Contents

[Installation and Release Notes 2](#_Toc8936999)

[survey app 2](#_Toc8937000)

[Software required 3](#_Toc8937001)

[Code Location 3](#_Toc8937002)

[Licensed Software used 3](#_Toc8937003)

[High Level Architecture 3](#_Toc8937004)

[Installations 4](#_Toc8937005)

[React Installation steps 4](#_Toc8937006)

[Python Installation steps 5](#_Toc8937007)

[MongoDB Installation steps 5](#_Toc8937008)

[Collection structure/Creation guideline: 6](#_Toc8937009)

[Deployment 7](#_Toc8937010)

[Application Startup 9](#_Toc8937011)

[Start and Stop the services in AWS 10](#_Toc8937012)

# Installation and Release Notes

# survey app

## Software required

* Python 3.5 (virtual environment)
* Flask Framework
* MongoDB
* React
* Graphs

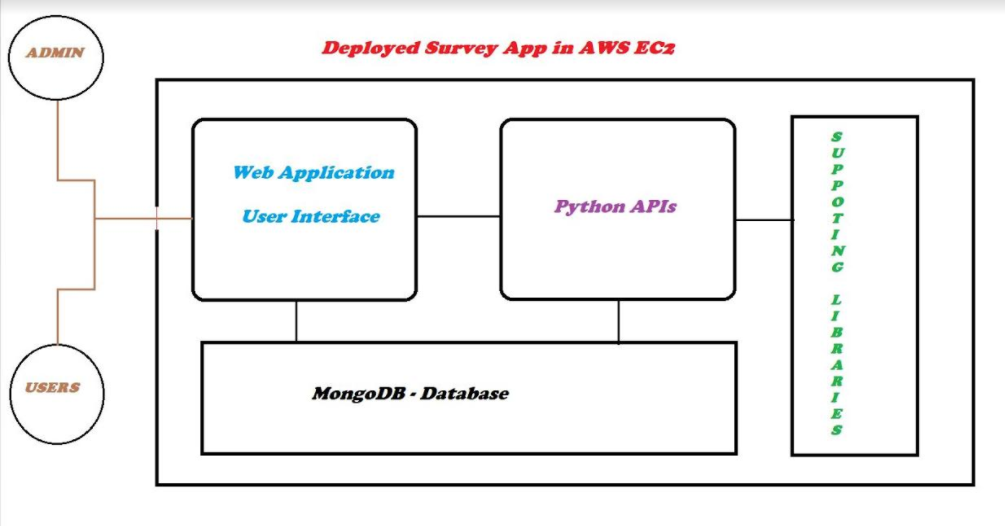
## Code Location

[**https://github.com/ITH-Survey/codebase**](https://github.com/ITH-Survey/codebase)

## Licensed Software used

* SMTP – mention if we would have used any licensed

## High Level Architecture



## Installations

### React Installation steps

* Install node.js
  + Command - yum install nodejs
* Install npm
  + Command - yum install npm

Please run below command <react\_app\_folder\_name> will be the folder name of your desire.

* npx create-react-app <react\_app\_folder\_name>
* cd <react\_app\_folder\_name>
* npm start

now copy the code provided to you(GitLocation) to <react\_app\_folder\_name>. now try running below command

* npm build
* npm start

## Python Installation steps

**Requirements:**

1. Python 3.5 (virtual environment)

2. Flask Framework

3. MongoDB

**Virtual environment:**

This isolates them cleanly from any “system” packages you have installed and yields the added bonus of not requiring root privileges to install additional Python packages.

Command to install virtual environment:

* pip install virtualenv

Command to initialize virtual environment:

* virtualenv -p python3 envname

This will create a Python 3.x compatible virtual space for your Python application. To start the environment on windows run the command .\env\Scripts\activate and to start it on linux machine run the command source env/bin/activate

Before running Metabase install all the required modules with the help of requirements.txt file packaged along with zip file. To-do so run the below command:

* pip install –r requirements.txt

We can run the application by python run.py runserver –host 0.0.0.0

* python3 run.py runserver –host 0.0.0.0

## MongoDB Installation steps

Windows:

Go to http://www.mongodb.org and download the latest production release zip file for Windows—choosing 32-bit or 64-bit depending on your system. Extract the contents of the zip file to a location like C:\mongodb and add the bin directory to your PATH.

RHEL:

* sudo yum install mongo-stable-server

Ubuntu / Debian:

* sudo apt-get update; sudo apt-get install mongodb

FreeBSD:

* sudo pkg\_add -r mongodb

Mac OS:

Go to http://www.mongodb.org and download the latest production release compressed tar file for OS X—choosing 32-bit or 64-bit depending on your

system. Extract the contents to a location like /usr/local/ or /opt and add the bin directory to your $PATH.

## Collection structure/Creation guideline:

Inserting into database:

db.collection.insert()

Inserts a document or documents into a collection.

· db refers to the current database.

· collection is the name of the collection.

Finding collection in database:

db.collection.find(query, projection) Selects documents in a collection and returns a cursor to the selected documents.

Updating in database:

db.collection.update(query, update, options)

Modifies an existing document or documents in a collection. The method can modify specific fields of an existing document or documents or replace an existing document entirely, depending on the update parameter.

Removing in database:

db.collection.remove()

Removes documents from a collection.

## Deployment

**Deployment:**

1. Open winscp or anyother client.

2. Transfer the extracted zip file into the target system, where the api has to be deployed.

3. Setup a virtual environment using python .

Installation of Python 3.6.3 wget http://www.python.org/ftp/python/3.6.3/python-3.6.3.tar.xz

**decode:**

* xz -d python -3.6.3.tar.xz
* tar –xvf python -3.6.3.tar
* cd python-3.6.3
* sudo ./configure –enable –optimization
* sudo make && make altinstall

**#Go inside the project folder and create virtual environment.**

* virtualenv env –p python3.6
* source env/bin/active
* python –version

**#Install requirement.txt via :**

* pip install –r requirement.txt
* pip install uwsgi

#After installing uwsgi create simple python file and name it as wsgi.py and save it. Copy the below program and paste it.

from config import app

if \_\_name\_\_==”\_\_main\_\_”

app.run()

#To test if uwsgi use the below

* uwsgi --socket 0.0.0.0:5000 –protocol = http –w wsgi:app

(OR)

* uwsgi --socket 0.0.0.0:8080 --protocol=http -w wsgi

#visit the server domain:

{{ip}}:5000/

#copy the provided ini and place it inside the project folder.

Create afile with .INI extension and paste the provided content inside this file .example like below

[uwsgi]

module = wsgi:application

master = true

processes = 5

uid = user

socket = /run/uwsgi/myapp.sock

chown-socket = user:nginx

chmod-socket = 660

vacuum = true

die-on-term = true

#To run metabase as service, paste service file uwsgi.service inside : /etc/systemd/system/

[Unit]

Description=uWSGI instance to serve myapp

[Service]

ExecStartPre=-/usr/bin/bash -c 'mkdir -p /run/uwsgi; chown user:nginx /run/uwsgi'

ExecStart=/usr/bin/bash -c 'cd /home/user/myapp; source myappenv/bin/activate; uwsgi --ini myapp.ini'

[Install]

WantedBy=multi-user.target

## Application Startup

Now, we can start the service by typing:

* sudo systemctl start uwsgi

Check that it started without issues by typing:

* systemctl status uwsgi

If there were no errors, enable the service so that it starts at boot by typing:

* sudo systemctl enable uwsgi

**Configure Nginx to Proxy to uWSGI**

First install Nginx:

* sudo yum install nginx
* sudo systemctl start nginx
* sudo systemctl status nginx

We will be modifying the existing nginx.conf file and adding a new server block. Open the file with sudo for editing:

Before the default server block, we will add our own server block:

sudo vim /etc/nginx/nginx.conf

server {

listen 80;

server\_name server\_domain\_or\_IP;

location / {

include uwsgi\_params;

uwsgi\_pass unix:/run/uwsgi/myapp.sock;

}

}

You can test to make sure that your Nginx configuration is valid by typing:

* sudo nginx -t

If this returns without any errors, start the service by typing:

* sudo systemctl start nginx

Start Nginx at boot by enabling the service:

* sudo systemctl enable nginx

You should be able to go to your server's domain name or IP address (without a port number) and see the application you configured:

## Start and Stop the services in AWS

**Start the services:**

1. Login to the AWS EC2 instance
2. Run below commands:

* *sudo su*
* *systemctl start nginx*
* *systemctl start uwsgi*

**Stop the services:**

1. Login to the AWS EC2 instance
2. Run below commands:

* *sudo su*
* *systemctl stop nginx*
* *systemctl stop uwsgi*

**To Start/Stop the EC2 Instance:**

1. Login to the AWS Administrative console
2. Select Services 🡪 EC2 🡪 Running Instances
3. Select the required EC2 Instance and click on start/stop as shown in below :

