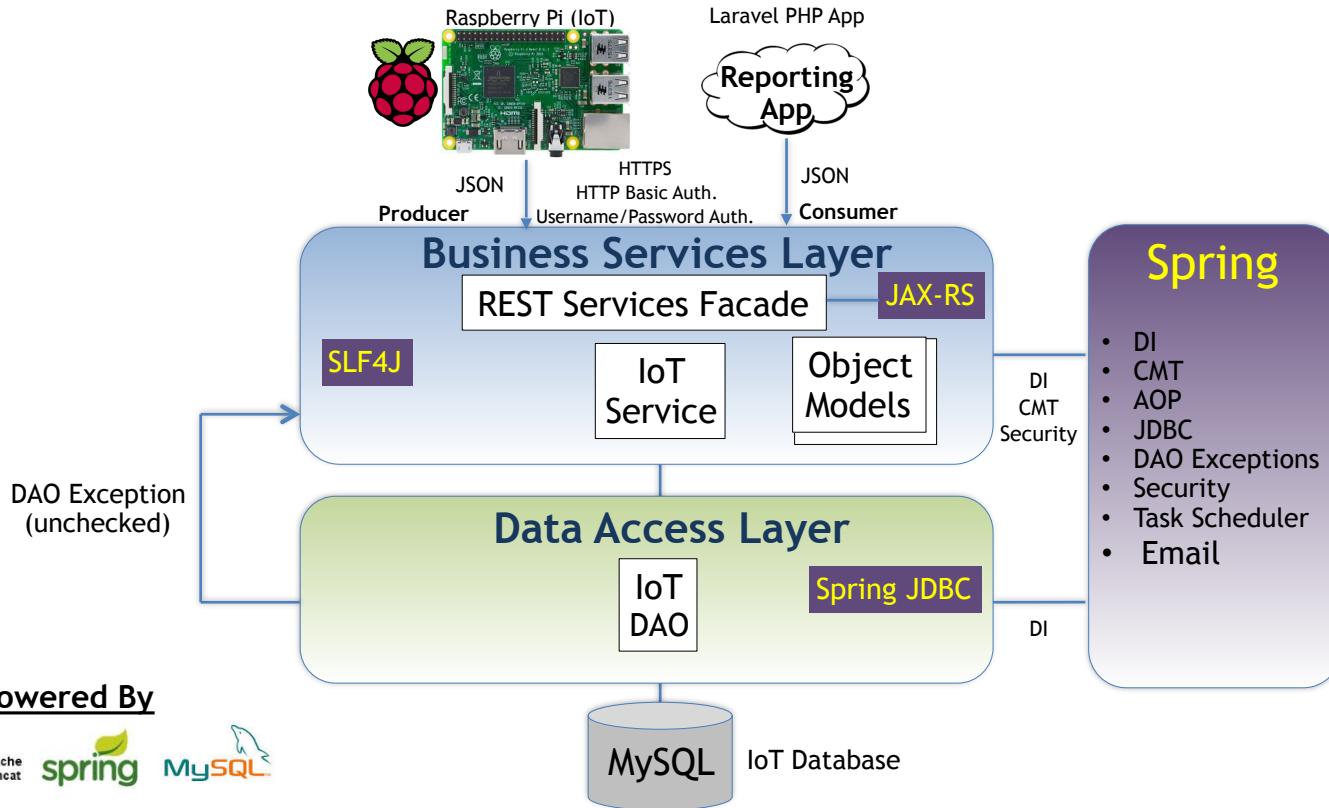
# Application Architecture



# IoT Services App Logical Architecture



# IoT Services App Logical Architecture



# Action Items / Next Steps





# Workshop Session 3

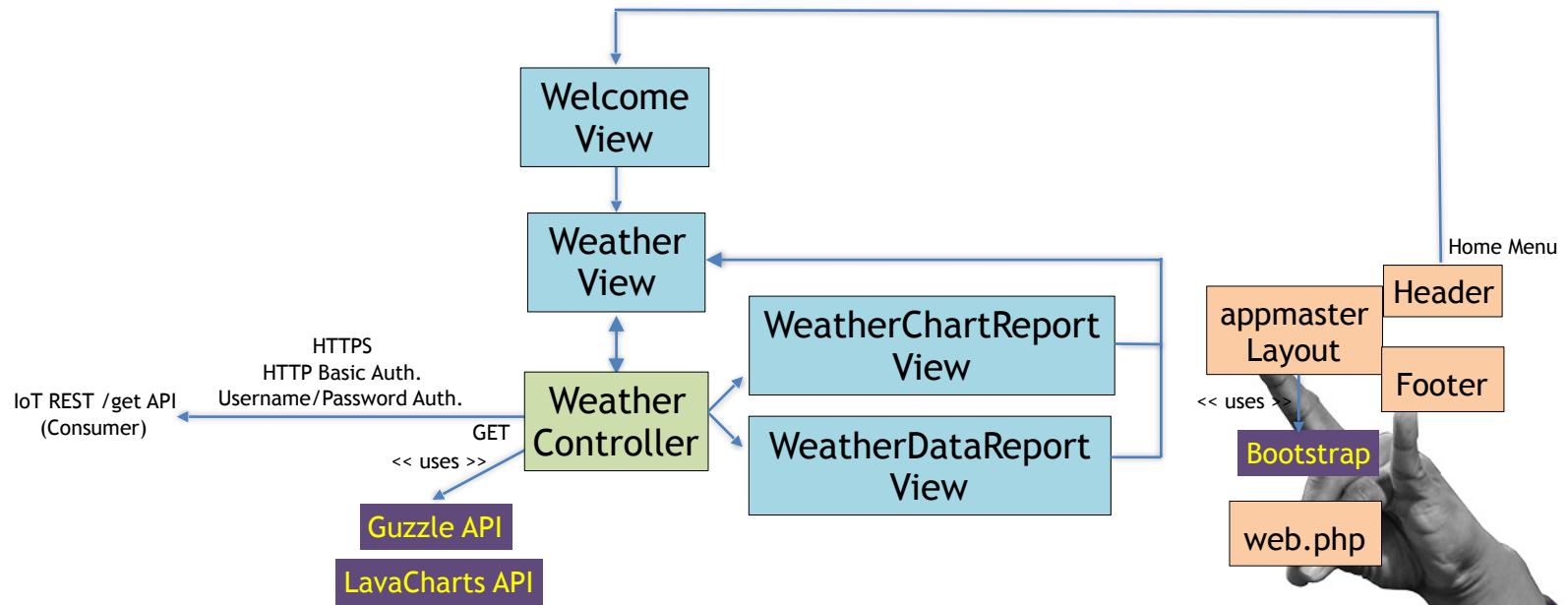
- IoT Reporting Reference Application:
  - Overview of the Application Architecture
  - Overview of Composer and Dependent Libraries
  - Code Walk Through
  - Development Tooling Setup and Build
  - Cloud Deployment
- Action Items/Homework



# Application Architecture



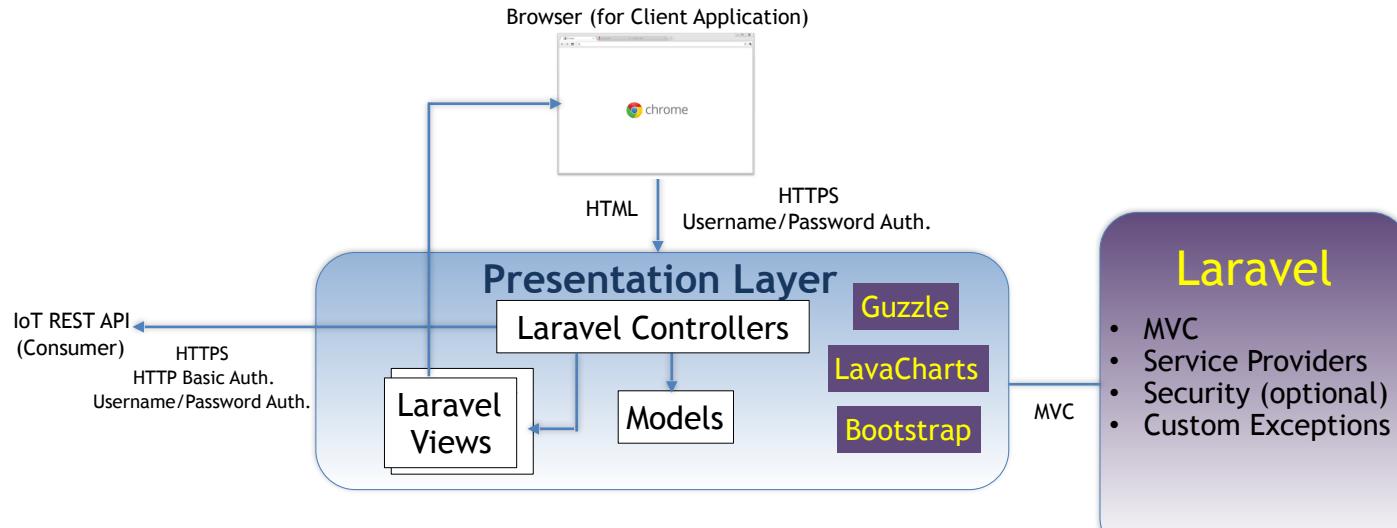
# IoT Reporting App Logical Architecture



Powered By



# IoT Reporting App Logical Architecture



NOTE: No backend services are required for this application!

Powered By



# Action Items / Next Steps

