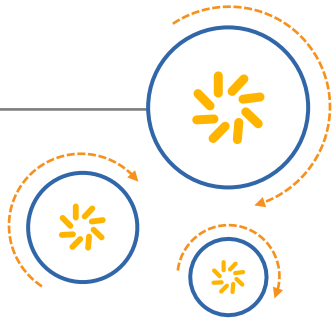
Qualcomm Technologies, Inc.



Lost & Stolen Device Subsystem 1.0.0

API Installation Guide

LSDS-API-Installation-Guide-1.0.0

June 19, 2018

Revision history

Revision	Date	Description
A		Initial release

Contents

Tables

1 Introduction

1.1 Purpose

This document provides:

- Installation instructions for the Lost & Stolen Device Subsystem (LSDS)
- Instructions for running test commands to verify LSDS API installation

1.2 Definitions, Acronyms & Abbreviations

Table 1- Definitions, Acronyms & Abbreviations

Term	Explanation
DIRBS	Device Identification, Registration & Blocking System
LSDS	Lost & Stolen Device Subsystem
OS	Operating System
PostgreSQL	PostgreSQL open source object-relational database system
Nginx	An open source, lightweight, high-performance web server or proxy server
uWSGI	uWSGI is used for serving Python web applications
API	Application Program Interface

1.3 References

N.A

1.4 Getting Started

The instructions provided in this document assume that the required equipment (hardware, software) has been installed and configured with Ubuntu 16.04. Refer to the Ubuntu Installation Guide for additional installation help.

The installer should be familiar with Linux command line.

2 Installation

NOTE: The reader acknowledges and agrees that he is entirely and solely responsible for the selection and use of all third-party software modules downloaded and installed by this installation method, including securing all appropriate and proper rights of use to any of such third-party software modules and to comply fully with any terms of use that may apply to or accompany any such third-party software modules.

Qualcomm Technologies, Inc. does not undertake any obligations, duties, or other responsibilities in connection with the selection or use by the reader of any of such third-party software modules.

2.1 System Requirements

2.1.1 Software Requirements

- Python 3.X
- Ubuntu 16.0
- PostgreSQL 10
- Nginx 1.14.X
- uWSGI 2.0

2.1.2 Hardware Requirements

Minimum hardware requirements

- 512 MB of RAM
- 1G of available disk space

2.1.3 Operating System

This system will be installed and configured with Ubuntu 16.04. Refer to the Ubuntu Installation Guide for additional installation help.

2.1.4 Database Support

NOTE: Create a new database on PostgreSQL 10 instance.

A complete guide for PostgreSQL installation and configuration can be found on PostgreSQL website

2.2 Extracting Software Release

The LSDS software release is distributed as a tar.gz file. To extract the contents of the distribution, run:

```
tar xvzf dirbs-lsds-api-1.0.0.tar.gz
```

Copy the contents to the web root directory e.g. /var/www/html (default Nginx web root directory)

2.3 Manual Installation

- Ensure the APT package index is updated
  ```
  apt-get update --fix-missing
  ```

- Install basic required packages
  ```
  apt-get install nginx git python3 virtualenv libpython3-dev
  python3-pip python3-dev
  ```

- Go to path /var/www/html/dirbs-lsds-api-1.0.0
  ```
  pushd /var/www/html/dirbs-lsds-api-1.0.0
  ```

- Create virtual environment and install requirements
  ```
  virtualenv –p python3 venv
  source venv/bin/activate
  pip3 install -r requirements.txt
  ```

- Nginx does not support python application so we need to install uWSGI to run python application through Nginx, below is the command to install uWSGI

  ```
  pip3 install uwsgi
  deactivate
  ```


3 Configuration

3.1 Nginx Configuration

Remove Nginx default configuration and create new configuration file for the LSDS app

```
rm /etc/nginx/sites-enabled/default
```

- Now create a new configuration file in the root path

```
nano /var/www/html/dirbs-lsds-api-1.0.0/lsds.conf
```

- Copy the below lines

```
server {
listen      80;
server_name localhost;
charset     utf-8;
client_max_body_size 75M;
location / {try_files $uri @dirbs-lsds-api-1.0.0;}
location @dirbs-lsds-api-1.0.0
{
include uwsgi_params;
uwsgi_pass unix:/var/www/html/dirbs-lsds-api-1.0.0/uwsgi.sock;
  }
}
```

- Symlink the new created file to Nginx's configuration files directory and restart Nginx

```
ln -s /var/www/html/dirbs-lsds-api-1.0.0/lsds.conf \
/etc/nginx/conf.d/
```

- Verify nginx configuration

```
nginx –t
```

- Restart Nginx Service

```
service nginx restart
```


3.2 uWSGI Configuration

- Create a new configuration file in the root path and copy the below lines
  `nano /var/www/html/dirbs-lsds-api-1.0.0/uwsgi.ini`

- Add below lines in this configuration file:

```
[uwsgi]

#application's base folder
base = /var/www/html/dirbs-lsds-api-1.0.0

#python module to import
app = run
module = %(app)
chdir = %(base)
home = %(base)/venv
pythonpath = %(base)
master = true
processes = 10
cheaper = 2
cheaper-initial = 5
cheaper-step = 1
cheaper-algo = spare
cheaper-overload = 5

#socket file's location
socket = /var/www/html/dirbs-lsds-api-1.0.0/%n.sock

#permissions for the socket file
chmod-socket = 666
chown-socket = www-data:www-data

#ownership of uwsgi service
uid = www-data
gid = www-data

#the variable that holds a flask application inside the module imported at
line #6
callable = app

#location of log files
logto = /var/log/uwsgi/%n.log
```

- Create a directory vassals in /etc/uwsgi/
  `mkdir -p /etc/uwsgi/vassals`

- Create Symlink in this directory to uwsgi ini config file
  `ln -s /var/www/html/dirbs-lsds-api-1.0.0/uwsgi.ini \`
  `/etc/uwsgi/vassals/uwsgi.ini`

- Create a new directory for log files
  `mkdir -p /var/log/uwsgi`

- Change ownership of the web root directory and logs directory to the web-user


```
chown -R www-data:www-data /var/www/html/dirbs-lsds-api-1.0.0/
chown -R www-data:www-data /var/log/uwsgi/
```

3.3 uWSGI Service Configuration

Configure the uWSGI to run as a service on the server.

- Create an init script at location
  `nano /etc/systemd/system/uwsgi.service`

- Copy below lines in to the script file

```
[Unit]
Description=uWSGI Emperor service
After=syslog.target

[Service]
ExecStart=/var/www/html/dirbs-lsds-api-1.0.0/venv/bin/uwsgi \-emperor \
/etc/uwsgi/vassals/
Restart=always
KillSignal=SIGQUIT
Type=notify
StandardError=syslog
NotifyAccess=all

[Install]
WantedBy=multi-user.target
```

- Reload system defaults to update the script in system services
  `systemctl daemon-reload`

- Start uwsgi to start the application
  `service uwsgi start`

- Go to the web-browser and enter the URL of the server to check the service running

3.4 LSDS Configuration and Initialization

To configure file according to database server and credentials, edit config.ini in directory /var/www/html/dirbs-lsds-api-1.0.0 and edit config.ini.

Change the hostname, port, username, password and database name as per requirements. Within the same directoy of LSDS root, activate the virtual environment.

To activate the virtual environment created earlier at `/var/www/html/dirbs-lsds-api-1.0.0` and start database initialization.

```
cd /var/www/html/dirbs-lsds-api-1.0.0
source venv/bin/activate
```

- Follow below steps for database initialization and migration

```
python manage.py db init
python manage.py db migrate
```



```
python manage.py db upgrade
python manage.py DbTrigger
python manage.py CreateView
python manage.py Seed
```

- Restart uWSGI service

```
service uwsgi restart
```


4 Testing

- To check nginx configuration for errros

```
nginx –t
```

- To get detailed logs of uWSGI service. uWSGI can be run without service command in foreground

```
uwsgi --ini /var/www/dirbs-lsds-api-1.0.0/uwsgi.ini
```

