Software INSTALLATION INSTRUCTIONS

NATIONAL GEOTHERMAL DATA SYSTEM - ngds

Version 0.6

*07/24/2013*

Version History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Author** | **Date** | **Reason** |
| 0.1 | Roberto Silva Filho | 05/28/2013 | Initial Draft Created |
| 0.2 | Monica McKenna | 06/11/2013 | Minor updates |
| 0.3 | Monica McKenna | 06/24/2013 | Combining comments from a few people |
| 0.4 | Monica McKenna | 06/25/2013 | Added appendix with summary of development.ini changes |
| 0.5 | Monica McKenna | 07/22/2013 | A little re-organization, more hints, and added gdal |
| 0.6 | Monica McKenna | 07/24/2013 | Updating with feedback |
|  |  |  |  |

Table of Contents

[1 INTRODUCTION 6](#_Toc362454638)

[1.1 Purpose 6](#_Toc362454639)

[2 Prerequisites 7](#_Toc362454640)

[2.1 Choosing a virtualized environment 7](#_Toc362454641)

[2.2 Creating a VM and installing linux in a Windows environment 7](#_Toc362454642)

[2.2.1 First, download and install Oracle VM VirtualBox Manager 7](#_Toc362454643)

[2.2.2 Then, download the Desktop CD for PC(Intel x86) for Ubuntu: 7](#_Toc362454644)

[2.2.3 Start Oracle VirtualBox and Create a VM for Linux Ubuntu 8](#_Toc362454645)

[2.2.4 Configure the new VM 11](#_Toc362454646)

[2.2.5 Mount the Linux installation .ISO file in the VM before it starts 12](#_Toc362454647)

[2.2.6 Finally, install Linux Ubuntu 12.04 13](#_Toc362454648)

[3 Configuring if behind a corporate firewall (OPTIONAL) 17](#_Toc362454649)

[3.1 Install CNTLM 17](#_Toc362454650)

[3.2 Configure the Linux VM Proxy to use CNTLM as its proxy 17](#_Toc362454651)

[3.3 What to do if cntlm and proxy continue to cause issues 19](#_Toc362454652)

[4 NGDS Installation Procedure 20](#_Toc362454653)

[4.1 Installing Git 20](#_Toc362454654)

[4.2 Obtaining the installation program and script 20](#_Toc362454655)

[4.3 Setting Parameters in my\_config.cfg 21](#_Toc362454656)

[4.4 Installing required programs 21](#_Toc362454657)

[4.4.1 Installing Apache Solr 21](#_Toc362454658)

[4.4.2 Installing Geoserver 22](#_Toc362454659)

[4.4.3 Installing GDAL 23](#_Toc362454660)

[4.5 Run the Installation program 23](#_Toc362454661)

[4.6 Setup Solr in the middle of the installation 24](#_Toc362454662)

[4.6.1 Run solr 24](#_Toc362454663)

[4.7 Run the installation again 25](#_Toc362454664)

[4.8 Parameters for development.ini 25](#_Toc362454665)

[4.9 Start up celery 26](#_Toc362454666)

[4.10 Starting up NGDS 26](#_Toc362454667)

[4.11 Post-Installation steps 26](#_Toc362454668)

[4.11.1 Connecting geoserver with our datastore Postgres database 27](#_Toc362454669)

[4.12 Hints for when the installation doesn’t work 28](#_Toc362454670)

[4.13 Various Other Notes 29](#_Toc362454671)

[Build a PostGIS template DB 30](#_Toc362454672)

[Manually creating databases (other than via the installation script): 30](#_Toc362454673)

[Creates users for production and testing setups for **CKAN** with the **Datastore** enabled. The little d is not a typo. 30](#_Toc362454674)

[5 Appendix A: Summary of Changes to the Development.ini file 30](#_Toc362454675)

List of Appendices

**No table of figures entries found.**

List of Figures

[Figure 1 Create a new linux virtual machine 7](#_Toc358720721)

[Figure 2 Set the VM memory to at least 1024MB 7](#_Toc358720722)

[Figure 3 Create a virtual HD. The disk size can be changed later. 8](#_Toc358720723)

[Figure 4 Specify the image type 8](#_Toc358720724)

[Figure 5 Dynamically allocated disk space will allow the image to grow as new data is imported into NGDS 9](#_Toc358720725)

[Figure 6 Configure the hard drive size. We recommend large values, based on the amount of data the node should store. 9](#_Toc358720726)

[Figure 7 Oracle VM Linux machine configuration 10](#_Toc358720727)

[Figure 8 You may wish to enable the shared clipboard 11](#_Toc358720728)

[Figure 9 Mounting ISO image in the Linux VM 12](#_Toc358720729)

[Figure 10 Linux Ubuntu installation screen 13](#_Toc358720730)

[Figure 11 Login for the new VM 14](#_Toc358720731)

[Figure 12 Installing guest additions 15](#_Toc358720732)

[Figure 13 Configuring the proxy in Ubuntu Linux 17](#_Toc358720733)

[Figure 14 Install updates before continuing 18](#_Toc358720734)

[Figure 16 Geoserver gtda workspace 24](#_Toc358720735)

List of Tables

**No table of figures entries found.**

# INTRODUCTION

NGDS or National Geothermal Data System is a government funded project. The system supports the storage and search of geothermal data from different organizations. It is a hub and spoke network of data providers through its many node-in-a-box distributions, with the hub being a centralized node that supports the harvesting and search of this distributed data.

## Purpose

This document is a step by step tutorial to guide new developers and users in setup both node-in-the-box and harvesting catalogs instances of NGDS.

# Prerequisites

Before one can install and configure the individual components utilized in the NGDS project, a virtual machine with the required OS version must be available and properly configured (a non-virtual machine dedicated for use with this project is also acceptable).

Super User (Administrator) privileges are required.

Network access is required.

The current implementation of the NGDS project is developed in Python and Linux OS. In particular, we utilize 32-bit (i386) Ubuntu distributions 12.04 or higher (specifically, Xubunto, e.g. such as xubuntu-13.04-desktop-i386.iso).

This section describes the steps necessary to create your own virtual machine, and to install NGDS as both a catalog and a node-in-a-box.

If you are installing directly on a Linux OS, skip to the section .

## Choosing a virtualized environment

The first step is the installation of Linux in a virtual machine within a host system. Virtual machines are supported by virtualization software, that provides an abstract hardware representation emulating real host hardware. Virtualization allows the installation of a full operating system within a host OS. (Note that a virtual machine is not required, however, the set up of the Linux environment must be custom for this project).

Currently, two free virtual environment managers are available: Vmware Player, and Oracle VirtualBox. They can be downloaded on the links below:

* VMWare Player: <http://www.vmware.com/products/player/>
* Oracle VM VirtualBox: <https://www.virtualbox.org/wiki/Downloads>

In particular, this tutorial was developed using VirtualBox version 4.2.10 for windows, where we install Linux Ubuntu 12.04 LTS from Canonical.

## Creating a VM and installing linux in a Windows environment

Please follow those steps in order to install Ubuntu Linux (or Xubuntu) within a virtual machine in your Windows desktop:

### First, download and install Oracle VM VirtualBox Manager

This tutorial utilized version 4.2.10 of Oracle VM VirtualBox. Newer versions can be utilized.

Download the software from <https://www.virtualbox.org/wiki/Downloads>

### Then, download the Desktop CD for PC(Intel x86) for Ubuntu:

An .ISO image for Ubuntu is available at: <http://releases.ubuntu.com/12.04/>

This tutorial utilizes the Long Term Service version, or LTS, which long term support (3 years): <http://releases.ubuntu.com/12.04/ubuntu-12.04-desktop-i386.iso>

### Start Oracle VirtualBox and Create a VM for Linux Ubuntu

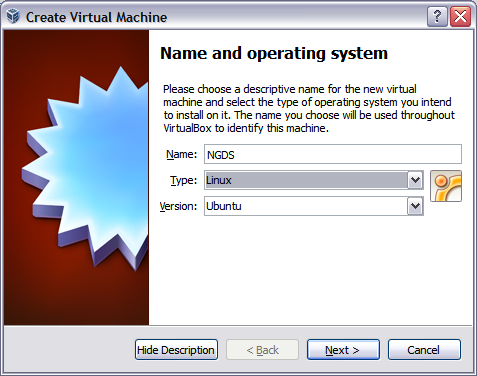


Figure : Create a new linux virtual machine

We recommend the machine to have at least 1024 MB of RAM.

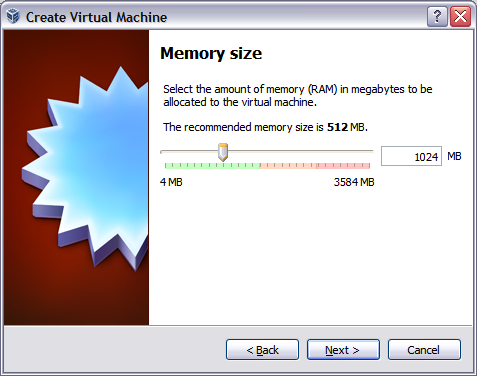


Figure : Set the VM memory to at least 1024MB

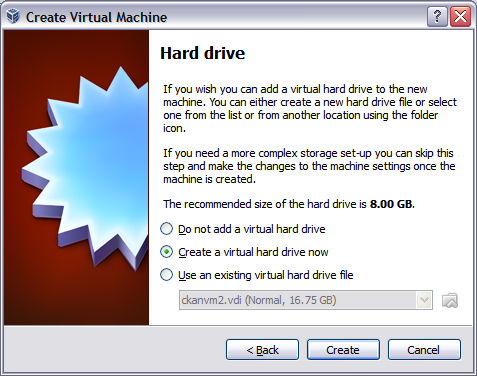


Figure : Create a virtual HD. The disk size can be changed later.

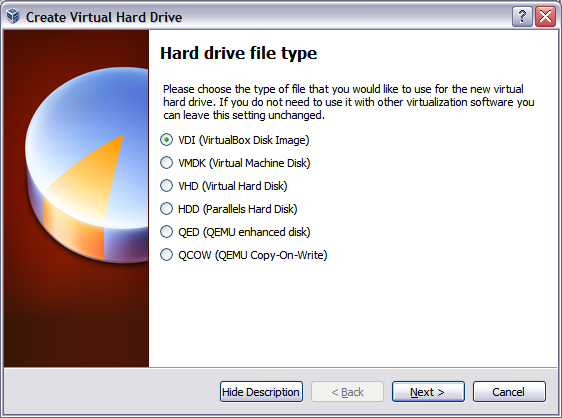


Figure : Specify the image type

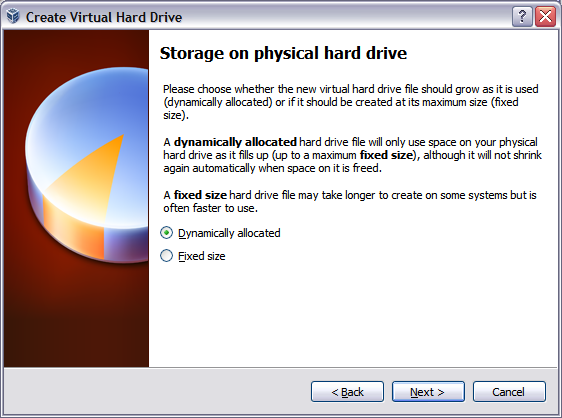


Figure 5: Dynamically allocated disk space will allow the image to grow as new data is imported into NGDS

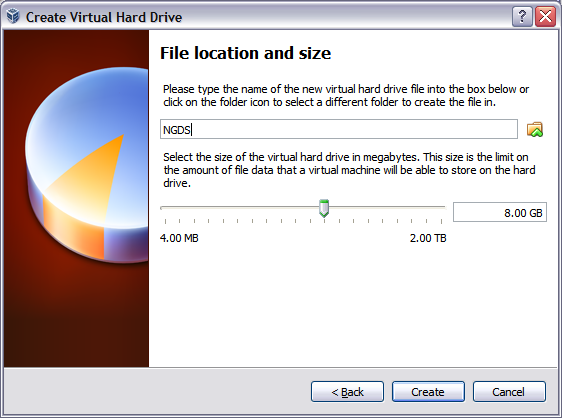


Figure : Configure the hard drive size. We recommend large values, based on the amount of data the node should store.

### 

Figure 7 Oracle VM Linux machine configuration

### Configure the new VM

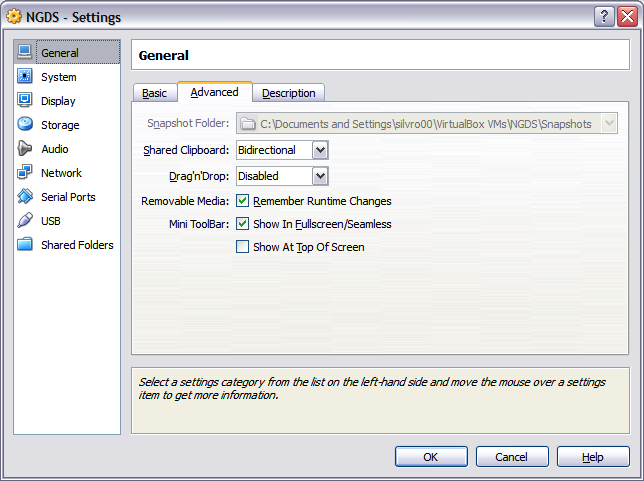


Figure You may wish to enable the shared clipboard

Now that the NGDS virtual machine is created, we have can install an existing Linux distribution.

### Mount the Linux installation .ISO file in the VM before it starts

After downloading a .ISO image as described in 2.2.2, we mount it and boot it up in our VM.

The ISO image contains the installation CD of the Linux OS to be installed in the VM created in the previous step.

Click on the recently created image, then on settings in the toolbar. A dialog box will show up. Select the storage tab on the left as shown in Figure 8. Select the .iso file by clicking on the DVD icon on the right side of CD/DVD drive drop down menu.

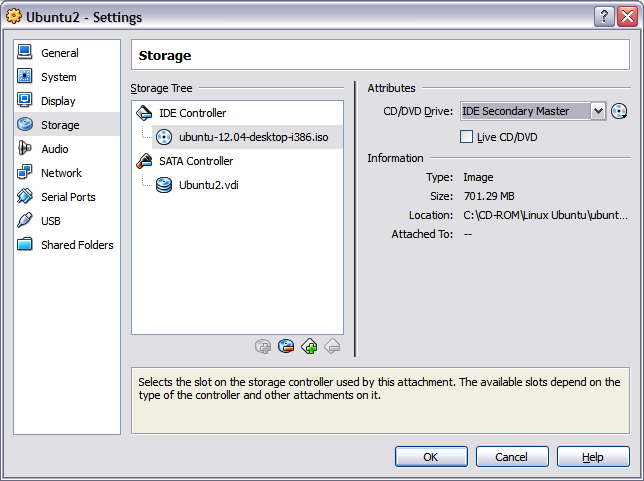


Figure Mounting ISO image in the Linux VM

Start the VM.

The first time the VM starts, the .iso image will be mounted and Linux Ubuntu installation prompt will come up.

### Finally, install Linux Ubuntu 12.04

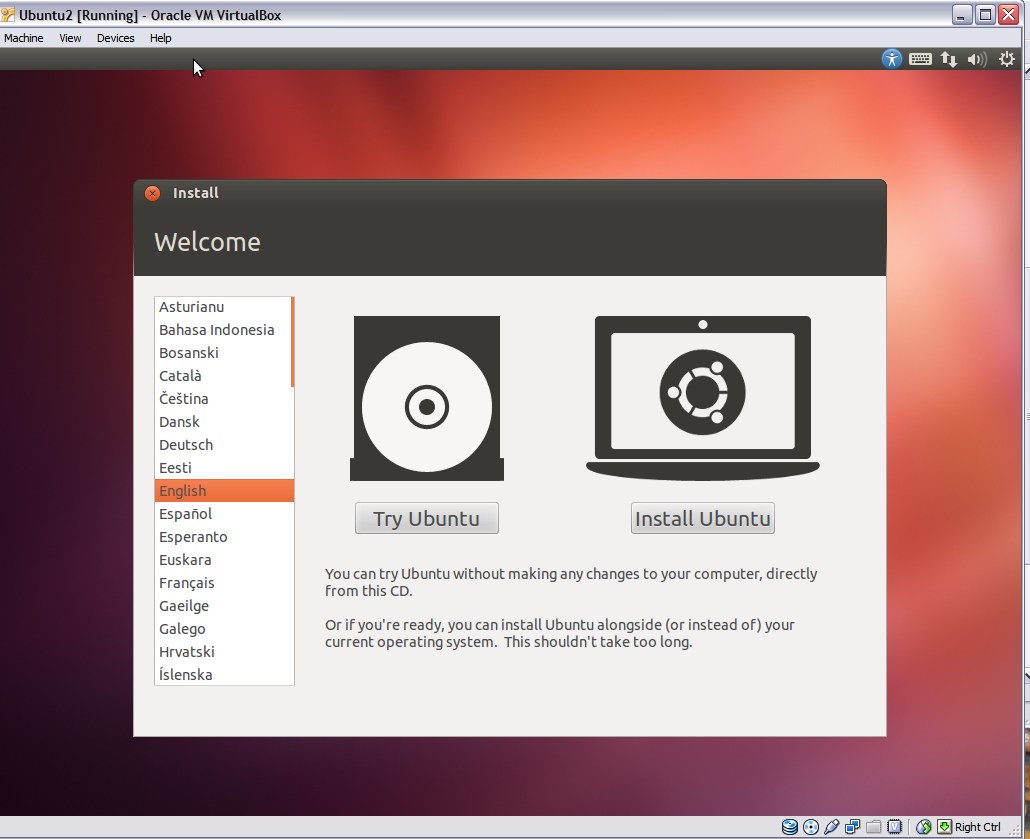


Figure Linux Ubuntu installation screen

Click on “Install Ubuntu” button and follow the installation procedure. Default installation parameters will suffice.

Create a regular user and password to use during setup and administration of the node.

After the installation is over, restart and log-in to the new VM.

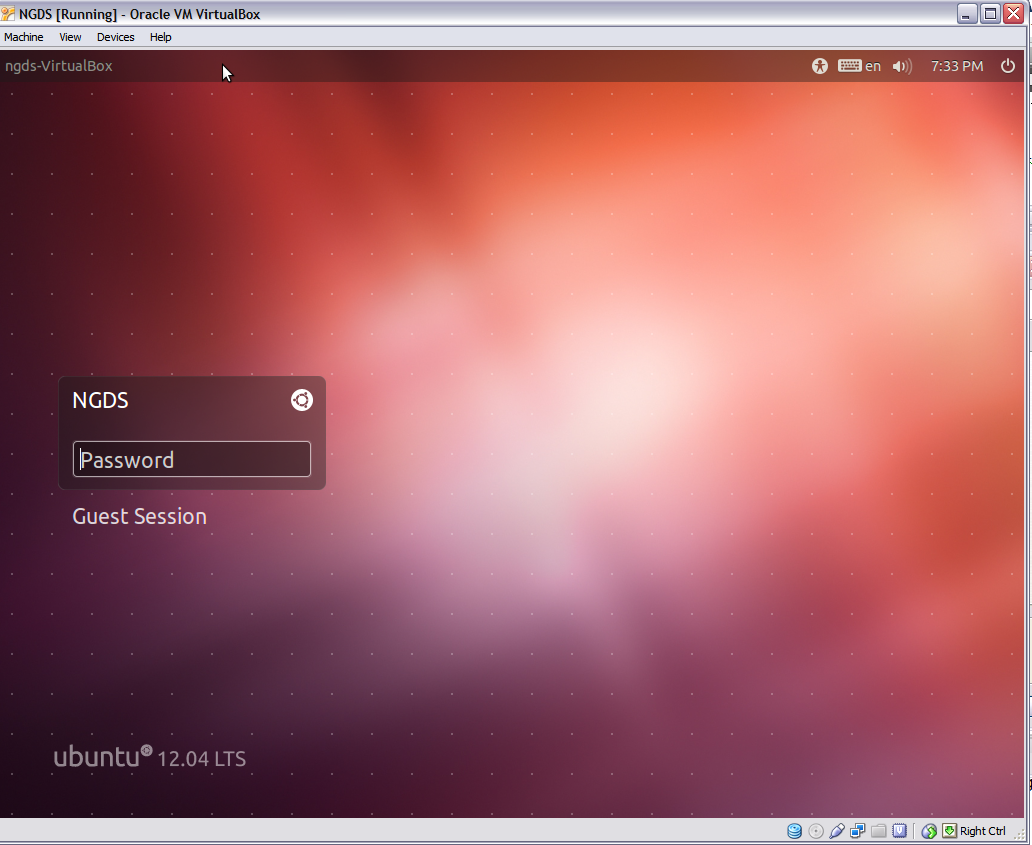


Figure Login for the new VM

In our example, we use the login: ngds and password: ngds.

We also recommend the installation of the VM guest additions.

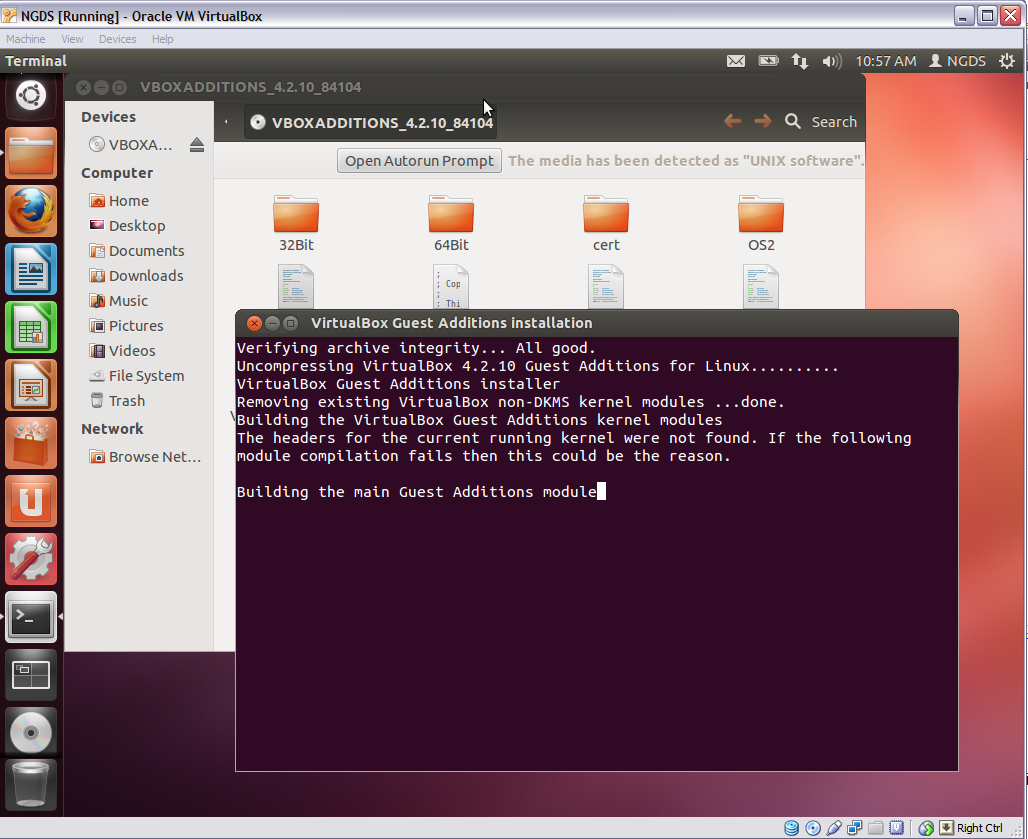


Figure Installing guest additions

# Configuring if behind a corporate firewall (OPTIONAL)

If you are behind a corporate firewall, you may not have immediate Internet access. Internet connectivity is required in order to install Python and Linux packages as will be discussed in the next section.

This section discusses how to configure the proxy within CT RTC.

## Install CNTLM

CNTLM is a proxy that authenticates the user with a log-in and password, which is a typical requirement for corporate firewalls. If you are not behind a firewall that requires authentication, you can skip this step.

You can download the windows version of CNTLM at: <http://cntlm.sourceforge.net/>

Install it in your computer, for example at C:\Program Files\Cntlm and configure the cntlm.ini file content with your log-in, password and corporate network proxy information. For example:

|  |
| --- |
| Username yourcorporateproxyusernamehere  Domain us008  Password yourpasswordhere  # List of corporate proxies  Proxy proxyfarm-us.3dns.netz.sbs.de:84  Proxy 129.73.8.72:8080  Proxy 129.73.11.208:3128  NoProxy localhost, 127.0.0.\*, 10.\*, 192.168.\*  # local port used by CMTLM  Listen 3128 |

In order to use CNTLM, the virtual machine needs to be configured to use the localhost version of this proxy. We use the default cntlm port which is 3128.

Run CNTLM whenever the VM is executing, otherwise, there will be no network connection. CNTLM can be run as either a service or directly in the command prompt. In a development environment, we recommend the use of the command prompt approach as you can manually restart CNTLM in case it freezes or crashes (which is pretty common).

## Configure the Linux VM Proxy to use CNTLM as its proxy

In the Linux VM, edit the /etc/environment file and add the following environment variables:

|  |
| --- |
| http\_proxy=http://10.0.2.2:3128/  https\_proxy=http://10.0.2.2:3128/  ftp\_proxy=http://10.0.2.2:3128/  no\_proxy="localhost,127.0.0.1,192.168.50.1,192.168.50.2"  HTTP\_PROXY=http://10.0.2.2:3128/  HTTPS\_PROXY=http://10.0.2.2:3128/  FTP\_PROXY=http://10.0.2.2:3128/  NO\_PROXY="localhost,127.0.0.1,192.168.50.1,192.168.50.2" |

Note that 10.0.2.2. is the IP address used by VirtualBox to represent the host machine, where, in our case, CNTLM should be running.

In Ubuntu Linux, this configuration may not work. Alternatively, one can configure the proxy by using the Network configuration application from Ubuntu as follows.

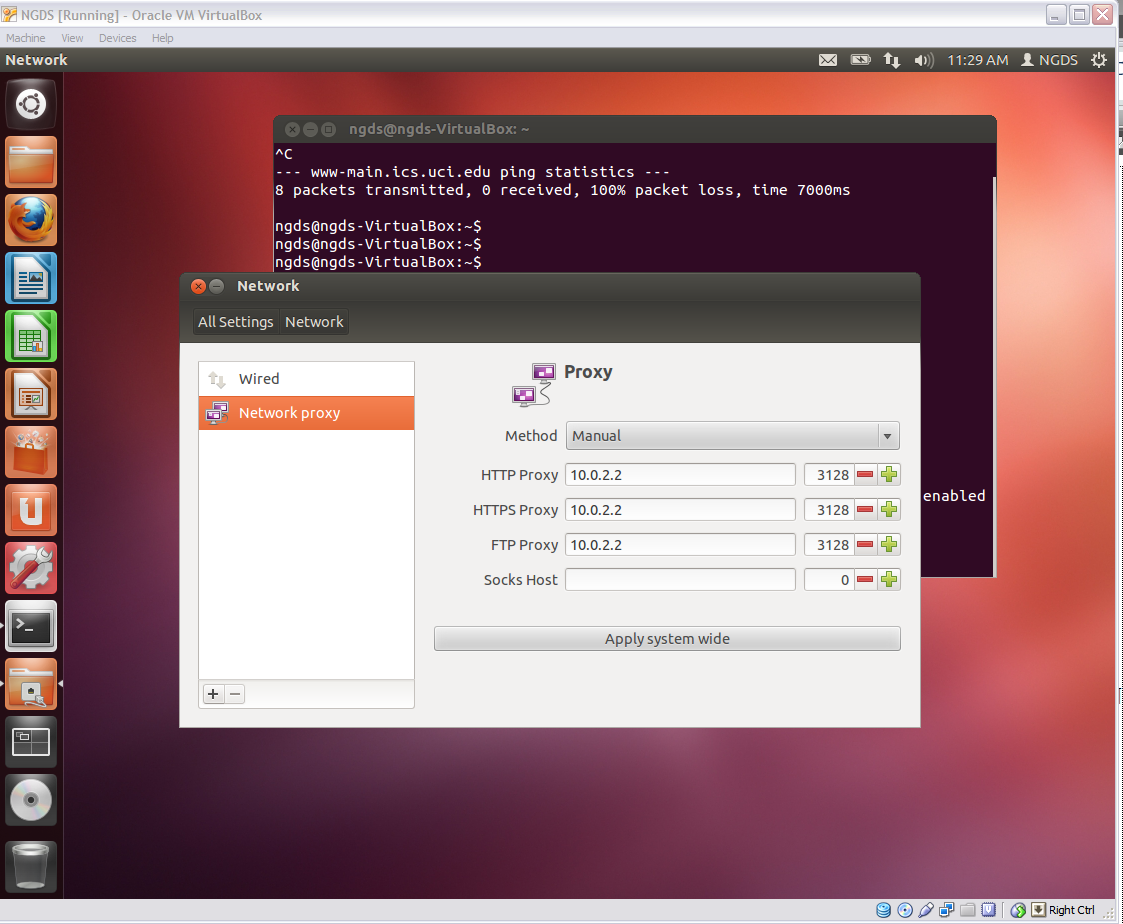


Figure : Configuring the proxy in Ubuntu Linux

Install applicable application updates before continuing as per below.

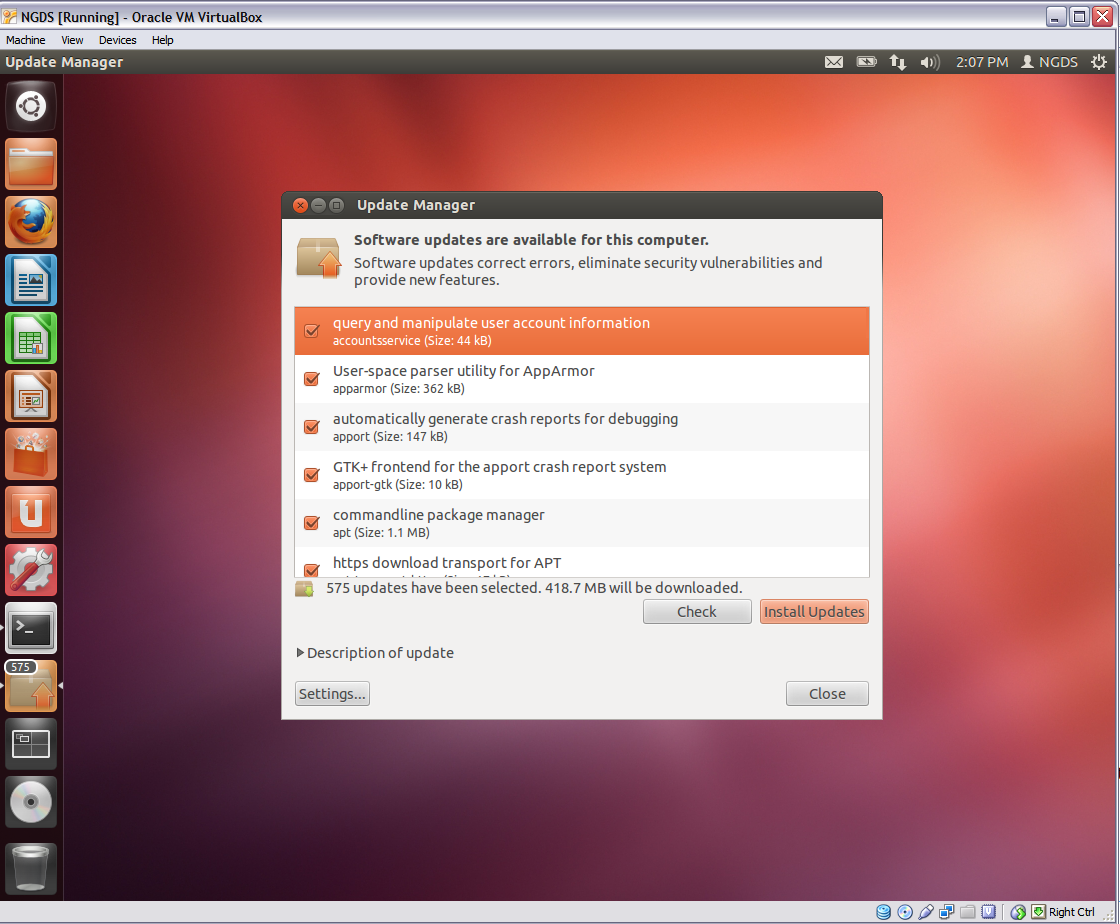


Figure : Install updates before continuing

## What to do if cntlm and proxy continue to cause issues

1. If possible finish install while on the internet instead of an intranet. If this is not possible, you will have to fight with your settings until at least apt get works. This may require installation of ctlm within your virtual machine.
2. If you are able to install the software, but then when you try to open the web sites locally hosted and your proxy software then causes issues, use port forwarding within the virtual machine to forward the ports of interest (e.g. 5000, etc) to your physical machine, and browse the web sites on your physical machine. At least at CT RTC this solves the issues with the proxy.

# NGDS Installation Procedure

After setting up a Linux physical or virtual machine to receive the installation package, different OS components must be installed. These components include

* Git
* Apache Solr
* PostgreSQL database
* PostgreSQL extensions for Geographical Information Systems (POSTGIS)
* Geoserver
* Different python extensions
* gdal

The good news is that the installation of many of these components is automated by an installation script (exceptions are Git, Apache Solr and Geoserver). The installation script is located in the NGDS project repository. The installation script is rather interactive in how it currently functions, so ***reading this whole chapter first before proceeding is recommended***.

Be aware that you will probably end up running the installation script more than one time. When running it again after running it the first time, answer N to the question if you want to overwrite the development.ini file. Also, please be aware the script has little smarts – e.g. it will keep asking you to set parameters and add plugins even if they are already there at various steps in the installation when it is run a second time.

Note that if you are behind a firewall, you must be configured to reach the download sites necessary. You may need ctlm for this, if so, go back first to the section .

Before running, execute:

|  |
| --- |
| % sudo apt-get update |

This verifies that apt-get is working and runs any needed updates.

## Installing Git

Open a terminal in Linux Ubuntu, get root access with the command ‘sudo su’ and execute the following commands to install Git in your system:

|  |
| --- |
| % sudo su  % apt-get install git  % apt-get install git-core |

## Obtaining the installation program and script

Once git is installed, one can check out the contents of the <https://github.com/ngds/dev-info.git> repository, which contains the source and installation programs for NGDS:

|  |
| --- |
| % git clone https://github.com/ngds/dev-info ./ngds-install |

The command above will create a ./ngds-install folder with a copy of the dev-info repository. Note that this requires a user account and password for this repository.

If you do not have a user account and password, please contact Ryan Clark at ryan.clark@azgs.az.gov.

## Setting Parameters in my\_config.cfg

After checking out the code from git hub, one needs to configure the installation parameters.

The first step in setting installation parameters is to modify the my\_config.cfg file within the ./ngds-install/installation folder and fill out the appropriate values. One can use pine editor for that (or pico, or whichever editor comes pre-installed with the operating system you chose). The configuration parameters are:

pyenv\_dir=/home/kaffeine/pyenv (The path where you want your pyenv virtual env. installed)

source\_dir=/home/kaffeine/pyenv/src ( The source directory for all the projects we will clone)

ckan\_port=5000

geoserver\_port=8080

ckan\_database=testdb (The database used by ckan)

datastore\_database=test\_datastore (The database used by the ckan datastore.)

ckan\_user=testuser (The username to the ckan database that ckan will use to connect to postgres)

ckan\_pass=pass (The password for the ckan\_user role)

verbose=true (As the script runs, do you want to see a bunch of output. Say true. )

pip\_proxy=10.0.2.2:3128 (Some people need to set this explicitly. It's usually 10.0.2.2:3128)

Later when you will run the installer.sh script, it will prompt you in several places to update other various items (don’t start the installer.sh script yet, just set the parameters in my\_config.cfg for now).

## Installing required programs

There is no need to install required programs as java, postgresql, python and others. The installation script will take care of those dependencies utilizing the apt-get program. Some dependencies such as Solr and Geoserver must be installed separately. Follow all the instructions below for each program.

### Installing Apache Solr

Solr is a web search appliance around Apache Lucene. Lucene is a search engine/indexer for unstructured data as text documents. Solr also provides a web interface, allowing other programs to search for content in a lucene repository using HTTP protocol.

#### Downloading Solr

First, we have to download Solr distribution from Apache web site. We need version 4.0 or later, that adds support for geo-based queries.

Download solr from: <http://lucene.apache.org/solr/>. Click on the DOWNLOAD button for version 4.0 and download the .zip file from one of the mirrors listed at: http://www.apache.org/dyn/closer.cgi/lucene/solr/

In particular, we downloaded solr-4.2.1.tgz from: http://archive.apache.org/dist/lucene/solr/4.2.1/solr-4.2.1.tgz

We install Solr at /opt/solr using the following steps.

|  |
| --- |
| % sudo su  % cd /opt  % mkdir solr  % cd solr  % wget http://archive.apache.org/dist/lucene/solr/4.2.1/solr-4.2.1.tgz  % tar xvfz solr-4.2.1.tgz |

If wget does not work for you, see the section to make sure your internet access is configured, or use your browser to navigate to the archive.apache.org web site to download the solr tgz file, then proceed with un-tarring it with the tar xvfz command.

To start solr, go to /opt/solr/solr-4.2.1/example and run:

java –jar start.jar

However, see a later step that will change the schema file first…

### Installing Geoserver

Geoserver is a web application that allows users to view and edit geospatial data using a web interface defined by the Open Geospatial Consortium (OGC). It can render geo-located data on top of existing mapping applications such as Google Maps, Google Earth, Yahoo Maps and Microsoft Virtual Earth.

Geoserver is deployed as a Web Archive (war file) within tomcat 7. Hence, in order to install geoserver, one must first download geoserver from the website, and unpack its content within /var/lib/tomcat7/webaps folder or run it as standalone.

First, download the stable distribution of geoserver from: <http://geoserver.org/display/GEOS/Download>. In particular, we downloaded the binary distribution: <http://downloads.sourceforge.net/geoserver/geoserver-2.1.4-bin.zip>.

One option is to run geoserver as a standalone application with runs it within a jetty container. To do so you may find that you have to first set JAVA\_HOME, e.g.

export JAVA\_HOME=/usr/lib/jvm/java-6-openjdk-i386

|  |
| --- |
| % cd /geoserver/install/dir  % start.sh |

Test with <http://localhost:8080/geoserver>.

Default log-in is ‘admin’, the password is ‘geoserver’

NOTE: You must then login and have your login cached by your browser (admin/geoserver).

*Alternatively,* an option is to run geoserver as a web application within tomcat, in this case, one has to copy the geoserver.war file to a tomcat installation after installing tomcat if it’s not yet installed.

$ sudo apt-get install tomcat6 unzip

$ wget http://downloads.sourceforge.net/geoserver/geoserver-2.3.3-war.zip

$ unzip geoserver-2.3.3-war.zip

$ sudo cp geoserver.war /var/lib/tomcat6/webapps/

$ rm -rf \*

**Tomcat** is now available on :8080, and **Geoserver** at http://localhost:8080/geoserver/. The default configuration uses admin:geoserver as username:password.

### Installing GDAL

Run the following commands:

sudo apt-add-repository ppa:ubuntugis/ubuntugis-unstable

sudo apt-get update

sudo apt-get install libgdal-dev

## Run the Installation program

Before running the installation program, run the following commands in the command window that you run the installer script from (this is for compiling gdal, one of the needed components):

export C\_INCLUDE\_PATH=/usr/include/gdal

export CPLUS\_INCLUDE\_PATH=/usr/include/gdal

Run the installer script. ./installer.sh ngds from the ./ngds/installation folder.

|  |
| --- |
| % ./installer.sh ngds |

When asked for passwords, the recommended password is “pass”.

Be patient – it takes quite a while to run and asks several questions along the way.

When you are asked to update the sqlalchemy URL, update the development.ini as requested, also see the appendix to make additional modifications (except for updating the plugins list).

Then before answering y to continue after setting the sqlalchemy URL, do the solr setup described in the next step.

If you run the installation script more than once (and you will probably end up doing so), answer N to the question if you want to overwrite the development.ini file, so that its content is stored. Be aware it is not context sensitive, so it will ask you to do things that you did before when you run it a second time, you can just continue the script when it does so.

Also, as you go along, making a backup of your development.ini file is recommended.

The development.ini file can be found in ~/pyenv/src/ckan/development.ini file after the installation has gone part way through.

If you want to stop the installation and do something else in that same command window before proceeding, you can use CTRL-Z to suspend the process, then do what you like, and the use “fg %1” or the job number of the suspended process to restart the installation where it left off. Alternatively you can do other tasks in separate command windows.

## Setup Solr in the middle of the installation

After starting the installation program and having it run part way through, a pyenv folder with src/ckan will be created. It will follow the name conventions provided in the my\_config.cfg file (See ).

Under example/solr/collection1/conf of your solr installation dir, you will find a schema.xml file. Replace this file with the file schema.xml from ngds/installation/schema.xml. This will allow solr to index imported ckan files. Do not skip this step!

|  |
| --- |
| % cd ~/pyenv/src/ckan/ckan/config/solr  % sudo cp ~/ngds-install/installation/schema.xml /opt/solr/solr-4.2.1/example/solr/collection1/conf/schema.xml |

### Run solr

Inside the /opt/solr/solr-4.2-1 directory, you will find an example directory which has a start.jar file. Run it via command line using java -jar start.jar. (See in previous section how to set the schema file correctly first).

Verify that it works by visiting localhost:8983/solr via a web browser.

To stop solr later, Ctrl-C this process. You can also Ctrl-Z it to background it and then kill it via a “kill” command.

Note that solr can also be installed to run as a service that starts whenever you login. If you choose to do this for convenience, see the solr documentation.

Solr MUST be running for anything else to work from this point forward.

## Run the installation again

Since we had to stop or pause the installation in the middle to configure Solr, you might need to restart the installation script again. When asked whether to overwrite some context reply with “N” since we have modified the development.ini and do not want those changes over-written.

As the installation runs, you have to keep editing it files as it tells you, as the installer will pause and ask you to make the edits (though it’s not aware of previous edits you made, so if it asks you to make the same edit again, just continue the installation).

If things appear to go wrong, start the installation over, though using the “N” above when asked about the context overwrite. Some things it does even if already installed, but most things it will bypass if already installed.

Remember that after the first part of running installation.sh, solr should be running during the rest of installation.

The installation will ask several questions, so be sure to monitor the installation and read the questions carefully.

Once the installer script is finished, check in the appendix of this file to see that all the settings are set that are listed in the appendix for the development.ini.

Note that you can’t just start out with all the plugins listed out in the development.ini, they have to be added as the installation proceeds as the install script prompts for them (until their dependencies are installed – this is why if you run it a second time, there is no need to delete plugins who have had their dependencies installed).

One question the installer is not asking at this time is to set the resources directory before it needs it, so add this line to your development.ini file. You should so this early in the process:

ngds.resources\_dir=/home/yourloc/pyenv/src/ckanext-ngds/ckanext/ngds/base/resources/

## Parameters for development.ini

The initial development.ini file is automatically generated by the installation script. The NGDS plug-ins list that is in this file needs to be modified. This list needs to be manually updated. The installation script will have prompted you to update this list of plugins and other values in the development.ini file. See the appendix for a summary of all changes to the development.ini

Important note: development.ini has various sections, where the […] statements indicate the sections. Be sure that all parameters described in this document go into the [app:main] section!

Also, note that having or not having the trailing / as on URLs can cause the system to not work, so watch these carefully.

So once installation is complete, check the appendix and compare to your development.ini.

Then possibly run the installation again if it did not seem to complete successfully.

## Start up celery

To do data import, you must start celery. To do so, activate pydev (. ~/pydev/bin/activate) and then in the ckan directory,

celeryd

This will start celery running and you should do this whenever running NGDS.

## Starting up NGDS

To start up NGDS, first make sure pydev is activated and that you then working within that activated environment (the prompt should change).

% . ~/pyenv/bin/activate

To start up, first make sure solr is running (and geoserver also if you are planning to make use of that functionality, and celery as well) then run

% paster serve development.ini –-reload

The –reload means that if you change your development.ini, the changes will be picked up with a restart.

To visit the website, use: localhost:5000/ngds

But first see post-installation steps below.

## Post-Installation steps

Create a sysadmin account. Navigate to /your\_pyenv/src/ckan and run -

(pyenv)$ paster sysadmin add admin

The command will prompt you to create an account called admin if one does not exist.

Suggested password is “pass”.

Temporarily remove ngdsui from the plugins list in the development.ini file (this starts ckan in “default” mode instead of in ngds mode).

Restart the server (better yet “paster serve development.ini –reload” and it will restart automatically whenever the development.ini file is changed)

Browse to http://localhost:5000

Login to the ckan ui with the admin account and create an organization called “public”.

Add ngdsui back in to the list of plugins in the development.init file . Restart the server if not using the –reload option.

Then to the development.ini, add the following settings if you have not yet added them, and again restart if necessary with yet “paster serve development.ini –reload”.

Make sure any directories referenced below exist (create them if needed). Be very careful to have or not have the trailing “/” on path names as in the text below. See also the appendix for all development.ini changes.

ckan.storage.bucket = subdir

#(Subdirectory under ofs.storage\_dir. Can be anything)

ofs.impl = pairtree

#(Do not modify this)

ofs.storage\_dir = /home/my\_username/wherever/i/want/

ngds.deployment=node

#(If you want a node - node, if central – central for type of web site)

ngds.default\_group\_name=public

#(Do not modify)

ngds.resources\_dir = /home/my\_username/my/path/to/ckanext-ngds/ckanext/ngds/base/resources/

ngds.logo\_text=CONTRIBUTING GEOTHERMAL DATA

#(Do not modify)

ngds.bulk\_upload\_dir=/home/my\_username/some/directory/i/created

ckan.site\_logo = /images/NODE\_BOX.png # this is a modification of an existing line

ngds.facets\_config=/home/my\_username/pyenv/src/ckanext-ngds/facet-config.json

See also the appendix to review that all needed parameters are set in your development.ini file.

### Connecting geoserver with our datastore Postgres database

Note that the ngds workspace should be automatically added to geoserver by the installation, and the following should not be needed. But should something have gone wrong, the following details are included so that you can do this step manually if needed.

Geoserver externalize database information in the form of layers. Layers are tables in a database where each row has at least one geo-located shape. A shape is either a point, polygon and any other standard GML type. GML types are supported by the GIS extensions to the postgres database that we previously installed.

Geoserver provides a web services interface for these layers. Layers can be connected to database tables stored in postgres or to queries performed in that database.

Note that shapes are automatically produced, based on human readable longitude and latitude information, at import time, using stored procedures(or functions) implemented in the gtda database.

We can, for example, export the core schema entries from the datastore database as a layer in geoserver.

First, we open geoserver main UI at: <http://localhost:8080/geoserver>, and fill in the database information including:

The standard postgres port: 5432, and the standard user name, which was configured during installation

Then, we create a workspace in the geoserver. A workspace is a collection of layers. Within the workspace, we create a layer that connects to the core schema in the datastore database.

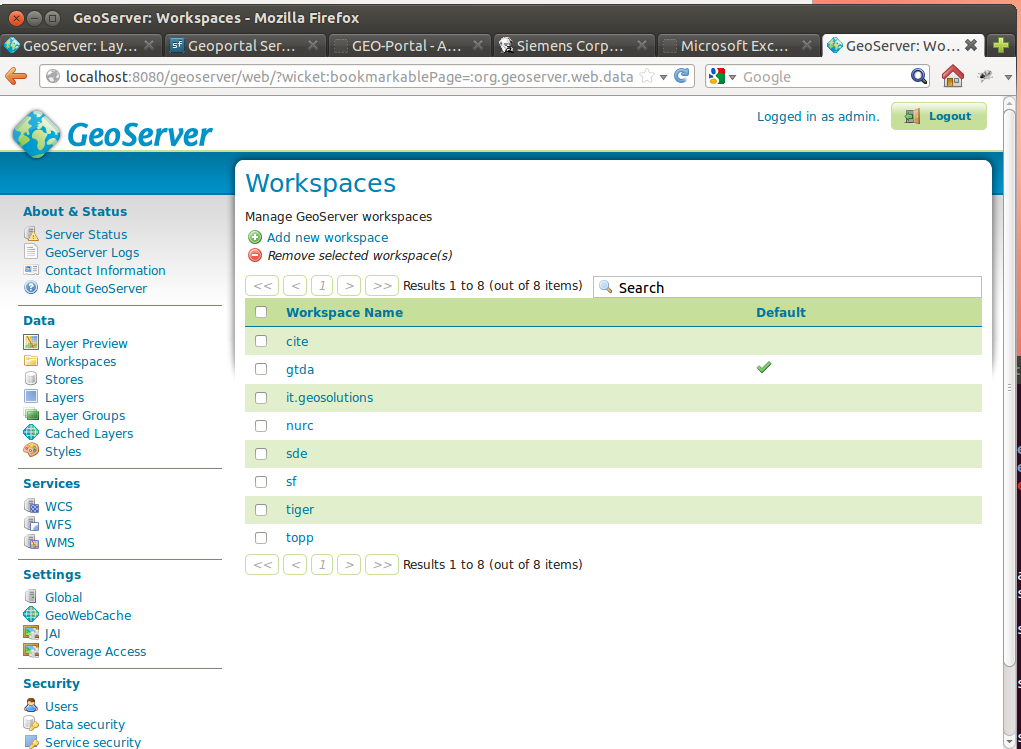


Figure Geoserver NGDS workspace

There should be a NGDS workspace in the workspaces tab. Create one if needed.

The workspace name is referenced in the development.ini file.

## Hints for when the installation doesn’t work

If the installation seems to just stall out, read back for a ways in the output of the installation script. At this time, the error messages are difficult to differentiate from the status output.

Most common errors:

1. Typos – typos can be very difficult to spot and can lead to unclear error messages in some cases. Check your text and paths carefully. (To address, the installation is going to be changed to generate some of the development.ini file either automatically or by copying from a provided file).
2. URLs and other file locations specified in the development.ini file must have exactly the expected / or have no / at the end as per the examples given.
3. Be aware where you are installing as sudo and where you are not. Watch for permission errors – if there are some, sudo and open up permissions on the directories involved.
4. If all the installation seemed to have proceeded ok, and yet the NGDS website won’t start up, then follow some of the wiki instructions for setting up eclipse for this project, and then attempt to start the paster command from within eclipse. This allows you to debug through what is not working and in some cases can immediately point out the issue (which is so far usually been one of the two issues above). Eclipse is a useful development tool in any case.

At the time of this writing, you might see a message near the end of the installation “Could not create CSW tables. Please make sure that you’ve added the csw plugin to your CKAN config.init file” and re-run the installation again if necessary.

**NOTE: When running NGDS in “node” mode, you need the plugin “datastorer”. When running NGDS in “central” mode, you need the plugin “harvest”. These two plugins cannot be the same instance!**

## Various Other Notes

If the environment changed due to further development, and you want to update it, here are some useful commands to update an existing environment.

To update requirements:

Goto directory ckanext-ngds/ckanext

Activate pyenv (. ~/pyenv/bin/activate)

pip install –r pip-requirements.txt

You can update the database using the following run from the src/ckan directory:

paster --plugin=ckanext-ngds ngds initdb

To fix up permissions if you have changed users in the development.ini file, you can use the following, also from the src/ckan directory:

paster datastore set-permissions postgres -c test.ini

See: http://docs.ckan.org/en/ckan-2.0/datastore-setup.html#create-users-and-databases

To start up postsgres in psql mode as super user:

sudo su postgres

psql

If you want to set up various PostGIS databases without using the installer to do so, you can make a PostGIS template database.

Build a PostGIS template DB

$ sudo su postgres

postgres $ createdb -E utf8 template\_postgis

postgres $ psql -d template\_postgis -f /usr/share/postgresql/9.1/contrib/postgis-1.5/postgis.sql

postgres $ psql -d template\_postgis -f /usr/share/postgresql/9.1/contrib/postgis-1.5/spatial\_ref\_sys.sql

postgres $ psql -d template\_postgis -c "GRANT ALL on geometry\_columns to PUBLIC;"

postgres $ psql -d template\_postgis -c "GRANT ALL on geography\_columns to PUBLIC;"

postgres $ psql -d template\_postgis -c "GRANT ALL on spatial\_ref\_sys to PUBLIC;"

postgres $ psql -d postgres -c "UPDATE pg\_database SET datistemplate='true' WHERE datname='template\_postgis';"

postgres $ exit

$

This template database should be used to create any new PostGIS-enabled database.

Manually creating databases (other than via the installation script):

$ sudo su postgres

postgres $ createuser -S -D -R -P ckan\_user

postgres $ createuser -s -P ckan\_tester

postgres $ createuser -S -D -R -P datastore\_reader

postgres $ createuser -S -D -R -P datastore\_writer

postgres $ createuser -S -D -R -P datastore\_test\_reader

postgres $ createuser -S -d -R -P datastore\_test\_writer

Creates users for production and testing setups for **CKAN** with the **Datastore** enabled. The little d is not a typo.

postgres $ createdb -E utf8 -O ckan\_user -T template\_postgis ckan\_main

postgres $ createdb -E utf8 -O ckan\_tester -T template\_postgis ckan\_test

postgres $ createdb -E utf8 -O datastore\_writer -T template\_postgis datastore

postgres $ createdb -E utf8 -O datastore\_test\_writer -T template\_postgis datastore\_test

postgres $ exit

$

# Appendix A: Summary of Changes to the Development.ini file

Here is a summary of all changes needed for the development.ini file. Watch the trailing slashes closely. Be sure also to not set all the plugins initially, but just as you are prompted by the installation script. Once the script is finished, verify that the plugins in the plugin list below are all in your plugins list.

sqlalchemy.url = postgresql://testuser:pass@localhost/testdb

ckan.datastore.write\_url = postgresql://testuser:pass@localhost/test\_datastore

ckan.datastore.read\_url = postgresql://readonlyuser:pass@localhost/test\_datastore

ngds.resources\_dir=/home/yourname/pyenv/src/ckanext-ngds/ckanext/ngds/base/resources/

ckan.geoserver.url=http://localhost:8080/geoserver/rest

ckan.geoserver.workspace\_name=NGDS

ckan.geoserver.workspace\_URL=http://localhost:5000/ngds

ckan.site\_id = ckan\_instance

solr\_url = <http://localhost:8983/solr>

ckan.storage.bucket=subdir

ofs.impl=pairtree

ofs.storage\_dir=/home/yourname/storage/

ngds.deployment=node

ngds.default\_group\_name=public

ngds.logo\_text=CONTRIBUTING GEOTHERMAL DATA

ngds.bulk\_upload\_dir=/home/yourname/upload

ngds.facets\_config=/home/yourname/pyenv/src/ckanext-ngds/facet-config.json

## plugins must be incrementally set during the installation!

## you can’t start out with the whole line

## also, either set datastorer or harvest, but not both, depending on if

## ngds.deployment is node or central

ckan.plugins = stats json\_preview recline\_preview datastore datastorer ngdsui ckan\_harvester metadata geoserver csw