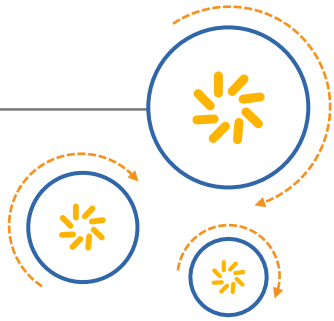
Qualcomm Technologies, Inc.



DIRBS View 1.0.0

Installation Guide

DIRBS-View-Installation-Guide-1.0.0

Sep 5, 2019

Revision history

Revision	Date	Description
A		Initial release

Contents

Tables

1 Introduction

1.1 Purpose & Scope

This document provides:

- Installation instructions for the DIRBS View
- Instructions for running test commands to verifyDIRBS View API installation

1.2 Definitions, Acronyms & Abbreviations

Table 1- Definitions, Acronyms & Abbreviations

Term	Explanation
DIRBS	Device Identification, Registration & Blocking System
OS	Operating System
uWSGI	uWSGI is used for serving Python web applications
API	Application Program Interface
Elasticsearch	Elasticsearch is a search engine based on the Lucene library. It provides a distributed, multitenant-capable full-text search engine with an HTTP web interface and schema-free JSON documents.

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
- uWSGI 2.0
- virtual env
- Java 8

2.1.2 Hardware Requirements

Minimum hardware requirements for API Installation

- At least 1 GB of RAM
- At least 8G of disk space

Minimum hardware requirements for elasticsearch

- At least 8 GB of RAM
- At least 100 G of disk space

2.1.3 Operating System

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

- Ubuntu 16.04
- non-root user

You should have a regular, non-root user account on your server with sudo privileges (in this installation guide the user is ‘user’)

2.2 Elasticsearch Installation on Ubuntu 16.04

- Ensure the APT package index is updated

```
sudo apt-get update
```

- Install basic required packages

```
sudo apt-get install curl openjdk-8-jdk zip unzip wget -y
```

- Create "JAVA_HOME" environment variable by following commands:

```
echo "export JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk-amd64" >>
~/.bashrc
```

```
source .bashrc
```

```
echo $JAVA_HOME
```

- Install below package and import GPG key for Elasticsearch packages.

```
sudo apt-get install apt-transport-https
```

- Import GPG key for Elasticsearch packages through following command

```
wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch |
```

```
sudo apt-key add –
```

- Then configure the “APT” repository on your Ubuntu system. The below command will add a repository to install latest Elasticsearch on your system.

```
sudo add-apt-repository "deb
https://artifacts.elastic.co/packages/7.x/apt stable main
```

- Run the following commands to update cache

```
sudo apt-get update
```

- Install Elasticsearch packages on your system.

```
sudo apt-get install elasticsearch
```

2.3 Configuration of Elasticsearch

- Edit configuration file in your favorite text editor and update it

```
sudo nano /etc/elasticsearch/elasticsearch.yml
```

- Change the following values

```
network.host: 0.0.0.0
cluster.name: myCluster1
node.name: "myNode1"
cluster.initial_master_nodes: ["node-1", "node-2"]
```


2.3.1 Start Elasticsearch

- To start automatically when the system boots up, run the following commands

```
sudo systemctl enable elasticsearch.service
```

- To start elasticsearch sevices

```
sudo systemctl start elasticsearch.service
```

- To stop elasticsearch services

```
sudo systemctl stop elasticsearch.service
```

- To test Elasticsearch working properly on server, run below mentioned command

```
curl -X GET "http://localhost:9200/?pretty"
```

2.4 Extracting Software Release

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

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

Copy the contents to the user home directory e.g. /home/user (you may have different home directory according to user)

2.5 API Manual Installation

- Ensure the APT package index is updated

```
sudo apt-get update --fix-missing
```

- Install basic required packages

```
sudo apt-get install nginx git python3 virtualenv libpython3-dev
python3-pip python3-dev
```

- Go to path /home/user/dirbs-view-api-1.0.0

```
pushd /home/user/dirbs-dps-view-1.0.0
```

- Create virtual environment and activate it

```
virtualenv –p python3 venv
source venv/bin/activate
```

- Install all libraries from requirements.txt in virtual environment

```
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 the virtual environment:

```
Deactivate
```


3 Configuration for API

To make API working properly on server, configure the uWSGI services first.

3.1 uWSGI Configuration

- Create a new configuration file in the root path and copy the below lines
  `nano /home/user/dirbs-view-api-1.0.0/uwsgi.ini`

- Add below lines in this configuration file:

```
[uwsgi]
#application's base folder
base = /home/user/dirbs-view-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 = /home/user/dirbs-view-api-1.0.0/%n.sock

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

#ownership of uwsgi service
uid = user
gid = user

#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/
  `sudo mkdir -p /etc/uwsgi/vassals`

- Create Symlink in this directory to uwsgi ini config file


```
sudo ln -s /home/user/dirbs-view-api-1.0.0/uwsgi.ini \
/etc/uwsgi/vassals/uwsgi.ini
```

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

- Change ownership of the logs directory to dps-user
  `sudo chown -R user:user /var/log/uwsgi/`

3.2 uWSGI Service Configuration

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

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

- Copy below lines in to the script file

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

[Service]
ExecStart=/home/user/dirbs-view-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
  `sudo systemctl daemon-reload`

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

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

4 Testing

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

```
uwsgi --ini /home/user/dirbs-view-api-1.0.0/uwsgi.ini
```