ckan 2.0 Installation Notes

1.0

Sol Lee

2016 09 14
## Contents

1 CKAN .................................................. 3
  1.1 .................................................. 3
  1.2 2. Virtual environment ............................................. 3
  1.3 3. CKAN .................................................. 3
  1.4 4. .................................................. 3
  1.5 5. CKAN .................................................. 4
  1.6 6. solr .................................................. 4
  1.7 7. .................................................. 5
  1.8 8. who.ini link .............................................. 6
  1.9 9. CKAN .................................................. 6
  1.10 10. .................................................. 6

2 CKAN .................................................. 7
  2.1 1. production.ini .............................................. 7
  2.2 2. production.ini .............................................. 7
  2.3 3. uwsgi .................................................. 7
  2.4 4. .................................................. 8
  2.5 5. nginx .................................................. 8
  2.6 6. nginx .................................................. 8
  2.7 7. .................................................. 9

3 DataStore DataPusher .......................................... 11
  3.1 .................................................. 11
  3.2 .................................................. 11
  3.3 DataStore .................................................. 11
  3.4 DataPusher .................................................. 12
  3.5 DataPusher .................................................. 13

4 ckanext-spatial .......................................... 15
  4.1 .................................................. 15
  4.2 .................................................. 17
  4.3 .................................................. 17

5 ckanext-geoview .......................................... 19
  5.1 .................................................. 19
  5.2 .................................................. 19

6 ckanext-harvest .......................................... 21
  6.1 .................................................. 21
6.2 ................................................................. 24
6.3 ................................................................. 24
7  Linked Data and RDF ........................................ 25
   7.1 ................................................................. 25
   7.2  Schema Mapping .......................................... 25
8  Indices and tables ............................................ 27
CKAN Data Portal Platform CKAN Association
data repository visualization search tag revisions share organization... plugins
CKAN data.gov.uk
CKAN Pylons template jinja2 Babel PostgreSQL ORM Pylons SQLAlchemy Apache Solr
CKAN2 Ubuntu 14.04 LTS Install from Source
1.1 1. CKAN

$ sudo apt-get install python-dev postgresql libpq-dev python-pip python-virtualenv git-core

1.2 2. Virtual environment

1. (virtualenv) CKAN

$ sudo mkdir -p /usr/lib/ckan/default
$ sudo chown "whoami" /usr/lib/ckan/default
$ virtualenv --no-site-packages /usr/lib/ckan/default

2. $ . /usr/lib/ckan/default/bin/activate

: deactivate . /usr/lib/ckan/default/bin/activate

1.3 3. CKAN

github checkout source ( 2.4.1

(pyenv) $ pip install -e 'git+https://github.com/okfn/ckan.git@ckan-2.4.1#egg=ckan'

: (pyenv)

Python

(pyenv) $ pip install -r /usr/lib/ckan/default/src/ckan/requirements.txt

1.4 4.

1. CKAN PostgreSQL
$ sudo -u postgres createuser -S -D -R -P ckan_default

$ sudo -u postgres createdb -O ckan_default ckan_default -E utf-8

1.5 5. CKAN

1. CKAN

  $ sudo mkdir -p /etc/ckan/default
  $ sudo chown -R `whoami` /etc/ckan/

2. paster

  : paster

  (pyenv) $ paster make-config ckan /etc/ckan/default/development.ini

3. development.ini db 4. db

   sqlalchemy.url = postgresql://ckan_default:pass@localhost/ckan_default

   : ckan_default pass db ckan_default db

1.6 6. solr

   : How To Install Solr 5.2.1 on Ubuntu 14.04 (DigitalOcean)

1. solr:

   : CKAN schema  Solr 5.2.1  Solr 5.1.0

   $ cd ~
   $ wget http://archive.apache.org/dist/lucene/solr/5.1.0/solr-5.1.0.tgz
   $ tar xzf solr-5.1.0.tgz solr-5.1.0/bin/install_solr_service.sh --strip-components=2

2. solr:

   $ sudo bash ./install_solr_service.sh solr-5.1.0.tgz

3. CKAN  solr configset:

   $ sudo -u solr mkdir -p /var/solr/data/configsets/ckan/conf
   $ sudo ln -s /usr/lib/ckan/default/src/ckan/ckan/config/solr/schema.xml /var/solr/data/configsets/ckan/conf/schema.xml
   $ sudo -u solr cp /opt/solr/server/solr/configsets/basic_configs/conf/solrconfig.xml /var/solr/data/configsets/ckan/conf/solrconfig.xml
   $ sudo -u solr touch /var/solr/data/configsets/ckan/conf/protwords.txt
   $ sudo -u solr touch /var/solr/data/configsets/ckan/conf/synonyms.txt
4. mmsg4j jar solr (/opt/solr/server/solr-webapp/webapp/WEB-INF/lib)

5. CKAN ()

    schema.xml fieldType name="text"

    <fieldType name="text" class="solr.TextField" positionIncrementGap="100">
        <analyzer type="index">
            <tokenizer class="com.chenlb.mmseg4j.solr.MMSegTokenizerFactory" mode="max-word"/>
            <filter class="solr.SynonymFilterFactory" synonyms="synonyms.txt" ignoreCase="true" expand="true"/>
            <filter class="solr.WordDelimiterFilterFactory" generateWordParts="1" generateNumberParts="1" catenateWords="0" catenateNumbers="0" catenateAll="0" splitOnCaseChange="1"/>
            <filter class="solr.SnowballPorterFilterFactory" language="English" protected="protwords.txt"/>
            <filter class="solr.LowerCaseFilterFactory"/>
            <filter class="solr.ASCIIFoldingFilterFactory"/>
        </analyzer>
        <analyzer type="query">
            <tokenizer class="com.chenlb.mmseg4j.solr.MMSegTokenizerFactory" mode="max-word"/>
            <filter class="solr.SynonymFilterFactory" synonyms="synonyms.txt" ignoreCase="true" expand="true"/>
            <filter class="solr.WordDelimiterFilterFactory" generateWordParts="1" generateNumberParts="1" catenateWords="0" catenateNumbers="0" catenateAll="0" splitOnCaseChange="1"/>
            <filter class="solr.SnowballPorterFilterFactory" language="English" protected="protwords.txt"/>
            <filter class="solr.LowerCaseFilterFactory"/>
            <filter class="solr.ASCIIFoldingFilterFactory"/>
        </analyzer>
    </fieldType>

: schema.xml /usr/lib/ckan/default/src/ckan/ckan/config/solr/schema.xml

6. solr:

    $ sudo service solr restart

7. CKAN solr core (ckan):

    http://127.0.0.1:8983/solr/admin/cores?action=CREATE&name=ckan&configSet=ckan


9. /etc/ckan/default/development.ini solr

    solr_url = http://127.0.0.1:8983/solr/ckan

: “ckan” solr core

1.7 7.

1. paster CKAN

    (pyenv) $ paster db init -c /etc/ckan/default/development.ini

2. Initialising DB: SUCCESS
1.8 8. **who.ini link**

```
$ ln -s /usr/lib/ckan/default/src/ckan/who.ini /etc/ckan/default/who.ini
```

1.9 9. **CKAN**

```
paster CKAN
```

```
(pyenv) $ paster sysadmin add admin -c /etc/ckan/default/development.ini
```

: admin

1.10 10.

1. paster serve CKAN

```
(pyenv) $ paster serve /etc/ckan/default/development.ini
```

2. **http://127.0.0.1:5000/** CKAN
How To Set Up uWSGI and Nginx to Serve Python Apps on Ubuntu 14.04 (DigitalOcean)

2.1 1. production.ini

$ cp /etc/ckan/default/development.ini /etc/ckan/default/production.ini

2.2 2. production.ini

production.ini [app:main]

[app:main]
ckan.site_url = http://site.domain

[uwsgi]
socket = /tmp/ckan_socket.sock
master = true
processes = 1
chmod-socket = 664
vacuum = true
die-on-term = true
logto = ()

2.3 3. uwsgi

uwsgi

$ . /usr/lib/ckan/default/bin/activate
(pyenv) $ pip install uwsgi
2.4 4.  

1. Upstart

$ sudo vi /etc/init/ckan.conf

2. vi

description "uWSGI instance to serve CKAN"
start on runlevel [2345]
stop on runlevel [!2345]
setuid ( /usr/lib/ckan/default )
setgid www-data

script
cd /etc/ckan/default
  . /usr/lib/ckan/default/bin/activate
  uwsgi --ini-paste /etc/ckan/default/production.ini
end script

3. $ sudo start ckan

4. $ ps aux | grep ckan

   demo 12575  0.0  0.5 249060  85144 ?  S   Sep15 0:41 uwsgi --ini-paste /etc/ckan/default/production.ini

5. $ sudo stop ckan

2.5 5. nginx

$ sudo apt-get install nginx

2.6 6. nginx

1. /etc/nginx/sites-available/ckan

   proxy_cache_path /tmp/nginx_cache levels=1:2 keys_zone=cache:30m max_size=250m;

   server {
     listen 80;
     server_name server_domain_or_IP;
     client_max_body_size 1000M;
     access_log /var/log/nginx/ckan_access.log;
     error_log /var/log/nginx/ckan_error.log error;

     location / {
       include uwsgi_params;
       uwsgi_pass unix://tmp/ckan_socket.sock;
       uwsgi_param SCRIPT_NAME ";
     }
   }
2. alias sites-enabled
   
   $ sudo ln -s /etc/nginx/sites-available/ckan /etc/nginx/sites-enabled/ckan

3. nginx
   
   $ sudo service nginx restart

2.7 7.

http://127.0.0.1/
3.1

- DataStore
  - DataStore API RESTful API JSON
- DataPusher
  - DataStore
  - DataStore DataStore

3.2

- CKAN (>=2.1)
- PostgreSQL (>=9.0)

3.3 DataStore

1. DataStore
   CKAN /etc/ckan/default/ ckan.plugins
can.plugins = datastore

2. DataStore PostgreSQL
3. DataStore

$ sudo -u postgres createuser -S -D -R -P -l datastore_default

4. DataStore

$ sudo -u postgres createdb -O ckan_default datastore_default -E utf-8

<table>
<thead>
<tr>
<th>CKAN db db</th>
</tr>
</thead>
<tbody>
<tr>
<td>ckan.datastore.write_url = postgresql://ckan_default:pass@localhost/datastore_default</td>
</tr>
<tr>
<td>ckan.datastore.read_url = postgresql://datastore_default:pass@localhost/datastore_default</td>
</tr>
</tbody>
</table>

: write_url ckan_default CKAN pass db datastore_default db read_url

5. DataStore

(pyenv) $ paster --plugin=ckan datastore set-permissions -c /etc/ckan/default/development.ini

6. CKAN

7. DataStore

$ curl -X GET "http://127.0.0.1/api/3/action/datastore_search?resource_id=_table_metadata"

## 3.4 DataPusher

1. $ sudo apt-get install python-dev python-virtualenv build-essential libxslt1-dev libxml2-dev git

2. DataPusher

   $ sudo mkdir -p /usr/lib/ckan/datapusher
   $ sudo chown `whoami` /usr/lib/ckan/datapusher
   $ virtualenv --no-site-packages /usr/lib/ckan/datapusher

3. $ . /usr/lib/ckan/datapusher/bin/activate

4. github checkout source

   $ cd /usr/lib/ckan/datapusher/src
   $ git clone https://github.com/ckan/datapusher.git
   (pyenv) $ pip install -e .

5. Python

   (pyenv) $ pip install -r requirements.txt

6. DataPusher

   (pyenv) $ JOB_CONFIG="/usr/lib/ckan/datapusher/src/datapusher/deployment/datapusher_settings.py"

7. DataPusher http://127.0.0.1:8800

8. DataPusher CKAN

   CKAN /etc/ckan/default/ ckan.plugins
ckan.plugins = datapusher

9. CKAN

### 3.5 DataPusher

DataPusher Production CKAN nginx + uWSGI

---

: How To Set Up uWSGI and Nginx to Serve Python Apps on Ubuntu 14.04 (DigitalOcean) Serving Flask With Nginx (Vladik Khononov)

1. **uwsgi**

   (pyenv) $ pip install uwsgi

2. **wsgi.py**

   uwsgi wsgi.py

   /usr/lib/ckan/datapusher/src/datapusher/wsgi.py

   ```python
   import ckan.serviceprovider.web as web
   import datapusher.jobs as jobs
   import os

   # check whether jobs have been imported properly
   assert (jobs.push_to_dataset)

   os.environ['JOB_CONFIG'] = '/usr/lib/ckan/datapusher/src/datapusher/deployment/datapusher_settings.py'

   web.init()
   web.app.run(web.app.config.get('HOST'), web.app.config.get('PORT'))
   ```

3. **uwsgi**

   /etc/ckan/default/datapusher.ini

   ```ini
   [uwsgi]
   wsgi-file = /usr/lib/ckan/datapusher/src/datapusher/wsgi.py
   socket = /tmp/datapusher.sock
   master = true
   processes = 1
   chmod-socket = 664
   vacuum = true
   die-on-term = true
   logto = /etc/ckan/default/log/datapusher.log
   ```

4. **Upstart**

   $ sudo vi /etc/init/datapusher.conf

5. **vi**

   ```
   description "uWSGI instance to serve DataPusher"
   start on runlevel [2345]
   ```

---

**3.5. DataPusher**
stop on runlevel \([-2345]\)
setuid ( /usr/lib/ckan/datapusher )
setgid www-data

script
cd /etc/ckan/default
  . /usr/lib/ckan/datapusher/bin/activate
  uwsgi --ini /etc/ckan/default/datapusher.ini
end script

6. DataPusher

$ sudo start datapusher

7. DataPusher

$ ps aux | grep datapusher

demo 1009  0.0  0.2  266332  37512 ?  S1  Sep14  2:49 uwsgi --ini /etc/ckan/default/datapusher.ini

: sudo stop datapusher  DataPusher kill

8. CKAN /etc/ckan/default/ ckan.datapusher.url

ckan.datapusher.url = http://0.0.0.0:8800/

9. CKAN
4.1

4.1.1 Spatial Metadata

4.1.2 Spatial Search Widget

“spatial” “Filter by location”

4.1.3 Dataset Extent Map

“spatial” (geojson) spatial geojson Dataset extent
4.1.4 CSW Server

CSW

4.1.5 Spatial Harvesters

harvesters CSW, WAF, spatial metadata document CKAN

harvester ckanext-harvest harvester interface

ckanext-harvest ckan.plugins csw_harvester, doc_harvester waf_harvester

:  
  • ckanext-spatial plugins “CKAN, CSW Server, Web Accessible Folder (WAF), Single spatial metadata document” source
  • ckanext-spatial havester (0.2) 11,400 3 source source harvester python library python exception
  • CSW WAF ISO 19139

CKAN harvester ckanext-harvest

• TGOSCSW source harvesting
  TGOS CSW spatial harvester CKAN list out of range exception

spatial harvester OWSLib python library csw source xml tag gmd:identificationInfo tag MD_DataIdentification TGOS MD_DataIdentification python

TGOS ISO19139 ISO tag
4.2

- CKAN (>=1.8)
- solr (>=3.1) PostGIS (>=1.3)

Spatial Harvesters ckanext-harvest:

- CKAN PostGIS
- Dataset Extent Map Spatial Search Widget snippets CKAN>=2.0

4.3

1. PostGIS
   PostgreSQL
2. ckanext-spatial

   (pyenv) $ pip install -e "git+https://github.com/ckan/ckanext-spatial.git@egg=ckanext-spatial"

3. ckanext-spatial
   ckanext-spatial solr 5.x
   /usr/lib/ckan/default/src/ckanext-spatial/ckanext/spatial/plugin.py
   search_params['fq_list'].append('spatial_geom:"Intersects({minx} {miny} {maxx} {maxy})"'.format(minx=bbox['minx'],miny=bbox['miny'],maxx=bbox['maxx'],maxy=bbox['maxy']))
   search_params['fq_list'].append('spatial_geom:"Intersects(ENVELOPE({minx}, {maxx}, {maxy}, {miny})"'.format(minx=bbox['minx'],miny=bbox['miny'],maxx=bbox['maxx'],maxy=bbox['maxy']))

: CKAN

4. Python

   (pyenv) $ pip install -r pip-requirements.txt
5. **JTS 1.13 jar solr**

   ```
   ```

6. **CKAN**

   **CKAN /etc/ckan/default/**

   ```
   ckanext.spatial.search_backend = solr-spatial-field
   ```

   **ckan.plugins**

7. **solr schema**

   **solr schema /usr/lib/ckan/default/src/ckan/ckan/config/solr/schema.xml**

   ```
   <fields>
   <!-- ... -->
   <field name="spatial_geom" type="location_rpt" indexed="true" stored="true" multiValued="true"/>
   </fields>
   ```

8. **solr**

   ```
   $ (pyenv) paster --plugin=ckan search-index rebuild -c /etc/ckan/default/development.ini
   ```

9. **Spatial Search Widget**

   **CKAN source ./ckan/templates/package/search.html {% block secondary_content %}**

   ```
   {% snippet "spatial/snippets/spatial_query.html" %}
   ```

10. **Dataset Extent Map (widget)**

    **CKAN source ./ckan/templates/package/read.html {% block secondary_content %}**

    ```
    {{ super() }}
    {% set dataset_extent = h.get_pkg_dict_extra(c.pkg_dict, 'spatial', '') %}
    {% if dataset_extent %}
    {% snippet "spatial/snippets/dataset_map_sidebar.html", extent=dataset_extent %}
    {% endif %}
    {% endif %}
    ```
5.1

5.1.1 OpenLayers Viewer

WMS, WFS, GeoJSON, KML, Google Fusion Tables /
ckan.plugins.resource_proxy.geo_view (2.2 geo_preview)
CKAN ckanext.geoview.ol_viewer.formats

5.1.2 Leaflet GeoJSON Viewer

GeoJSON, geojson, gjson
ckan.plugins.resource_proxy.geojson_view (2.2 geojson_preview)

5.1.3 Leaflet WMTS Viewer

WMTS, wmts
ckan.plugins.resource_proxy.wmts_view (2.2 wmts_preview)

5.2

github checkout source

$ cd /usr/lib/ckan/default/src
$ git clone https://github.com/ckan/ckanext-geoview.git
$ cd ckanext-geoview
(ipyenv) $ pip install -e .
ckanext-harvest

ckanext-harvest  CKAN extension interface metadata CKAN
harvest  harvesting interface:

• gather: harvest source id,
• fetch: source object metadata
• import: metadata CKAN package

6.1

6.1.1 harvest source

SITE_URL/harvest “Add Harvest source” source source

6.1.2 harvest

1. virtualenv gather fetch handler

(pyen) $ paster --plugin=ckanext-harvest harvester gather_consumer -c /etc/ckan/default/production.ini
(pyen) $ paster --plugin=ckanext-harvest harvester fetch_consumer -c /etc/ckan/default/production.ini

: handler

2. SITE_URL/harvest harvest source Reharvest harvest

3. virtualenv run handler

(pyen) $ paster --plugin=ckanext-harvest harvester run -c /etc/ckan/default/production.ini

: harvest paster harvester run harvest harvest run d.
6.1.3 harvest

production harvesting Supervisor cron

- **Supervisor**: harvest gather_consumer fetch_consumer
- **cron**: unix/linux harvest run

1. Supervisor

   ```
   $ sudo apt-get install supervisor
   
   Supervisor
   
   $ ps aux | grep supervisord
   
   Supervisor
   
   root 9224 0.0 0.3 56420 12204 ? Ss 15:52 0:00 /usr/bin/python /usr/bin/supervisord
   
   2. Supervisor /etc/supervisor/conf.d ckan_harvesting.conf

   ; ===============================
   ; ckan harvester
   ; ===============================
   
   [program:ckan_gather_consumer]
   
   command=/usr/lib/ckan/default/bin/paster --plugin=ckanext-harvest harvester gather_consumer -c /etc/ckan/default/production.ini
   
   ; user that owns virtual environment.
   
   user=okfn
   
   numprocs=1
   stdout_logfile=/var/log/ckan/default/gather_consumer.log
   stderr_logfile=/var/log/ckan/default/gather_consumer.log
   autostart=true
   autorestart=true
   startsecs=10
   
   [program:ckan_fetch_consumer]
   
   command=/usr/lib/ckan/default/bin/paster --plugin=ckanext-harvest harvester fetch_consumer -c /etc/ckan/default/production.ini
   
   ; user that owns virtual environment.
   
   user=okfn
   
   numprocs=1
   stdout_logfile=/var/log/ckan/default/fetch_consumer.log
   stderr_logfile=/var/log/ckan/default/fetch_consumer.log
   autostart=true
   autorestart=true
   startsecs=10
   
   user=okfn python virtual environment /var/log/ckan/default virtualenv
   
   3. Supervisor

   $ sudo supervisorctl reread
   $ sudo supervisorctl add ckan_gather_consumer
   $ sudo supervisorctl add ckan_fetch_consumer
   ```
$ sudo supervisorctl start ckan_gather_consumer
$ sudo supervisorctl start ckan_fetch_consumer

$ sudo supervisorctl status

Supervisor

<table>
<thead>
<tr>
<th>service</th>
<th>status</th>
<th>pid</th>
<th>uptime</th>
</tr>
</thead>
<tbody>
<tr>
<td>ckan_fetch_consumer</td>
<td>RUNNING</td>
<td>6983</td>
<td>0:22:06</td>
</tr>
<tr>
<td>ckan_gather_consumer</td>
<td>RUNNING</td>
<td>6968</td>
<td>0:22:45</td>
</tr>
</tbody>
</table>

4. run

$ sudo crontab -e -u okfn

okfn virtualenv

5. crontab

# m h dom mon dow command
*/15  *  *  *  * /usr/lib/ckan/default/bin/paster --plugin=ckanext-harvest harvester run -c /etc/ckan/default/production.ini

6.1.4 harvest

harvest source  harvest

6.1.5 harvesting interface

ckanext-harvest  interface harvester

- SRDA harvester SRDAHarvester
- CKAN csv harvester DataLondonGovUkHarvester
6.2

- RabbitMQ Redis

6.3

1. RabbitMQ Redis Redis

   $ sudo apt-get install rabbitmq-server
   $ sudo apt-get install redis-server

2. ckanext-harvest

   (pyenv) $ pip install -e git+https://github.com/okfn/ckanext-harvest.git#egg=ckanext-harvest

3. Python

   (pyenv) $ pip install -r pip-requirements.txt

4. CKAN ckan.plugins

   ckan.plugins = harvest ckan_harvester
   ...
   ckan.harvest.mq.type = redis

5. harvest

   (pyenv) $ paster --plugin=ckanext-harvest harvester initdb -c /etc/ckan/default/production.ini
Resource Description Framework RDF Schema W3C metadata
ckan 1.7 RDF

7.1 RDF

7.1.1

```
```

7.1.2

```
curl -L http://thedatahub.org/dataset/gold-prices.rdf
```

7.2 Schema Mapping

ckan RDF Schema Mapping
Indices and tables

• search