Infrastructure Setup

# GitHub.com

GitHub is a web-based Git repository hosting service, which offers all of the distributed revision control and source code management (SCM) functionality of Git as well as adding its own features. Unlike Git, which is strictly a command-line tool, GitHub provides a web-based graphical interface and desktop as well as mobile integration. It also provides access control and several collaboration features such as wikis, task management, and bug tracking and feature requests for every project.

1. If you do not have a GitHub.com account, create one.
2. Login to GitHub with your account
3. Create a new repo named: FDADI

# JIRA

JIRA is an issue tracking product, developed by Atlassian. It provides bug tracking, issue tracking, and project management functions.

## Create JIRA Project

1. If you do not have a JIRA account, create one.
2. Login to your JIRA domain.
3. Click on Projects > Create Project
   1. Select Project Type: SCRUM Software Development
   2. Provide Name: FDADI
   3. Provide Key: FDADI
   4. Check options to link Bamboo and Confluence
   5. Click Submit

## Set Workflow

1. Go to Project Administration > Workflows > Add Workflow > Choose from Marketplace
2. Find and select ‘Software Development Workflow’
3. Click Submit

## Update SCRUM Board to include Software Development Workflow Transitions

1. Click the Board dropdown in the upper right side of page
2. Click Board > Configure
3. Click the ‘Add column’ button and name the column ‘Waiting for QA’
4. Click the ‘Add column’ button and name the column ‘In QA’
5. Drag the ‘In Development’ status to the ‘In Progress’ column
6. Drag the ‘Waiting for QA’ status to the ‘Waiting for QA’ column
7. Drag the ‘In QA Review’ status to the ‘In QA’ column

## Set Story Points on Bugs and Sub-tasks

1. Go to Project Administration > Fields
2. Click on the Edit pencil on the right side of the screen
3. Click on ‘Custom Fields’ on the left side column
4. Click on the Gear and select Configure on the ‘Story Points’ field
5. Click on the ‘Edit Configuration’ link
6. Select ‘Any issue type’ in the applicable issues types list box
7. Select ‘Global context’ in the applicable context list box
8. Click the Modify button.

# Link GitHub and JIRA

## Create OAuth access token for GitHub Account

1. Login to GitHub.com
2. Click on your username in the box at the top right
3. Choose ‘Edit Your Profile’
4. Select ‘Applications’
5. Choose ‘Register New Application’
6. Enter **JIRA DVCS** for the Application Name
7. Enter the **full url** of your JIRA domain in both the URL and Callback URL fields.  
   i.e. **https://clearavenue.atlassian.net**
8. Press the ‘Register Application’ button
9. A OAuth key and secret will be displayed—keep a copy of them

## Authorize JIRA to use GitHub using OAuth key/secret

1. Log in to JIRA as a user with administrative rights.
2. From the JIRA dashboard click (settings) icon.
3. Choose Add-ons.
4. If the Marketplace banner appears, skip through by choosing the Manage add-ons link.
5. Locate the Source Control section and choose DVCS Accounts.
6. The Manage DVCS Accounts page displays.
7. Click Link a Bitbucket or GitHub account.
8. The Add New Account page displays.
9. Choose a Host value: GitHub
10. Enter a Team or Account name: clearavenue
11. Copy the OAuth Key and OAuth Secret from your DVCS site into the dialog.
12. Check Auto Link New Repos and Enable Smart Commits
13. Click the ‘Add’ button.
14. Click ‘Authorize Application’

# AWS

## Setup AWS Tomcat bamboo user

1. Login to an AWS EC instance that has tomcat installed and running.
2. Edit the <server>/tomcat8/conf/tomcat-users.xml file and add the following:  
   <role rolename="manager-gui"/>  
   <role rolename="manager-script"/>  
   <role rolename="manager-jmx"/>  
   <role rolename="manager-status"/>  
   <user username="bamboo" password="bamboosPwd" roles="manager-gui,manager-script,manager-jmx,manager-status"/>
3. Restart tomcat.

## Setup AWS Tomcat SSL

The SSL configuration of Tomcat follows the standard instructions found at https://tomcat.apache.org/tomcat-8.0-doc/ssl-howto.html

# Bamboo

Bamboo is a continuous integration server from Atlassian, the makers of JIRA, Confluence and Crowd. Bamboo is free for philanthropic and open-source projects.

Bamboo supports builds in any programming language using any build tool, including Ant, Maven, make, and any command line tools. Build notifications can be customized based on the type of event, and received via email, instant message, RSS, or pop-up windows in Eclipse-based IDEs and IntelliJ IDEA.

## Setup Clover global license

1. Login to Bamboo and click Administration > Add-Ons > Clover Plugin
2. Enter the global or evaluation license key for Clover

## Build profile

A build plan in Bamboo is where you configure the work that you want done for the build. A plan defines everything about your build process, including what gets built, how the build is triggered and what jobs are executed.

1. Login to Bamboo and click the Create > Create a New Plan
   1. Project: New Project
   2. Project Name: FDA Drug Interactions
   3. Project Key: FDAD
   4. Plan Name: FDAD-BuildPlan
   5. Plan Key: FDAD
   6. Link a New repository
   7. Click the Other box and select GitHub
   8. Display Name: FDADI
   9. Enter the GitHub username
   10. Enter the GitHub password
   11. Click the ‘Load Repositories’ button and select FDADI from the dropdown
   12. Click ‘Configure Plan’ button
2. Configure Plan
   1. Notifications tab > Add notification
      1. Event: Failed Builds and First Successful
      2. Recipient Type: group
      3. Group: jira-users
      4. Click the ‘Add’ button
   2. Triggers tab > Add trigger
      1. Repository polling : check FDADI
      2. Polling strategy: Periodically
      3. Click the ‘Save trigger’ button
   3. Stages tab > Default Job > Tasks tab
      1. Add Task Source Code Checkout
         1. Repository: FDADI
         2. Click ‘Save’ button
      2. Add Task Maven 3.x
         1. Goal: clean package
         2. JDK: JDK 1.7
         3. Check: The build will produce test results
         4. Check: Look in the standard test results directory
         5. Click the ‘Save’ button
   4. Stages tab > Default job > Miscellaneous tab
      1. Check: Use Clover to collect Code Coverage for this build
      2. Check: Automatically integrate Clover into this build
      3. Check: Generate a Clover Historical Report
      4. Click the ‘Save’ button

## Deployment profile

A deployment project in Bamboo is a container for holding the software project you are deploying: releases that have been built and tested, and the environments to which releases are deployed.

1. Click on the FDADI-BuildPlan > Plan Configuration > Stages tab
2. Click on ‘Deployment for FDA Drug Interactions’ link in the Related deployment projects section
3. Click on ‘Edit Project’
4. Setup ‘Environment’, name it ‘AWS’
5. Setup ‘Deploy Tasks’ by clicking on the ‘Edit Tasks’ button
   1. Add Task Source Code Checkout
      1. Repository: FDADI
      2. Click ‘Save’ button
   2. Add Task Maven 3.x
      1. Goal: clean package findbugs:check
      2. JDK: JDK 1.7
      3. Check: The build will produce test results
      4. Check: Look in the standard test results directory
      5. Click the ‘Save’ button
   3. Add Task Deploy Tomcat Application
      1. Enter Tomcat Manager URL: <tomcaturl>/manager/
      2. Enter Tomcat manager username/password that has access
      3. Application context: /FDADI
      4. WAR File: target/fdadi.war
      5. Click the ‘Save’ button

# Docker Installation

## Install Docker

1. Log in as a user with sudo privileges.
2. Verify that you have wget installed.  
   $> which wget
3. If wget isn’t installed, install it after updating your manager:  
   $> sudo apt-get update  
   $> sudo apt-get install wget
4. $ wget -qO- https://get.docker.com/ | sh

The system prompts you for your sudo password. Then, it downloads and installs Docker and its dependencies.

## Verify docker is installed correctly

$> docker run hello-world

Unable to find image 'hello-world:latest' locally

511136ea3c5a: Pull complete

31cbccb51277: Pull complete

e45a5af57b00: Pull complete

hello-world:latest: The image you are pulling has been verified.

Important: image verification is a tech preview feature and should not be relied on to provide security.

Status: Downloaded newer image for hello-world:latest

Hello from Docker.

This message shows that your installation appears to be working correctly.

## Docker images

Our docker deployment consists of two images:

1. Docker Hub Tomcat image: https://registry.hub.docker.com/\_/tomcat and is inherited by our image through our Dockerfile
2. Docker Hub MongoDB image: https://registry.hub.docker.com/\_/mongo and is run during the Build and Deploy step #5 below.

## Dockerfile

Below are the contents of our Dockerfile used to deploy the application to Docker (inherits Tomcat):

FROM tomcat:8  
MAINTAINER "Jeff Heath <jeff.heath@clearavenue.com>"  
ADD FDADI.war /usr/local/tomcat/webapps/

## Build and Deploy

During the Build and Deploy step of the application, the following commands are used to load the tomcat and mongoDB images.

1. $> docker build –t clearavenue/mymeds .  
   *This step will build the Docker container using the Dockerfile that provides a running instance of Tomcat8*
2. $> docker run –name clearavenue-mymeds-mongo –d mongo  
   *This step will run the Docker Hub mongodb image*
3. $> docker run –p 8080:8080 –link clearavenue-mymeds-mongo:mongo –name clearavenue-mymeds –d clearavenue/mymeds  
   *This step starts our container along with the Tomcat instance and links in the running mongodb instance*