National Geothermal Data System

Node-In-A-Box Software Installation Instructions

Arizona Geological Survey

Christoph Kuhmuench, Siemens Corp.

version 1.3; 3/18/2014

Edit History

Version:	Author:	Date:	Details
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
0.7	Christoph Kuhmuench	12/26/2013	Updating to latest installer.
0.8	Jordan Matti	2/7/2014	Many changes; tracked changes best viewed *without* showing format changes (**Review** pane -> **Show Markup** -> **Formatting**)
0.9	Christy Caudill	2/19/2014	Moved Window OS/ Oracle VM install to Appendix A.
0.91	Jordan Matti	2/20/2014	Minor changes (formatting, headers, comments)
1.0	Christy Caudill	2/21/2014	Formatting changes, minor edits and figure updates
1.1	Christy Caudill	3/7/2014	Edits from test install with VM
1.2	Christy Caudill	3/13/2014	Serious revisions, updating the installer script used.
1.2.1	Christy Caudill	3/14/2014	Incorporating changes per Matt MacKenzie
1.3	Christy Caudill	3/18/2014	Update to installer script, edits per Matt MacKenzie. Added Section 5.

Contents

List of Figures

1. Preface

National Geothermal Data System (NGDS) was a Department of Energy-funded effort to facilitate public access to information about geothermal resources from public and private sources. NGDS data is available through a distributed, scalable network of data providers.

1.1 Purpose and Audience

This document is a step-by-step tutorial to help new developers and users setup an instance of the **NGDS Software Stack for an NGDS node**. It describes a version 1, beta installation method in advance of a more friendly user interface to be rolled out in late 2014, but is also intended to give system administrators a more thorough understanding of the components and architecture of NGDS Node-In-A-Box (NIAB).

This document is intended for a technical audience who need to understand the concepts and the reasoning of the installation process. Targeted audiences include:

- **NGDS System Administrators**
- Software Architects
- Software Developers

This document purports to:

- Describe NGDS and how it works
- Identify and describe the basic components that are necessary to install the NGDS Software Stack
- Provide step-by-step installation instructions for the NGDS Software Stack on an Ubuntu Linux operating system
- Outlines the process of installing the NGDS Software Stack in **production** mode, as compared to **development** mode

This fulfills one of the main goals of NGDS: to provide a basis for a sustainable open-source software project that is attractive for an open source team to maintain. With this documentation, system administrators will be able to quickly understand the system and become a productive node in that system.

1.2 Document Roadmap

This document outlines the architecture of NGDS and is structured in the following way:

- Section 2: NGDS Software Stack prerequisites
- Section 3: Installing the NGDS Software Stack on an Ubuntu Linux operating system
- Section 4: NGDS Software Stack installation troubleshooting
- Appendix A: Installation guide for an Ubuntu Linux virtual machine in VirtualBox
- Appendix B: An overview of the **development.ini** file
- Appendix C: NGDS architecture and diagrams and notes

1.3 System Scope and Background

NGDS is a distributed data-sharing network. NGDS data providers host data using their own computing resources and submit metadata describing their data to web-accessible NGDS metadata repositories (referred to hereafter as **publisher** nodes).

Metadata submitted to registered **publisher** nodes is regularly harvested by the NGDS **aggregator** node, which creates web-accessible metadata catalogs. A metadata catalog can be used to find and access any data described by metadata records in the catalog. Thus, the aggregator node becomes the one-stop search interface for the entirety of the system. NGDS publisher nodes are provided by the NGDS Software Stack; the NGDS aggregator node can be currently accessed at www.geothermaldata.org.

What is the NGDS Software Stack?

The **NGDS Software Stack** is a collection of applications designed to interact with NGDS data, metadata, and interchange formats.
When installed, the NGDS Software Stack allows the computer on which it is installed to become an NGDS node. There are two types of NGDS nodes:

- **Publisher** nodes: When installed on a server and configured to act as a publisher node, the NGDS Software Stack provides a web-accessible interface that can be used to submit and manage metadata records. Metadata that has been added to a publisher node that has been *registered* with an NGDS aggregator node will be harvested by the aggregator node at regular intervals.
- **Aggregator** nodes: When installed on a server and configured to act as an aggregator node, the NGDS Software Stack provides a web-accessible metadata catalog to which harvested metadata records are added. An NGDS aggregator node will harvest metadata from any registered catalog service (CSW) or any registered NGDS publisher node. This same installation can produce either a node or and aggregator, but this document focuses on the installation of a node.

Note that the NGDS Software Stack can also be installed in two *modes*:

- **Production** mode: a stable release of the software stack
- **Development** mode: used by developers to create new versions of the software stack for subsequent release

2. Prerequisites

Installing and configuring the individual components utilized by the NGDS Software Stack requires a physical or virtual computer with the following properties:

- Network access
- A properly configured *clean* Ubuntu Linux distribution 12.04 or higher operating system installed (example: Xubuntu 13.04 desktop-i386.iso)
- A user account with Super-User (Administrator) privileges
- At least 1024 megabytes of RAM; a physical computer that will be used to host a virtual machine should have sufficient RAM to allocate at least 1024 MB of RAM to a virtual machine

Continue to Section 3 of this document which describes the steps necessary to install the NGDS Software Stack as a **publisher** node. For those strictly using Windows OS, Appendix A of this document describes preliminary steps to create your own virtual machine and install Ubuntu Linux using Oracle VM Virtual Box (free download).

3. Install the NGDS Software Stack

The NGDS Software Stack depends on a number of operating system components that must be installed on a computer before that computer can become an NGDS node. These components include:

- Java Development Kit (JDK)
- Git
- Apache SOLR
- PostgreSQL database
- PostgreSQL extensions for Geographical Information Systems (POSTGIS)
- Geoserver
- Apache Tomcat
- CKAN
- Python extensions
- gdal

To install these components, the computer on which they will be installed must have access to the Internet. All of these components (with the exception of Git) will be installed automatically by the NGDS Software Stack installation script.

Figure 1 provides a visual representation of the manner in which these components interact. Components near the top of the figure are *nested* within components near the bottom of the figure:

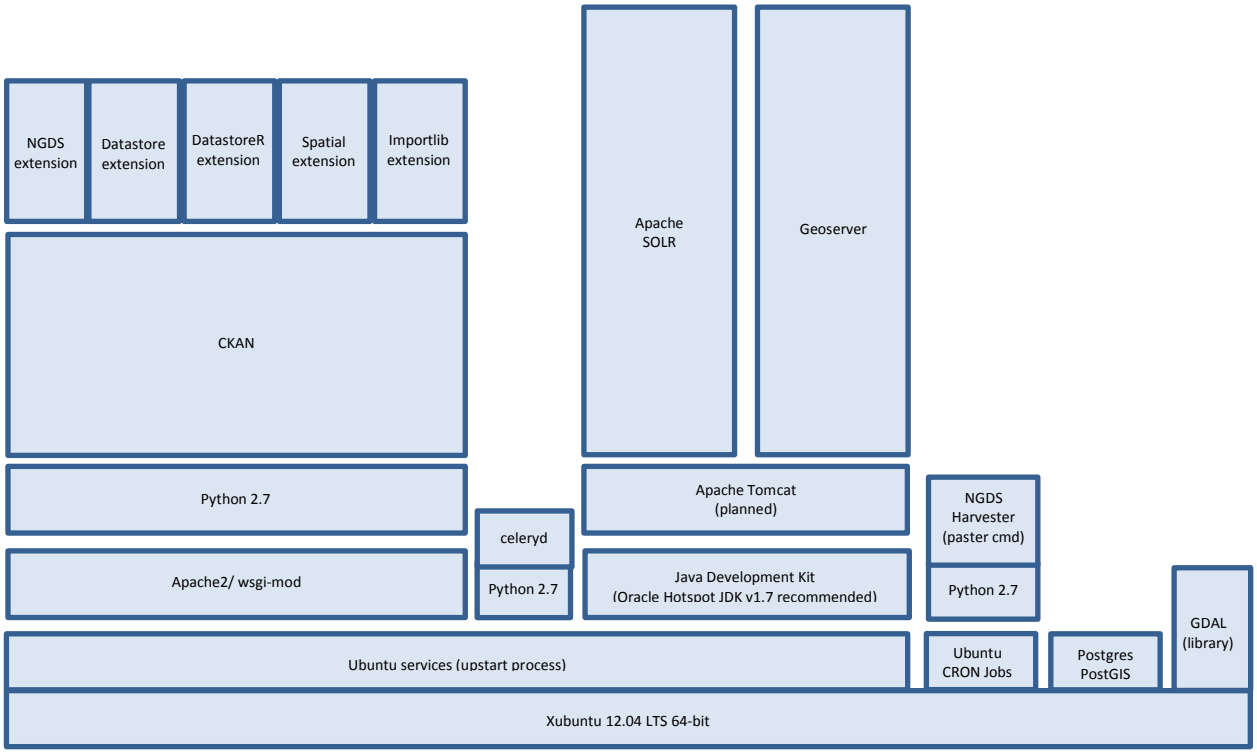

Figure 1: NGDS Software Stack in Production Mode

3.1 Install Git

To install **Git**, make sure you are logged in as **ngds**. Open an Ubuntu Linux terminal and execute the following commands:

```
%  cd ~ngds
%  sudo apt-get install git git-core
```

3.2 Obtain the NGDS Software Stack Installation Files

To obtain the installation files for the NGDS Software Stack, create a **tmp** directory and clone the git repository:

```
%  mkdir tmp
%  cd tmp
%  git clone https://github.com/ngds/ckanext-ngds.git
```

3.3 Set Installation Parameters

Before running the NGDS Software Stack installation script, you will need to ensure specific required installation paramaeters exist. To do so, navigate to the following directories and use a text editor to make the following changes:

```
%  cd ckanext-ngds/installation
%  sudo nano install-ngds.sh
```

The most important variables to specify are:

- depolyment_type: User may choose between *central* or *node* mode. Default is: *node*, and for the purposes of this document, *node* is the correct option.
- site_url='http://myservername_IPname'

The **site_url** should indicate the exact web-facing server or IP address on which this software is installed. Other variables such as the Apache Tomcat home directory can be configured in this file as well. Do not change anything beyond line 95, which reads, "DO NOT CHANGE ANYTHING BELOW THIS POINT".

Next, change to the following direcrtory to make changes:

```
%  cd ckanext-ngds/scripts
%  sudo nano ngds_config_file.py
```

The same **site_url** indicated above must be indicated in the **ngds_config_file.py** file. Scroll down to **node_params** and the line:

```
("geoserver.rest_url", "geoser er://admin:geoserver@localhost:8080/geoserver/
rest","This is Geoserver rest URL"),
```

which must be edited to remove 'localhost', entering the exact web-facing server or IP address on which this software is installed as in the example that follows. This will allow web services published through the installed GeoServer instance to be web-accessible.

```
("geoserver.rest_url", "geoserer://admin:geoserver@myservername_IPname:8080/
geoserver/rest","This is Geoserver rest URL"),
```

3.4 Run the Installation Script

In an Ubuntu Linux terminal, in the **installation** directory, execute the following command:

```
%  sudo ./install_ngds.sh
```

The above script will take some time to install various features and functions.

Note: Tomcat will need to have sufficient memory (the JVM needs at least 2 GB). It may be necessary to edit Tomcat's **JAVA_OPTS** or **CATALINA_OPTS** variable; see Appendix C.3 for more information.

The NGDS Software Stack has now been installed; follow the additional steps below to complete configuration of your new node.

3.5 Final Steps

If the installation was performed correctly, the web-accessible interface provided by the NGDS Software Stack can be reached at:

http://127.0.0.1/

Having navigated to the above address, perform the following:

3.5.2. Log in with the following credentials:

- Username: admin
- Password: admin

3.5.3. Navigate to the following URL:

- http://127.0.0.1/organization

3.5.4. Create a new organization; make the name of the organization:

- Public

Note: **admin** is not a secure password. Though we will use it in examples for the remainder of this document, we recommend that you change the password to something more secure.

The system has now been configured and is ready for use.

4. Troubleshooting your NGDS Installation

If the installation seems to stall out, check the output of the installation script to look for error messages. If the site http://127.0.0.1/ seems slow or does not load correctly, give it a few extra minutes and try again. Likely, you will need to restart apache and wait a few more minutes.

The most common errors are:

1) **Typos**: typos appearing in commands or paths can be very difficult to spot and can sometimes lead to unclear error messages. Check your text and paths carefully. Some scripts (such as BASH) are case-sensitive, so a lower-case or upper-case letter in the wrong place can cause problems.
2) **Permission Errors**: Permission errors occur when you try to perform an action without super-user capabilities. If you notice permission errors, use the **sudo** command ("super-user do") to open up permissions on the directories involved.
3) **Path Errors**: Some paths in the **development.ini** file terminate in a **slash** character (/) and some do not; the presence or absence of a **slash** character can impede an installation. Compare your **development.ini** file with the entries listed in Appendix A of this document to make sure your **development.ini** file has **slash** characters in the right places. See Appendix B for more information on the development.ini file.

After evaluating the output of the installation script and fixing any errors you find, re-run the installation script.

5. Short tutorial on using the node installation

This section is intended to be a brief introduction to uploading data to a new installation of NIAB, perhaps more appropriate for a database manager than technical staff or software developer. Additional information can be found at the NGDS help site http://geothermaldata.org/ngds/data.

5.1. Tiers of NGDS data delivery

5.1.1. Tier 1

The simplest and most common access to resources is provided by simple Web links that result in a file download. Information contained in files can be accessed by users who have software that can recognize and open these files. This is the standard model for files accessible on the web, supported by HTTP servers and desktop web browser software.

Unstructured data requires user interpretation before it can be used for analysis. Users can utilize the information if they can understand the encoding and language, but the system provides no support for this understanding, and little or no automation is possible. Audio files must be transcribed; text files must be parsed and mined for data that is then broken down and structured in ways that can be processed by computers; images must be scanned, interpreted, and often georeferenced. Preparing Tier 1 data for analysis can be a painstaking and time-consuming process.

5.1.2. Tier 2

Tier 2 interoperability indicates that information content is structured (consistently organized) in a spreadsheet or database file such that it is amenable to computer processing; that said, Tier 2 data does not use a shared, documented interchange format. Data in this tier must be transformed by the data consumer on a case-by-case basis for integration with other datasets, requiring them to study each new data source to figure out how to extract the information they need. Obtaining data in a structured format is a step towards interoperability because once the format is understood, computer programs can be instructed to extract the desired information.

5.1.3. Tier 3

Tier 3 data is structured data that conforms to an NGDS information exchange. Data that is published according to the exchange specification (content model, interchange format, service protocol) is interoperable with any other data published using that exchange. This is referred to as Tier 3 interoperability. This is the most valuable data in the system, as it allows end users (researchers or computer programs) to retreive and manipulate data from any source in a predictable and expedient way. When a data file (CSV, for example) is said to be "schema-valid", this refers to the file as is conforming to the standards of a given information exhange schema. The blank Excel files for a given information exchange are found at http://schemas.usgin.org/models/. When data conforms to these specifications (field headings, data types, etc.) the file can be validated at http://schemas.usgin.org/validate/cm before uploading to the node as a tier 3, structured dataset (see Section 5.3).

5.2. How to upload tier 1 and 2 data

If a structured flat file (like CSV or Excel file) is uploaded, the user can preview the data table. Any file type that is uploaded will be available for users to download after the following steps:

- Go to the **Contribute** page.
- Fill in Title for the resource, keywords (Tags), and other metadata. Click **Next**.
- Click **Upload a file** and navigate to a file or other resource.
- Choose **Unstructured** or **Offline Resource** and enter in all metadata. Click **Add**.
- Enter all metadata. Click **Finish**.

5.3. How to upload teir 3 data and publish web services

This upload requires using only schema-valid CSV data files (see Section 5.1.3). Uploading **Structured** data files and publishing them as web services creates GIS points in WFS and WMS formats which is the greatest of utility within the system.

- Go to the **Contribute** page.
- Title will be the title of the web service. Find the required names of services at https://github.com/ngds/system-design/wiki. Fill in keywords (Tags) and other metadata. Click **Next**.

- Click **Upload a file** and navigate to the schema-valid CSV file.
- Choose **Structured Resource** and enter in all the metadata.
- Choose the appropriate content model from the drop-down list. Always use the latest version. Click **Next**.
- Fill out all metadata here, click **Finish**.
- The metadata for that dataset is now published, but the web service is not yet published. To publish the web service, click **Publish as OGC**.
- Choose the correct layer name from the drop-down list.
- Choose the Latitude and Longitude fields from the drop-down list. Click **Confirm**.

5.4. How to upload tier 3 metadata

As a beta-version of the NAIB, this functionality does not yet exist. Please see Section 5.2 instead for how to register resources and create metadata in the system.

Appendix A Installing a Virtual Machine

If you are working with an operating system other than Linux (e.g., Windows) and will be installing the Ubuntu Linux operating system on a virtual machine, you will need to choose appropriate virtualization software. 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.

In other words: a virtual machine is a computer that is created by a software application running on a host computer; a virtual machine therefore exists *entirely* within the memory of the host machine on which it is hosted. The obvious advantage here is that the resources of a single powerful host machine can be allocated to host many virtual machines for different purposes.

Though a virtual machine is not required for this project, this project requires a specific configuration of the Ubuntu Linux OS; a virtual environment is ideal for the installation of this configuration.

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

This tutorial was developed using VirtualBox version 4.2.10 for Windows. Here, we install Linux Ubuntu 12.04 LTS from Canonical.

A.1 Creating an Ubuntu Linux Virtual Machine using VirtualBox

The steps in Appendix A of this document describe the installation of Ubuntu Linux (or Xubuntu) on a virtual machine supported by version 4.2.10 of Oracle VM VirtualBox. Newer versions of VirtualBox can be utilized.

A.1.1 Download and install Oracle VM VirtualBox Manager

Download and install version 4.2.10 of Oracle VM VirtualBox on a Windows computer of your choice. Doing so allows your Windows computer to support one or more virtual machines.

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

Run the installer and follow the on-screen instructions to install VirtualBox.

A.1.2 Create an Ubuntu Linux Virtual Machine

Run the VirtualBox application installed previously and use it to create a virtual machine:

1. Run the VirtualBox application
2. Create a new virtual machine
3. Specify the following (Figure 2):
    a. **Name**: NGDS

   b. **Type**: Linux
   c. **Version**: Ubuntu
4. The maximum stack configured for Java is 2048 MB, so choose to allocate at least 3072 MB of ram to your virtual machine
5. Create a hard drive for your new virtual machine (Figure 3)
6. Specify the type of hard drive used by your virtual machine (Figure 4); the drive type you select determines the compatibility of the virtual hard disk you create with different virtualization software
7. Specify disk space allocation (Figure 5); dynamic allocation allows your virtualization software to allocate more hard drive space from the virtualization platform to this virtual hard drive as-needed
8. Allocate disk space to your virtual hard drive (Figure 6); this allocates a specified amount of hard drive space from the virtualization platform to the virtual machine

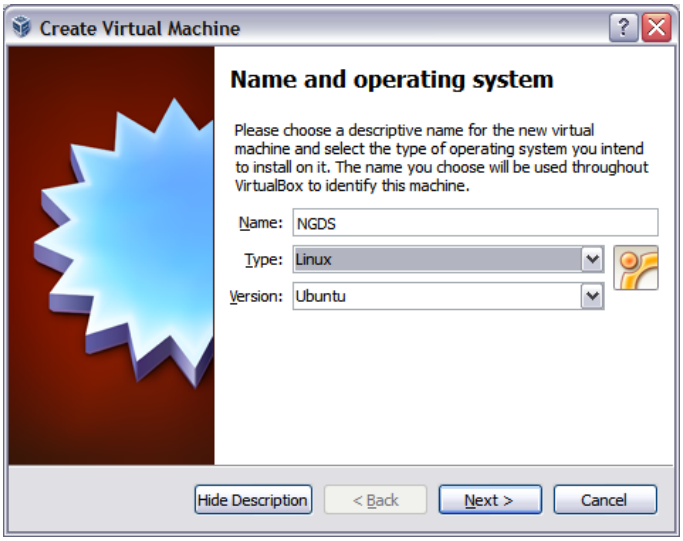

Figure 2: Create a new Linux virtual machine

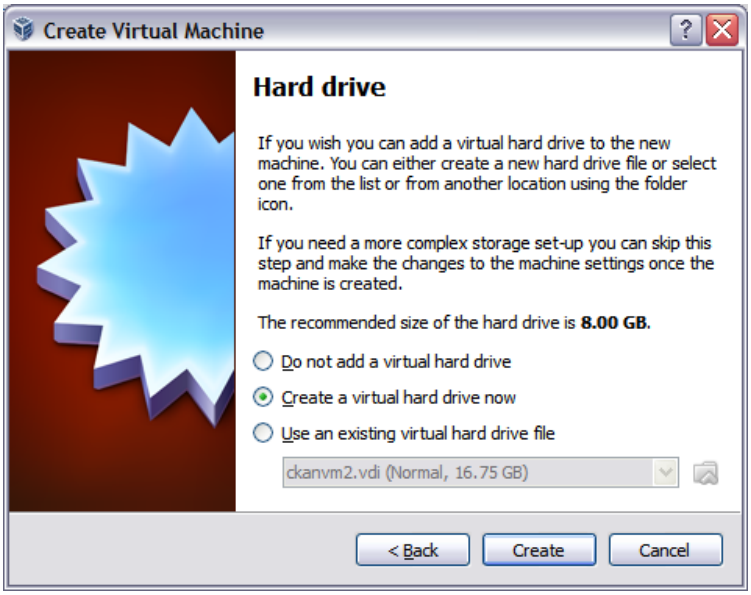

Figure 3: Create a virtual hard disk

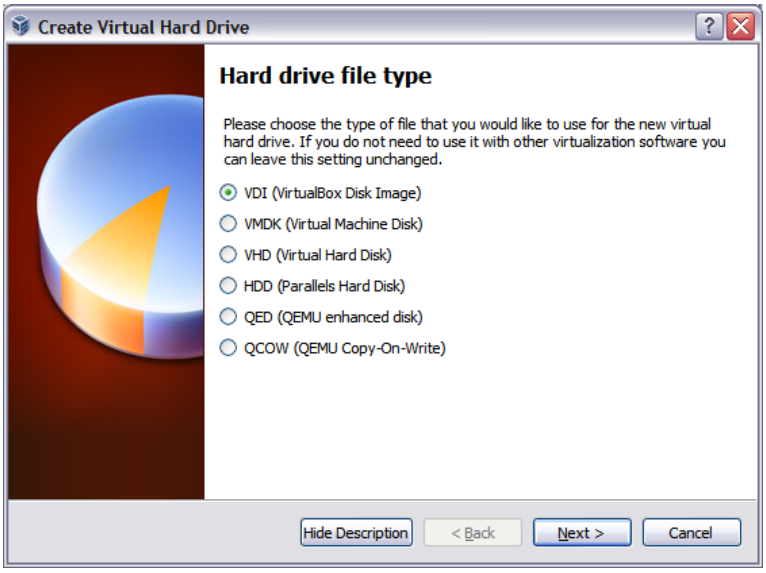

Figure 4: Specify the image type

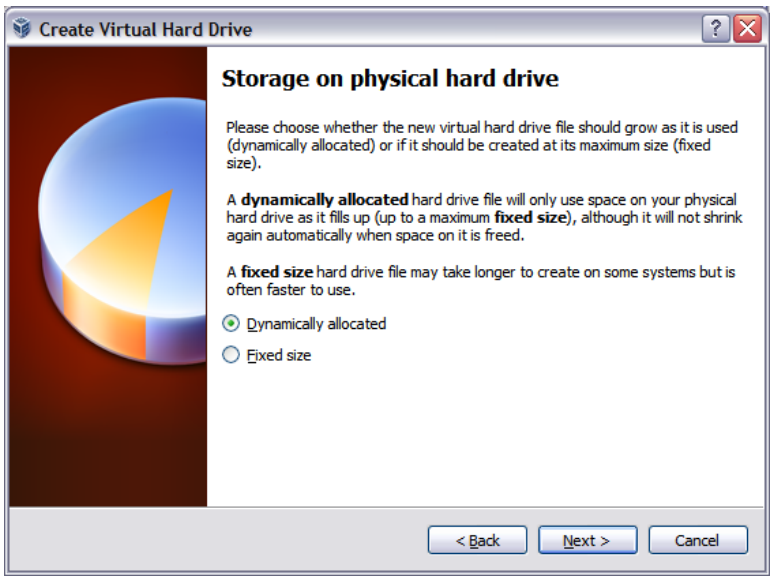

Figure 5: Specify storage allocation

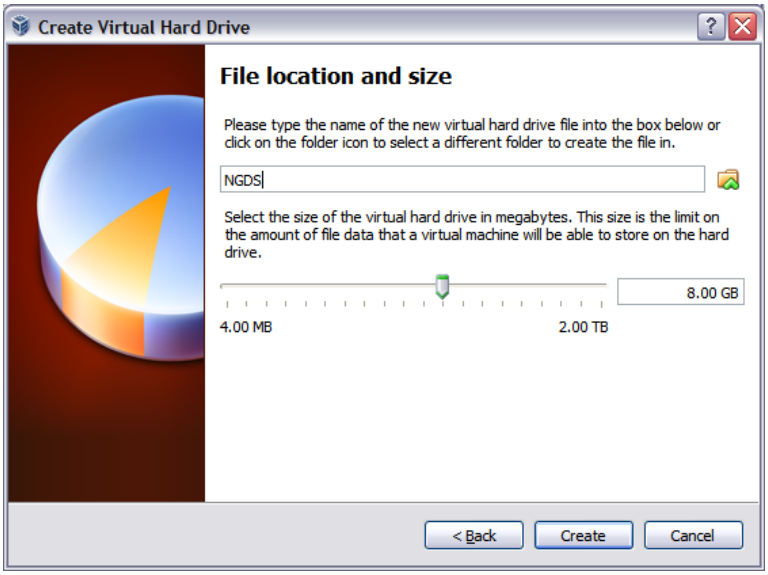

Figure 6: Configure virtual hard drive

A.1.3 Configure your Virtual Machine

Open the **Oracle VM VirtualBox Manager** (Figure 7); select your virtual machine and click **Settings**.

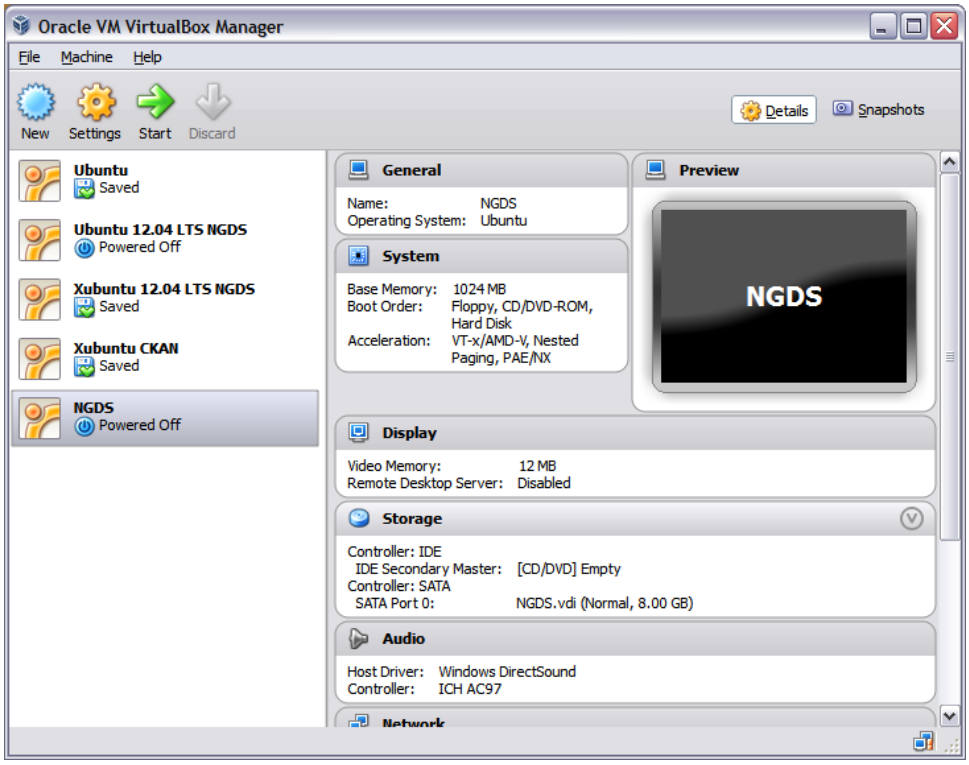

Figure 7: Configuring a virtual machine in VirtualBox

First, enable the **Shared Clipboard**:

1. Select **General** settings
2. Select the **Advanced** tab (Figure 8)
3. Click the **Shared Clipboard** dropdown menu
4. Select **Bidirectional**

This will enable anyone who connects to this virtual machine to copy and paste between the virtual machine and the computer used to connect to the virtual machine (including the computer on which the virtual machine is hosted).

Virtual machines, being virtual, are distinct from the computer that is used to connect to them and therefore do not necessarily share the same clipboard by default.

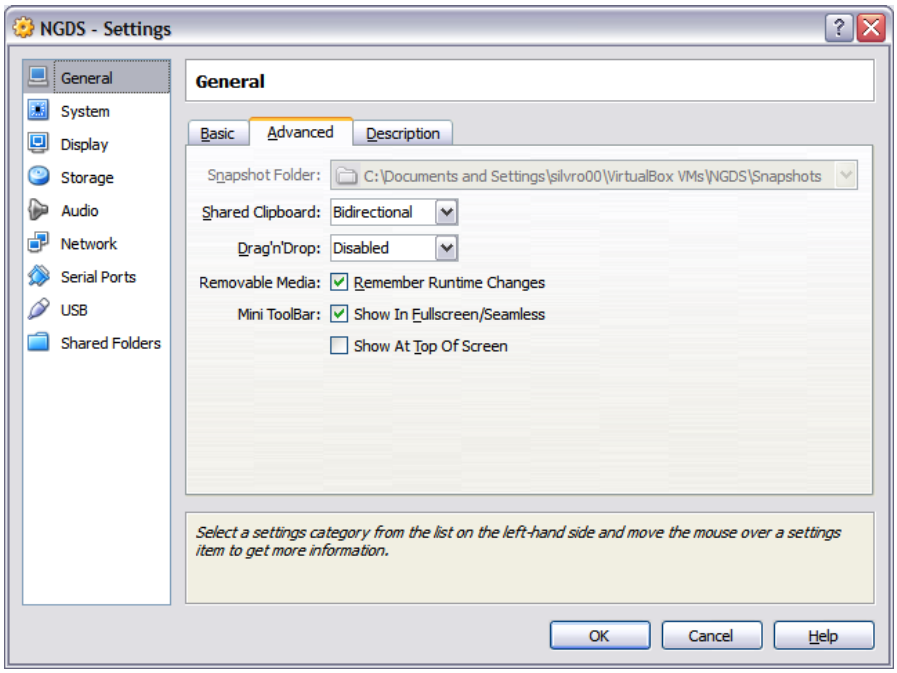

Figure 8: Enabling the shared clipboard

Now that your virtual machine is created and configured, you can install an existing Linux distribution.

A.1.4 Download an Ubuntu ISO image

An ISO image is a type of virtual CD (ISO stands for *International Standards Organization*; an ISO image is an image of an ISO-standard CD).

CD images are files that can be loaded and read by virtual CD drives. Virtual CD drives are software applications that emulate a CD-ROM drive in much the same way that an entire computer can be emulated by virtualization software.

To install Ubuntu on a virtual machine, you will need an ISO image of an Ubuntu installation file, available at: http://releases.ubuntu.com/12.04/

The site listed above provides multiple downloads; this tutorial utilizes the Long Term Service (LTS) version of Ubuntu, which features long-term support (3 years), which is the following download: http://releases.ubuntu.com/12.04/ubuntu-12.04-desktop-i386.iso

A.1.5 Mount the Linux installation .ISO file in your virtual machine

After downloading an ISO image but *before* starting it, mount it within the VirtualBox environment and use it to install the Ubuntu operating system on your virtual machine:

1. In the **Oracle VM VirtualBox Manager**, select the virtual machine you created in Section 3.2.2 and click **Settings**

2. In the **Settings** window, click **Storage** (Figure 9).
3. In the **Storage** panel under **Attributes**, click the **CD** icon next to the **CD/DVD Drive** dropdown menu on the far right.
4. Navigate to the ISO image you downloaded in Section A.1.4; select the ISO file
5. In the **Storage** panel, click **OK** to mount the image

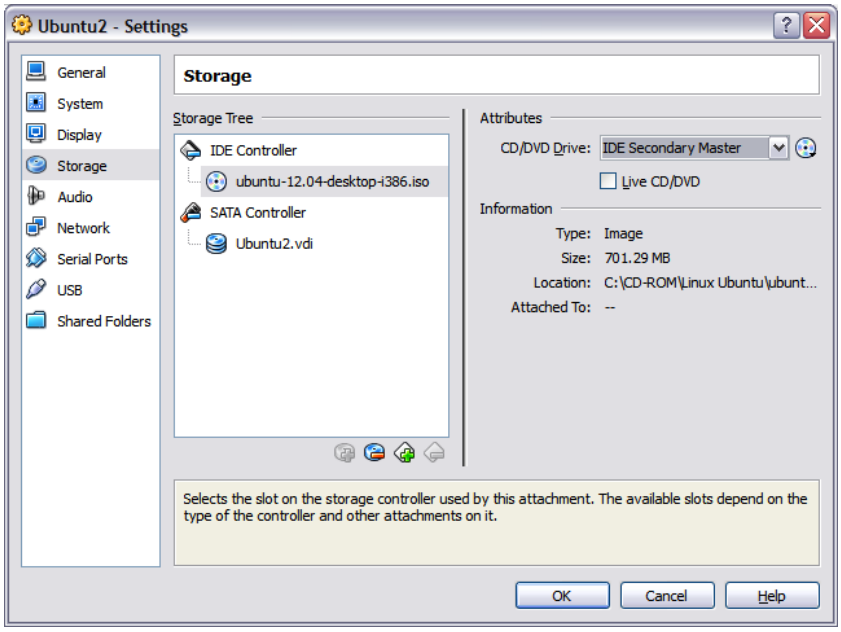

Figure 9: Mounting the Ubuntu ISO image in the VM

A.1.6 Install Ubuntu Linux 12.04

In the **Oracle VM VirtualBox Manager**, select your virtual machine and click **Start**.When started, your virtual machine will prompt you to install the operating system loaded in the image in much the same manner as you would on a physical computer.

Click **Install Ubuntu** to begin; follow the on-screen instructions (Figure 10).

When you are prompted to do so, create a user ngds . Enter **ngds** for **Your name** as well as for **Pick a username**; specify a password of your choice.

When the installation is complete, you will be prompted to restart. Once the virtual machine is shutting down, press Enter when prompted. When your machine is restarted, log in using the username **ngds** and the password you specified during the Ubuntu installation process (Figure 11).

In addition to the above, it is recommended that you install the **Guest Additions** module. Choose **Device** drop-down from the top left. Choose **Install Guest Additions** and follow the installation steps.

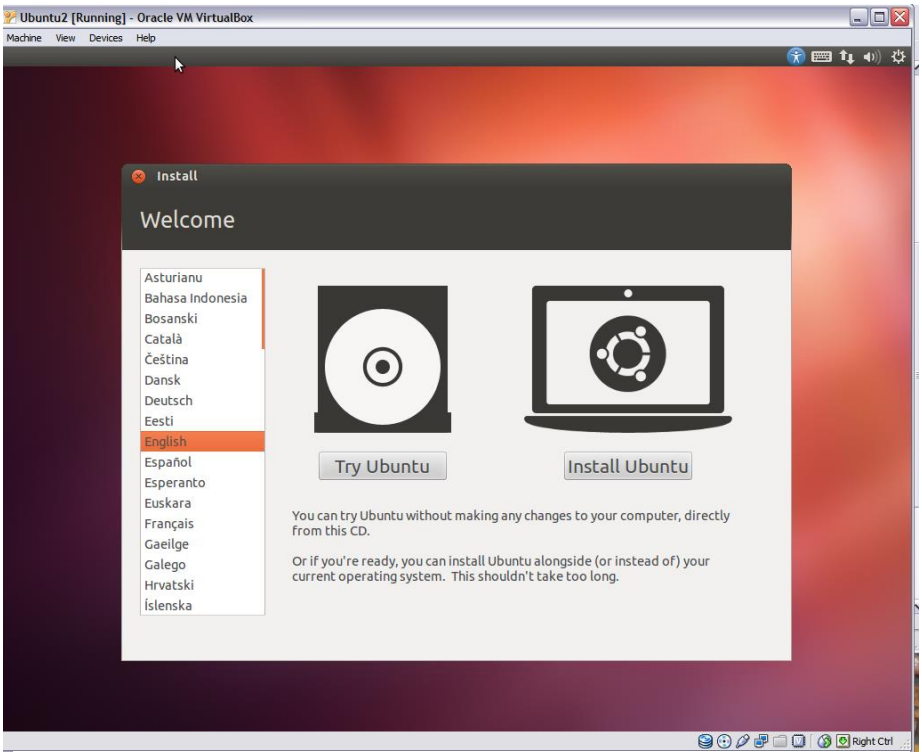

Figure 10: The Ubuntu Linux installation screen

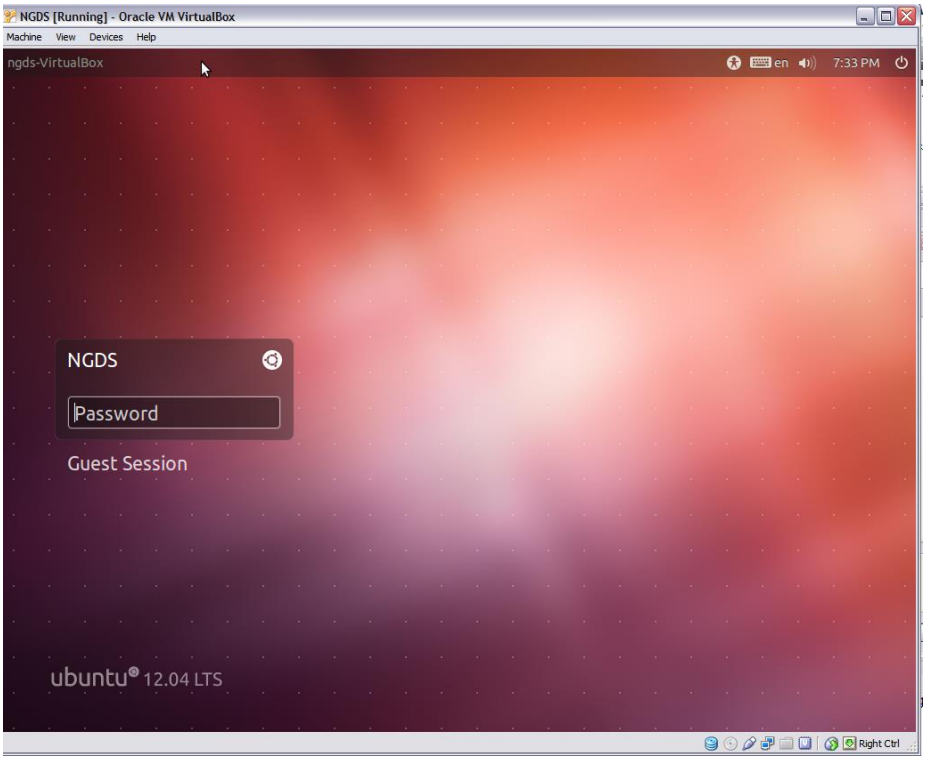

Figure 11: Logging on to Ubuntu Linux

A.1.7 Take a Snapshot

Take a Snapshot of your virtual machine before continuing. A Snapshot is a record of the virtual machine that can be used to restore it to its condition at the time the Snapshot was taken. Snapshots are typically used as precautions against failure at a later date.

A Snapshot can be taken via the **VirtualBox Manager** or from the **Machine** drop-down on the top left

A.2 Accommodating a corporate firewall (OPTIONAL)

If the computer you are using to host your virtual machine is behind a corporate firewall, your virtual machine may not have immediate Internet access. Internet connectivity is required in order to install NGDS Software Stackcomponents on your virtual machine (as will be discussed in Section 3).

A.2.1 Install and Configure CNTLM (OPTIONAL)

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

CNTLM is available at: http://cntlm.sourceforge.net/

After installing CNTLM on your host machine, use a text editor to modify the **cntlm.ini** file; here, specify the credentials your host machine uses to bypass your corporate firewall. An example appears in Table 1 below:

```
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 the example above, text strings preceded by a pound sign or hash symbol (#) are *comments* for the benefit of human operators; comments are not interpreted by any program reading the **cntlm.ini** file.

When configuring CNTLM, be sure to specify a localhost (**NoProxy**) entry with appropriate IP addresses and an appropriate port. The default CNTLM port is 3128. Asterisks (*) are *wildcard characters* which indicate the range of available possibilities for a given character – so 10.* can be 10.0, 10.1, or 10.2, all the way up to 10.9.

To use CNTLM, make sure CNTLM is running on your host machine whenever you run the virtual machine you created previously. If CNTLM is not running on the host machine, your virtual machine will be unable to establish an Internet connection.

CNTLM can be executed by command prompt or set to run as a Windows service. Starting CNTLM from a command prompt is useful within a development environment because doing so allows you to manually restart CNTLM in response to freezes or crashes.

A.2.2 Configure your virtual machine environment to use CNTLM as its proxy (OPTIONAL)

Log in to your virtual machine; navigate to the **etc** directory and use a text editor to manually edit the **environment** file. Add the proxies specified above to the **environment** file; an example appears below:

```
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"
```

Alternatively, you can use the Ubuntu Network Configuration application to manually specify the desired proxies (Figure 12).

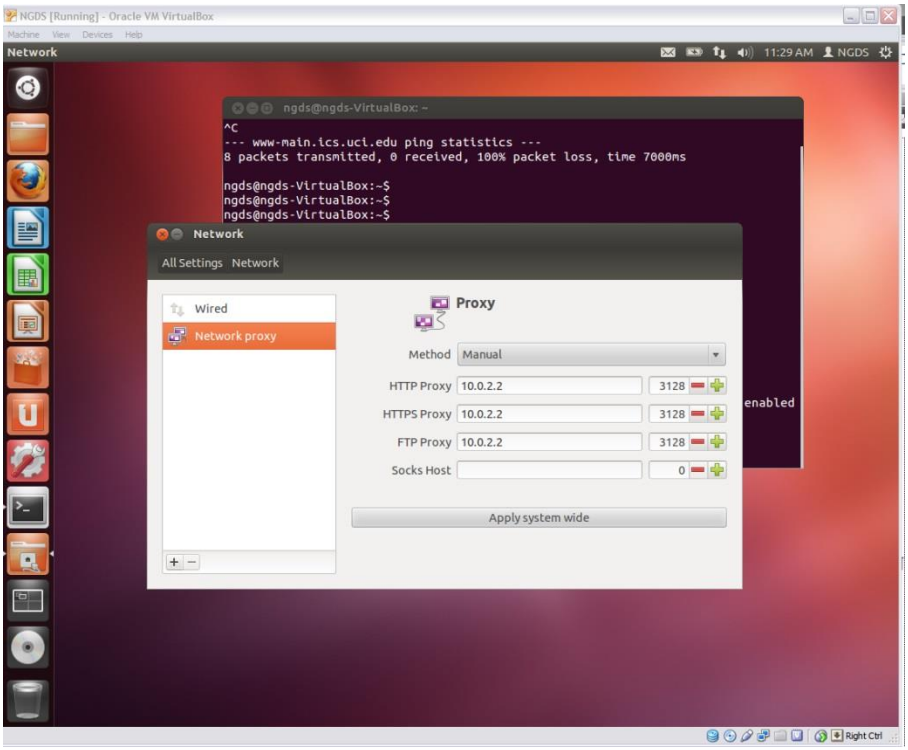

Figure 12: Configuring a proxy in Ubuntu Linux

A.2.3 What to do if cntlm and proxy continue to cause issues

1) If possible, finish the install on a virtual machine connected to the Internet instead of a local intranet. If this is not possible, you will need to configure your virtual machine's settings in such a way that you are able to use the **apt get** command; negotiating an intranet may require installation of CNTLM within your virtual machine, as well.
2) If CNTLM causes issues after you have successfully installed the software, but then when you try to open the web sites locally hosted and CNTLM then causes issues, establish port forwarding within your 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.

When Oracle VM Virtual Box install is complete, return to Section 3 of this document to continue installation of the NGDS node.

Appendix B The development.ini file

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.

Important note: **development.ini** is divided into various sections; section headers are demarcated by [brackets]. All CKAN parameters described in this document must be entered into the [app:main] section!

The code block below contains a summary of all changes that need to be made to the **development.ini** file during the NGDS CKAN Extension installation process.

```
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



  ckan.plugins = stats json_preview recline_preview datastore datastorer ngdsui
ckan_harvester metadata geoserver csw
```

B.1 Development.ini Considerations

- Plugins must be set *incrementally* during installation; they cannot be added all at once. Instead, plugins must be added *after* their dependencies have been added.
- You need to choose between the **datastorer** and **harvest** plugins depending on your CKAN installation:
    - When running NGDS in **node** mode (**ngds.deployment = node)**, you need the **datastorer** plugin
    - When running NGDS in **central** mode (**ngds.deployment = central)**, you need the **harvest** plugin
    - These two plugins *cannot* be installed in the same instance of the NGDS CKAN Extension!
- Trailing slashes on paths must be identical to those specified in the code block above.

Appendix C Architectural and Deployment Diagrams

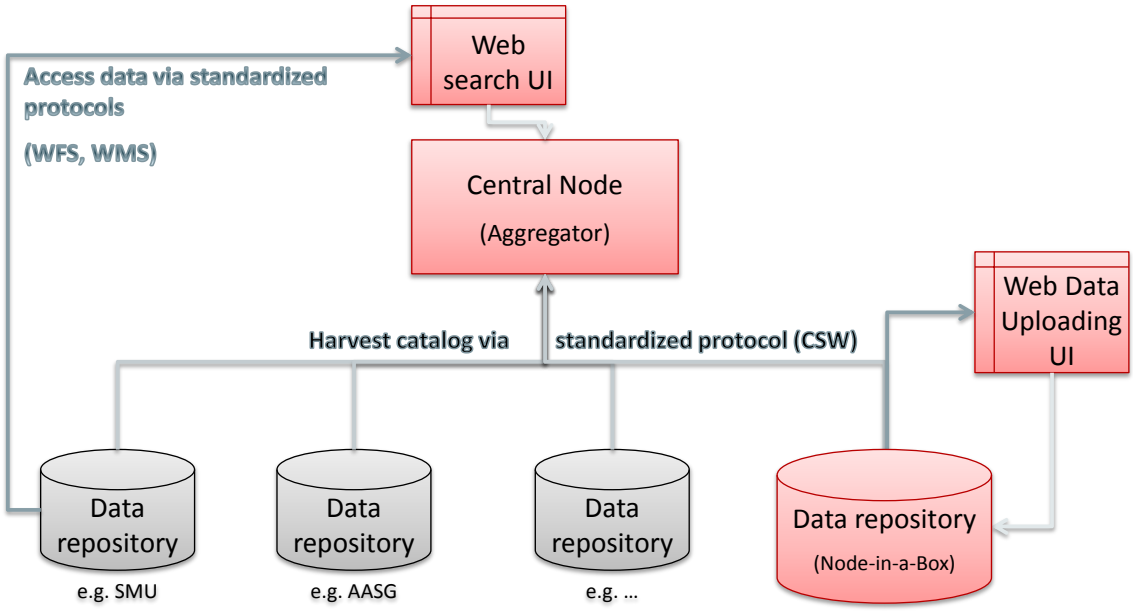

Figure 13: A diagram of NGDS

C.1 What is CKAN?

CKAN stands for **Comprehensive Knowledge Archive Network**.

CKAN is modular free-and-open-source data portal software. When properly installed on a server, CKAN provides a web-accessible interface by which users can submit and manage metadata records. The CKAN user interface also allows users to configure automated metadata harvesting from registered CKAN instances (an *instance* is a specific installation of the CKAN software); metadata harvested in this way is used to generate a web-accessible catalog. These traits are well-suited to the requirements of NGDS.

A CKAN *extension* is a user-generated modification of the CKAN software. The NGDS CKAN Extension is a CKAN extension designed to interact with NGDS data, metadata, and interchange formats. See Figure 17 for an overview of the components of CKAN as developed for use in NGDS.

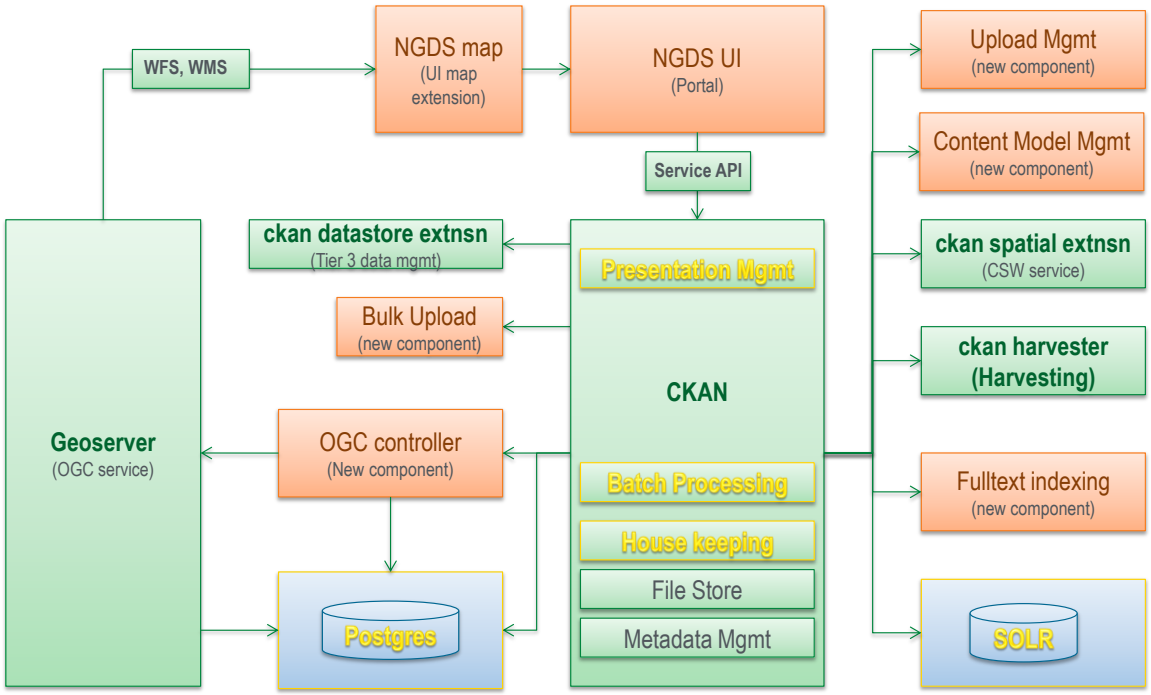

Figure 14: NGDS High-level Components

C.2 Domain Model

The Domain Model of NGDS can be represented as a class diagram (Figure 15). This shows the relationships of the separate entities that comprise the system; boxes on the left and bottom represent end users accessing the system, which results in discovering datasets, OGC-compliant web services, and other resources.

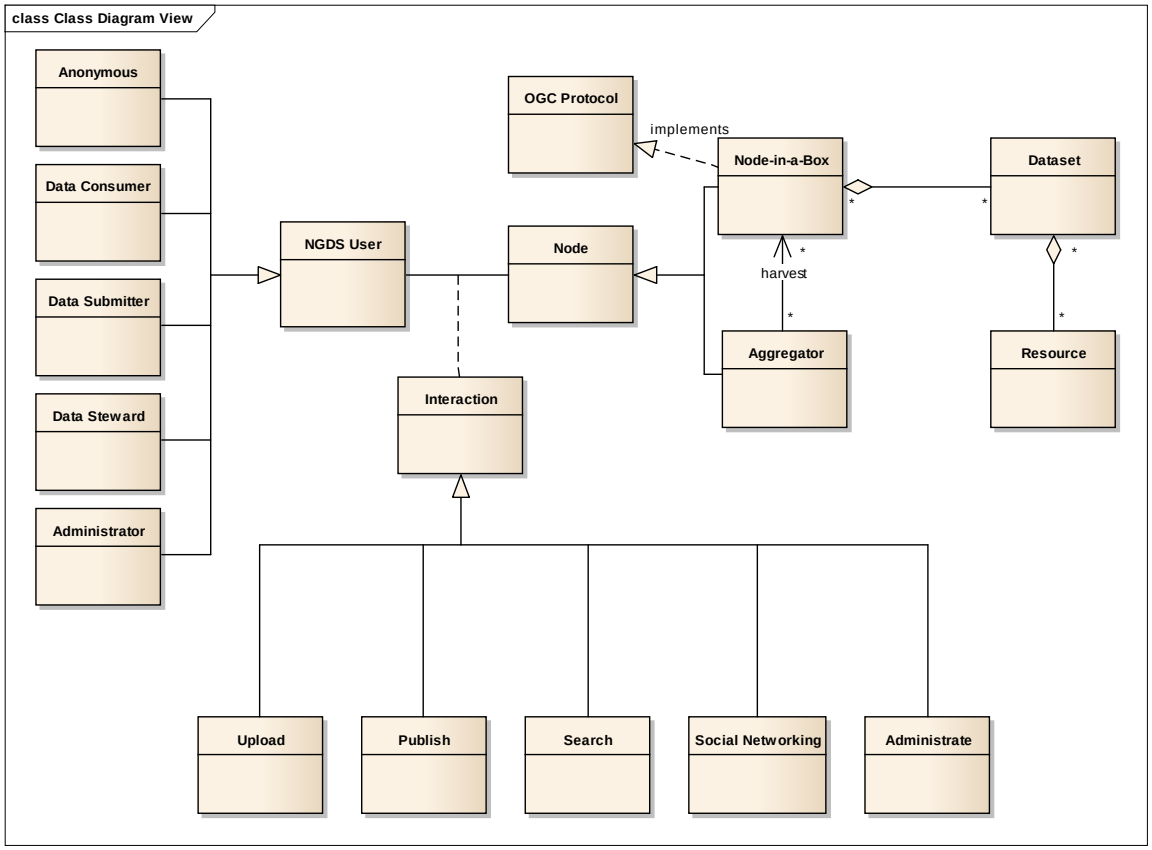

Figure 15: NGDS Domain Model as a Class Diagram

C.3 Additional Notes on CKAN in Production Mode

When running CKAN in **production** mode, consider the following:

- The celeryd runs as a service; you can control it with the following command:

  sudo service ngds-celeryd start|stop|restart|status

- If Tomcat needs to be started manually, do so with the following command:

  cd /opt/ngds/tomcat/bin; ./catalina.sh run

- Both SOLR and GeoServer are hosted by the same Tomcat instance in order to reduce the amount of resources needed to run the system. Configure the CATALINA_OPTS variable to provide more stackspace for Tomcat (the default values are too low for production mode).

```
%  cd /opt/ngds/tomcat/bin
%  cd nano catalina.sh
JAVA_OPTS=-Dfile.encoding=UTF-8 -server -Xms512m -Xmx2048m -XX:NewSize=256m -
XX:MaxNewSize=256m -XX:PermSize=256m -XX:MaxPermSize=512m -
XX:+DisableExplicitGC
```

- CKAN runs as wsgi job behind apache2; you can control it with the following command:

  sudo service apache2 start|stop|restart|status

- CKAN should not be run behind apache2. Instead, we recommend the following commands:

```
% sudo service apache2 stop
% cd /opt/ngds/bin/default
% . ./bin/activate
% cd ckan
% ln -s /opt/ngds/etc/ckan/default/production.ini ./development.ini
% paster serve development.ini
```