

GIS - Support #7853

Install OpenDroneMap and WebODM

08/03/2019 12:13 - Philippe May

Status:	New	Start date:	08/03/2019
Priority:	Normal	Due date:	
Assignee:	Philippe May	% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:		Spent time:	0.00 hour
Description			
We need to understand the ODM software's capabilities:			
<ul style="list-style-type: none">• use a virtual machine (set up with Windows' Hyper-V on one of the CAD machines)• use pictures taken by Chennai's drone surveyor in Feb. 2019 (reference?)			
This ticket involves exploring the software, before eventually using it in production, which would require a more solid process established after this experience			

History

#1 - 08/03/2019 12:21 - Philippe May

Quite few steps have been done so far, not documented in details, unfortunately.

Summary:

- installed a Debian Stretch VM on Hyper-V: IP: 192.168.1.31, name (set in local DNS): odm1.csr.av
- installed (on root account of that VM, quick and dirty but hey it's a VM accessible only locally) ODM from github, in /root/OpenDroneMap
- setup ODM, which has quite a few dependencies: not too easy, not too difficult. Mostly, using packages from Debian when available, otherwise with Python/pip
- setup project_path in settings.yml to /usr/local/odm

So far, the installation seems to be successful:

```
cd /root/OpenDroneMap
./run.sh TestProject1
```

Next steps:

- Install WebODM
- Setup a share with Windows to access the images

#2 - 08/03/2019 16:33 - Philippe May

Installing WebODM:

Install Debian packages (~ found in requirements.txt and trying to launch the web server):

```
apt install python3-venv nginx redis-server celeryd grass
```

Create a virtual environment (should have done it for ODM, will redo probably later on):

```
cd ~/WebODM
python3 -m venv /usr/local/venv/webodm
pip install -r requirements.txt
```

#3 - 08/03/2019 17:04 - Philippe May

Requires a postgres server, let's install it locally as i don't want to touch our DB server in this phase:

```
apt install postgres
```

Create a superuser (yark):

```
su - postgres
createuser -s -P webodm
```

Configure webodm/settings.py to use this connection.

Try to start the server:

```
./start.sh
```

The script has populated the database, run nginx as a process (bad: doesn't play well with the system).

Next, it wants node and npm. Let's try as found on https://linuxhint.com/install_npm_debian/.

OK (after temporarily deactivating the Aurinoco apt cacher config), npm is installed.

Running ./start.sh again, the node packages are fetched and should be able to be build.

Next episode tomorrow as now is mullu time :)

#4 - 08/03/2019 17:11 - Philippe May

Just before going:

```
FileNotFoundError: [Errno 2] No such file or directory: 'webpack-cli'
```

#5 - 11/03/2019 12:45 - Philippe May

Installed webpack and webpack-cli:

```
npm install -g webpack webpack-cli
```

Then (in WebODM base dir):

```
npm install
```

Then:

```
webpack
```

Launch the application (nginx frontend started by the ./start.sh script: `nginx -c /root/WebODM/nginx/nginx.conf`):

```
cd /root/WebODM  
gunicorn webodm.wsgi --bind unix:/tmp/gunicorn.sock --timeout 300000 --max-requests 250 --preload
```

Check at <http://odm1.csr.av:8000> : server running, but static files aren't served properly.

#6 - 11/03/2019 12:46 - Philippe May

Next step: configure the nginx frontend correctly.

#7 - 11/03/2019 16:22 - Philippe May

Installed config with standard nginx location (/etc/nginx). Tweaked quick and dirty permissions for the /root/WebODM directory: <http://odm1.csr.av:8000> works as expected.

Created an account "admin" with a password.

WebODM asks for "Add processing node" with hostname, port and token, so ODM needs to be run as a service.

#8 - 11/03/2019 16:47 - Philippe May

- Assignee changed from Philippe May to Giulio Di Anastasio

Apparently, we also need nodeodm: <https://www.opendronemap.org/nodeodm/>

1) Install PotreeConverter and LASzip dependency

```
mkdir /usr/local/staging
git clone https://github.com/pierotofy/LAStools /usr/local/staging/LAStools
cd /usr/local/staging/LAStools/LASzip
mkdir build
cd build
cmake -DCMAKE_BUILD_TYPE=Release ..
make

git clone https://github.com/pierotofy/PotreeConverter /usr/local/staging/PotreeConverter
cd /usr/local/staging/PotreeConverter
mkdir build
cd build
cmake -DCMAKE_BUILD_TYPE=Release -DLASZIP_INCLUDE_DIRS=/usr/local/staging/LAStools/LASzip/dll -DLASZIP_LIBRARY
=/usr/local/staging/LAStools/LASzip/build/src/liblaszip.a ..
make && sudo make install
```

2) Install gdal2tiles.py script, node.js and npm dependencies

```
cd /usr/local/
git clone https://github.com/OpenDroneMap/NodeODM
cd NodeODM
npm install
```

3) Start NodeODM

```
cd /usr/local/NodeODM
node index.js --odm_path=/root/OpenDroneMap
info: Authentication using NoTokenRequired
info: No tasks dump found
info: Checking for orphaned directories to be removed...
info: Server has started on port 3000
```

Could add the node in the admin.

I created a user "giulio" with superuser rights, so one should be able to start exploring ODM and WebODM at <http://odm1.csr.av:8000>

Quite some work to be done to probably fix, and rationalize later the installation: please consider it to be a playground only, and start filling tickets...

- There's a (required) component called "NodeODM" which i haven't set up to start automatically when the VM starts.
To start it: see point 3 just above

- Need also to start a gunicorn process: see comment #5

#9 - 12/03/2019 11:15 - Philippe May

An idea for later, since the architecture of ODM, nodeODM and WebODM would allow to have a distributed computing environment.

I'm thinking of: WebODM running on a VM of our main server, and "computing nodes" in VMs across our network (hyper-V on Windows, etc).

#10 - 22/03/2019 17:12 - Philippe May

Created systemd services (in /etc/systemd/system) for:

- gunicorn (<https://docs.gunicorn.org/en/stable/deploy.html>)
- node (<https://thomashunter.name/posts/2016-09-27-running-a-node-js-process-on-debian-as-a-systemd-service>)

WebODM OK.

The node service errors, and manually we're getting an error:

```
(webodm) root@odm1:/usr/local/NodeODM# /usr/bin/node index.js --odm_path=/root/OpenDroneMap
error: Error during startup: Could not load list of options from OpenDroneMap. Is OpenDroneMap installed in /r
oot/OpenDroneMap? Make sure that OpenDroneMap is installed and that --odm_path is set properly: Unexpected tok
en F in JSON at position 2
```

Will follow up later.

#11 - 23/03/2019 12:50 - Philippe May

- Assignee changed from Giulio Di Anastasio to Philippe May

Added nodeODM service, with a config file /etc/nodeODM.json.

Web site seems OK, node is shown, but the test task doesn't start.

#12 - 23/03/2019 12:57 - Philippe May

Giulio tried another task and got a misleading error message, like 'only jpg can be added', but the files are actually jpg

#13 - 23/03/2019 13:07 - Philippe May

Most probably a bad error message, related to <https://github.com/OpenDroneMap/WebODM/issues/615>

In the journal log, got this error:

```
Mar 23 12:59:02 odm1 gunicorn[136464]: PermissionError: [Errno 13] Permission denied: '/root/WebODM/app/media/
project/4'
```

So, better than tweaking permissions, let's spend a bit of time configuring the web server properly.

#14 - 23/03/2019 16:09 - Philippe May

It will takes more time to understand how webODM can be properly configured, so another quick and dirty trick:

```
chown -R www-data: /root/WebODM/app/media/project/
```

Also, found in node's logs (/var/log/nodeODM/node-OpenDroneMap.log):

```
2019-03-23T10:30:00.005Z - error: Could not dump tasks: EACCES: permission denied, open 'data/tasks.json'
```

Yet another quick and dirty trick:

```
root@odm1:~# chown -R www-data: /usr/local/nodeODM/data/tasks.json
```

#15 - 23/03/2019 16:49 - Philippe May

One can access a web interface for odm-node, eg. at <http://odm1.csr.av:3000>

So i tried to create a task there, uploading one file, and after changing again some ownership to www-data (/usr/local/nodeODM/[tmp|data]), i could start the processing.

It failed with the error:

```
File "/root/OpenDroneMap/scripts/dataset.py", line 16, in make_odm_photo
force_ccd)
File "/root/OpenDroneMap/opendm/types.py", line 35, in __init__
self.parse_exif_values(path_file, force_focal, force_ccd)
File "/root/OpenDroneMap/opendm/types.py", line 103, in parse_exif_values
ccd_widths = system.get_ccd_widths()
File "/root/OpenDroneMap/opendm/system.py", line 15, in get_ccd_widths
with open(context.ccd_widths_path) as f:
IOError: [Errno 2] No such file or directory: '/root/OpenDroneMap/SuperBuild/src/opensfm/opensfm/data/sensor_data.json'
```

Enough for today, my mind is too full of errors. Will continue next week.

#16 - 29/03/2019 16:54 - Philippe May

Quick and dirty fix for openSFM, which was installed with normal pip and ODM isn't a nice player because it assumes the packages are installed to its predefined location:

```
root@odml:~/OpenDroneMap/SuperBuild/src/opensfm# ln -s /usr/local/lib/python2.7/dist-packages/opensfm-0.4.1a7-py2.7.egg/opensfm .
```

For the opensfm script (hardcoded in /root/OpenDroneMap/opendm/system.py, why do they execute it as subprocess when it's a python module??):

```
ln -s /usr/local/lib/python2.7/dist-packages/opensfm-0.4.1a7-py2.7.egg/EGG-INFO/scripts /root/OpenDroneMap/SuperBuild/src/opensfm/bin
```

#17 - 29/03/2019 17:03 - Philippe May

Next error:

```
ImportError: No module named pyopengv
```

Building from source:

```
cd /root
mkdir libs
git clone --recursive https://github.com/laurentkneip/opengv.git # Recursive is for python/pybind11
cd opengv
cmake -D BUILD_PYTHON=ON . # This was actually done using ccmake (ncurses interface for cmake)
make
make install
```

Installed in standard location (/usr/local/lib/, /usr/local/include/opengv), so probably need to be installed in /root/OpenDroneMap. But now, mullu time.

#18 - 03/04/2019 15:54 - Philippe May

For opengv: added in /etc/systemd/system/odm-node.service:

```
Environment=PYTHONPATH=/usr/local/lib/python2.7/dist-packages
```

Also, changed log file locations of both services (odm-gunicorn and odm-node) to /var/log/openDroneMap

Found that the node service executes python through a shell script, in /var/log/openDroneMap/node-OpenDroneMap.log:

```
2019-04-03T06:27:42.578Z - info: About to run: /root/OpenDroneMap/run.sh --project-path /usr/local/nodeODM/data-d1837535-25b2-4cb3-8e15-e236f93a9a03
```

Let's see:

```
www-data@odml:/root/OpenDroneMap/opendm$ cat /root/OpenDroneMap/run.sh
#!/bin/bash
```

```
RUNPATH="$( cd "$( dirname "${BASH_SOURCE[0]}" )" && pwd )"
export PYTHONPATH=$RUNPATH/SuperBuild/install/lib/python2.7/dist-packages:$RUNPATH/SuperBuild/src/opensfm:$PYTHONPATH
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$RUNPATH/SuperBuild/install/lib
python $RUNPATH/run.py "$@"
```

So, that script tells us that python modules run by node is probably expected in the (messy) ODM installation in that **SuperBuild** directory, but the python executable is set by the environment. Ideally, it should be in a venv.

#19 - 03/04/2019 17:06 - Philippe May

Some progress: after installing (correctly?) pyopengv, it's available:


```
www-data@odml:~$ python -c 'import pyopengv' && echo OK
OK
```

I think it's moving on, as WebODM now displays quickly some activity when but WebODM is still stuck with "Resizing images..."

Trying to run manually gives some clue:

```
www-data@odml:~$ /root/OpenDroneMap/run.sh --project-path /usr/local/nodeODM/data/2d97d042-0427-4fe4-9696-f10dec0d1d3c
[INFO] Initializing OpenDroneMap app - Wed Apr 03 17:03:39 2019
[INFO] Running ODM Load Dataset Cell
[DEBUG] Loading dataset from: /usr/local/nodeODM/data/2d97d042-0427-4fe4-9696-f10dec0d1d3c/images
[INFO] Loading images database: /usr/local/nodeODM/data/2d97d042-0427-4fe4-9696-f10dec0d1d3c/images.json
[INFO] Found 2 usable images
[DEBUG] running /root/OpenDroneMap/build/bin/odm_extract_utm -imagesPath /usr/local/nodeODM/data/2d97d042-0427-4fe4-9696-f10dec0d1d3c/images/ -imageListFile /usr/local/nodeODM/data/2d97d042-0427-4fe4-9696-f10dec0d1d3c/img_list.txt -outputCoordFile /usr/local/nodeODM/data/2d97d042-0427-4fe4-9696-f10dec0d1d3c/odm_georeferencing/coords.txt -logFile /usr/local/nodeODM/data/2d97d042-0427-4fe4-9696-f10dec0d1d3c/odm_georeferencing/odm_georeferencing_utm_log.txt
/bin/sh: 1: /root/OpenDroneMap/build/bin/odm_extract_utm: not found
[WARNING] Could not generate coordinates file. Ignore if there is a GCP file
[WARNING] Could not find file /usr/local/nodeODM/data/2d97d042-0427-4fe4-9696-f10dec0d1d3c/odm_georeferencing/coords.txt
[INFO] Running ODM Load Dataset Cell - Finished
[INFO] Running ODM OpenSfM Cell
[WARNING] Found a valid OpenSfM reconstruction file in: /usr/local/nodeODM/data/2d97d042-0427-4fe4-9696-f10dec0d1d3c/opensfm/reconstruction.json
[DEBUG] running PYTHONPATH=/root/OpenDroneMap/SuperBuild/install/lib/python2.7/dist-packages /root/OpenDroneMap/SuperBuild/src/opensfm/bin/export_bundler /usr/local/nodeODM/data/2d97d042-0427-4fe4-9696-f10dec0d1d3c/opensfm
/bin/sh: 1: /root/OpenDroneMap/SuperBuild/src/opensfm/bin/export_bundler: not found
Traceback (most recent call last):
  File "/root/OpenDroneMap/run.py", line 47, in <module>
    plasm.execute(niter=1)
  File "/root/OpenDroneMap/scripts/run_opensfm.py", line 178, in process
    (context.pyopencv_path, context.opensfm_path, tree.opensfm))
  File "/root/OpenDroneMap/opendm/system.py", line 34, in run
    raise Exception("Child returned {}".format(retcode))
Exception: Child returned 127
```

- odm_extract_utm source is in /root/OpenDroneMap/modules/odm_extract_utm, it seems it was not compiled and installed.
- /root/OpenSfM/bin/export_bundler is OK, but not found in the expected location.

Will proceed tomorrow, i have appointment with mullus now.

#20 - 06/04/2019 12:18 - Philippe May

Upgraded (git pull) /root/OpenDroneMap: new error coming:

```
FATAL: password authentication failed for user "postgres"
```

Resolution (password hidden):

```
root@odml:~/WebODM/webodm# git diff settings.py
diff --git a/webodm/settings.py b/webodm/settings.py
index 69d41e0..2934842 100644
--- a/webodm/settings.py
+++ b/webodm/settings.py
@@ -124,8 +124,8 @@ DATABASES = {
     'default': {
         'ENGINE': 'django.contrib.gis.db.backends.postgis',
         'NAME': 'webodm_dev',
-        'USER': 'postgres',
-        'PASSWORD': 'postgres',
+        'USER': 'webodm',
+        'PASSWORD': '***',
         'HOST': 'db',
         'PORT': '5432',
     }
}
```

#21 - 06/04/2019 12:46 - Philippe May

Improve the installation by pulling WebODM settings out of git code base:

```
root@odml:~/WebODM/webodm# mkdir -p /etc/openDroneMap/webodm
root@odml:~/WebODM/webodm# diff settings.py /etc/openDroneMap/webodm/settings.py
21c21
< BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))
---
> BASE_DIR = '/root/WebODM'
27,41c27
< try:
<     from .secret_key import SECRET_KEY
< except ImportError:
<     # This will be executed the first time Django runs
<     # It generates a secret_key.py file that contains the SECRET_KEY
<     from django.utils.crypto import get_random_string
<
```

```

<     current_dir = os.path.abspath(os.path.dirname(__file__))
<     chars = 'abcdefghijklmnopqrstuvwxyz0123456789!@#%&^*(_=+)'
<     secret = get_random_string(50, chars)
<     with open(os.path.join(current_dir, 'secret_key.py'), 'w') as f:
<         f.write("SECRET_KEY='{}'".format(secret))
<     SECRET_KEY=secret
<
<     print("Generated secret key")
---
> SECRET_KEY = 'e*a680p5m!g!q88q@%r#cgwmqv@-naq1lw=1t!w470p%m*!gry'
127,128c113,114
<     'USER': 'postgres',
<     'PASSWORD': 'postgres',
---
>     'USER': 'webodm',
>     'PASSWORD': '****',
338c324
< except ImportError:
---
> except (ImportError, SystemError):

```

Consequently, added to /etc/systemd/system/odm-gunicorn.service:

```

Environment=PYTHONPATH=/etc/openDroneMap/webodm
Environment=DJANGO_SETTINGS_MODULE=settings

```

#22 - 06/04/2019 13:06 - Philippe May

Next error:

[Errno 13] Permission denied: '/root/WebODM/plugins/contours/public/webpack.config.js'

The git pull above created this new plugin.

Fix (another bad practice to give write permissions to the www-data user, again):

```

chgrp -R www-data /root/WebODM
chmod -R g+w /root/WebODM

```

#23 - 06/04/2019 16:09 - Philippe May

Quick note in-between, for using the virtual env for WebODM:

```
su - www-data -s /bin/bash
. /usr/local/venv/webodm/bin/activate
```

#24 - 06/04/2019 16:22 - Philippe May

Next: found in WebODM/README.md that i missed a component: celery.

To start it (in the python virtual env):

```
celery -A worker worker --autoscale $(grep -c '^processor' /proc/cpuinfo),2 --max-tasks-per-child 1000 --loglevel=warn
```

Error is: password authentication failed for user "postgres" => need to configure it to use Django's settings.

=> created the odm-celery.service, so should be enabled/started automatically.

#25 - 07/04/2019 13:16 - Philippe May

Some visible progress: now, we see "running" in the tasks (Dashboard page) so the step "Image Resize / Upload" is passed. But, nothing moves after that.

#26 - 08/04/2019 16:31 - Philippe May

Got a bit more clarity (from <https://github.com/OpenDroneMap/WebODM#architecture-overview>):

We use Celery workers to do background tasks such as resizing images and processing task results, but we use an ad-hoc scheduling mechanism to communicate with NodeODM (which processes the orthophotos, 3D models, etc.). The choice to use two separate systems for task scheduling is due to the flexibility that an ad-hoc mechanism gives us for certain operations (capture task output, persistent data and ability to restart tasks mid-way, communication via REST calls, etc.).

So, celery and node are 2 different and required "workers", celery runs first in the workflow, then only node. I wish that it would be more clear, as, eg. WebODM/worker.sh deals with celery only.

In practice, this seems to unlock the pending tasks:

```
(webodm) www-data@odm1:/root/WebODM$ ./worker.sh start
```

I don't understand why the main WebODM start script launches ./worker.sh scheduler start.

So, it's just a matter starting correctly that celery worker from systemd to get this right.

#27 - 13/04/2019 13:13 - Philippe May

- Tracker changed from Bug to Support

Finally, i'm giving a try with docker:

- installed docker in debian, with docker apt repo
- created user phil with sudo group

Run:

```
cd /usr/local/WebODM ; docker-compose up
```

```
phil@odm1:/usr/local$ docker run -p 3000:3000 opendronemap/node-opendronemap
```

- Open web interface at port 8000 => OK
- Register node with hostname: odm1.csr.av:3000
- Create task with few images: opensfm seems to be OK.

4GB RAM wasn't enough (cache hit): gave 16GB, rebooted the VM. Processing, will see after a while.

#28 - 30/04/2019 13:06 - Philippe May

I still don't really understand how to manage the docker images, and specifically to have them started at boot time. Currently, it seems that the node is started, but the webapp isn't. So, after start of the VM, one needs to:

```
cd /usr/local/WebODM ; docker-compose up
```

#29 - 09/05/2019 15:57 - Philippe May

For the second time, the disk of the VM was filled with docker "overlay" (/var/lib/docker/overlay2: 42GB). Even no space left at all on device for the docker to start.

The first time, i have run "docker system prune", which freed all that space but got to recreate the docker image, with possible data loss (?).

Doesn't work now:

```
root@odm1:/usr/local/WebODM# docker system prune -f
Cannot connect to the Docker daemon at unix:///var/run/docker.sock. Is the docker daemon running?
```

Of course it cannot run because that damn stupid thing filled up the whole filesystem, the machine was rebooted and it cannot start anymore.

Enough for today.