

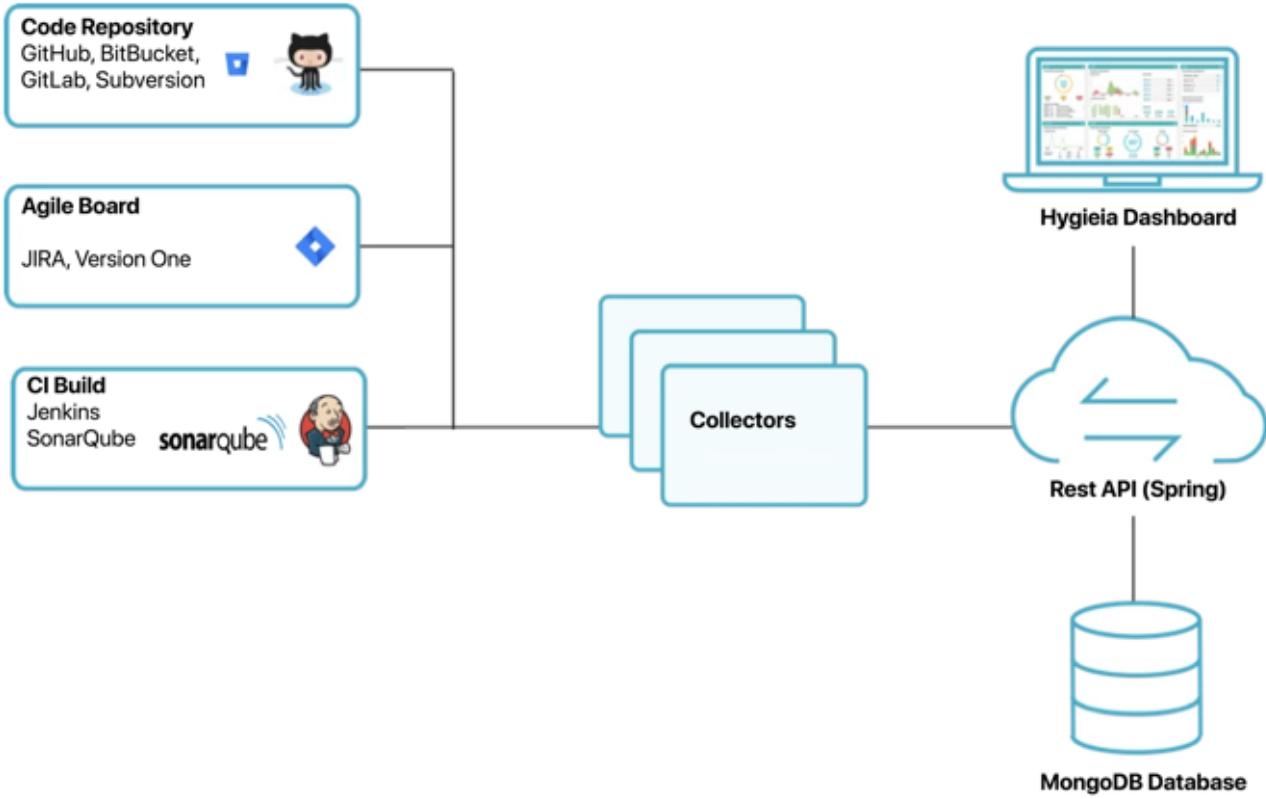
Setup and configuration for Hygieia DevOps Dashboard

The purpose of the documentation is to setup and configure the DevOps dashboard tool called Hygieia.

- [Introduction](#)
- [Pre-requisite](#)
 - [Install and configure Java JDK 1.8](#)
 - [Setup the Java_HOME](#)
 - [Install Apache Maven](#)
 - [Install Node.js](#)
 - [Install NPM](#)
 - [Install Bower](#)
 - [Install Glup](#)
 - [Install MongoDB NoSQL Database](#)
 - [Setup the Database MongoDB](#)
 - [Install Git Source Control](#)
 - [Install Jenkins 2.164.1 LTS \(Long-Term Support\)](#)
- [Install Hygieia](#)
 - [Hygieia API](#)
 - [Hygieia UI](#)
 - [Jenkins Collector for Hygieia](#)
 - [GitHub Collector for Hygieia](#)
- [User Interface Configuration of Hygieia](#)
 - [Using Template Dashboard](#)
 - [Using Team Dashboard](#)
- [Starting Hygieia](#)
- [Working Screenshot](#)

Introduction

Hygieia is an open source dashboard tool developed by CapitalOne that provides visibility to developers and product owners. It integrates with code repositories, allows a view of the complete delivery pipeline in near real time, and provides analysis of quality and performance. The document provides instructions and step-by-step details for setting up the Hygieia in Linux or MacOS operating system. The Hygieia is a distributed component that have User interface, REST APIs and Collectors along with Mongo Database. It is an solution that will stand up on the Infrastructure as a service (IaaS) layer from a cloud computing / cloud provider perspective.



Pre-requisite

Following packages are required to be installed on the Operating system before Hygieia can be installed

1. Java JDK 1.8
2. MongoDB Database
3. Apache Maven
4. Node.js
5. Jenkins
6. NPM
7. Bower
8. Glup
9. Git Command Line
10. Wget
11. CURL

Install and configure Java JDK 1.8

1. Download the Oracle JDK 1.8

```
$ wget --no-cookies --no-check-certificate --header "Cookie: gpw_e24=http%3A%2F%2Fwww.oracle.com%2F; oraclelicense=accept-securebackup-cookie"
"https://download.oracle.com/otn-pub/java/jdk/8u201-b09/42970487e3af4f5aa5bca3f542482c60/jdk-8u201-linux-x64.rpm"
```

2. Use yum to install the Oracle RPM package that was downloaded.

```
$ sudo yum localinstall -y jdk-8u201-linux-x64.rpm
```

3. Check the Java Installation

```
$ java -version
```

The expected output to be following

```
java version "1.8.0_161"
Java(TM) SE Runtime Environment (build 1.8.0_161-b12)
Java HotSpot(TM) 64-Bit Server VM (build 25.161-b12, mixed mode)
```

Setup the Java_HOME

1. After the Java JDK 1.8 is installed. It is required to set the environment variable JAVA_HOME
2. Open the .bash_profile file and insert the following lines. Make sure that the JAVA_HOME is pointing to the installed directory of the Java JDK 1.8

```
export JAVA_HOME=/Library/Java/JavaVirtualMachines/jdk1.8.0_161.jdk
/Contents/Home
export PATH="$JAVA_HOME:$PATH"
```

3. Source the .bash_profile

```
$ cd ~
$~/.bash_profile
```

4. Confirm the JAVA_HOME Variable by issuing following command and see the expected output

```
$ echo $JAVA_HOME
/Library/Java/JavaVirtualMachines/jdk1.8.0_161.jdk/Contents/Home
```

Install Apache Maven

The required version to install is 3.5.2 or above of the Apache Maven. Apache Maven is a software project management and comprehension tool. Hygieia uses the maven for building the REST API layer using the POM.xml

1. Download and obtain the Maven Tar from official repository

```
$ wget http://mirror.olnevhost.net/pub/apache/maven/binaries/apache-maven-3.2.1-bin.tar.gz
```

2. Run command above from the dir you want to extract maven to (e.g. /usr/local/apache-maven) run the following to extract the tar

```
$ tar xvf apache-maven-3.2.1-bin.tar.gz
```

3. Setup the Maven Home environment variables such as

```
export M2_HOME=/usr/local/apache-maven/apache-maven-3.2.1
export M2=$M2_HOME/bin
export PATH=$M2:$PATH
```

4. Verify the mvn version

```
$ mvn -version
```

The sample output should be similar to following

```
Apache Maven 3.5.2 (138edd61fd100ec658bfa2d307c43b76940a5d7d; 2017-10-18T03:58:13-04:00)
Maven home: /Users/tapan/apache-maven-3.5.2
Java version: 1.8.0_161, vendor: Oracle Corporation
Java home: /Library/Java/JavaVirtualMachines/jdk1.8.0_161.jdk/Contents/Home/jre
Default locale: en_US, platform encoding: UTF-8
OS name: "mac os x", version: "10.13.5", arch: "x86_64", family: "mac"
```

Install Node.js

1. Enable node.js yum repository in your system provided by the Node.js official website

```
$ yum install -y gcc-c++ make
$ curl -sL https://rpm.nodesource.com/setup_10.x | sudo -E bash -
```

2. Installation After adding yum repository in your system lets install Node.js package.

```
$ sudo yum install nodejs
```

3. Check Node.js and NPM Version

```
$ node -v  
v8.9.1
```

Install NPM

1. NodeJS provides a setup script that must run before you install it with yum

```
$ curl -sL https://rpm.nodesource.com/setup | bash -
```

2. Then the yum command should work

```
$ yum install -y nodejs
```

3. Verify the version

```
$ npm -version  
5.5.1
```

Install Bower

1. Need to install node.js on your system first. If you already have node.js and NPM installed on your system skip this step. Install bower

```
$ npm install -g bower
```

2. Verification check the installed version on your system using the following command.

```
$ bower -v  
1.8.8
```

Install Glup

1. Need to install node.js on your system first. If you already have node.js and NPM installed on your system skip this step and Install Gulp. The global installation will allow all users to use Gulp on your system.

```
$ sudo npm install -g gulp
```

2. Verification check the installed version on your system using the following command.

```
$ gulp --version  
CLI version 2.0.1
```

Install MongoDB NoSQL Database

Hygieia uses MongoDB as the database for storage and retrieval of data. For user with MacOsX or Linux, execute following commands

1. Download Latest MongoDB

```
$ wget https://fastdl.mongodb.org/osx/mongodb-osx-ssl-x86_64-4.0.3.tgz
```

2. Extract MongoDB download that was downloaded before

```
$ tar -xvf mongodb-osx-ssl-x86_64-4.0.3.tgz
```

3. Create the data directory in MongoDB. Go inside the MongoDB directory. Start the Mongo Database access the interactive shell of MongoDB.

```
$ mv mongodb-osx-x86_64-4.0.3 mongodb4.0.3  
$ cd mongodb4.0.3  
$ sudo mkdir -p /data/db/  
$ sudo chmod 700 /data/db
```

4. Provide proper permission and Start the MongoDB service.

```
$ sudo chmod 700 /data/db  
$ sudo bin/mongod
```

5. In another terminal section, access the interactive shell of MongoDB. Define the database:

```
$ bin/mongo
```

Setup the Database MongoDB

1. The step will be only one-time activity. We need to define the database where the Hygieia will store the information. Define the MongoDB database

```
> use dashboarddb
```

2. Deliver and create a user with permission to write in the dashboarddb database. store the username and password in notes as it will be needed in the API, UI and Collectors to configure.

```
> db.createUser(  
{  
  user: "dashboarduser",  
  pwd: "dbpassword",  
  roles:  
  [{  
    role: "readWrite",  
    db: "dashboarddb"  
  }]  
})
```

3. When execute the command get out of the MongoDB shell.

```
> exit
```

Install Git Source Control

1. Install Git from the system default repositories, and make sure that your system is up-to-date with the latest version of packages and then install Git by typing the following command:

```
$ yum install git
```

2. After git successfully installed, you can issue the following command to display the version of Git installed:

```
$ git --version  
git version 2.17.2
```

Install Jenkins 2.164.1 LTS (Long-Term Support)

1. Download and Install the Jenkins on Mac take the LTS (Long-Term Support). Go to the URL <https://jenkins.io/download/> and select the version as per the operating system.

2. Set the default port for Jenkins to start at 7070 (This is to avoid port conflict as Hygieia start on port 8080).

```
sudo defaults write /Library/Preferences/org.jenkins-ci httpPort 7070
```

3. Update the Java Home location in the Jenkins file. Jenkins 2.XXX version needs Java JDK 1.8

```
sudo nano /Library/LaunchDaemons/org.jenkins-ci.plist  
sudo sublime /Library/LaunchDaemons/org.jenkins-ci.plist
```

Add the appropriate JAVA_HOME environment variable in the org.jenkins-ci.plist file.

```
<dict>  
<key>JENKINS_HOME</key>  
<string>/Users/Shared/Jenkins/Home</string>  
<key>JAVA_HOME</key>  
<string>/Library/Java/JavaVirtualMachines/jdk1.8.0_161.jdk/Contents/Home</string>  
</dict>
```

4. Jenkins Start

```
sudo launchctl unload /Library/LaunchDaemons/org.jenkins-ci.plist
```

5. Jenkins Stop

```
sudo launchctl load /Library/LaunchDaemons/org.jenkins-ci.plist
```

6. Check for error logs

```
tail -f /var/log/jenkins/jenkins.log  
ps -e | grep jenkins
```

7. In the .bash_profile set the alias

```
cd ~  
sudo nano .bash_profile  
alias jenkins-start="sudo launchctl load /Library/LaunchDaemons/org.jenkins-ci.plist"  
alias jenkins-stop="sudo launchctl unload /Library/LaunchDaemons/org.jenkins-ci.plist"  
exit  
source .bash_profile
```

8. If needed to uninstall Jenkins

```
sudo su  
'/Library/Application Support/Jenkins/Uninstall.command'
```

9.In the Jenkins User Interface

1. Click on top left "Jenkins" Manage Jenkins Configure Global Security Check Mark on the logged-in user can do anything and "tick" on Allow anonymous read access.

The screenshot shows the Jenkins 'Configure Global Security' configuration page. At the top, there is a 'Configure Global Security' title with a lock icon. Below it, there are several sections: 'Enable security' (checkbox checked), 'Disable remember me' (checkbox unchecked), 'Access Control' (checkbox checked), 'Security Realm' (radio button selected for 'Jenkins' own user database'), 'Authorization' (radio button selected for 'Logged-in users can do anything' and 'Allow anonymous read access' checkbox checked). On the right side of the page, there are several help icons (blue question marks).

Install Hygieia

Hygieia promotes a single, configurable, easy to use dashboard to visualize near real-time status of the entire delivery pipeline.

Hygieia API

1. Use the official open source Git Repository of Hygieia. <https://hygieia.github.io/Hygieia/setup.html> and GitHub <https://github.com/Hygieia/Hygieia.git>

```
$ git clone https://github.com/Hygieia/Hygieia.git  
$ cd Hygieia  
$ mvn clean install package
```

2. Configure the Hygieia with the api.properties to setup the components.

```
cd Hygieia  
touch api/api.properties
```

3. Update the api.properties file with the MongoDb connection properties

```
# api/api.properties
server.contextPath=/api
server.port=8080
dbname=dashboarddb
dbhost=localhost
dbport=27017
dbusername=dashboarduser
dbpassword=dbpassword
```

4. Update the application.properties(Path: Hygieia\api\src\main\resources\application.properties)

```
# api/api.properties
server.contextPath=/api
server.port=8080

dbname=dashboarddb
dbhost=localhost
dbport=27017
dbreplicaset=false
dbhostport=localhost:27017
```

Hygieia UI

1. To access the Hygieia UI, we need to start a frontend project. Access the UI directory. The project next needs Gulp to execute the command that will start the server. We can install Gulp with npm.

```
$ cd Hygieia
$ cd UI/
$ npm install -g gulp
```

2. Install the frontend projects dependencies

```
$ npm install
$ bower install
```

Jenkins Collector for Hygieia

1. We need to update the Jenkins configuration so that Hygieia can connect and obtain the information.

```
$ cd Hygieia/collectors/build/jenkins
$ touch api/application.properties
```

2. Update the application.properties file with following content

```
# jenkins/application.properties

dbname=dashboarddb
#dbhost=localhost
#dbport=27017
dbusername=dashboarduser
dbpassword=dbpassword

jenkins.servers[0]=http://localhost:7070/
jenkins.saveLog=true

# root 11d6e6d371c597ce6389da705b501f84a7

#Collector schedule (required)
jenkins.cron=0 * * * *
jenkins.pageSize=1000
```

GitHub Collector for Hygieia

1. To configure the Hygieia GitHub collector.

```
$ cd Hygieia
$ cd collectors/scm/github
$ mvn clean install
$ touch api/application.properties
```

2. Update the application.properties file with following content

```
# Github.com application.properties
dbname=dashboarddb
dbhost=localhost
dbport=27017
dbusername=dashboarduser
dbpassword=dbpassword
dbhostport=localhost:27017
# Logging File location
logging.file=./logs/github.log
# Collector schedule (required)
github.cron=0 0/5 * * *
github.host=github.com
# Maximum number of previous days from current date, when fetching commits
github.commitThresholdDays=15
```

3. Configure the Code Repository Widget in Github

Configure Code Repo Widget

Repo Type

GitHub

Repo URL

https://github.com/tapanbanker/maven-project

Branch

master

Credentials (Optional, required for private repos)

Username :

Enter your repo username

Password :

Enter your repo password

(OR)

Personal Access Token

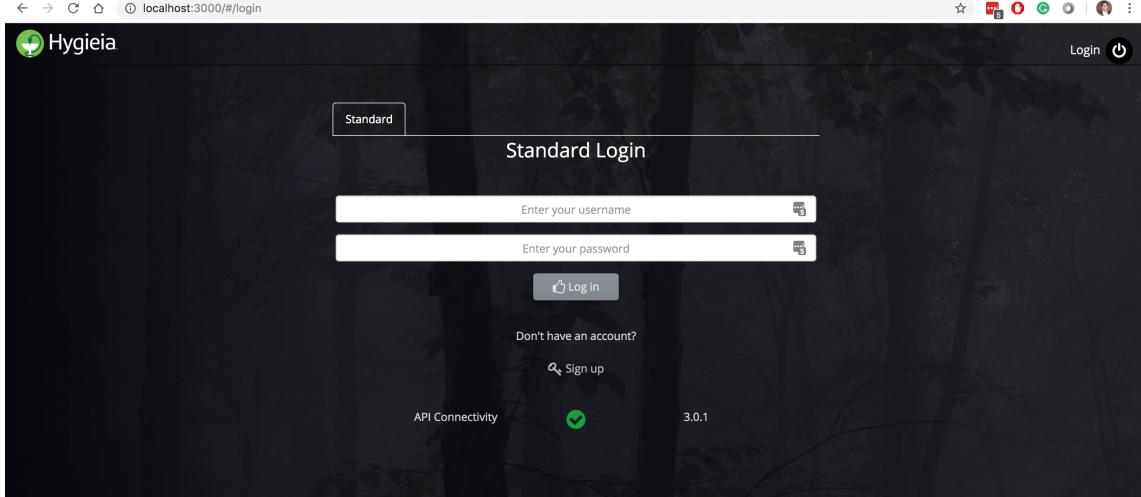
Enter your Github personal access token

Save

User Interface Configuration of Hygieia

Using Template Dashboard

1. Setup the account with username and password. Click on Login (Top right hand corner) and then click on "Sign Up" (Center bottom)



2. After login click on Create New Dashboard

3. Fill in the following details

Create a New Dashboard

Team dashboard

Select Widgets Select Templates [i](#)

Cap One [i](#)

DevSecOps [i](#)

CMS

Select a Business Service (Optional) [i](#)

Select a Business Application (Optional) [i](#)

Enable Score [i](#)

Display in Header Display in Widget [i](#)

Create

4. Click on the Create button, It will display following

Hygieia Score: N/A [i](#)

DevSecOps-CMS

Widget Pipeline Cloud

Feature Build Monitor

Configure widget

Configure widget

Configure widget

Code Repo Quality Performance Deploy

Configure widget

Configure widget

Configure widget

5. Click on the Build "Configure Widget". Then select the line and drop-down of the Jenkins Job will come. Click on Save

Configure Build Widget

Build Job

Search a build job in format <Server Name> : <Job Name>

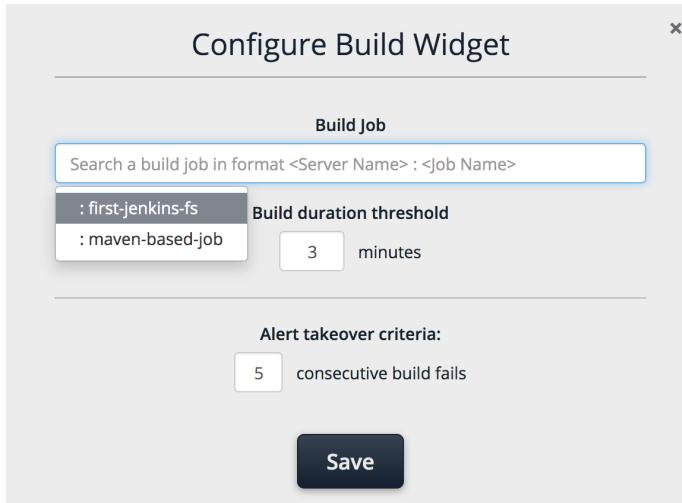
: first-jenkins-fs **Build duration threshold**

: maven-based-job 3 minutes

Alert takeover criteria:

5 consecutive build fails

Save



6. Click on the Repo "Configure Widget". Then select the line and drop-down and select GitHub. Provide the repository URL. Provide the branch details. Click on Save

Configure Code Repo Widget

Repo Type

GitHub

Repo URL

https://github.com/tapanbanker/maven-project

Branch

master

Credentials (Optional, required for private repos)

Username :

Enter your repo username

Password :

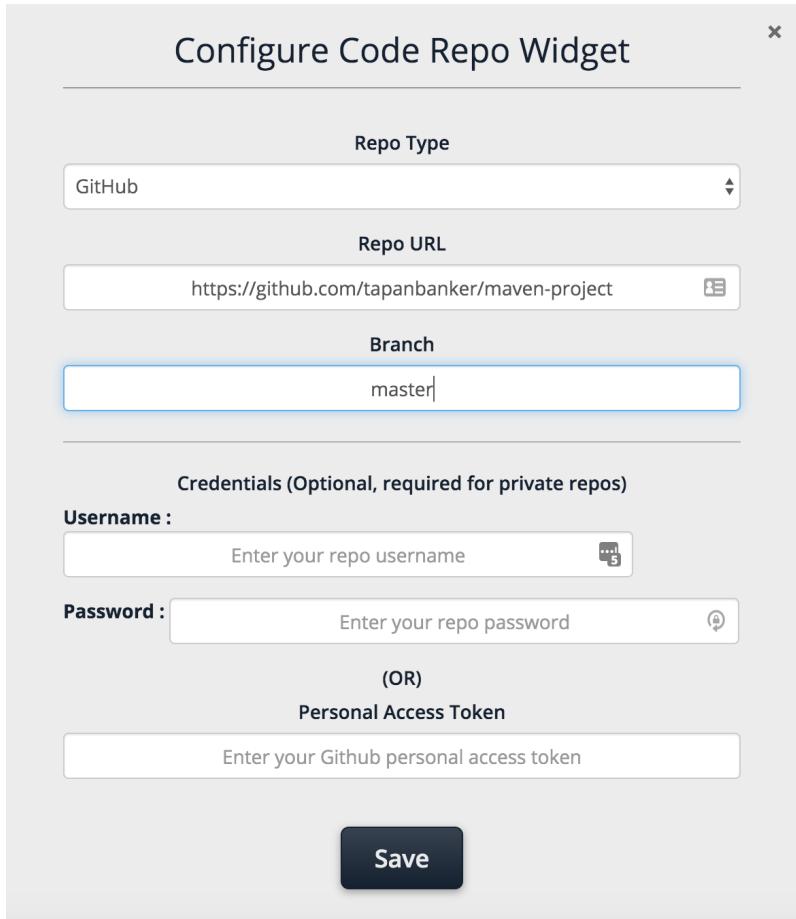
Enter your repo password

(OR)

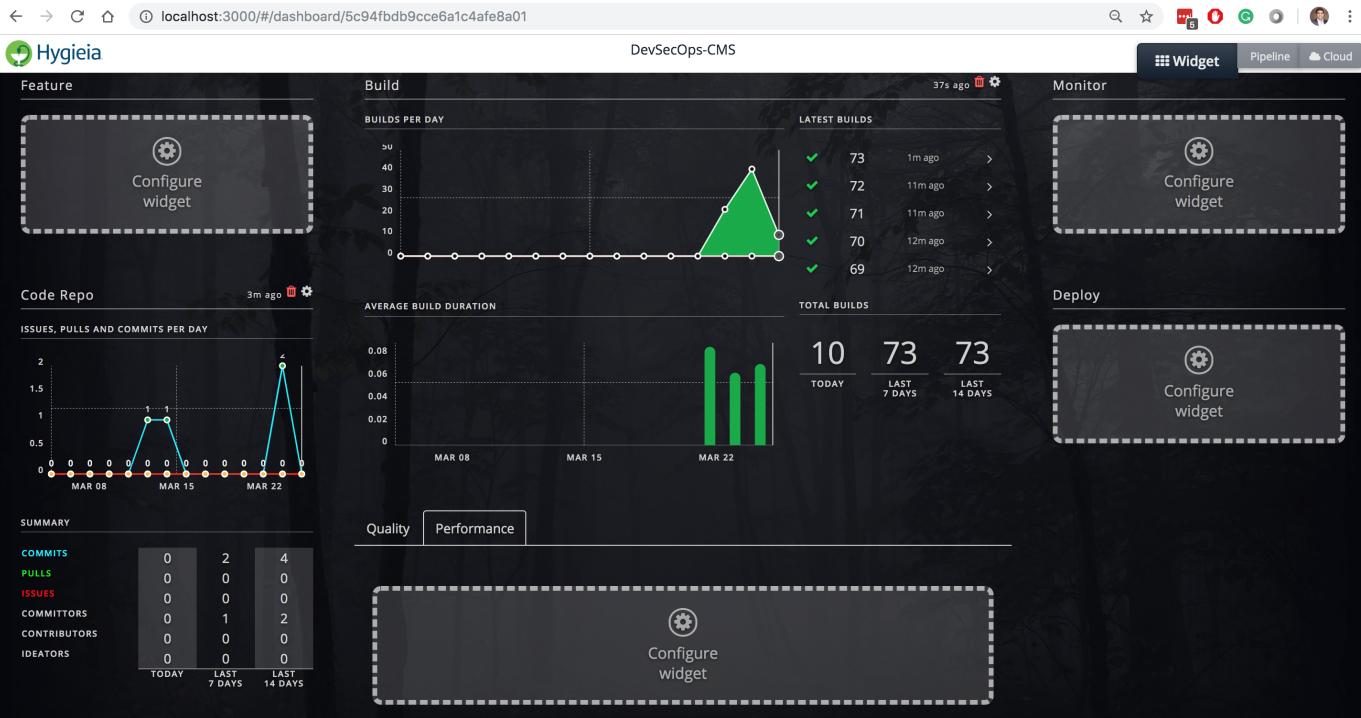
Personal Access Token

Enter your Github personal access token

Save



7. The dashboard will display like following



Using Team Dashboard

The team dashboard provide a custom dashboard feature instead of using the pre-baked template. The team dashboard allows us to configure the dashboard in a custom manner.

Click on Create New Dashboard

Create a New Dashboard

Team dashboard

Select Widgets Select Templates

DevOps

CMS

Select a Business Service (Optional)

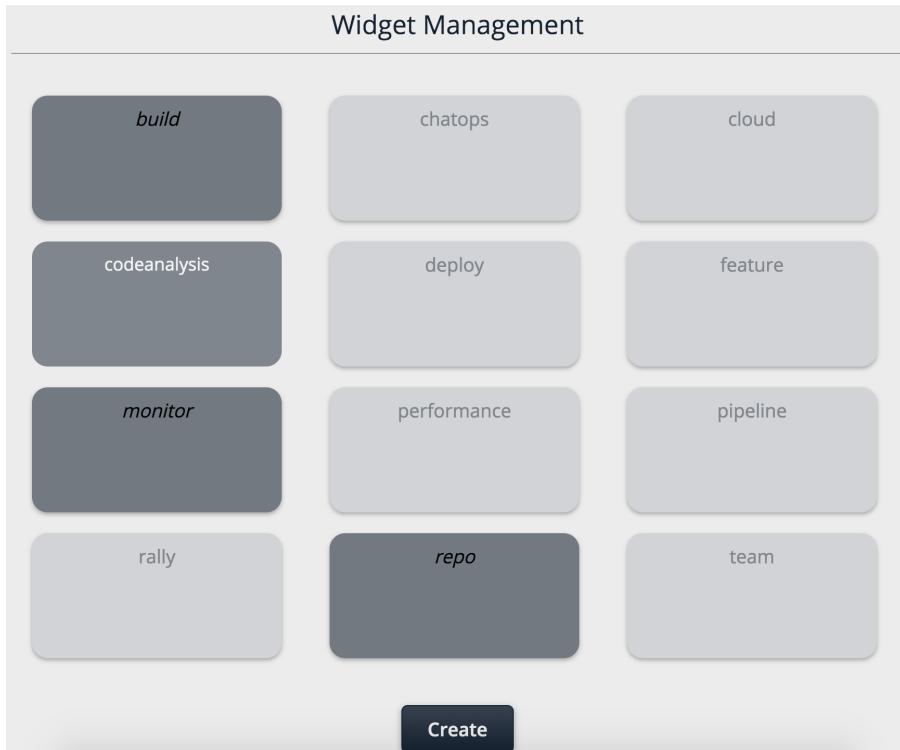
Select a Business Application (Optional)

Enable Score

Display in Header Display in Widget

Create

2. Select the Build, Repo and Monitor and click on Create



Click on Build and configure the widget, provide following information

Configure Build Widget

Build Job

Build duration threshold

3 minutes

Alert takeover criteria:

5 consecutive build fails

Save

This is a configuration dialog for a 'Build Widget'. It has a title bar 'Configure Build Widget' with a close button. Below it is a section titled 'Build Job' containing a text input field with the placeholder ': maven-based-job'. Underneath is a 'Build duration threshold' section with a numeric input set to '3' followed by 'minutes'. Another section below is 'Alert takeover criteria:' with a numeric input set to '5' followed by 'consecutive build fails'. At the bottom is a large 'Save' button.

Click on the Repo Widget

Configure Code Repo Widget

Repo Type

GitHub

Repo URL

https://github.com/tapanbanker/maven-project

Branch

master

Credentials (Optional, required for private repos)

Username :

Enter your repo username

Password :

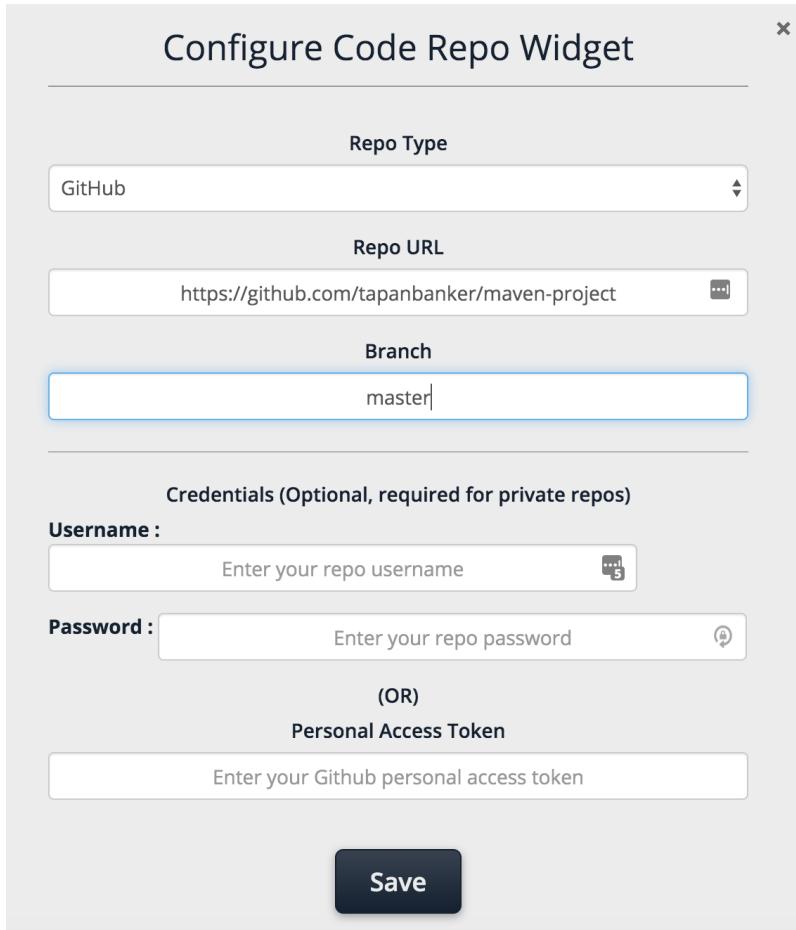
Enter your repo password

(OR)

Personal Access Token

Enter your Github personal access token

Save



For Monitoring, Hygieia will use a URL and see if the status is returned as 200 OK.

Monitor Configuration

OUR SERVICES

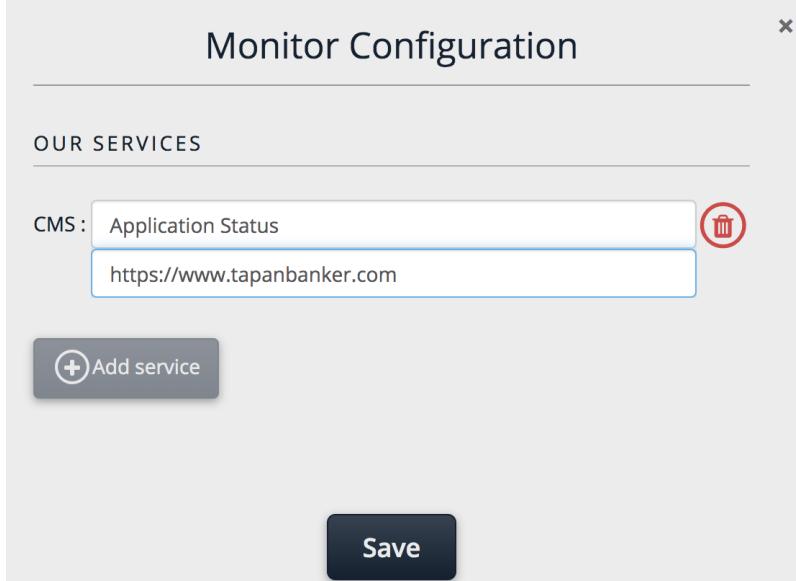
CMS :

Application Status

https://www.tapanbanker.com

Add service

Save



Starting Hygieia

1. Start the Jenkins on your local machine



```

Option #1 Use alias to start the Jenkins as setup in .bash_profile
    // alias jenkins-stop="sudo launchctl unload /Library
/LaunchDaemons/org.jenkins-ci.plist"
    $ jenkins-start
Option #2 Execute following command to run the Jenkins
    $ sudo launchctl unload /Library/LaunchDaemons/org.jenkins-ci.plist
// Jenkins will start on Port 7070 at URL http://localhost:7070

```

2. Open a new terminal and start the MongoDB database

```

$ cd mongodb4.0.3
$ sudo bin/mongod

```

3. Open a new terminal and start the REST API of Hygieia

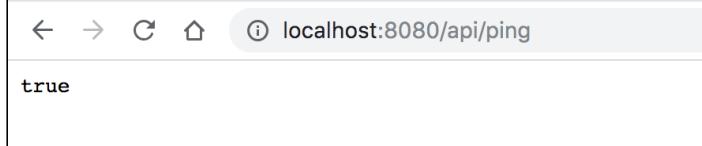
```

$ cd Hygieia
$ java -jar api/target/api.jar --spring.config.location=api/api.properties
-Djasypt.encryptor.password=hygieiasecret

It will display following in the console (Will take few minutes)
-----
Started Application in 9.498 seconds (JVM running for 10.152)

```

3. Test if Hygieia is running, accessing the API URL at <http://localhost:8080/api/ping>



4. Open a new terminal, The Jenkins collector will be needed to start. Hygieia uses collectors to read the data from Jenkins.

```

$ cd Hygieia
$ cd collectors/build/jenkins
$ mvn install
$ java -jar target/jenkins-build-collector-3.0.1.jar --spring.config.
name=jenkins --spring.config.location=application.properties

----- It should display following (It will take few minutes) -----
com.capitalone.dashboard.Application - Started Application in 11.16
seconds (JVM running for 12.55)
INFO c.c.d.collector.CollectorTask - Running Collector: Hudson
INFO c.c.d.collector.CollectorTask -
-----
INFO c.c.d.collector.CollectorTask - http://*****:****@localhost:7070/
INFO c.c.d.collector.CollectorTask -
-----
INFO c.c.d.collector.DefaultHudsonClient - Fetching jobs 0/2 pageSize
1000...

```

```
INFO c.c.d.collector.CollectorTask - Fetched jobs 0s
INFO c.c.d.collector.CollectorTask - New jobs 2 0s
INFO c.c.d.collector.CollectorTask - New builds 0 0s
INFO c.c.d.collector.CollectorTask - New configs 0 0s
INFO c.c.d.collector.CollectorTask - Finished 0s
```

5. Open a new terminal to start the Github collector

```
$ cd Hygieia
$ cd collectors/scm/github
$ java -jar target/github-scm-collector-3.0.1.jar --spring.config.name=github --spring.config.location=application.properties

----- It should display following (It will take few minutes)-----
INFO c.c.d.collector.CollectorTask - Repo Count 1 0s
INFO c.c.d.collector.CollectorTask - New Commits 0 0s
INFO c.c.d.collector.CollectorTask - New Pulls 0 0s
INFO c.c.d.collector.CollectorTask - New Issues 0 0s
INFO c.c.d.collector.CollectorTask - Finished 0s
```

4. Start the User Interface, open a new terminal window.

```
$ cd Hygieia
$ cd UI/
$ gulp serve
```

5. Open a new tab or browser window should open with the URL to access the project. If not, access the following URL: <http://localhost:3000/>

Working Screenshot

The screen shot is taken on March 20, 2019

