**Cloud Native Workshop – E&Y**

**Lab Setup guide for the developers :-**

## Software Prerequisites

* Access to the Internet
* Git bash >= 2.14.0
* Gradle >= 4.1
* Maven >= 3.5.0
* Java JDK >= 1.8u144
* CF CLI >= 6.29.0+ff886fa93.2017-07-24

To build the applications in this workshop, you’ll need a couple of things:

* [Git](https://help.github.com/articles/set-up-git/#setting-up-git)
* [Java JDK 8](http://www.oracle.com/technetwork/java/javase/downloads/index.html)
* Your favorite IDE or editor (e.g., [Eclipse](http://www.eclipse.org/) or [Spring Tool Suite](https://spring.io/tools), [IntelliJ IDEA](https://www.jetbrains.com/idea), [NetBeans](https://netbeans.org/), Visual Studio, etc)

We’ll be pushing applications and creating services in Pivotal Cloud Foundry (PCF). This workshop uses Pivotal Web Services, an instance of PCF hosted by Pivotal.

* Login or signup for a free [Pivotal Web Services](http://run.pivotal.io/) account
* Click the *Tools* link and…​
  + download and install the CLI matching your operating system
  + login to the CF CLI (cf login -a api.run.pivotal.io)

**IDE Setup and tips**

|  |  |
| --- | --- |
| 💡 | This section shares some optional tips for configuring your Eclipse based IDE for an optimal experience during the workshop! |

**Intellij will be used in the workshop . In case you are not comfortable using it you can use Eclipse / Spring Tool Suite with the following configuration**

***Exclude java.awt.\* from auto-complete suggestions***

Preferences → Java → Appearance → Type Filters → Add…​ → java.awt.\*

This way when you need to auto import something with List you don’t get the dialog box that asks if you want java.awt.List when you really want java.util.List

***Configure Maven Auto Update***

Preferences → Maven → [✔] Automatically update Maven projects configuration

Allows you to change a pom.xml and have the eclipse classpath automatically change without having to trigger the change manually.

***Open pom.xml in XML view***

Preferences → Maven → User Interface → [✔] Open XML Page in the POM editor by default

This will get you straight to the XML when you first open your pom.xml

***Show line numbers***

Preferences → General → Editors → Text Editors → Show Line Numbers

Very useful when collaborating and you need to explicitly state which line number you are referring to.

***Automatically refresh resources changed outside of Eclipse***

Preferences → General → Workspace → Refresh using native hooks or polling

This enables Eclipse to recognize changes to files that have been modified outside of Eclipse. Pretty handy.

***Close all views you don’t need***

Give yourself more space to view/write code by closing any views in the perspective that you don’t use, such as: Outline, Spring Explorer, and Servers  
***Boot Dashboard*** Use it, it’s awesome :)