

HERRAMIENTA CKAN

SERVIDOR:172.18.20.233

DESPLIEGUE

Install and activate the CentOS Release Repository

```
yum install centos-release
```

Update and reboot your system

```
yum update
```

```
shutdown -r now
```

Install wget and policycoreutils-python, which we'll need later.

```
yum install wget policycoreutils-python
```

Install and activate the Extra Packages for Enterprise Linux (EPEL) Repository (it may already be installed)

```
rpm -Uvh http://dl.fedoraproject.org/pub/epel/7/x86_64/e/epel-release-7.5.noarch.rpm
```

Install the packages

```
yum install xml-commons git subversion mercurial postgresql-server
```

```
postgresql-devel \
```

```
postgresql python-devel libxslt libxslt-devel libxml2 libxml2-devel python-virtualenv \
```

```
gcc gcc-c++ make java-1.6.0-openjdk-devel java-1.6.0-openjdk redis tomcat
```

```
tomcat-webapps \
```

```
tomcat-admin-webapps xalan-j2 unzip policycoreutils-python mod_wsgi httpd
```

2. Install CKAN

First, create a CKAN User. The `ckan` user is created with a shell of `/sbin/nologin` and a home directory of `/usr/lib/ckan` to mirror what is shown in the [CKAN Deployment](#) documentation.

```
useradd -m -s /sbin/nologin -d /usr/lib/ckan -c "CKAN User" ckan
```

Open the newly created directory up for read access so that the content will eventually be able to be served out via `httpd`.

```
chmod 755 /usr/lib/ckan
```

Switch to the `ckan` user.

```
su -s /bin/bash - ckan
```

Install an isolated Python environment, called `default`, to host CKAN from.

```
virtualenv --no-site-packages default
```

Activate the newly installed Python environment.

```
. default/bin/activate
```

Check for the latest release version of CKAN: <https://github.com/ckan/ckan/blob/master/CHANGELOG.rst>

Download and install CKAN. For example, for version 2.4.1:

```
pip install --ignore-installed -e git+https://github.com/okfn/ckan.git@ckan-2.4.1#egg=ckan
```

Download and install the necessary Python modules to run CKAN into the isolated Python environment

```
pip install --ignore-installed -r default/src/ckan/pip-requirements-docs.txt
```

Return back to root user by typing `exit` or pressing `Ctrl + D`.

3. Configure PostgreSQL

Enable PostgreSQL to start on system boot

```
systemctl enable postgresql.service
```

Initialize the PostgreSQL database

```
service postgresql initdb
```

Edit `/var/lib/pgsql/data/pg_hba.conf` so it will accept passwords for login while still allowing the local postgres user to manage via ident login. The relevant changes to `pg_hba.conf` are as follows:

```
local all      postgres      ident
local all      all            md5
# IPv4 local connections:
host all      all      127.0.0.1/32      md5
# IPv6 local connections:
host all      all      ::1/128         md5
```

Start PostgreSQL

```
systemctl start postgresql.service
```

Switch to postgres user

```
su - postgres
```

List existing databases:

```
psql -l
```

Check that the encoding of databases is UTF8, if not internationalisation may be a problem. Since changing the encoding of PostgreSQL may mean deleting existing databases, it

is suggested that this is fixed before continuing with the CKAN install.

Next you'll need to create a database user if one doesn't already exist. Create a new PostgreSQL database user called `ckan_default`, and enter a password for the user when prompted. You'll need this password later

```
createuser -S -D -R -P ckan_default
```

Create a new PostgreSQL database, called `ckan_default`, owned by the database user you just created.

```
createdb -O ckan_default ckan_default -E utf-8
```

Exit the `postgres` user environment with `Ctrl + D` or `exit`

4. Create a CKAN Configuration

Switch back to root user and create a directory to contain the site's config files:

```
mkdir -p /etc/ckan/default
```

```
chown -R ckan /etc/ckan/
```

Switch to `ckan` user and create a CKAN config file:

```
su -s /bin/bash - ckan
```

```
. default/bin/activate
```

```
cd /usr/lib/ckan/default/src/ckan
```

```
paster make-config ckan /etc/ckan/default/development.ini
```

Edit the `development.ini` file in a text editor, changing the following options:

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

```
ckan.site_id = default
```

```
solr_url = http://127.0.0.1:8080/solr/ckan-schema-2.3
```

Exit from running as the `ckan` user with `Ctrl + D` or `exit`.

5. Setup Apache SOLR

CKAN can not use the latest version of Apache SOLR and requires version 1.4.1.

Download and extract Apache SOLR

```
curl http://archive.apache.org/dist/lucene/solr/1.4.1/apache-solr-1.4.1.tgz | tar xzf -
```

Create directories to hold multiple SOLR cores.

```
mkdir -p /usr/share/solr/core0 /usr/share/solr/core1 /var/lib/solr/data/core0 \
/var/lib/solr/data/core1 /etc/solr/core0 /etc/solr/core1
```

Copy the Apache SOLR war to the desired location.

```
cp apache-solr-1.4.1/dist/apache-solr-1.4.1.war /usr/share/solr
```

Copy the example Apache SOLR configuration to the core0 directory.

```
cp -r apache-solr-1.4.1/example/solr/conf /etc/solr/core0
```

Edit the configuration file, `/etc/solr/core0/conf/solrconfig.xml`, as follows:

```
<dataDir>${dataDir}</dataDir>
```

Copy the core0 configuration to core1.

```
cp -r /etc/solr/core0/conf /etc/solr/core1
```

Create a symbolic link between the configurations in `/etc` and `/usr`.

```
ln -s /etc/solr/core0/conf /usr/share/solr/core0/conf
```

```
ln -s /etc/solr/core1/conf /usr/share/solr/core1/conf
```

Remove the provided schema from the two configured cores and link the schema files in the CKAN source.

```
rm -f /etc/solr/core0/conf/schema.xml
```

```
ln -s /usr/lib/ckan/default/src/ckan/ckan/config/solr/schema.xml /etc/solr/
core0/conf/schema.xml
```

```
rm -f /etc/solr/core1/conf/schema.xml
```

```
ln -s /usr/lib/ckan/default/src/ckan/ckan/config/solr/schema-1.4.xml /etc/solr/
core1/conf/schema.xml
```

Create a new file, called `/etc/tomcat/Catalina/localhost/solr.xml`, with the following contents:

```
<Context docBase="/usr/share/solr/apache-solr-1.4.1.war" debug="0"
privileged="true" allowLinking="true" crossContext="true">
  <Environment name="solr/home" type="java.lang.String" value="/usr/
share/solr" override="true" />
</Context>
```

Note: Check that the directory is `/etc/tomcat` and not `/etc/tomcat6`.

Create a new file, called `/usr/share/solr/solr.xml`, with the following contents:

```
<solr persistent="true" sharedLib="lib">
  <cores adminPath="/admin/cores">
    <core name="ckan-schema-2.3" instanceDir="core0"> <property
name="dataDir" value="/var/lib/solr/data/core0" /></core>
    <core name="ckan-schema-1.4" instanceDir="core1"> <property
name="dataDir" value="/var/lib/solr/data/core1" /></core>
```

```
</cores>
</solr>
```

Set Permissions

Make tomcat the owner of the Solr directories.

```
chown -R tomcat:tomcat /usr/share/solr /var/lib/solr
```

Enable Tomcat

Configure Tomcat to start on system boot.

```
systemctl enable tomcat.service
```

Start Tomcat

```
systemctl start tomcat.service
```

If Tomcat installation was successful, you may find its web interface at:

```
http://www.yourdomain.com:8080/
```

If Apache Solr installation was successful, you may find its web interface at:

```
http://www.yourdomain.com:8080/solr
```

6. Create the Database Tables

Switch back to running as the ckan user, activate the isolated Python environment, and change to the CKAN source directory.

```
su -s /bin/bash - ckan
```

```
. default/bin/activate
```

```
cd default/src/ckan
```

Initialize the CKAN database.

```
paster db init -c /etc/ckan/default/development.ini
```

You may see a few errors but then **Initialising DB: SUCCESS**.

7. Setup the Datastore (Optional)

Follow the instructions in [Setting up the DataStore](#) to create the required databases and users, set the right permissions and set the appropriate values in your CKAN config file.

Note: You'll need to run the `paster --plugin=ckan datastore set-permissions -c /etc/ckan/default/development.ini` command as root user, since we've not set a sudo password for the ckan user.

Note: Setting up the DataStore is optional.

8. Link to who.ini

You should still be in the python virtualenv for this step, if not, do the following:

```
su -s /bin/bash - ckan
. default/bin/activate
cd default/src/ckan
```

`who.ini` (the Repoze.who configuration file) needs to be accessible in the same directory as your CKAN config file, so create a symlink to it:

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

9. Create a WSGI file

Create your site's WSGI script file `/etc/ckan/default/apache.wsgi` with the following contents:

```
import os
activate_this = os.path.join('/usr/lib/ckan/default/bin/activate_this.py')
execfile(activate_this, dict(__file__=activate_this))
```

```
from paste.deploy import loadapp
config_filepath = os.path.join(os.path.dirname(os.path.abspath(__file__)),
                              'development.ini')
from paste.script.util.logging_config import fileConfig
fileConfig(config_filepath)
application = loadapp('config:%s' % config_filepath)
```

The modwsgi Apache module will redirect requests to your web server to this WSGI script file. The script file then handles those requests by directing them on to your CKAN instance (after first configuring the Python environment for CKAN to run in).

Exit the ckan user with Ctrl + D or `exit`.

10. Create the Apache config file

Create your site's Apache config file at `/etc/httpd/conf.d/ckan_default.conf`, with the following contents:

```
WSGISocketPrefix /var/run/wsgi
```

```
<VirtualHost 0.0.0.0:80>
```

```
    ServerName default.yourdomain.com
```

```
    ServerAlias www.default.yourdomain.com
```

```
    WSGIScriptAlias / /etc/ckan/default/apache.wsgi
```

```
    # Pass authorization info on (needed for rest api).
```

```
    WSGIPassAuthorization On
```

```
    # Deploy as a daemon (avoids conflicts between CKAN instances).
```

```
    WSGIDaemonProcess ckan_default display-name=ckan_default  
    processes=2 threads=15
```

```
    WSGIProcessGroup ckan_default
```

```
    # Add this to avoid Apache show error:
```

```
    # "AH01630: client denied by server configuration: /etc/ckan/default/  
apache.wsgi"
```

```
    <Directory /etc/ckan/default>
```

```
        Options All
```

```
        AllowOverride All
```

```
        Require all granted
```

```
    </Directory>
```

```
    ErrorLog /var/log/httpd/ckan_default.error.log
```

```
    CustomLog /var/log/httpd/ckan_default.custom.log combined
```

```
</VirtualHost>
```

Replace `default.ckanhosted.com` and

www.default.ckanhosted.com with the domain name for your site.

This tells the Apache modwsgi module to redirect any requests to the web server to the WSGI script that you created above. Your WSGI script in turn directs the requests to your CKAN instance.

11. Configure Apache

Enable httpd to start on system boot

```
chkconfig httpd on
```

Start httpd
service httpd start

12. Configure iptables

Edit the file `/etc/sysconfig/iptables` by inserting the following line near the middle of the file:

```
-A INPUT -m state --state NEW -m tcp -p tcp --dport 80 -j ACCEPT
```

Restart iptables
service iptables restart

Connect to CKAN

Start your web browser
systemctl start httpd.service
and head to your domain and you should see CKAN running.
For customization, CKAN is located at `/usr/lib/ckan/default/src/ckan/ckan`