On The Edge Software Consulting

Always On The Edge of Technology

Enterprise Application Development in the Cloud Workshop Training

Copyright© 1998-2018 On The Edge Software Consulting

v0.6

Welcome / Introduction

Copyright© 1998-2018 On The Edge Software Consulting

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 could consist of the following Cloud based components:
  - ★ IDE
  - ★ Development Runtimes
  - ★ Production Runtimes
  - ★ Source Control System
  - ★ Automated DevOps and Build System

Copyright© 1998-2018 On The Edge Software Consulting

Expectations

- ✓ We will meet 1-2 times a week for 1-2 hours.
- ✓ We will meet for 4 weeks.
- ✓ All meetings will be held remotely via Zoom Conferencing.
  - We will review a Cloud based architecture, technologies, and walk through the code for an IoT Reference Application.
  - You will be given basic instruction and also the resources (tutorials, web sites, and sample code) so that you can then build your own IoT apps using Cloud technologies and platforms.
  - You are expected review the workshop materials and build your IoT apps. outside of the workshop.
  - You are expected to setup your own tools and cloud accounts.
  - I have setup a Google Group and Padlet for Questions and Collaboration.
  - We will also use the workshop to review questions and answers.

Copyright© 1998-2018 On The Edge Software Consulting

What New Stuff Will You Learn?

Domain	Details
Java	Spring Framework (Core, Security, JDBC) - Alternative to Java EE.
Java	JAX-RS API - To build a REST API based Server App in Java.
Java	Apache Tomcat Java Server / Redhat Wildfly- Java Application Server.
Java	Maven - Build and Dependency Management Tool.
PHP	LavaChart, Plotly, and Guzzle - Libraries to build the Reporting App.
PHP	Composer - Dependency Management Tool.
Cloud	GitHub - Cloud based Version Control System.
Cloud	Codenvy - Cloud based IDE (or you can use Eclipse).
Cloud	OpenShift - PaaS based Cloud.

Use this experience as foundation to build other apps or for your Capstone project!

Copyright© 1998-2018 On The Edge Software Consulting

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.

Copyright© 1998-2018 On The Edge Software Consulting

Planned Workshop Sessions

Session	Topic	Week - Day
1	Overview of SDK, Reference Architecture, and the IoT Device App. (Python)	1 - Tues.
2	IoT Services App. (JAX-RS, Spring, Maven)	2 - Tues.
3	Build and deploy the IoT Services App.	2 - Thurs.
4	IoT Reporting App. (PHP, LavaCharts, Guzzle	3 - Tues.
5	Build and deploy the IoT Reporting App.	3 - Thurs.
6	General Q/A and Troubleshooting	4 - Mon.

Copyright© 1998-2018 On The Edge Software Consulting

Questions?
Are You Ready?

Copyright© 1998-2018 On The Edge Software Consulting

Workshop Session 1

Copyright© 1998-2018 On The Edge Software Consulting

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**

Copyright© 1998-2018 On The Edge Software Consulting

SDK and Reference Architecture

Copyright© 1998-2018 On The Edge Software Consulting

Reference SDK

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

The SDK will be used as guide and a reference for the workshop 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 imported into Eclipse).
- Laravel Application application template (can be imported into Eclipse).
- 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.

Copyright© 1998-2018 On The Edge Software Consulting

Reference Applications

- The following applications will be built in the Workshop using the Cloud based Development Platform:
  - ★ IoT Services App - Using Apache Tomcat or Redhat Wildfly or Apache Jetty written in Java leveraging JAX-RS, JSON, and the Spring Framework
  - ★ Reporting App - Responsive browser based written in PHP leveraging Bootrap (for responsive support), Guzzle (for HTTP Client for REST API), LavaCharts (for charting), and the Laravel Framework
    - ✳ An IoT App using a Raspberry Pi 3 written in Python consuming a REST API published on the IoT Server will be available for use in the Development Platform. The IoT application will be developed on the standard Raspberry Pi development environment outside of the Workshop.

Copyright© 1998-2018 On The Edge Software Consulting

IoT Logical Architecture

Copyright© 1998-2018 On The Edge Software Consulting

How do I get the code for the Reference Applications?

- **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.

Copyright© 1998-2018 On The Edge Software Consulting

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.

Folders of interest:

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

Let's Walk Through the SDK!

Copyright© 1998-2018 On The Edge Software Consulting

Development Tools

- For the Desktop you will need to install and setup a few tools……..
  - ✓ Eclipse Neon for EE and Tomcat or Wildfly.
  - ✓ Eclipse Neon for PHP.
  - ✓ MAMP Stack.
- For the Cloud you won'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 quickly "rebuild" your environment if it gets messed up!
- You will also need to use a GIT client, such as SourceTree, or you can use a browser or you can use the GIT client built into Eclipse and Codenvy.

Copyright© 1998-2018 On The Edge Software Consulting

Deployment Environment

- You (or IT) won't need to install anything……..
  - ✓ We will use a Cloud PaaS.
  - ✓ We will use the public and free Redhat OpenShift Cloud and/or Microsoft Azure Cloud.
  - ✓ Once you have "procured" your Cloud 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) won't need to install anything to support application development!

Copyright© 1998-2018 On The Edge Software Consulting

Cloud Based Development Platform

Copyright© 1998-2018 On The Edge Software Consulting

Physical Cloud Architecture

Copyright© 1998-2018 On The Edge Software Consulting

IoT Device Reference Application

Copyright© 1998-2018 On The Edge Software Consulting

IoT Device

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

Copyright© 1998-2018 On The Edge Software Consulting

IoT Device Reference Application

- IoT Device Reference Application is written in Python 3.
  - 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 keyboard and mouse via USB.
- However, now I just connect to the Raspberry Pi remotely using VNC.

Copyright© 1998-2018 On The Edge Software Consulting

IoT Device Application Logic

Initialize logging, read config file, init local variables.

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

Loop Forever.

Display Activity Dot (.) on the Console.

Read IoT Data from Sense HAT.
Round data 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.

Copyright© 1998-2018 On The Edge Software Consulting

IoT Device Application Code Walkthrough

Sense HAT Raspberry Pi 3

Connected via VNC

Let's Walk Through the Code!

Copyright© 1998-2018 On The Edge Software Consulting

Action Items / Next Steps

Copyright© 1998-2018 On The Edge Software Consulting

How Do I Clone the Workshop Repositories?

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

2.

3. Fill in the Clone Repository dialog as follows.

Source Repo URL
Local Repo Path
Local Repo Name

3. Click the Clone button.

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

Copyright© 1998-2018 On The Edge Software Consulting

How Can I 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.
   - Then 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.

Copyright© 1998-2018 On The Edge Software Consulting

How Do I Report a Bug?

1. Go the appropriate Github repository that has a bug.

2. Report an Issue as follows:

Click the Issues Tab

Click the New Issue button

3. Fill out the New Issue form and then click the Submit new issue button.

Copyright© 1998-2018 On The Edge Software Consulting

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.

Copyright© 1998-2018 On The Edge Software Consulting

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

Copyright© 1998-2018 On The Edge Software Consulting

Next Up

- Workshop Session 2 will be scheduled for next week.
- Agenda:
  - Q/A.
  - Review the IoT Services App. Architecture.
  - Overview of Spring, JAX-RS, and Maven.
  - Review the IoT Services App. Code.

- We will be monitoring our Padlet for questions.
  - Post your tool/account questions in the Padlet.
  - Post your technical questions in the Padlet.

Copyright© 1998-2018 On The Edge Software Consulting

Workshop Session 2

Copyright© 1998-2018 On The Edge Software Consulting

Workshop Session 2

- **IoT Services Reference Application:**
  - Overview of the Application Architecture
  - Overview of the Spring Framework
  - Overview of JAX-RS
  - Code Walk Through
  - Maven Walk Through
- **Action Items/Homework**

Copyright© 1998-2018 On The Edge Software Consulting

Application Architecture

Copyright© 1998-2018 On The Edge Software Consulting

IoT Services App Logical Architecture

Powered By

Copyright© 1998-2018 On The Edge Software Consulting

IoT Services Application Logical Architecture

Powered By

Copyright© 1998-2018 On The Edge Software Consulting

IoT Weather Service REST API's

1. /test: Testing Method

HTTP Method: GET

Consumes: Nothing

Produces: JSON of ResponseModel

2. /save: Saves IoT Weather Data

HTTP Method: POST

Consumes: JSON of WeatherSensorModel

Produces: JSON of ResponseModel

3. /get/{device}/{from}/{to}: Returns IoT Weather Data

HTTP Method: GET

Consumes: URI parameters for Device ID, From Date, and To Date

Produces: JSON of ResponseDataModel

See JavaDoc and Swagger API in the SDK!

Copyright© 1998-2018 On The Edge Software Consulting

Code Walk Through

Copyright© 1998-2018 On The Edge Software Consulting

What Is The Spring Framework?

- Spring is an alternative Java stack for building web and enterprise applications.
- Spring has essentially replaced Enterprise Java in the industry.
- Spring consists of a number of “modules”. We will use Spring Core, Spring Security, and Spring JDBC.
- Spring supports a “light weight” POJO based programming model using annotations and/or XML configuration.
- All POJO classes are referred to and defined as SpringBeans.
- SpringBeans can configured and injected using IoC.
- Spring apps can be deployed on Apache Tomcat and do not require a full J2EE compliant app. server.

Copyright© 1998-2018 On The Edge Software Consulting

What Is JAX-RS?

- JAX-RS is a specification for building REST based services in Java.
- An implementation of the specification is included in all Enterprise Java stacks.
- There are a number of other implementations that are available if you are not using a full Enterprise Java stack (for example Spring).
- We are going to use Jersey (and Jackson) for our implementation.
- JAX-RS makes building REST based services easy by simply using a few annotations in your code.

Copyright© 1998-2018 On The Edge Software Consulting

JAX-RS Code Example

URI to Service Class

Simply a POJO class

HTTP GET (or @POST)

URI to Service Method

MIME Type for return type (also see @Consumes)

Jackson converts to JSON for you

```
@Path("weather")
public class RestService
{
    @GET
    @Path("/test")
    @Produces("application/json")
    public ResponseModel test()
    {
      // Return a Test Response
      ResponseModel response = new ResponseModel(0, "This is a test");
      return response;
    }
}
```

Create a POJO and markup your code using JAX-RS Annotations.
Import annotations from javax.ws.rs.* package.
You will also have to do some one time configuration.

Copyright© 1998-2018 On The Edge Software Consulting

Jersey JAX-RS Configuration

```
<!-- Jersey Servlet -->
<servlet>
    <servlet-name>Jersey Web Services</servlet-name>
     <servlet-class>org.glassfish.jersey.servlet.ServletContainer</servlet-class>
     <init-param>
        <param-name>jersey.config.server.provider.packages</param-name>
        <param-value>edu.gcu.web.service</param-value>
    </init-param>
</servlet>
<servlet-mapping>
    <servlet-name>Jersey Web Services</servlet-name>
    <url-pattern>/rest/*</url-pattern>
</servlet-mapping>
```

Any URI with /rest/ will get handled by the Jersey Servlet

This configuration goes in your web.xml file!

Copyright© 1998-2018 On The Edge Software Consulting

REST Service URI

/ cloudservices / rest / weather / test

HTTP GET

App Name

Request
Handled by
Jersey Servlet

RestService Class
Instantiated

RestService.test()
Invoked

No Parameters!

Jackson converts all service method parameters and return values to JSON for you.

Copyright© 1998-2018 On The Edge Software Consulting

What is Spring Security?

Spring Security is a powerful framework to secure Spring applications. We will configure Spring to force HTTP Basic Authentication scheme for all of our REST API's.

```
<!-- Security for REST Services -->
<filter>
    <filter-name>springSecurityFilterChain</filter-name>
    <filter-class>org.springframework.web.filter.DelegatingFilterProxy</filter-class>
</filter>
<filter-mapping>
    <filter-name>springSecurityFilterChain</filter-name>
    <url-pattern>/rest/*</url-pattern>
</filter-mapping>
```

Any request URI with /rest/ will get handled by the Spring Proxy Filter

This configuration goes in your web.xml file!

Copyright© 1998-2018 On The Edge Software Consulting

Spring Security Configuration

```
<sec:authentication-manager alias="casAuthenticationManager" >
    <sec:authentication-provider>
        <sec:user-service>
      <sec:user name="CloudWorkshop" password="dGVzdHRlc3Q=" authorities="ROLE_SYSTEM" />
  </sec:user-service>
</sec:authentication-provider>
</sec:authentication-manager>
<sec:http auto-config="true">
    <sec:intercept-url pattern="/rest/**" access="ROLE_SYSTEM" />
    <sec:http-basic />
</sec:http>
```

Use HTTP Basic Authentication

Any URI with /rest/ will require HTTP Basic Authentication

Security Credentials

This configuration goes in your securityContext.xml file!

Copyright© 1998-2018 On The Edge Software Consulting

How Does Spring Support IoC?

Spring Core contains an IoC Container and is really the only way to develop in Spring (leveraging SpringBeans).

SpringBean Definition

Inject Business Service SpringBean into this SpringBean

```
<bean id="weatherService" class="edu.gcu.business.WeatherService">
    <property name="dao">
        <ref bean="weatherDao" />
    </property>
</bean>
<bean id="restServicesBean" class="edu.gcu.web.service.RestService">
    <property name="service">
        <ref bean="weatherService" />
    </property>
</bean>
```

This configuration goes in your applicationContext.xml file!

Copyright© 1998-2018 On The Edge Software Consulting

Spring IoC Code Example

Variable to inject
(note an interface)

Use setter injection

```
public class RestService
{
    static WeatherServiceInterface service;

    // ***** Dependencies and Helper Functions *****
    public void setService(WeatherServiceInterface service)
    {
        RestService.service = service;
    }
}
```

You can also use Annotations versus using XML!

Copyright© 1998-2018 On The Edge Software Consulting

What Is Spring JDBC?

Spring JDBC is a simple framework to execute SQL statements using Spring JDBC API's.

Action	Spring	You
Define connection parameters.		X
Open the connection.	X	
Specify the SQL statement.		X
Declare parameters and provide parameter values		X
Prepare and execute the statement.	X	
Set up the loop to iterate through the results (if any).	X	
Do the work for each iteration.		X
Process any exception.	X	
Handle transactions.	X	
Close the connection, statement and resultset.	X	

No coding required to handle connections, cleanup, etc.

Once you do a bit of configuration you can then focus on writing your database CRUD code!!

Copyright© 1998-2018 On The Edge Software Consulting

Spring JDBC Configuration

SpringBean

Use Apache DB Connection Pool

Use MySQL DB JDBC Driver

```
<bean id="dataSource" destroy-method="close" class="org.apache.commons.dbcp.BasicDataSource">
    <property name="driverClassName" value="com.mysql.jdbc.Driver" />
    <property name="url" value="jdbc:mysql://${db.hostname}/iot" />
    <property name="username" value="${db.username}" />
    <property name="password" value="${db.password}" />
</bean>
```

SpringBean

Inject Datasource SpringBean into this SpringBean

```
<bean id="weatherDao" class="edu.gcu.dao.WeatherDAO">
    <property name="dataSource">
        <ref bean="dataSource" />
    </property>
</bean>
```

This configuration goes in your applicationContext.xml file!

Copyright© 1998-2018 On The Edge Software Consulting

Spring JDBC Code Example

Use setter injection

Variable to inject

Create JDBC Template using this Datasource

```
@SuppressWarnings("unused")
public class WeatherDAO implements WeatherDAOInterface
{
    private DataSource dataSource;
    private JdbcTemplate jdbcTemplateObject;

    public void setDataSource(DataSource dataSource)
    {
        this.dataSource = dataSource;
        this.jdbcTemplateObject = new JdbcTemplate(dataSource);
    }
}
```

Just like Prepared Statements

```
String sql = "SELECT * FROM weather WHERE device_id=? AND date >= ? AND date <= ?";
SqlRowSet srs = jdbcTemplateObject.queryForRowSet(sql, deviceID, from, to);
while(srs.next())
{}
```

Use queryForRowSet() API for SQL queries and to bind your data to SQL

See update() API for doing a SQL insert using Spring JDBC!

Copyright© 1998-2018 On The Edge Software Consulting

A Few Misc. Things

- Handling time and dates in Java needs to be done properly due to timezones
- Remember the time zone by default and unless you configure or code to a specific timezone will be set by the JVM and obtained from the OS.
- If you deploy your application to a Cloud Server that is running in Europe or EST then your time and dates may need to be adjusted for a desired timezone to meet your requirements.
- I had to do this for the IoT Reporting Application!

Let's look at the code to handle PST timezone.

Copyright© 1998-2018 On The Edge Software Consulting

Maven Overview

Copyright© 1998-2018 On The Edge Software Consulting

What Is Maven?

- Maven is a build tool. Maven is a dependency management system.
- You configure your application using an XML file (i.e. pom.xml) that goes in the root of your project.
- The pom.xml file defines plugins used during build process, what files to build/package, what libraries and their versions that you want to build and package with, and how to package your project.
- Maven is driven by goals that define that tasks that you want to perform during a build.

Let's look at the Maven file for the Template and Reference App!

Copyright© 1998-2018 On The Edge Software Consulting

How Do I Do a Maven Build in Eclipse?

Run Name

Workspace Name

Maven Goals and Profile

Use Maven built into Eclipse

Under Run->Run Configurations…..menu. Select Maven Build.

Copyright© 1998-2018 On The Edge Software Consulting

How Do I Do a Maven Build in Codenvy?

```
mvn clean package -f ${current.project.path} -Pdev
```

Maven CLI Maven Goals Working Directory Maven Profile

OpenShift
also uses
Maven for its
builds!!

Setup a Custom Command and invoke via CLI.

Copyright© 1998-2018 On The Edge Software Consulting

Reference App. Code Walk Through

Copyright© 1998-2018 On The Edge Software Consulting

Action Items / Next Steps

Copyright© 1998-2018 On The Edge Software Consulting

Action Items / Homework

1. **Use the SDK and Reference Apps:**
   - Clone the IoT Service Reference app or the Template app.
   - Push the code to your Github repository.
   - Review the code.
   - Review the README.md in the ‘cloudservices’ Github repository.
   - Review the README.md in the *docs/development* directory from the SDK.
   - Import your app into Eclipse and/or the Codenvy IDE.
2. **Get ready for Workshop Session #3:**
   - Go thru JAX-RS Tutorials (see Reference Resources slide).
   - Go thru Codenvy Tutorials (see Reference Resources slide).
   - Go thru OpenShift Tutorials. (see Reference Resources slide).

Copyright© 1998-2018 On The Edge Software Consulting

Reference Materials

- Codenvy Tutorials:
  - ✓ Tutorial from Codenvy
- OpenShift Tutorials:
  - ✓ Docs from Redhat
- JAX-RS Tutorials:
  - ✓ Jersey + Jackson Tutorial
  - ✓ JAX-RS Using Jersey Tutorial
- Spring Documentation:
  - ✓ Spring JDBC Docs

Copyright© 1998-2018 On The Edge Software Consulting

Next Up

- Workshop Session 3 will be scheduled for this coming Thursday.
- Agenda:
  - Q/A.
  - Build the IoT Services App. in Eclipse and Codenvy.
  - Test the IoT Service App. API's.
  - Deploy the IoT Services App. to OpenShift.

- We will be monitoring our Padlet for questions.
  - Post your tool/account questions in the Padlet.
  - Post your technical questions in the Padlet.

Copyright© 1998-2018 On The Edge Software Consulting

Workshop Session 3

Copyright© 1998-2018 On The Edge Software Consulting

Workshop Session 3

- **IoT Services Reference Application:**
  - Development Tooling Setup and Build
  - Testing and Documenting your API's
  - Cloud Deployment (time permitting)
- **Action Items/Homework**

Copyright© 1998-2018 On The Edge Software Consulting

Build The IoT Services App.

Copyright© 1998-2018 On The Edge Software Consulting

How Do I Build In Eclipse?

- See the README.md file in the *docs/development* directory of the SDK!

Let's import the Reference App!

Copyright© 1998-2018 On The Edge Software Consulting

How Do I Build In Codenvy?

- See the README.md file in the *docs/development* directory of the SDK!

Let's import the Reference App!

Copyright© 1998-2018 On The Edge Software Consulting

How Do I Build & Deploy in OpenShift?

- See the README.md file in the *docs/development* directory of the SDK!

Let's build and deploy the Reference App!

Copyright© 1998-2018 On The Edge Software Consulting

REST API Tools

Copyright© 1998-2018 On The Edge Software Consulting

What Tools Do I Need For Developing REST API's?

- **Testing:**
  - Use Postman!
  - You can use this tool during your development to simply test out your API's (either GET or POST).
  - You can also use this tool to setup Unit Tests as a Test Suite to test one or more of your API's.
- **Documentation:**
  - Use Swagger!
  - You should always document your API's in JavaDoc and/or Swagger.
  - See the example in the SDK that I built and published.

Copyright© 1998-2018 On The Edge Software Consulting

Hands On in the Cloud

Copyright© 1998-2018 On The Edge Software Consulting

Action Items / Next Steps

Copyright© 1998-2018 On The Edge Software Consulting

Action Items / Homework

1. **Use the SDK and Reference Apps:**
   - Get your app in Eclipse and/or the Cloudenvy IDE to build and deploy on Tomcat.
   - Test your code using Postman.
2. **Get ready for Workshop Session #4:**
   - Go thru PHP Tutorials (see Reference Resources slide).
   - Clone the IoT Reporting Reference Application from the ‘cloudapp’ Github repository.
   - Review the IoT Reporting Reference App. code from the ‘cloudapp’ Github repository.

Copyright© 1998-2018 On The Edge Software Consulting

Reference Materials

- Laravel PHP Tutorials:
  - ✓ Guzzle Library Docs
  - ✓ Lavacharts Library Docs
- Composer Tutorials:
  - ✓ Online Tutorial 1
  - ✓ Online Tutorial 2

Copyright© 1998-2018 On The Edge Software Consulting

Next Up

- Workshop Session 4 will be scheduled for next week.
- Agenda:
  - Q/A.
  - Review the IoT Reporting App. Architecture.
  - Overview of Guzzle, Lavacharts, and Composer.
  - Review the IoT Reporting App. code.

- We will be monitoring our Padlet for questions.
  - Post your tool/account questions in the Padlet.
  - Post your technical questions in the Padlet.

Copyright© 1998-2018 On The Edge Software Consulting

Workshop Session 4

Copyright© 1998-2018 On The Edge Software Consulting

Workshop Session 4

- **IoT Reporting Reference Application:**
  - Overview of the Application Architecture
  - Overview of Guzzle
  - Overview of Lavacharts
  - Code Walk Through
  - Composer Walk Through
- **Action Items/Homework**

Copyright© 1998-2018 On The Edge Software Consulting

Application Architecture

Copyright© 1998-2018 On The Edge Software Consulting

IoT Reporting Application Logical Architecture

Powered By

Copyright© 1998-2018 On The Edge Software Consulting

IoT Reporting Application Logical Architecture

Powered By

Copyright© 1998-2018 On The Edge Software Consulting

Code Walk Through

Copyright© 1998-2018 On The Edge Software Consulting

What Is Guzzle?

- Guzzle is PHP library that implements a HTTP Client.
- Guzzle makes it easy to send HTTP requests for making REST based API calls.
- Supports POST and GET.
- Supports HTTP Basic Authentication.
- Supports setting HTTP Parameters.
- Supports handling HTTP Responses.
- Is easy to use.

Copyright© 1998-2018 On The Edge Software Consulting

What Is Lavacharts?

- Lavacharts is PHP library that implements charting functionality.
- Lavacharts is a wrapper for Google's powerful Javascript Chart API.
- The Lavacharts library wraps the work of generating the proper Javascript so you can focus on the data and not the Javascript in page.
- Supports Data Tables, Line Charts, Bar Charts, Pie Charts, and more.
- Is pretty easy to use. Also see Plotly!!

Copyright© 1998-2018 On The Edge Software Consulting

Guzzle Code Example

Guzzle HTTP Client

Build REST API URI

Invoke REST API

Check HTTP Response Code

Convert JSON to Object

```
// Call Web API to get Weather Data
$serviceURL = "http://cloudservices-workshop.1d35.starter-us-east-1.openshiftapps.com/cloudservices/rest/weather/
$client = new Client(['base_uri' => $serviceURL]);

$api = "get";
$device = 0;
$uri = $api . "/" . $device . "/" . $from . "/" . $to;

$username = "CloudWorkshop";
$password = “dGVzdHRlc3Q=";
$response = $client->request('GET', $uri, ['auth' => [$username, $password]]);

// Process API Response
if($response->getStatusCode() == 200)
{
    // Convert JSON to Objects and render report
    $jsonObj = json_decode($json = $response->getBody(), true);
```

This code can go in one of your Laravel Controllers.
Import Guzzle Library from GuzzleHttp\Client namespace.
You will also have to configure Composer.

Copyright© 1998-2018 On The Edge Software Consulting

Lavacharts Code Example

Convert JSON to Object

```
$jsonObj = json_decode($json = $response->getBody(), true);
```

Lavacharts instance, DataTable instance, and Chart Columns

```
// Build a Lavachart Datatable, Line Chart, and pass the Chart to the View to render
$lava = new Lavacharts;
$weatherData = $lava->DataTable();
$weatherData->addStringColumn(‘Date’)
             ->addNumberColumn(‘Temp')
             ->addNumberColumn(‘Pressure’)
             ->addNumberColumn(‘Humidity');
```

Add Chart Data to DataTable

```
foreach ($jsonObj['data'] as $item)
{
    $weatherData->addRow([$item['date'],
            $item['temperature'],
    $item['pressure'],
    $item['humidity']]);
}
```

Create Line Chart and pass to View

```
$lava->LineChart('Temps', $weatherData, ['title' => 'Weather in Date Range']);
return view("weatherReportChart")->with("lava", $lava);
```

Render Chart View

```
{!! $lava->render('LineChart','Temps', 'weather_div1') !!}
```

This code can go in one of your Laravel Controllers and View.
Import Lavacharts Library from Khill\Lavacharts\Lavacharts namespace.
You will also have to configure Composer.

Copyright© 1998-2018 On The Edge Software Consulting

jQuery Time-picker Code Example

Initialize each of the time-pickers

Bind each time-picker to each edit text control

```
<script>
   $('#fromDate').datetimepicker({
      dateFormat: 'yy-mm-dd',
      timeFormat: 'HH:mm:ss',
      separator: ' ',
      showTimezone: false
   });
   $('#toDate').datetimepicker({
      dateFormat: 'yy-mm-dd',
      timeFormat: 'HH:mm:ss',
      separator: ' ',
      showTimezone: false
   });
   $('#fromDate').datetimepicker();
   $('#toDate').datetimepicker();
</script>
```

This code can go in one of your Laravel Views.
Import the jQuery Timepicker JS files from the Reference App.
Remember your end user and the device they are using.
Make your apps easy to use!

Copyright© 1998-2018 On The Edge Software Consulting

Composer Overview

Copyright© 1998-2018 On The Edge Software Consulting

What Is Composer?

- Composer is a dependency manager for PHP.
- Composer will manage the dependencies you require on a project by project basis. This means that Composer will pull in all the required libraries, dependencies and manage them all in one place.
- You configure your application using a JSON file (i.e. composer.json) that goes in the root of your project.
- There are also plug-ins available for Eclipse.

Copyright© 1998-2018 On The Edge Software Consulting

How Do I Configure Dependencies in Composer?

Add your dependencies and version to the "require" section

composer.json

```
"require": {
        "php": ">=5.6.4",
        "guzzlehttp/guzzle": "~6.0",
        "khill/lavacharts": "^3.1",
        "laravel/framework": "5.4.*",
        "laravel/tinker": "~1.0"
    },
```

To install the components into your project run composer from the command line using the install option

```
php composer.phar install
```

OpenShift also uses Composer for its "builds"!!

Let's look at the Composer file for the Reference App

Copyright© 1998-2018 On The Edge Software Consulting

Reference App. Code Walk Through

Copyright© 1998-2018 On The Edge Software Consulting

Action Items / Next Steps

Copyright© 1998-2018 On The Edge Software Consulting

Action Items / Homework

1. **Use the SDK and Reference Apps:**
   - Clone the IoT Reporting Reference app or the Template app.
   - Push the code to your Github repository.
   - Review the code.
   - Review the README.md in the ‘cloudapp’ Github repository.
   - Review the README.md in the *docs/development* directory from the SDK.
   - Import your app into Eclipse and/or the Cloudenvy IDE.
2. **Get ready for Workshop Session #5:**
   - Go thru Composer Tutorials (see Reference Resources slide).
   - Go thru Codenvy Tutorials (see Reference Resources slide).
   - Go thru OpenShift Tutorials. (see Reference Resources slide).

Copyright© 1998-2018 On The Edge Software Consulting

Reference Materials

- Codenvy Tutorials:
  - ✓ Tutorial from Codenvy
- OpenShift Tutorials:
  - ✓ Docs from Redhat
- Composer Docs:
  - ✓ Composer Online Docs
- jQuery Timepicker Docs:
  - ✓ jQuery Timepicker Github Docs
  - ✓ jQuery Timepicker Docs

Copyright© 1998-2018 On The Edge Software Consulting

Next Up

- Workshop Session 5 will be scheduled for this coming Thursday.
- Agenda:
  - Q/A.
  - Build the IoT Reporting App. in Eclipse and Codenvy.
  - Test the IoT Reporting App. (end to end).

- We will be monitoring our Padlet for questions.
  - Post your tool/account questions in the Padlet.
  - Post your technical questions in the Padlet.

Copyright© 1998-2018 On The Edge Software Consulting

Workshop Session 5

Copyright© 1998-2018 On The Edge Software Consulting

Workshop Session 5

- **IoT Reporting Reference Application:**
  - Development Tooling Setup and Build
  - Cloud Deployment
- **Action Items/Homework**

Copyright© 1998-2018 On The Edge Software Consulting

Build The IoT Reporting App.

Copyright© 1998-2018 On The Edge Software Consulting

How Do I Build In Eclipse?

- See the README.md file in the *docs/development* directory of the SDK!

Let's import the Reference App!

Copyright© 1998-2018 On The Edge Software Consulting

How Do I Build In Codenvy?

- See the README.md file in the *docs/development* directory of the SDK!

Let's import the Reference App!

Copyright© 1998-2018 On The Edge Software Consulting

How Do I Build & Deploy in OpenShift?

- See the README.md file in the *docs/development* directory of the SDK!

Let's build and deploy the Reference App!

Copyright© 1998-2018 On The Edge Software Consulting

Hands On in the Cloud

Copyright© 1998-2018 On The Edge Software Consulting

Next Up

- Workshop Session 6 will be scheduled for next week.
- Agenda:
  - Introduction to DevOps.
  - Next Steps / Wrap Up.
  - Troubleshooting.
  - Q/A.

Copyright© 1998-2018 On The Edge Software Consulting

Workshop Session 6

Copyright© 1998-2018 On The Edge Software Consulting

Workshop Session 6

- **Agenda:**
  - Introduction to DevOps
  - Next Steps
  - Troubleshooting Issues
  - Wrap Up and Q/A

Copyright© 1998-2018 On The Edge Software Consulting

What is DevOps?

DevOps is the combination of cultural philosophies, practices, and tools that increases an organization's ability to deliver applications and services at high velocity: evolving and improving products at a faster pace than organizations using traditional software development and infrastructure management processes. This speed enables organizations to better serve their customers and compete more effectively in the market.

Reference: https://aws.amazon.com/devops/what-is-devops/

Think automation!
Netflix, Google, Amazon, and Facebook are leading the way.
There's more to software development than coding!

Copyright© 1998-2018 On The Edge Software Consulting

Can you imagine this?

Google Speed and Scale

- >30,000 developers in 40+ offices
- 13,000+ projects under active development
- 30k submissions per day (1 every 3 seconds)
- Single monolithic code tree with mixed language code
- Development on one branch - submissions at head
- All builds from source
- 30+ sustained code changes per minute with 90+ peaks
- 50% of code changes monthly
- 150+ million test cases / day, > 150 years of test / day
- Supports continuous deployment for all Google teams!

Google Confidential and Proprietary

Learn these automated development and testing tools!

Copyright© 1998-2018 On The Edge Software Consulting

Or can you imagine this?

Only 2 Years!

Automation is how these companies pull this off!

Copyright© 1998-2018 On The Edge Software Consulting

DevOps Reference Architecture

- Many principles of DevOps are demonstrated in the IoT Reference Applications via the DevOps Reference Architecture.
  - Logging: Setup SLF4J and LOG4J in your application.
  - CI/CD: Setup a build and deployment pipeline in GitLab or Jenkins.
  - Log Analysis: View your logs and setup alerts in Loggly.
  - Monitoring: Ensure your application is running in Uptime Robot.

See SDK for instructions how to setup Loggly

Powerful Services and Tools are a necessity for DevOps!

Copyright© 1998-2018 On The Edge Software Consulting

DevOps Reference Architecture

Copyright© 1998-2018 On The Edge Software Consulting

Cloud Powered CI/CD Pipeline

Automated Jenkins or GitLab CI/CD Pipeline to build, deploy, execute unit tests, code coverage, and execute a load test using a dedicated OpenShift Cloud QA Test environment.

Manual or automated deployment to OpenShift Cloud Production environment once all unit tests and functional tests pass in OpenShift Cloud QA Test environment.

See Appendix A for instructions on how to setup CI/CD using GitLab.

See Appendix B for instructions on how to setup CI/CD using Jenkins.

Copyright© 1998-2018 On The Edge Software Consulting

Cloud Powered Code Coverage

Each build in the Automated Jenkins or GitLab CI/CD Pipeline includes running a Unit Test Suite along with validating that Code Coverage metrics are still being met.

Coverage Report - All Packages

Package	# Classes	Line Coverage		Branch Coverage		Complexity
All Packages	36	61%	1375/2238	55%	343/618	2.552
com.ontheedgesc.nflpool.beans	20	77%	376/488	67%	19/28	1.079
com.ontheedgesc.nflpool.business	3	55%	337/612	45%	135/296	4.574
com.ontheedgesc.nflpool.data	3	49%	236/477	43%	28/64	4.07
com.ontheedgesc.nflpool.exceptions	4	25%	4/16	N/A	N/A	1
com.ontheedgesc.nflpool.util	5	61%	153/250	53%	28/52	3.091
com.ontheedgesc.nflpool.web.service	1	68%	266/395	74%	133/178	9.538

Report generated by Cobertura 2.1.1 on 7/12/15 6:24 AM.

JUnit JACOCO Java Code Coverage GitLab

Copyright© 1998-2018 On The Edge Software Consulting

Cloud Powered Load Testing

Each build in the Automated Jenkins or GitLab CI/CD Pipeline includes running an N user JMeter Load Test using the Flood IO Cloud Load Test environment.

Copyright© 1998-2018 On The Edge Software Consulting

Cloud Powered Monitoring

Get notified via email whenever any of your (Cloud) Servers goes down. One Service to achieve this is Uptime Robot.

See SDK for instructions how to setup Uptime Robot

Measure and monitor critical business transactions and the performance of your JVM. One Tool to achieve this is New Relic.

Powerful Services and Tools are a necessity for DevOps!

Copyright© 1998-2018 On The Edge Software Consulting

Cloud Powered Log Analysis

Analyze and archive your Log file. One Service to achieve this is Loggly.

Log files can be uploaded to Loggly either automatically or via a scheduled job.

The Log files can be be viewed without having to log into your Cloud Server.

Get notified via email when conditions or text matching rules are met from the content in your Log files.

See SDK for instructions how to setup Loggly

Powerful Services and Tools are a necessity for DevOps!

Copyright© 1998-2018 On The Edge Software Consulting

What Do You Need To Learn Next?

	Domain	Details
	Java	JUnit - Unit Testing Framework.
	Java	Cobertura or Jacoco - Code Coverage Framework.
	Java	PMD or Findbugs - Static Code Analysis Framework.
	Java	JMeter - Performance Load Testing Framework.
See SDK	DevOps	Log4j and Loggly - Log Analysis.
See Appendix A &B	DevOps	Jenkins or GitLab - Build Server.
See Appendix A & B	DevOps	Jenkins or GitLab - CI/CD.
See SDK	DevOps	Uptime Robot - Server Monitoring.
	DevOps	New Relic - Application Monitoring.
	DevOps	Flood.io - Cloud based Java Load Testing.

Copyright© 1998-2018 On The Edge Software Consulting

Questions?
What is not working?

Copyright© 1998-2018 On The Edge Software Consulting

Thank You

Copyright© 1998-2018 On The Edge Software Consulting

Appendix A
CI/CD using GitLab

Refer to https://gitlab.com/help

Copyright© 1998-2018 On The Edge Software Consulting

GitLab CI/CD - Setup Environment

1) Setup GitLab Environment Variables:

- **OpenShift:**
  - OPENSHIFT_TOKEN = Use the OpenShift CLI to setup a Service Account and to get token (see example in the DevOps Setup Notes in the SDK).
- **Azure:**
  - AZURE_DEPLOY_USERNAME = Go to Azure Portal, open your App, enter a username and password under the Deployment Credentials menu. Use these credentials in your environment variables.
  - AZURE_DEPLOY_PASSWORD = See above.
- **Google:**
  - GOOGLE_TOKEN = Go to the Google Console, open your App, go to IAM & admin menu, create a Service Account named GitLab with an Editor Role with the 'Furnish new private key' JSON format checkbox, copy the entire downloaded JSON key to the clipboard for use in in you environment variable.
- **Heroku:**
  - HEROKU_TOKEN = Get from Heroku Account Settings or use Heroku CLI, login, run the command heroku auth:token. Use the generated token value in your environment variable.

2) Setup GitLab Projects:

See the example code in the SDK. You can just mirror your GitHub repo's!

GitLab

Copyright© 1998-2018 On The Edge Software Consulting

GitLab CI/CD - OpenShift Laravel App 1

Job Name
Build Image
(with OpenShift CLI)

Build Branch
Build Script
1. Use CLI
2. Login
3. Start Build
4. Log off

```
deploy_openshift:
  image: openshift/origin-cli
  stage: deploy
  environment: openshift
  only:
    - master
  script:
    - echo "Deploying for OpenShift Cloud Platform"
    - oc login --token=$OPENSHIFT_TOKEN https://console.starter-ca-central-1.openshift.com
    - oc start-build cloudapp
    - oc logout
  when: manual
```

Change to automatic if you want to build when code is checked into Repo

Refer to https://gitlab.com/help/ci/yaml/README.md

Copyright© 1998-2018 On The Edge Software Consulting

GitLab CI/CD - Azure Laravel App 2

Job Name
Build Image
(with Linux)
Build Branch
Build Script
1. Use ZipDeploy
2. SED .env
3. Install Zip
4. Zip up code
5. Send to Azure using Curl

```
deploy_azure:
  image: centos
  stage: deploy
  environment: azure
  only:
    - master
  script:
    - echo "Deploying for Azure Cloud Platform"
    - sed -i 's/^APP_ENV=.*/APP_ENV=azure/' .env
    - yum install zip -y
    - /usr/bin/zip -r cloudapp.zip .
    - curl -X POST -u $AZURE_DEPLOY_USERNAME:$AZURE_DEPLOY_PASSWORD --data-binary @
markwsapp.scm.azurewebsites.net/api/ZipDeploy
  when: manual
```

Change to automatic if you want to build when code is checked into Repo

Refer to https://gitlab.com/help/ci/yaml/README.md

Copyright© 1998-2018 On The Edge Software Consulting

GitLab CI/CD - Google Laravel App 3

Job Name → deploy_google:

Build Image (with Cloud SDK) → image: google/cloud-sdk

```
deploy_google:
  image: google/cloud-sdk
  stage: deploy
  environment: google
  only:
    - master
  script:
    - echo "Deploying for Google Cloud Platform"
    - echo $GOOGLE_TOKEN > /tmp/$CI_PIPELINE_ID.json
    - sed -i 's/^APP_ENV=.*/APP_ENV=google/' .env
    - gcloud auth activate-service-account --key-file /tmp/$CI_PIPELINE_ID.json
    - gcloud --quiet --project cloud-workshop-207715 app deploy
  after_script:
    - rm /tmp/$CI_PIPELINE_ID.json
  when: manual
```

Build Branch → only: - master

Build Script → script:

1. Use SDK Deploy
2. SED .env
3. Log into Google
4. Deploy App

Change to automatic if you want to build when code is checked into Repo

Refer to https://gitlab.com/help/ci/yaml/README.md

Copyright© 1998-2018 On The Edge Software Consulting

GitLab CI/CD - Heroku Laravel App 4

Job Name → deploy_heroku:

Build Image (with Linux) → image: centos

```
stage: deploy
environment: heroku
only:
```

Build Branch → - master

Build Script →

```
script:
  - echo "Deploying for Heroku Cloud Platform"
  - sed -i 's/^APP_ENV=.*/APP_ENV=heroku/' .env
  - yum install git -y
  - git remote add heroku https://heroku:$HEROKU_TOKEN@git.heroku.com/mark-cloudapp.git
  - git config --global user.email "markreha@mac.com"
  - git config --global user.name "Mark Reha"
  - git config --global push.default matching
  - git add -A
  - git status
  - git commit -m "Release from GitLab"
  - git push -f heroku HEAD:master
when: manual
```

1. Use GIT
2. SED .env
3. Install GIT
4. Setup Heroku Remote GIT
5. Add a changed file
6. Push and comment to Heroku

Refer to https://gitlab.com/help/ci/yaml/README.md

Copyright© 1998-2018 On The Edge Software Consulting

GitLab CI/CD - OpenShift Spring App 1

Job Name
Build Image
(with OpenShift CLI)

Build Branch
Build Script
1. Use CLI
2. Login
3. Start Build
4. Log off

```
deploy_openshift:
  image: openshift/origin-cli
  stage: deploy
  environment: openshift
  only:
    - master
  script:
    - echo "Deploying for OpenShift Cloud Platform"
    - oc login --token=$OPENSHIFT_TOKEN https://console.starter-us-west-2.openshift.com
    - oc start-build test
    - oc logout
  when: manual
```

Change to automatic if you want to build when code is checked into Repo

Refer to https://gitlab.com/help/ci/yaml/README.md

Copyright© 1998-2018 On The Edge Software Consulting

GitLab CI/CD - Azure Spring App 2

Job Name → deploy_azure:

Build Image (with Maven) → image: maven:latest

```
stage: deploy
environment: azure
only:
```

Build Branch → - master

Build Script → script:

1. Use Curl
2. Build with Maven
3. Send WAR to Azure using Curl

```
- echo "Deploying for Azure Cloud Platform"
- mvn -Pazure clean package
- curl -X PUT -u $AZURE_DEPLOY_USERNAME:$AZURE_DEPLOY_PASSWORD -H 'If-Match:*' -
cloudservices.war" https://markwsserve2.scm.azurewebsites.net/api/vfs/site/wwwroot/bin/apache-
cloudservices.war
when: manual
```

Change to automatic if you want to build when code is checked into Repo

Refer to https://gitlab.com/help/ci/yaml/README.md

Copyright© 1998-2018 On The Edge Software Consulting

GitLab CI/CD - Google Spring App 4

Job Name → deploy_google:

Build Image (with Cloud SDK) →
```
  image: google/cloud-sdk:latest
  stage: deploy
  environment: google
  only:
```

Build Branch →
```
    - master
```

Build Script →
```
  script:
    - echo "Deploying for Google Cloud Platform"
    - echo $GOOGLE_TOKEN > /tmp/$CI_PIPELINE_ID.json
    - gcloud auth activate-service-account --key-file /tmp/$CI_PIPELINE_ID.json
    - echo "deb http://http.debian.net/debian jessie-backports main" | tee /etc/apt/sources.list.d/doc
    - apt-get -y update && apt-get -y install -t jessie-backports openjdk-8-jdk
    - update-java-alternatives -s java-1.8.0-openjdk-amd64
    - export JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk-amd64
    - export PATH=$JAVA_HOME/bin:$PATH
    - java -version
    - javac -version
    - apt-get -y update && apt-get -y install maven
    - mvn -Pgoogle -Dapp.deploy.project=cloud-workshop-services clean appengine:deploy
```

1. Use SDK Deploy
2. Update to JDK 1.8
3. Install Maven
4. Build and Deploy App

Refer to https://gitlab.com/help/ci/yaml/README.md

GitLab

Copyright© 1998-2018 On The Edge Software Consulting

GitLab CI/CD - Heroku Spring App 5

Job Name
Build Image
(with Linux)
Build Branch
Build Script

1. Use GIT
2. Install GIT
4. Setup Heroku Remote GIT
5. Checkout master branch to force change
6. Push and commit to Heroku

```
deploy_heroku:
  image: centos
  stage: deploy
  environment: heroku
  only:
    - master
  script:
    - echo "Deploying for Heroku Cloud Platform"
    - yum install git -y
    - git remote add heroku https://heroku:$HEROKU_TOKEN@git.heroku.com/mark-servicesapp.gi
    - git config --global user.email "markreha@mac.com"
    - git config --global user.name "Mark Reha"
    - git config --global push.default matching
    - git checkout master
    - git commit --allow-empty -m "Heroku build trigger notification"
    - git push -f heroku HEAD:master
  when: manual
```

Change to automatic if you want to build when code is checked into Repo

Refer to https://gitlab.com/help/ci/yaml/README.md

Copyright© 1998-2018 On The Edge Software Consulting

GitLab CI/CD - Start a Build

1. From Project menu make sure your GitLab repo is up to date

2. From CI / CD Pipeline menu select your Build Job from the drop down

3. From CI / CD Pipeline menu select your Finished Build Job from the drop down, then click on the Job to view the details

Refer to https://gitlab.com/help

Copyright© 1998-2018 On The Edge Software Consulting

Appendix B
CI/CD using Jenkins

Refer to https://jenkins.io/doc/

Copyright© 1998-2018 On The Edge Software Consulting

Jenkins CI/CD - Setup Environment

Create a new Jenkins Project by selecting New Item from the Main Menu.

1. Give your project a name

2. Select a Freestyle Project type

3. Click OK

Jenkins

Copyright© 1998-2018 On The Edge Software Consulting

Jenkins CI/CD - Configure Project 1

Setup Source Code Management.

General | **Source Code Management** | Build Triggers | Build Environment | Build | Post-build Actions

Source Code Management

- None
- Git

Repositories

Repository URL: https://github.com/markreha/cloudapp

Credentials: - none - | Add

Advanced...

Repository URL: https://git.heroku.com/mark-cloudapp.git

Credentials: markreha@mac.com/****** | Add

Advanced...

Add Repository

Branches to build

Branch Specifier (blank for 'any'): */master

Add Branch

Repository browser: (Auto)

Save | Apply | Add

1. Select GIT
2. Enter source repo URL
3. Enter remote repo URL
4. Add Jenkins managed credential using your Heroku Login Username and your Heroku API Key for your password
5. Enter remote repo Name
6. Click Apply

Jenkins

Copyright© 1998-2018 On The Edge Software Consulting

Jenkins CI/CD - Configure Project 2

Setup Build Script.

General | Source Code Management | Build Triggers | **Build Environment** | Build | Post-build Actions

Poll SCM

Build Environment

- Use secret text(s) or file(s)
- Provide Configuration files
- Generate Release Notes
- Setup Kubernetes CLI (kubectl)

Build

Execute shell

Command

```
sed -i 's/^APP_ENV=.*/APP_ENV=heroku/' .env
git config --global user.email "markreha@mac.com"
git config --global user.name "Mark Reha"
git config --global push.default matching
git add -A
git status .
git commit -m "Release from Jenkins"
```

See the list of available environment variables

Advanced...

Add build step

Post-build Actions

Save | Apply

2. Enter any bash commands used to customize code or configuration

1. Add Build Step of type Execute Script

3. Click Apply

Jenkins

Copyright© 1998-2018 On The Edge Software Consulting

Jenkins CI/CD - Configure Project 3

Setup Post Build Actions.

General | Source Code Management | Build Triggers | Build Environment | Build | **Post-build Actions**

Post-build Actions

Git Publisher

Push Only If Build Succeeds ☑

Merge Results ☐

If pre-build merging is configured, push the result back to the origin

Force Push ☑

Add force option to git push

Tags: Add Tag

Tags to push to remote repositories

Branches

Branch to push: master

Target remote name: heroku

Add Branch

Branches to push to remote repositories

Notes: Add Note

Notes to push to remote repositories

Add post-build action

Save | Apply

1. Only deploy if build passes

2. Force GIT Push

3. Enter source and remote branches to push

4. Click Save

Jenkins

Copyright© 1998-2018 On The Edge Software Consulting

Jenkins CI/CD - Start a Build

Start a Build.

1. Start a Build
2. Validate code
3. Validate build

Jenkins

Copyright© 1998-2018 On The Edge Software Consulting