Software INSTALLATION INSTRUCTIONS

NATIONAL GEOTHERMAL DATA SYSTEM - ngds

Version 0.2

*06/11/2013*

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 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[1 INTRODUCTION 6](#_Toc358741161)

[1.1 Purpose 6](#_Toc358741162)

[2 Prerequisites 7](#_Toc358741163)

[2.1 Choosing a virtualized environment 7](#_Toc358741164)

[2.2 Creating a VM and installing linux in a Windows environment 7](#_Toc358741165)

[2.2.1 First, download and install Oracle VM VirtualBox Manager 7](#_Toc358741166)

[2.2.2 Then, download the Desktop CD for PC(Intel x86) for Ubuntu: 7](#_Toc358741167)

[2.2.3 Start Oracle VirtualBox and Create a VM for Linux Ubuntu 7](#_Toc358741168)

[2.2.4 Configure the new VM 11](#_Toc358741169)

[2.2.5 Mount the Linux installation .ISO file in the VM before it starts 12](#_Toc358741170)

[2.2.6 Finally, install Linux Ubuntu 12.04 13](#_Toc358741171)

[3 Configuring the VM behind a corporate firewall 17](#_Toc358741172)

[3.1 Install CNTLM 17](#_Toc358741173)

[3.2 Configure the Linux VM Proxy to use CNTLM as its proxy 17](#_Toc358741174)

[3.3 What to do if cntlm and proxy continue to cause issues 19](#_Toc358741175)

[4 NGDS Installation Procedure 20](#_Toc358741176)

[4.1 Installing Git 20](#_Toc358741177)

[4.2 Obtaining the installation program and script 20](#_Toc358741178)

[4.3 Modifying installation parameters 20](#_Toc358741179)

[4.4 Installing required programs 21](#_Toc358741180)

[4.5 Installing Apache Solr 21](#_Toc358741181)

[4.5.1 Downloading Solr 21](#_Toc358741182)

[4.6 Run the Installation program 22](#_Toc358741183)

[4.7 Setup Solr in the middle of the installation 22](#_Toc358741184)

[4.8 Run solr 22](#_Toc358741185)

[4.9 Run the installation again 23](#_Toc358741186)

[4.10 Installing Geoserver 23](#_Toc358741187)

[4.10.1 Connecting geoserver with our datastore Postgres database (optional) 24](#_Toc358741188)

[4.11 Parameters for development.ini 25](#_Toc358741189)

[4.12 Start up celery 26](#_Toc358741190)

[4.13 Starting up NGDS 26](#_Toc358741191)

[4.14 Post-Installation steps 26](#_Toc358741192)

[4.15 Hints for when the installation doesn’t work 27](#_Toc358741193)

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 enables a peer-to-peer network of data providers through its many node-in-a-box distributions, and a centralized node that supports the harvesting and search of this distributed data.

## Purpose

The purpose of this document is to provide 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).

The current implementation of the NGDS project is developed in Python and Linux OS. In particular, we utilize Ubuntu distributions 12.04 or higher (specifically, Xubunto).

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.

## 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 provide an abstract hardware representation emulating a 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

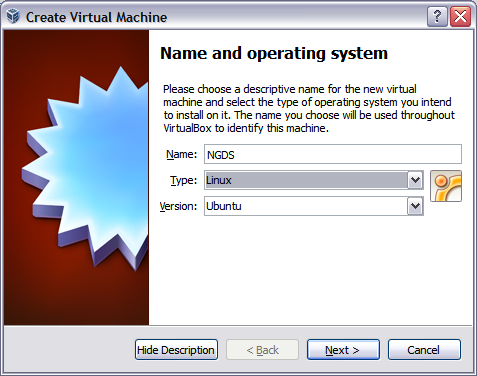


Figure 1 Create a new linux virtual machine

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

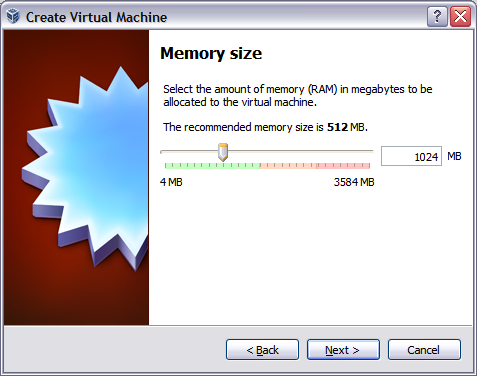


Figure 2 Set the VM memory to at least 1024MB

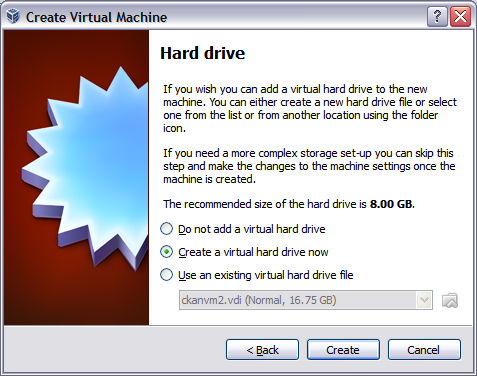


Figure 3 Create a virtual HD. The disk size can be changed later.

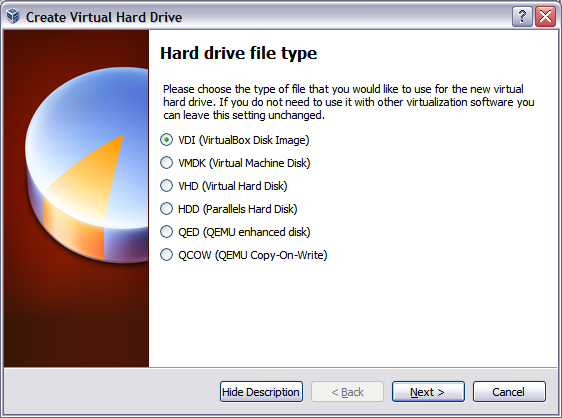


Figure 4 Specify the image type

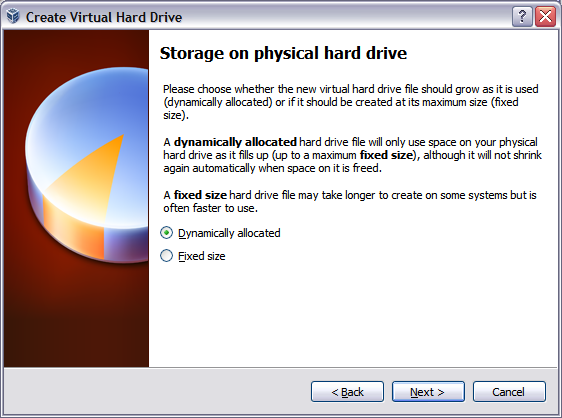


Figure 5 Dynamically allocated disk space will allow the image to grow as new data is imported into NGDS

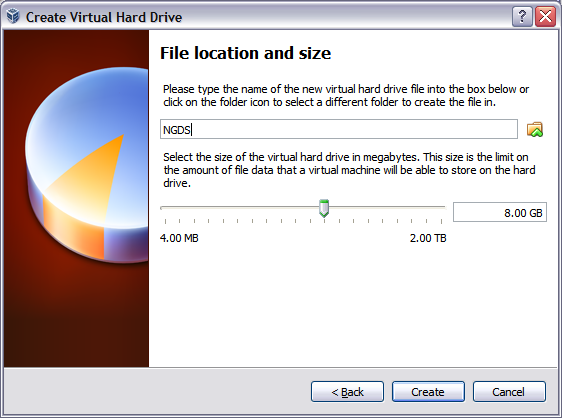


Figure 6 Configure the hard drive size. We recommend large values, based on the amount of data the node should store.

### 

Figure 7 Oracle VM Linux machine configuration

### Configure the new VM

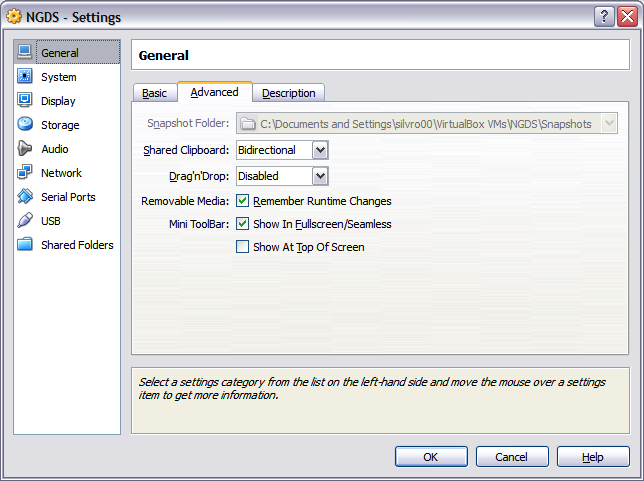


Figure 8 You may wish to enable the shared clipboard

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.

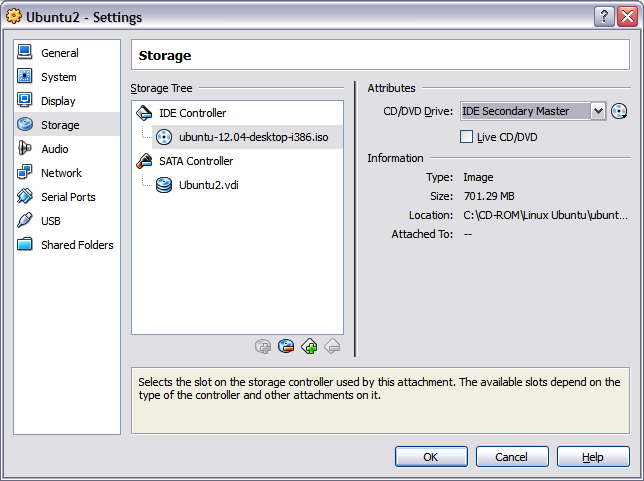


Figure 9 Mounting ISO image in the Linux VM

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

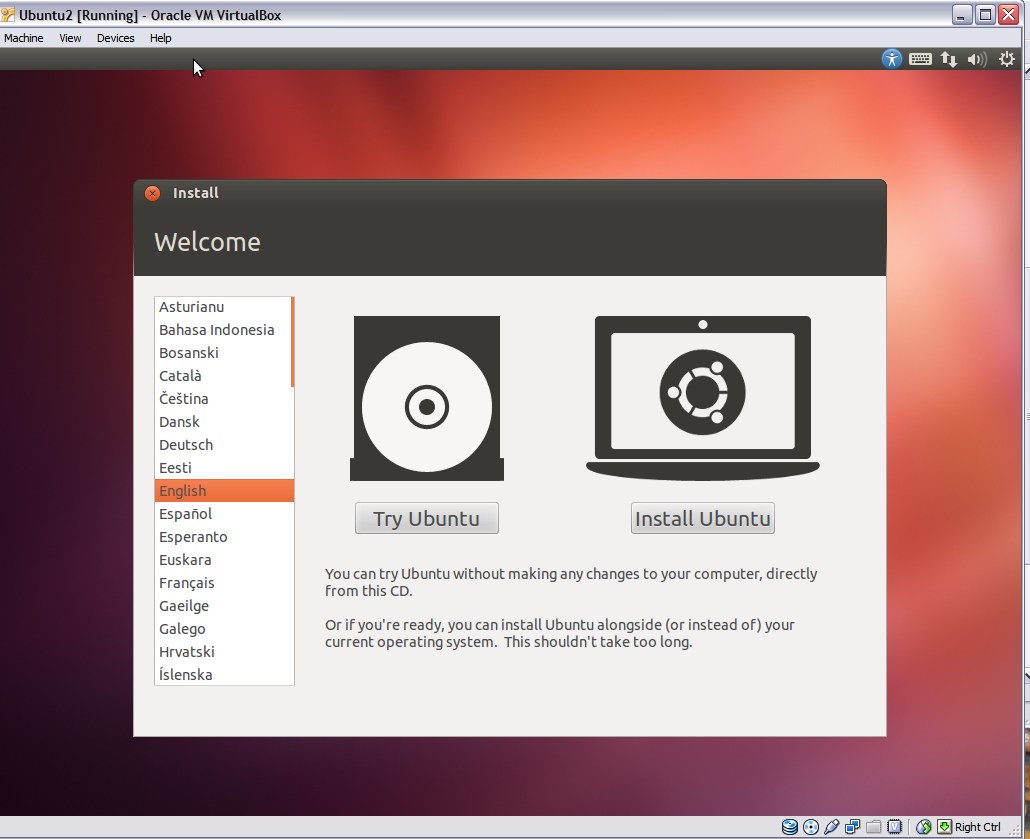


Figure 10 Linux Ubuntu installation screen

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.

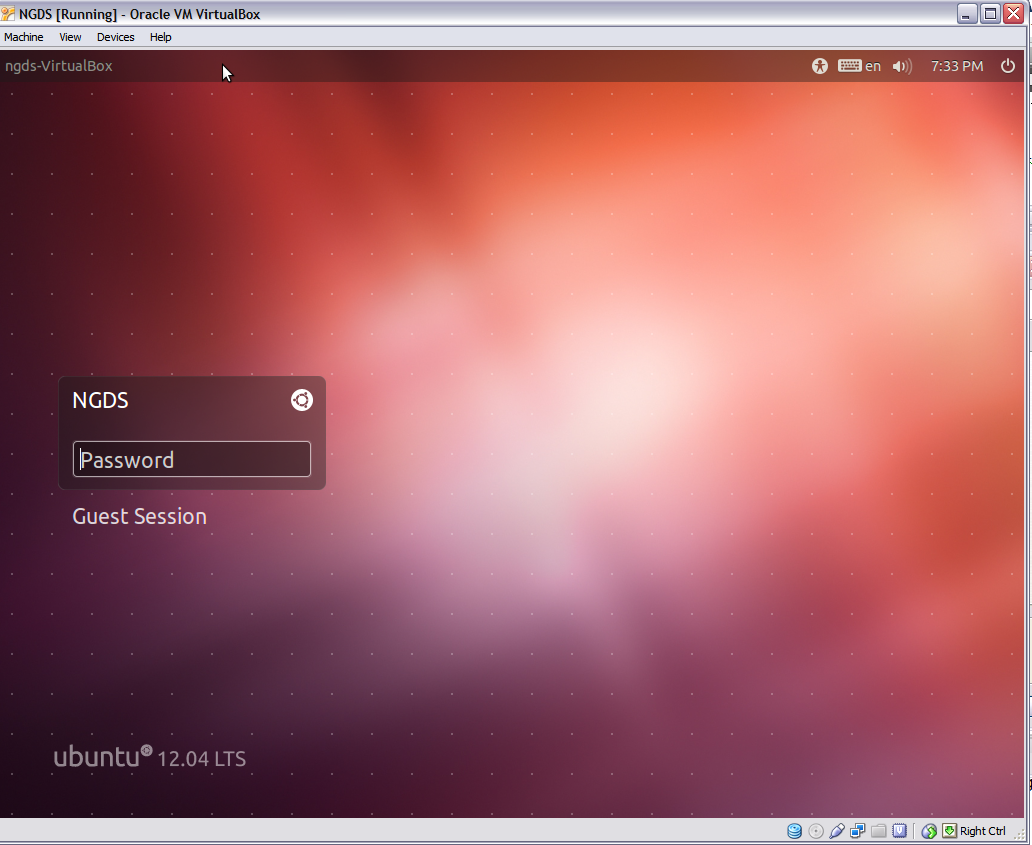


Figure 11 Login for the new VM

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

We also recommend the installation of the VM guest additions.

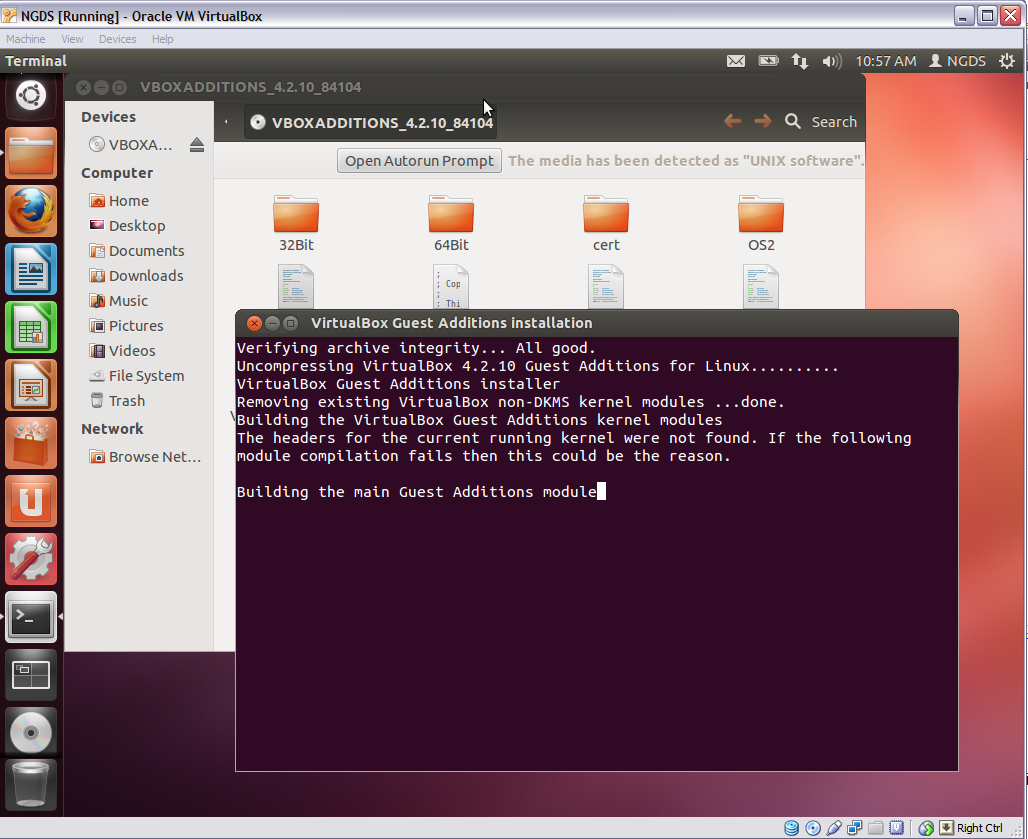


Figure 12 Installing guest additions

# Configuring the VM behind a corporate firewall

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.

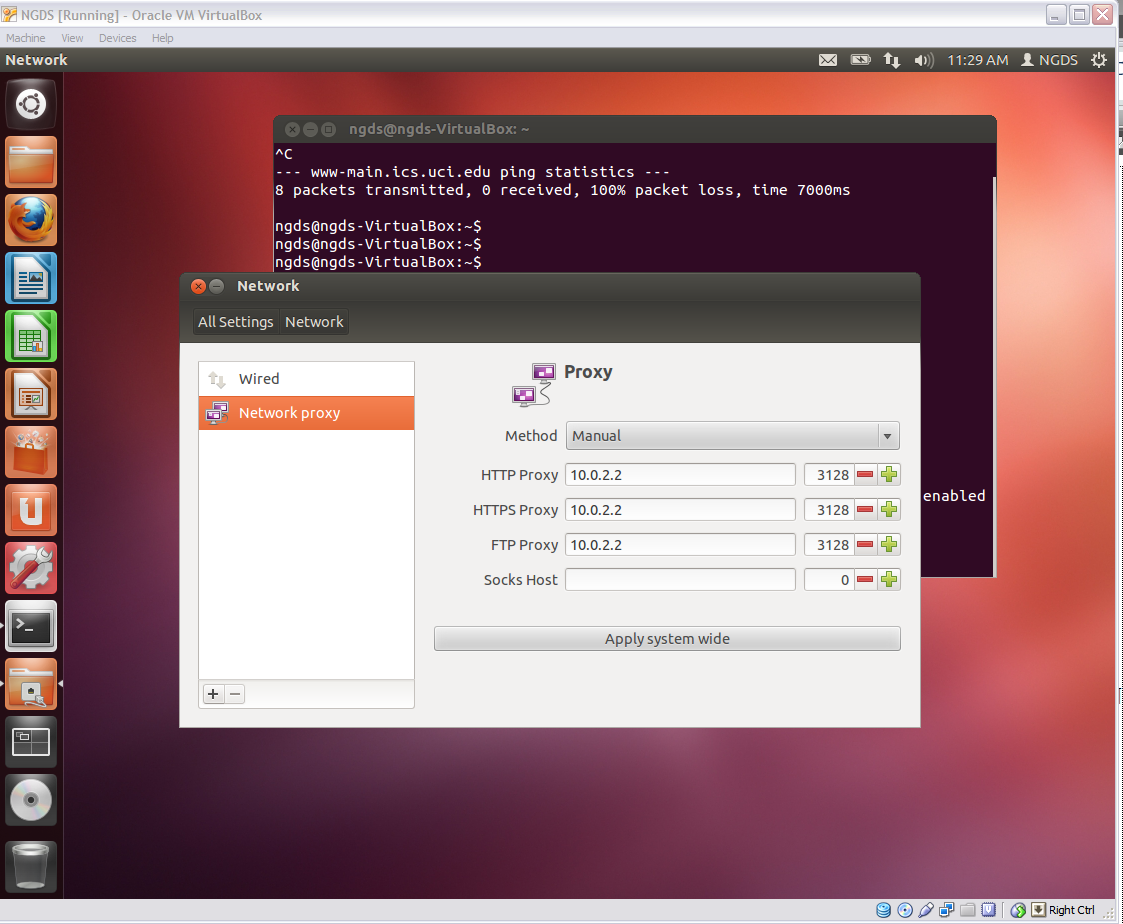


Figure 13 Configuring the proxy in Ubuntu Linux

Install updated before continuing.

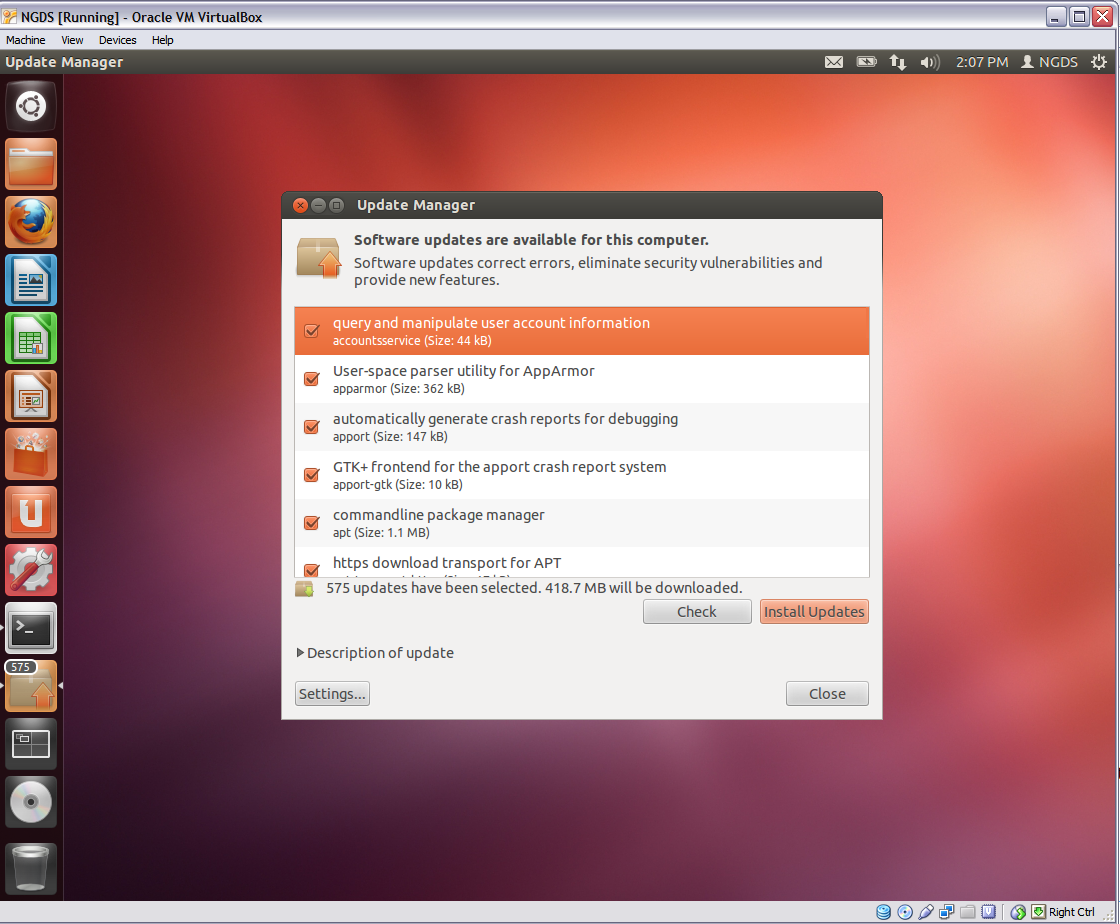


Figure 14 Install updates before continuing

## 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 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 (GIS)
* Jetty
* Geoserver
* Different python extensions

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 it rather interactive in how it currently functions, so it is recommend to read this whole chapter first before proceeding.

## 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 folder with a copy of the dev-info repository. Note that this requires a user account and password for this repository.

## Modifying installation parameters

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

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\_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)

While the installer.sh script runs, it will prompt you in several places to update various items.

For example, you will be asked to change settings like the following in the development.ini file (the development.ini file is under pyenv/src/ckan):

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

#ckan.site\_url = <http://localhost:5000> can just be commented out

solr\_url=<http://127.0.0.1:8983/solr>   # (no trailing slash)

ckan.site\_id=ckan\_instance         # (can be anything)

Obs. Do not run the installation script yet.

## 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. The only dependency currently not available in Ubuntu’s distribution is Solr and Geoserver. These components need to be installed separately.

## 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://www.trieuvan.com/apache/lucene/solr/4.2.1/

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

|  |
| --- |
| % mkdir /opt/solr  % cd /opt/solr  % wget http://www.trieuvan.com/apache/lucene/solr/4.2.1/solr-4.2.1.tgz  % tar xvfz solr-4.2.1.tgz |

## Run the Installation program

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

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

**Important Note:** As soon as the step where ckan is cloned from git is complete, the script will pause and prompt you to make changes to development.ini. At this point, stop solr if it is running, navigate to your pyenv/src/ckan/ckan/config/solr directory and copy schema-2.0.xml. Put it under your solr/example/solr/collection1/conf directory. Delete the schema.xml file in that directory and rename schema-2.0.xml to schema.xml. Restart solr.

|  |
| --- |
| % cd ~/pyenv/src/ckan/ckan/config/solr  % sudo cp schema-2.0.xml /opt/solr/solr-4.2.1/example/solr/collection1/conf/schema.xml |

## Setup Solr in the middle of the installation

After running the installation program, a pyenv folder with ckan will be created. It will follow the name conventions provided in the config file.

Under example/solr/collection1/conf of your solr installation dir, you will find a schema.xml file. Replace this file with schema-2.0.xml under pyenv/src/ckan/ckan/config/solr and rename it to schema.xml. This will allow solr to index imported ckan files.

## Run solr

Inside, you will find an example directory which has a start.jar file. Run it via command line using java -jar start.jar.

Verify that it works by visiting localhost:8983/solr.

## Run the installation again

Since we had to stop or pause the installation in the middle to configure Solr, we have to run the installation again if stopped. When asked whether to overwrite some context reply with “N”  
since we have modified the development.ini.

As the installation runs, you have to keep editing it files as it tells you, as the installer will pause and ask you.

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.

One question the installer is not asking at this time is to set the resources directory before it needs it, so add this line:

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

## 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 Gogle 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.

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 can run geoserver as a web application within tomcat, in this case, one has to copy the geoserver.war file to an existing tomcat installation.

|  |
| --- |
| % unzip geoserver-2.1.4-war.zip  % cp geoserver.war /var/lib/tomcat7/webaps |

After copying the .war file, check the proper installation of geoserver at: <http://localhost:8080/geoserver/>

Log-in as ‘admin’, the password is ‘geoserver’

Another option is to run geoserver as a standalone application. 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>.

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

### Connecting geoserver with our datastore Postgres database (optional)

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.

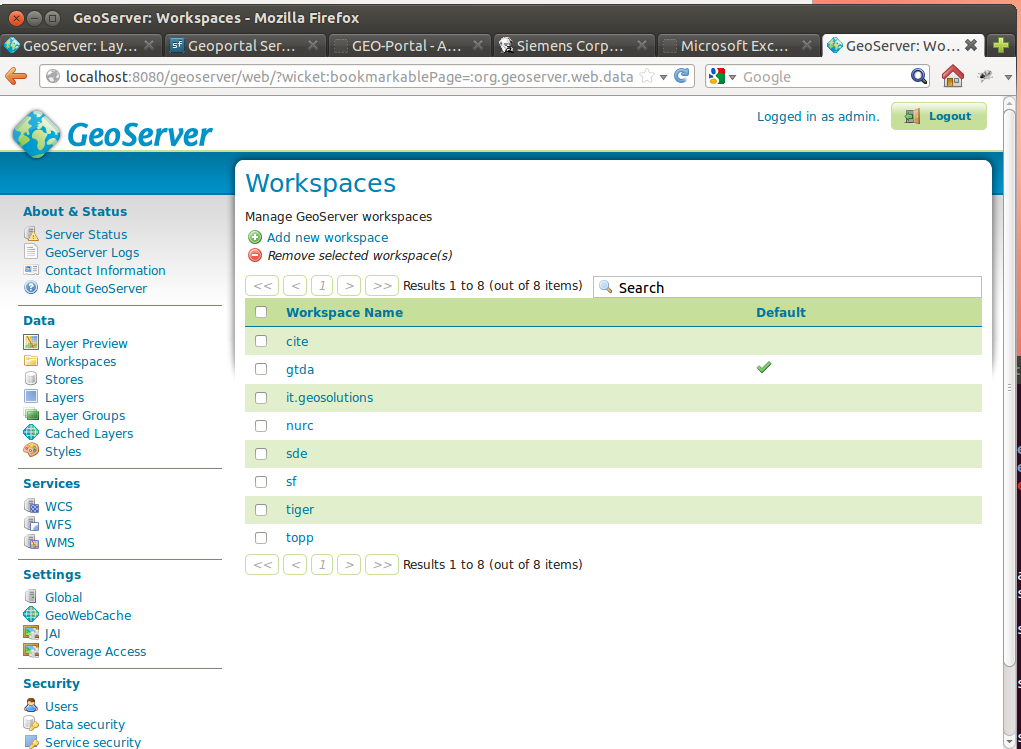


Figure 16 Geoserver gtda workspace

There should be a gtda or NGDS workspace in the workspaces tab.

## Parameters for development.ini

The development.ini file is automatically generated by the installation script. The NGDS plug-ins list in this file need to 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.

The geoserver integration also requires the following parameters which the installation will not prompt you for, plus the parameters mentioned below in the post installation steps.

ckan.geoserver\_url **=** http**:***//localhost:8080/geoserver/rest*

ckan.geoserver.workspace\_name **=** NGDS

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

Important note: development.ini has various sections, where the […] statements indicates 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.

## 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

## 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

Note that for some things you don’t want pyenv activated, so it’s not recommended to put it into your /etc/environment file.

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

Remove ngdsui from the plugins list (this starts ckan in “default” mode instead of in ngds mode).

Restart the server (better yet paster serve development.ini --reload and it will auto reload)

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

Add ngdsui back in to the list of plugins. Restart the server (if it hasn't restarted)

Then to the development.ini, add the following settings, and then restart the server if it wasn’t started with –reload option, using the paster command.

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

## 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.
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. 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”. This message can be safely ignored at this time.