ΕΜΕΊΣ

CMXM

CodusMaximus

Hyper Perform
User Manual

Repository: `https://github.com/claudioMDS/hyper-perform`

Developers:
Claudio Da Silva *14205892*
Rohan Chhipa *14188377*
Avinash Singh *14043778*
Jason Gordon *14405025*

Updated September 11, 2016

Contents

1 System Overview

Many different tools are available for measuring the quality of products made, but very few tools exist which assess the quality of the people making said products. People play a huge role in a project, and trying to monitor each and every one becomes a tedious task which diverts man power away from other more critical tasks. Whether it be for an end of year evaluation, or attempting to assess the current status of a project, generating a report on a staff member can help keep up productivity, as well as get them any help they need in order to resume quality performance. By ensuring that there is constant quality performance from each individual on a project, one can increase project quality as well as reduce project risks such as loss of an important team member during a critical stage of a project's life-cycle.

2 System Configuration

This guide has been made for users using a Linux based operating system. To install the HyperPerform system on the machine you will be required to have an active connection to the internet. Please note that high amounts of data might be consumed.

2.1 Docker installation

To install system with ease and avoid all configurations you can download Docker. Docker can be found at `www.docker.com` where guides are made available for installing docker on a particular operating system. If you intend to use Docker to install the HyperPerform system then please ensure you install Docker on your machine.

2.2 Manual Installation

The manual installation requires you to download the source code from the GitHub repository. The newest release is highly recommended. To carry out a manual installation please ensure you have Maven and the WildFly application server on your machine.

Maven can be downloaded from: `maven.apache.org`

WildFly can be downloaded from: `wildfly.org`

Please ensure you download WildFly 10. The HyperPerform system was fully tested on this version of WildFly. Any other version might produce unexpected behaviour.

3 Installation

3.1 Docker Installation

Assuming you have docker installed on your machine, simply run the following command in terminal:

```
docker run hyperperform/HyperPerform
```

This will download the HyperPerform Docker image from DockerHub and run it on you machine.

The front end component does not have a Docker image at this point in time. To install the front end component please look at section 3.2.5 for the manual installation.

3.2 Manual Installation

This installation guide assumes a Linux Server running Ubuntu 14 or higher:

3.2.1 WildFly

Once you have downloaded the WildFly application server please carry out the WildFly installations and add a user. Once this is done proceed to installing PostgreSQL.

3.2.2 PostgreSQL

The install PostgreSQL on your machine:

Install via terminal:

```
sudo apt-get update
sudo apt-get install postgresql postgresql-contrib
```

To configure PostgreSQL to connect remotely:

```
sudo nano /etc/postgresql/9.3/main/postgresql.conf
```

Edit the following lines:

```
listen_addresses = ”*”
```

Create database hyperperform and the tables

Run the following commands in terminal:

```
psql −c ’CREATE DATABASE hyperperform;’ –U postgres

psql –d hyperperform −c ’CREATE TABLE public.”GitPush” ( id
    integer NOT NULL, repository character varying(255), ”
    timestamp” timestamp without time zone, username
    character varying(255), commitsize integer, CONSTRAINT ”
    GitPush_pkey” PRIMARY KEY (id) ); CREATE SEQUENCE public.
    hibernate_sequence INCREMENT 1 MINVALUE 1 MAXVALUE
    9223372036854775807 START 1 CACHE 1;’ –U postgres

psql –d hyperperform −c ’CREATE TABLE public.”TravisEvent” (
     id integer NOT NULL, branch character varying(255),
    commiter character varying(255), repo character varying
    (255), status character varying(255), ”timestamp”
    timestamp without time zone, CONSTRAINT ”TravisEvent_pkey
    ” PRIMARY KEY (id));’ –U postgres

psql –d hyperperform −c ’CREATE TABLE public.”
    CalendarProject” ( projectid integer NOT NULL, calendarid
     character varying(255), collaborators bytea, creator
    character varying(255), duedate timestamp without time
    zone, eventid character varying(255), reponame character
    varying(255), ”timestamp” timestamp without time zone,
    CONSTRAINT ”CalendarProject_pkey” PRIMARY KEY (projectid)
    );’ –U postgres

psql –d hyperperform −c ’CREATE TABLE public.”
    CalendarMeeting” ( meetingid integer NOT NULL, calendarid
     character varying(255), creator character varying(255),
    duedate timestamp without time zone, eventid character
    varying(255), location character varying(255), ”timestamp
    ” timestamp without time zone, CONSTRAINT ”
    CalendarMeeting_pkey” PRIMARY KEY (meetingid));’ –U
    postgres

psql –d hyperperform −c ’CREATE TABLE public.”
    CalendarMeeting_attendees” ( ”CalendarMeeting_meetingID”
    integer NOT NULL, attendees integer, attendees_key
    character varying(255) NOT NULL, CONSTRAINT ”
    CalendarMeeting_attendees_pkey” PRIMARY KEY (”
    CalendarMeeting_meetingID”, attendees_key), CONSTRAINT
    fkn4q1pmj9vx3tfsaw9irp9voax FOREIGN KEY (”
    CalendarMeeting_meetingID”) REFERENCES public.”
    CalendarMeeting” (meetingid) MATCH SIMPLE ON UPDATE NO
    ACTION ON DELETE NO ACTION);' −U postgres

psql −d hyperperform −c 'CREATE TABLE public.”GitIssue”(id
    integer NOT NULL, action character varying(255), assignee
     character varying(255), createdby character varying(255)
    , issueid bigint, repository character varying(255), ”
    timestamp” timestamp without time zone, CONSTRAINT ”
    GitIssue_pkey” PRIMARY KEY (id));' −U postgres
```

3.2.3 ActiveMQ

To setup ActiveMQ on your server:

- Start up your WildFly application server
- Navigate to WildFly management console on localhost:9990
- Navigate to configurations tab and click on sub-systems
- Scroll down and search for Messaging-ActiveMQ and click on it
- Click on default, select queues/topics
- Click add and input the following information:
  - Name*: hyperperform
  - JNDI Names*: java:/jms/queue/hyperperform
- Click save

3.2.4 Deploying to WildFly

To deploy the HyperPerform system to the application you will need to build the system using from the source code.

- Ensure the WildFly server is running.
- Navigate to `https://github.com/HyperPerform/hyper-perform-server/releases` and download the newest release source code.
- Extract the source code

- Navigate to the root directory of the source code. A file named pom.xml should be clearly visible.
- Run the following command: mvn clean wildfly:deploy
- Maven will then ask you to provide your user name and password for the Wildfly Server.
- Thereafter Maven will automatically deploy the compiled code (war) to WildFly

3.2.5 Front-end Dashboard

Please not that there is no release yet for the dashboard and there might be a few bugs, or limitations to the software.
To start up the front end please ensure you have Node 6.4.0 or higher installed on your machine. Node can be found at `https://nodejs.org/en/`.

Please make sure that these commands execute successfully before attempting to run the system:

```
        npm install −g gulp
        npm install −g bower
        npm install −g sass
```

Once that has completed with no errors do the following.

- Download the Dashboard source code from `https://github.com/HyperPerform/hyper-perform-web-application`
- Navigate to the root directory of the source code

Run the following commands in terminal:

```
        npm install
        gulp build
        gulp serve
```

The front-end system will auto launch in your default browser in order to view the data in the front-end system the Wildfly application server must be running.