Seno Adiwicaksono K. (seno@irisiko.com)

# Deploying JakSAFE webapp on Ubuntu 14.04

### Install base libraries and virtualenv (Python Virtual Environment)

- cd ~
- sudo apt-get update
- # install the base libraries
- sudo apt-get -y install build-essential python-pip python-dev python-software-properties git-core
- sudo pip install virtualenv virtualenvwrapper
- nano ~/.bashrc
    - # add to the end
    - export WORKON_HOME=$HOME/.virtualenvs
    - source /usr/local/bin/virtualenvwrapper.sh
- . .bashrc

### Install MySQL database server

- sudo apt-get -y install mysql-server libmysqlclient-dev
    - # follow MySQL configuration and set the root user password
- # create the MySQL user and database for JakSAFE from MySQL CLI
- mysql -u root -p
    - create database jaksafe;
    - grant all privileges on jaksafe.* to 'jaksafe'@'localhost' identified by 'password';
    - set password for 'jaksafe'@'localhost' = PASSWORD('password');
    - flush privileges;
    - exit;

### Create a new virtualenv and pull the source code from the JakSAFE repo

- # create a new virtualenv called ‘jaksafe’
- mkvirtualenv jaksafe
- # cd to the virtualenv home directory
- cdvirtualenv
- pwd
- # default path is: ~/.virtualenvs/jaksafe/
- # initialize a new git repo and pull the source code
- git init
- git remote add origin https://irisiko@bitbucket.org/irisiko/jaksafe.git

- git fetch
- git checkout -t origin/master

## Install the Python package requirements in the virtualenv

- pip install -r requirements.txt
- # wait until installation completes
- # verify that the packages are installed
- pip list

## Sync the JakSAFE webapp database

- cd jaksafe
- cp jaksafe/settings.py.sample jaksafe/settings.py
- nano jaksafe/settings.py
    - # adjust the MySQL database connection settings in DATABASES
    - # adjust the jakservice dirs (use the default)
    - # check PYTHON_EXEC path (default is to use the 'JakSAFE' virtualenv Python binary)
- python manage.py migrate
- python manage.py createsuperuser
    - # create the admin account

### Run the JakSAFE SQL script to create the required tables

- cdvirtualenv
- mysql -u jaksafe -p jaksafe < ./jaksafe_etc/jaksafe.sql

### Configure JakSERVICE repo

- cdvirtualenv
- cd jaksafe/jaksafe/jaksafe/jakservice
- git init
- git remote add origin https://irisiko@bitbucket.org/irisiko/jakservice.git
- git fetch
- git checkout -f -t origin/master
- # perform the JakSERVICE deployment steps before running the web server (refer to jakservice/README.md)

### Optional: install phpMyAdmin for managing the MySQL database

- # install phpMyAdmin set it to listen on port 8080
- sudo apt-get -y install phpmyadmin
    - # during phpMyAdmin setup select:
        - # apache2
        - Yes => enter root user password => leave blank
- sudo php5enmod mcrypt
- sudo nano /etc/apache2/ports.conf
    - # comment: Listen 80 (example: #Listen 80)
    - # add below Listen 80: Listen 8080
- sudo service apache2 restart
- # open in browser http://SERVER_IP:8080/phpmyadmin

**Run the web server**

- # run the dev web server
- python manage.py runserver 0.0.0.0:8000
- # open in browser http://SERVER_IP:8000
- # OR run the web server with Supervisor
- sudo ./start_supervisord.sh
- # OR run Gunicorn web server directly
- sudo ./start_server.sh
- # open in browser http://SERVER_IP
- # to stop the web server (Gunicorn):
    - sudo pkill gunicorn

**Set the web server to always run on server startup**

- sudo crontab -e
    - # add the following entry
    - @reboot /path/to/jaksafe/virtualenv/dir/start_server.sh &

**Done!**

Seno Adiwicaksono K. (seno@irisiko.com)

# Deploying JakSERVICE on Ubuntu 14.04

### Install QGIS and other library requirements

- # install the newer QGIS version from QGIS repo
- cd ~
- sudo nano /etc/apt/sources.list
  - # add to the end
  - deb http://qgis.org/debian trusty main
  - deb-src http://qgis.org/debian trusty main
- gpg --keyserver keyserver.ubuntu.com --recv DD45F6C3
- gpg --export --armor DD45F6C3 | sudo apt-key add -
- sudo apt-get update
- sudo apt-get -y install qgis python-qgis
- # install locale library requirements for pdf report generation
- sudo apt-get -y install language-pack-id
- # install Matplotlib library requirements
- sudo apt-get -y install libpng-dev libfreetype6-dev libxft-dev
- # activate 'jaksafe' virtualenv
- workon jaksafe
- add2virtualenv "/usr/lib/python2.7/dist-packages/"
- cdvirtualenv
- cd jaksafe/jaksafe/jakservice

### Configure JakSERVICE repo (skip if already completed)

- git init
- git remote add origin https://irisiko@bitbucket.org/irisiko/jakservice.git
- git fetch
- git checkout -f -t origin/master

### install JakSERVICE Python package requirements in virtualenv

- pip install -r requirements.txt
- # wait until installation completes
- # verify that the packages are installed
- pip list

### Edit JakSERVICE configuration file

- cp global_conf.cfg.sample global_conf.cfg
- nano global_conf.cfg
    - # set [database configuration]: url_address, user, passwd, database_name, port
    - # set [dims_conf]: url_dims
    - # set [folder_conf]:
        - project_folder = /set/absolute/path/to/jakservice/dir (example: /home/user/.virtualenvs/jaksafe/jaksafe/jaksafe/jakservice)
        - auto_folder = relative path from jakservice dir (use default: ../uploaded/jakservice/auto)
        - adhoc_folder = relative path from jakservice dir (use default: ../uploaded/jakservice/adhoc)
    - # set [directory]:
        - # set absolute path for all, default settings below is provided for example:
        - resource = /home/user/.virtualenvs/jaksafe/jaksafe/jaksafe/uploaded/jakservice/auto/resource/
        - assumptions = /home/user/.virtualenvs/jaksafe/jaksafe/jaksafe/uploaded/jakservice/auto/input/assumptions/
        - aggregate = /home/user/.virtualenvs/jaksafe/jaksafe/jaksafe/uploaded/jakservice/auto/input/aggregat/
        - log = /home/user/.virtualenvs/jaksafe/jaksafe/jaksafe/uploaded/jakservice/auto/output/log/
        - impact = /home/user/.virtualenvs/jaksafe/jaksafe/jaksafe/uploaded/jakservice/auto/output/impact/
        - report = /home/user/.virtualenvs/jaksafe/jaksafe/jaksafe/uploaded/jakservice/auto/output/report/
        - hazard = /home/user/.virtualenvs/jaksafe/jaksafe/jaksafe/uploaded/jakservice/auto/output/hazard/
        - log_adhoc = /home/user/.virtualenvs/jaksafe/jaksafe/jaksafe/uploaded/jakservice/adhoc/output/log/
        - impact_adhoc = /home/user/.virtualenvs/jaksafe/jaksafe/jaksafe/uploaded/jakservice/adhoc/output/impact/

- report_adhoc = /home/user/.virtualenvs/jaksafe/jaksafe/jaksafe/uploaded/jakservice/adhoc/output/report/
- hazard_adhoc = /home/user/.virtualenvs/jaksafe/jaksafe/jaksafe/uploaded/jakservice/adhoc/output/hazard/

**Copy JakSERVICE input files to JakSAFE webapp uploaded directory**

- cp -R auto adhoc ../uploaded/jakservice/

**Run the script to populate fl_event table with past flood reports from DIMS flr API**

- python populate_dims.py

**Create cron jobs for automatic DALA calculation every 6 hours**

- sudo crontab –e
  - # add the following entries
  - 59 5 * * * /home/user/.virtualenvs/jaksafe/bin/python /home/user/.virtualenvs/jaksafe/jaksafe/jaksafe/jakservice/run_dalla_auto.py > /dev/null 2>&1
  - 59 11 * * * /home/user/.virtualenvs/jaksafe/bin/python /home/user/.virtualenvs/jaksafe/jaksafe/jaksafe/jakservice/run_dalla_auto.py > /dev/null 2>&1
  - 59 17 * * * /home/user/.virtualenvs/jaksafe/bin/python /home/user/.virtualenvs/jaksafe/jaksafe/jaksafe/jakservice/run_dalla_auto.py > /dev/null 2>&1
  - 59 23 * * * /home/user/.virtualenvs/jaksafe/bin/python /home/user/.virtualenvs/jaksafe/jaksafe/jaksafe/jakservice/run_dalla_auto.py > /dev/null 2>&1

**Done!**