SmartAT instruction

This document provides detailed instructions on how to deploy SmartAT and start it, please follow the steps one by one and make sure every last step get a positive response before moving on.

**Deploy SmartAT**

**Requirement list,**

1, PC with Ubuntu 12.04 LTS (Recommended) or 12.10 release

* 64 bit required
* The OS is up to date(apt-get upgrade)

2, Functional python 2.7.X environment.

3, Download the source code of SmartAT from Github or FTP server.

**Step 1,**

Install required runtime libraries.

Execute following commands in terminal.

sudo apt-get install libssl-dev python-dev libevent-dev

**Step 2,**

Install pip in order to install more python packages.

Execute following commands in terminal,

sudo apt-get install python-pip

**Step 3,**

Install needed python libraries

Since there are a bunch of them, we developed a script to do this, find requirements.txt under the root folder of SmartAT source code then execute following commands in terminal,

sudo pip install -r requirements.txt --use-mirrors

In the coming three steps, you are going to install DBMS needed by SmartAT, please be noticed that you can either install them on other PC or all on one as you wish, since you know SmartAT just needs theirs IP address.

**Step 4,**

Install mongodb,

Execute following commands in terminal,

sudo apt-get install mongodb

After mongodb is installed successfully, we need to modify its configuration(/etc/mongodb.conf),

* Locate “bind\_ip = 127.0.0.1” and replace it with “#bind\_ip = 127.0.0.1”
* Locate “#port = 27017” and replace it with “port = 27017”

Execute following commands to restart mongodb,

sudo service mongodb restart

**Start SmartAT**

Note: before any step in this chapter, please make sure that you have executed all the 6 steps in the last chapter successfully

**Step 1,**

Start web server,

We provide two ways to start web server of SmartAT,

Navigate to the root folder of SmartAT source code

**a. For testing and internal debug purpose**

execute following commands in terminal,

MONGODB\_URI=mongodb://localhost:27017 WEB\_PORT=8080 python app.py

**b. For deployment to end-user, use g-unicorn as container**

MONGODB\_URI=mongodb://localhost:27017 gunicorn -k "geventwebsocket.gunicorn.workers.GeventWebSocketWorker" --workers=2 --bind=:8080 app:app --pid /tmp/smart-web.pid --daemon --log-syslog --access-logfile ./server.log

**Step 2,**

Start worker server,

There are 2 ways to start worker server, execute commands below

Single node server,

MONGODB\_URI=mongodb://localhost:27017 celery worker --app=smartserver.v1.worker:worker --logfile ./worker.log &

Multi nodes server(4 in this example if CPU of the host PC has 4 cores),

MONGODB\_URI=mongodb://localhost:27017 celery multi start 4 --app=smartserver.v1.worker:worker -P gevent -l info -c:1-4 1000 --logfile ./worker.log

**Step 3,**

Start regular tasks,

execute following commands in terminal,

MONGODB\_URI=mongodb://localhost:27017 celery beat --app=smartserver.v1.worker:worker --pid=/tmp/periodic\_task.pid --detach --logfile ./beat.log

If you have installed the DBMS in different PCs, please replace "localhost" in the command-line with actual IP address.

If you know these servers, you are welcomed to customize the starting commands and configurations but please be sure you know what you are doing.