

National Geothermal Data System

Node-In-A-Box Software Installation Instructions

Arizona Geological Survey and Siemens Corp.

version 1.03; 3/26/2014



This work is licensed under a Creative Commons Attribution 3.0 Unported License.
Copyright © Arizona Geological Survey, 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- 07/24/2013	Minor updates, Combining comments from a few people, Added appendix with summary of development.ini changes, A little re-organization, more hints, and added gdal, Updating with feedback
0.7	Christoph Kuhmuench	12/26/2013	Updating to latest installer.
0.8	Jordan Matti	2/7/2014	Many changes
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)
1.3	Christy Caudill	2/21/2014- 3/26/2014	Formatting changes, minor edits and figure updates, Edits from test install with VM, Serious revisions, updating the installer script used, Incorporating changes per Matt MacKenzie, Update to installer script, edits per Matt MacKenzie. Added Section 5, added Section 6.

Contents

1.	Preface	2
1.1	Purpose and Audience	2
1.2	Document Roadmap	2
1.3	System Scope and Background.....	2
	The NGDS Software Stack.....	3
2.	Prerequisites	3
3.	Install the NGDS Software Stack	4
3.1	Install Git.....	5
3.2	Obtain the NGDS Software Stack Installation Files	5
3.3	Set Installation Parameters	5
3.4	Run the Installation Script	6
3.5	Final Steps	6
3.5.2.	Log in with the following credentials:	6
3.5.3.	Navigate to the following URL:	6
3.5.4.	Create a new organization; make the name of the organization:	7
3.5.5.	Navigate to the installed GeoServer:.....	7
4.	Troubleshooting your NGDS Installation	7
5.	Short tutorial on using a publisher node installation	7
5.1.	Tiers of NGDS data delivery	8
5.1.1.	Tier 1	8
5.1.2.	Tier 2	8

5.1.3.	Tier 3	8
5.2.	How to upload tier 1 and 2 data.....	8
5.3.	How to upload teir 3 data and publish web services	9
5.4.	How to upload tier 3 metadata	9
6.	Tips for using an aggregator node installation	9
Appendix A Installing a Virtual Machine.....		11
A.1	Creating an Ubuntu Linux Virtual Machine using VirtualBox ..	11
A.1.1	Download and install Oracle VM VirtualBox Manager.....	11
A.1.2	Create an Ubuntu Linux Virtual Machine	11
A.1.3	Configure your Virtual Machine.....	15
A.1.4	Download an Ubuntu ISO image	16
A.1.5	Mount the Linux installation .ISO file in your virtual machine	16
A.1.6	Install Ubuntu Linux 12.04	17
A.1.7	Take a Snapshot	18
A.2	Accommodating a corporate firewall (OPTIONAL).....	19
A.2.1	Install and Configure CNTLM (OPTIONAL)	19
A.2.2	Configure your VM to use CNTLM as its proxy (OPTIONAL)	20
A.2.3	What to do if cntlm and proxy continue to cause issues	21
Appendix B Architectural and Deployment Diagrams		22
B.1	What is CKAN?.....	22
B.2	Domain Model	23
B.3	Additional Notes on CKAN in Production Mode.....	24

List of Figures

Figure 1: NGDS Software Stack in Production Mode	4
Figure 2: Create a new Linux virtual machine	12
Figure 3: Create a virtual hard disk.....	13
Figure 4: Specify the image type.....	13
Figure 5: Specify storage allocation	14
Figure 6: Configure virtual hard drive.....	14
Figure 7: Configuring a virtual machine in VirtualBox.....	15
Figure 8: Enabling the shared clipboard.....	16
Figure 9: Mounting the Ubuntu ISO image in the VM.....	17
Figure 10: The Ubuntu Linux installation screen	18
Figure 11: Configuring a proxy in Ubuntu Linux.....	20
Figure 12: A diagram of NGDS	22
Figure 13: NGDS High-level Components	23
Figure 14: NGDS Domain Model as a Class Diagram.....	24

1. Preface

National Geothermal Data System (NGDS) was initiated as 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 includes:

- A description of NGDS and how it works
- Description and enumeration of the components that are necessary to install the NGDS Software Stack
- Step-by-step installation instructions for the NGDS Software Stack on an Ubuntu Linux operating system
- A comparison of installing the NGDS Software Stack in **production** mode, as opposed to **development** mode

One of the goals of the NGDS is to provide an open-source software stack for releasing open data on the World Wide Web project that is sustainable and a cost-effective option for data producers. With this documentation, system administrators will be able to quickly understand the system and deploy a productive node in the NGDS.

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: NGDS architecture and diagrams and notes

1.3 System Scope and Background

The National Geothermal Data System (NGDS) is a distributed data-sharing network. NGDS data providers (publishers) host data using their own computing resources and present web-accessible metadata describing their data holdings for harvest by aggregating catalog nodes (aggregators) in the data network.

Metadata presented by registered **publisher** nodes is regularly harvested into NGDS **aggregator** catalogs. The aggregator node(s) host web sites from which users can search the aggregated metadata catalog for datasets, documents and services. Thus, the aggregator node becomes the one-stop search interface for the entirety of the system. The central NGDS aggregator node can be accessed at <http://geothermaldata.org>.

The NGDS Software Stack

The **NGDS Software Stack** is a collection of applications that support release of data for the NGDS, creation and publication of metadata records, and search of the metadata catalog hosted by an NGDS node. 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 create and manage metadata records. Metadata held by 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 that can be configured to harvest metadata from NGDS publisher nodes. An NGDS aggregator node harvests metadata from any NGDS publisher node or Open Geospatial Consortium Catalog Service for the Web (CSW) with metadata that conforms to the USGIN ISO19115 profile. This same installation can produce either a publisher or and aggregator, but this document focuses on the installation of a publishing node.

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

- **Production** mode: a deployment to support day to day operation with minimal disruption and maximal performance.
- **Development** mode: used by software developers to update software components in the stack; generally installed in a development framework that enables debugging at the cost of performance.

2. Prerequisites

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

- Internet 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 the virtual machine running the NGDS software.

The continuation of this document describes the steps necessary to install the NGDS Software Stack as a **publisher** node or an **aggregator** 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:

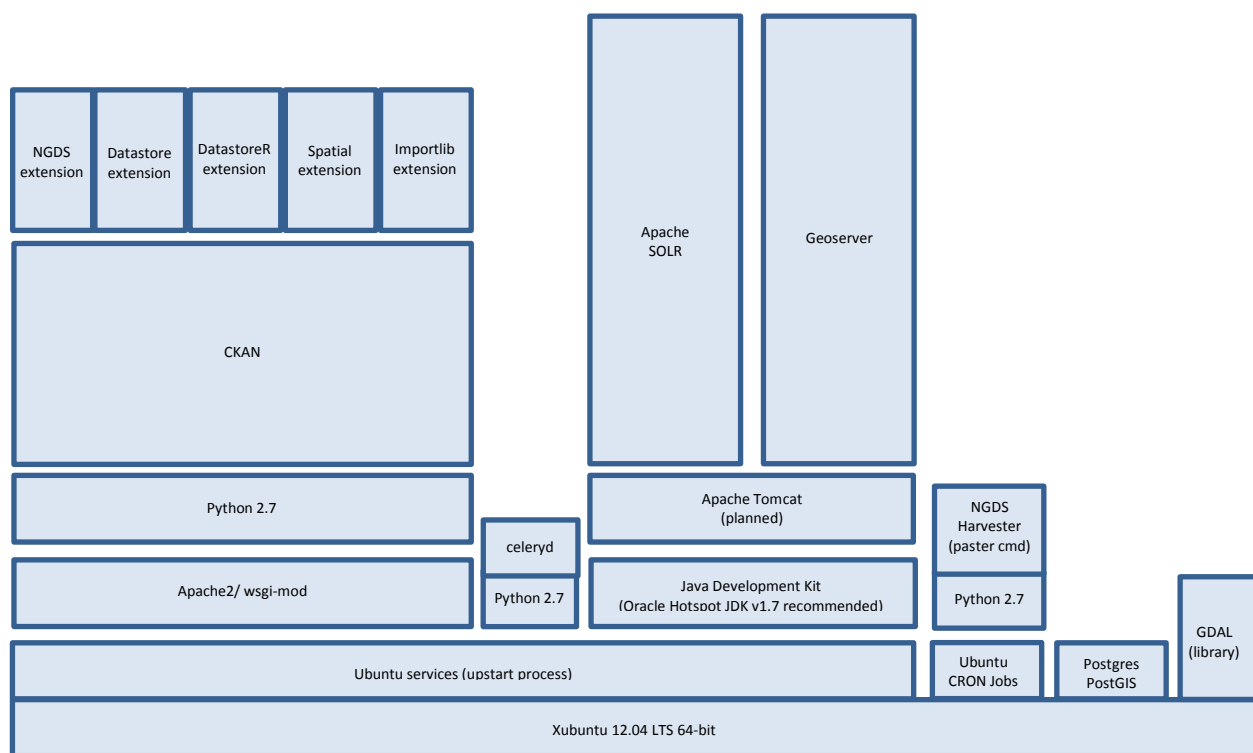


Figure 1: NGDS Software Stack in Production Mode

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 parameters 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 *node* or *central* mode:
 - *node* (default) will set up the installation as a **publisher** node
 - *central* will set up the installation as an **aggregator** node
- `site_url`='http://myservername_IPname'
- `SERVER_NAME` = Should be the same as the 'site_url' (ex. http://127.0.0.1)
- `SMTP_SERVER` = A server that receives outgoing email messages and routes them to recipients (ex. smtp.gmail.com:587)
- `SMTP_STARTTLS` = An extension which upgrades a plain text connection to an encrypted one (ex. True)
- `SMTP_USER` = An email address that will send automated emails from CKAN, through the SMTP server (ex. email@gmail.com)
- `SMTP_PASSWORD` = The password associated with the above email address
- `GEOSERVER_REST_URL` = Which contains the connection parameters for Geoserver, in the form:
"geoserver://{username}:{password}@{geoserver_rest_api_url}"

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 directory 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", "geoserver://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", "geoserver://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

Change the master and administrator passwords in GeoServer for security:

3.5.5. Navigate to the installed GeoServer:

- <http://127.0.0.1:8080/geoserver/web/>
- Log in with the following credentials:
 - Username: admin
 - Password: geoserver
- Follow the **Change it** instructions on the home page indicated by the  icon

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. You can also try running the following command:

```
% sudo a2dissite default
```

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.

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

5. Short tutorial on using a publisher node installation

This section is intended to be a brief introduction to uploading data to a new installation of a **publisher** node, 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 retrieve 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 conforming to the standards of a given information exchange 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 tier 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; this will enter automatically or prompt to enter to associated layer name (again, check <http://schemas.usgin.org/models/> if unsure which layer name to select). 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**.

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.

6. Tips for using an aggregator node installation

As the publisher node is used to upload and manage individual resources, the aggregator node is used as a harvester. As such, it has different functionality in the **Contribute** page than does a publisher node. Here, the UI is used to manage those harvest sources. The following are a few helpful hints for using the version 1 aggregator node as a harvester.

- Version 1 of this installation will support harvesting from:
 - an OGC CSW endpoint with ISO-standards-compliant metadata (<http://usgin.org/specifications>; <http://www.opengeospatial.org/standards/cat>)
 - publisher nodes using the CKAN Harvester (ckan_harvester plugin, which is CKAN's API for harvesting between CKAN instances)
- Harvest using the *NGDS CSW Harvester*

-
- To harvest a CSW, use only the string before the '?' in the URL, for example:

<http://catalog.usgin.org/geothermal/csw>

- Version 1 of this installation does not support implementation of a CSW endpoint from an aggregator node

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

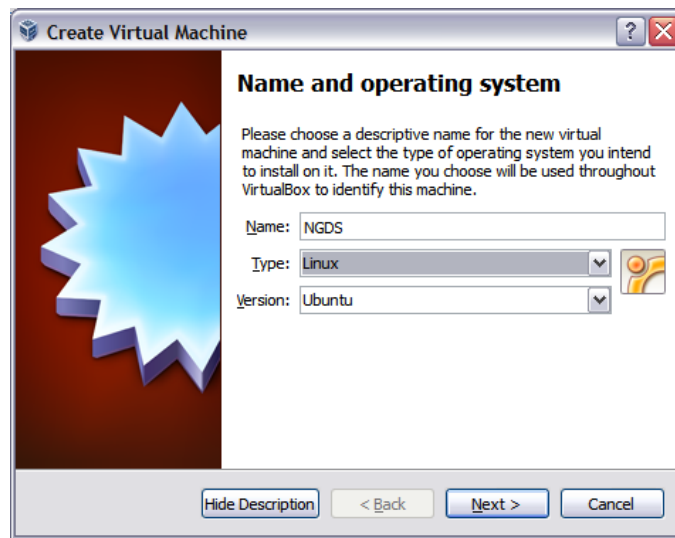


Figure 2: Create a new Linux virtual machine

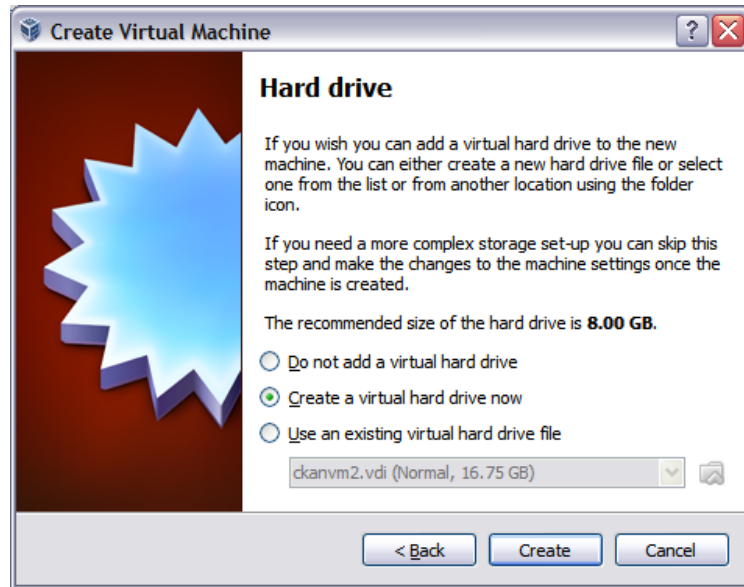


Figure 3: Create a virtual hard disk

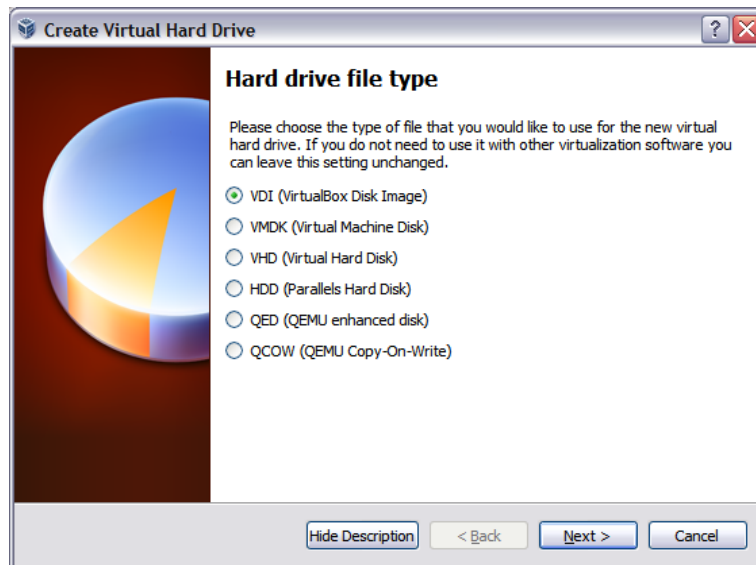


Figure 4: Specify the image type

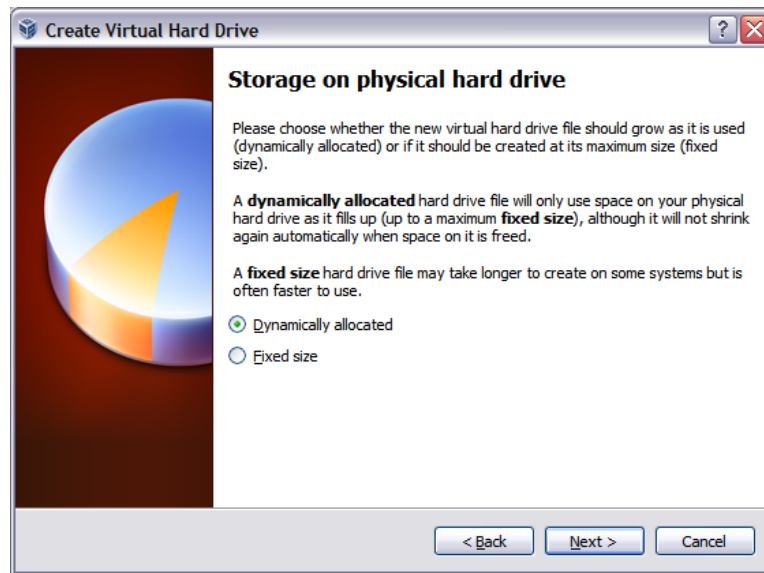


Figure 5: Specify storage allocation

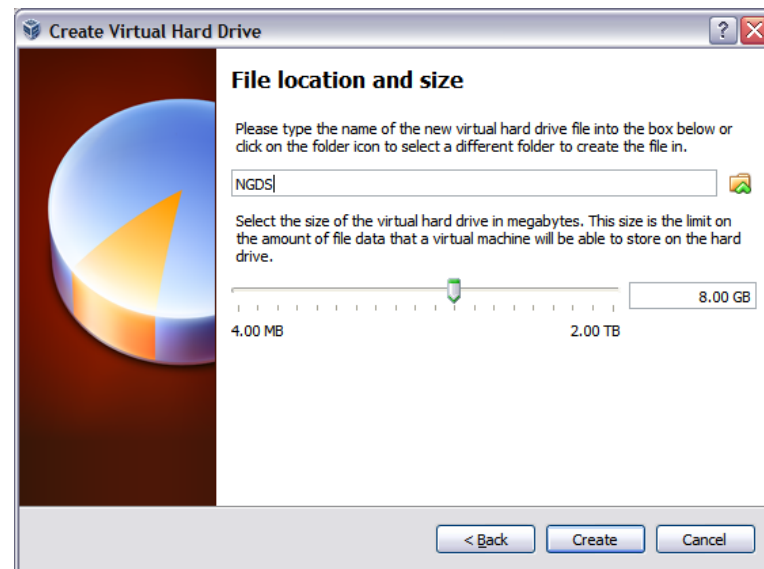


Figure 6: Configure virtual hard drive

A.1.3 Configure your Virtual Machine

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

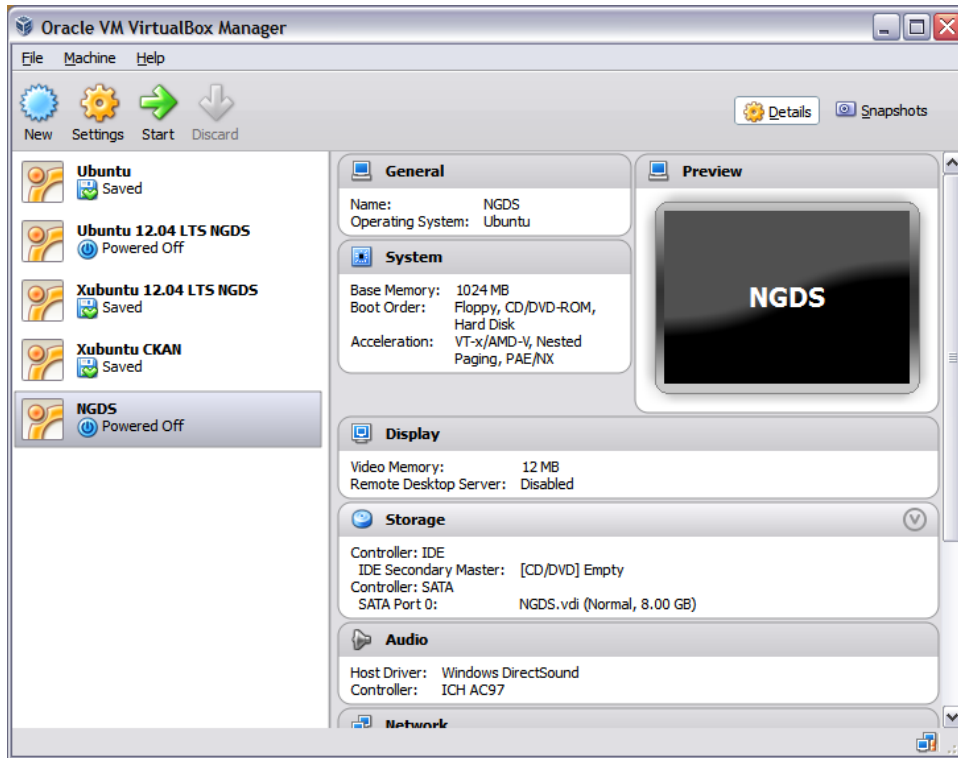


Figure 7: Configuring a virtual machine in VirtualBox

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.

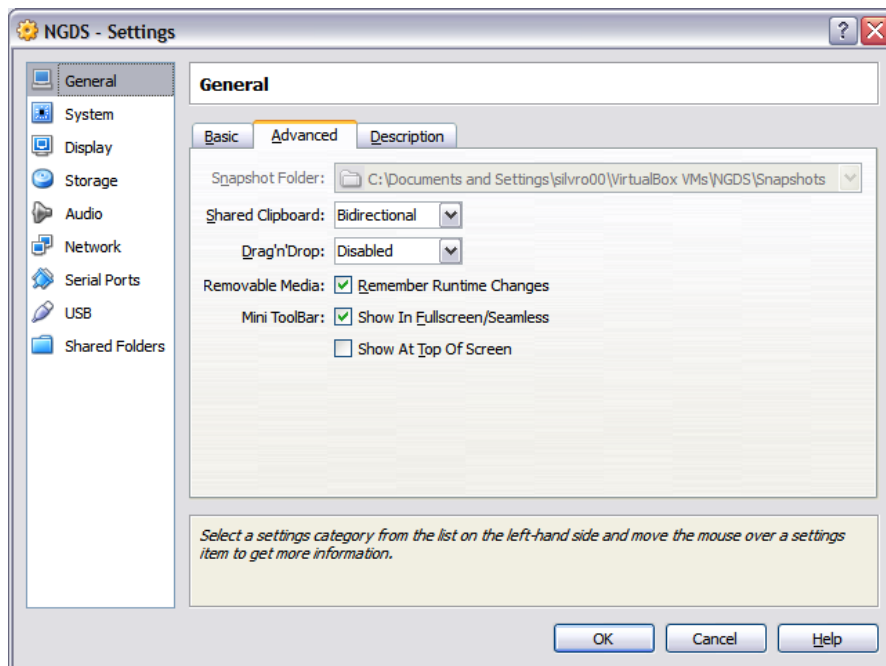


Figure 8: Enabling the shared clipboard

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

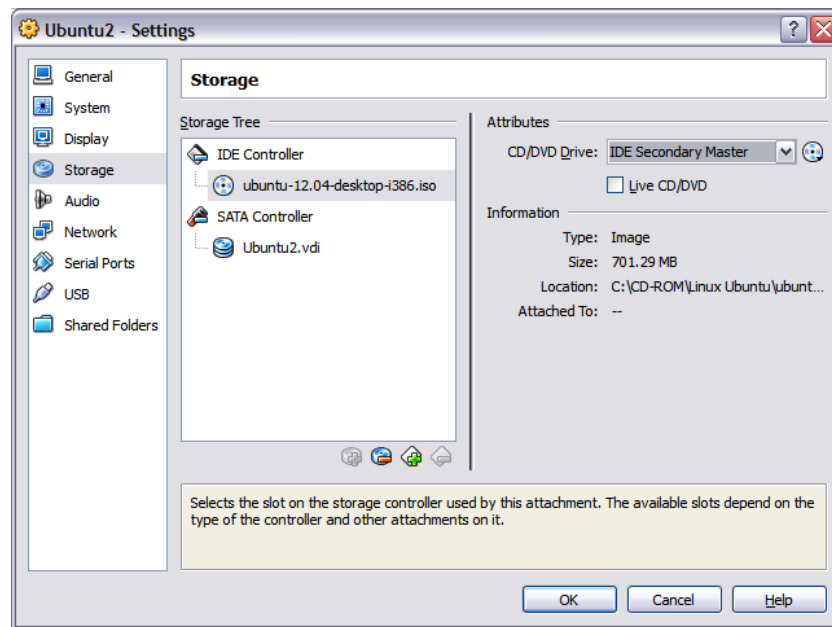


Figure 9: Mounting the Ubuntu ISO image in the VM

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).

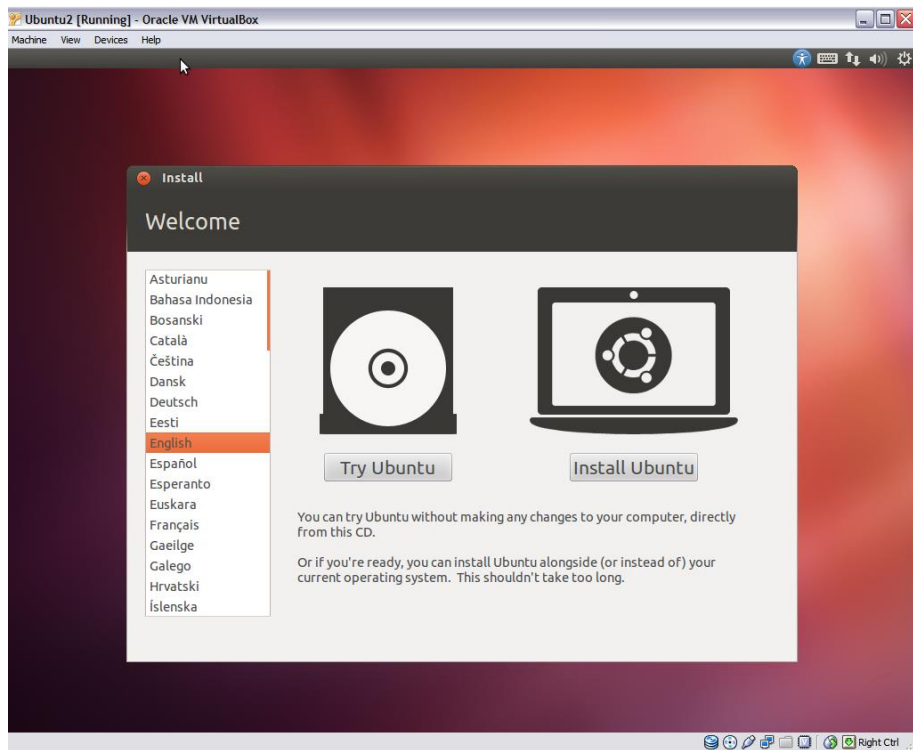


Figure 10: The Ubuntu Linux installation screen

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.

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.

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 Stack components 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 11).

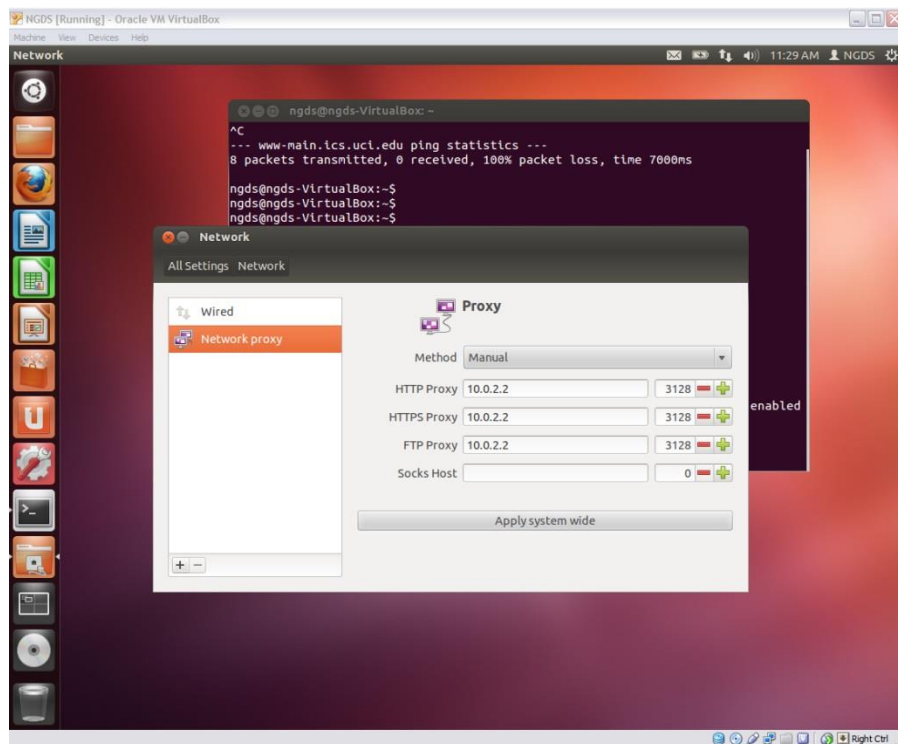


Figure 11: Configuring a proxy in Ubuntu Linux

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 Architectural and Deployment Diagrams

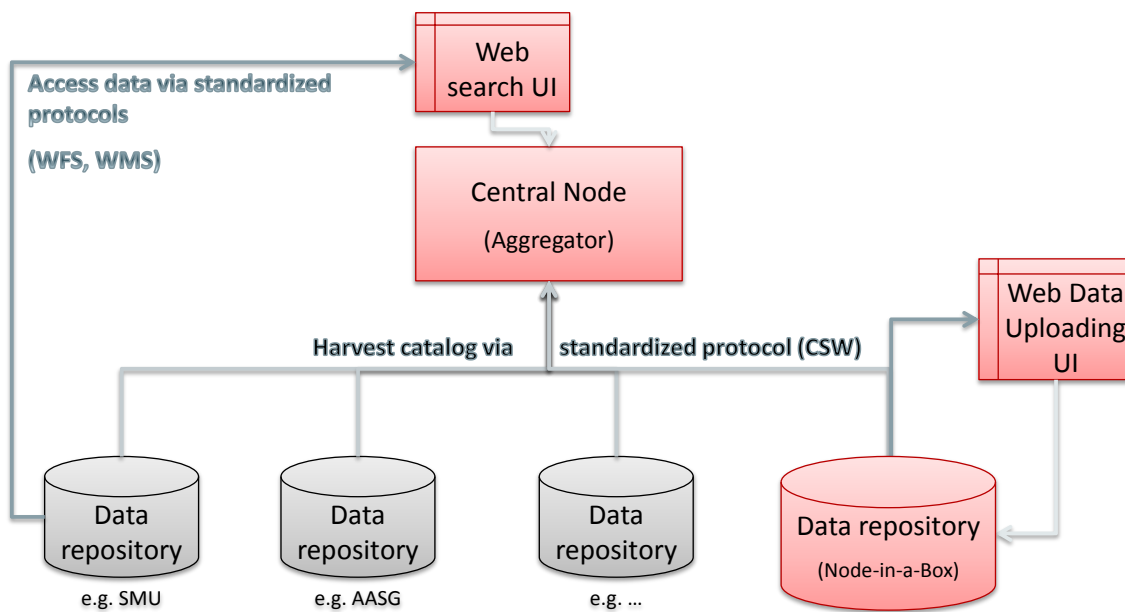


Figure 12: A diagram of NGDS

B.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 13 for an overview of the components of CKAN as developed for use in NGDS.

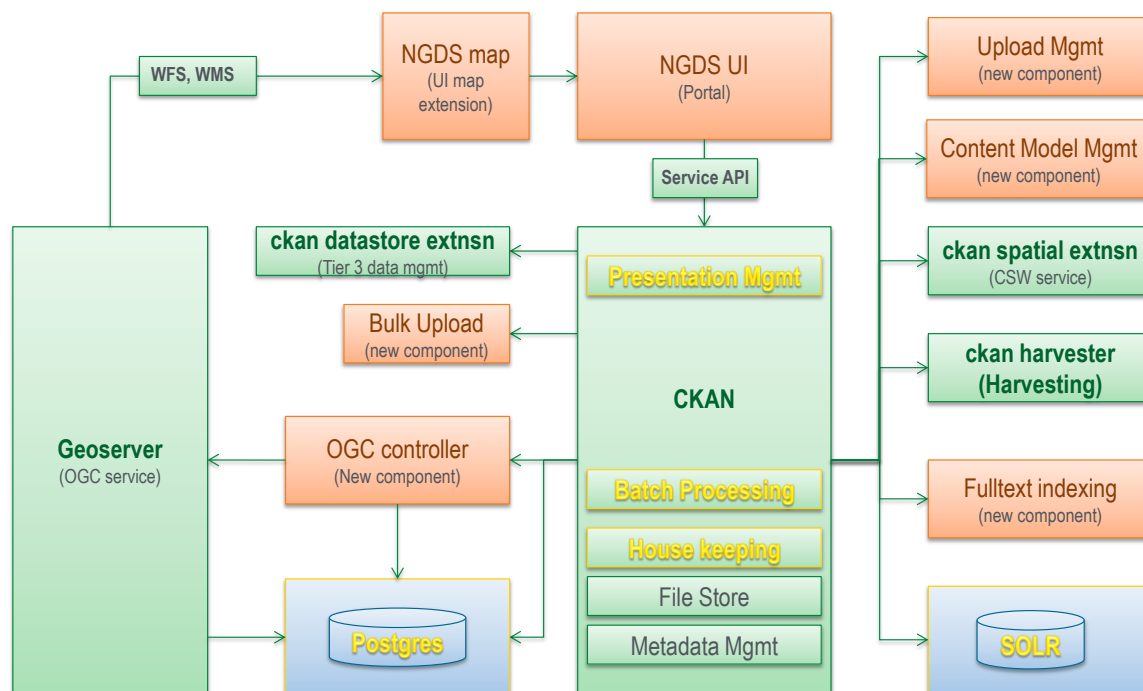


Figure 13: NGDS High-level Components

B.2 Domain Model

The Domain Model of NGDS can be represented as a class diagram (Figure 14). 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.

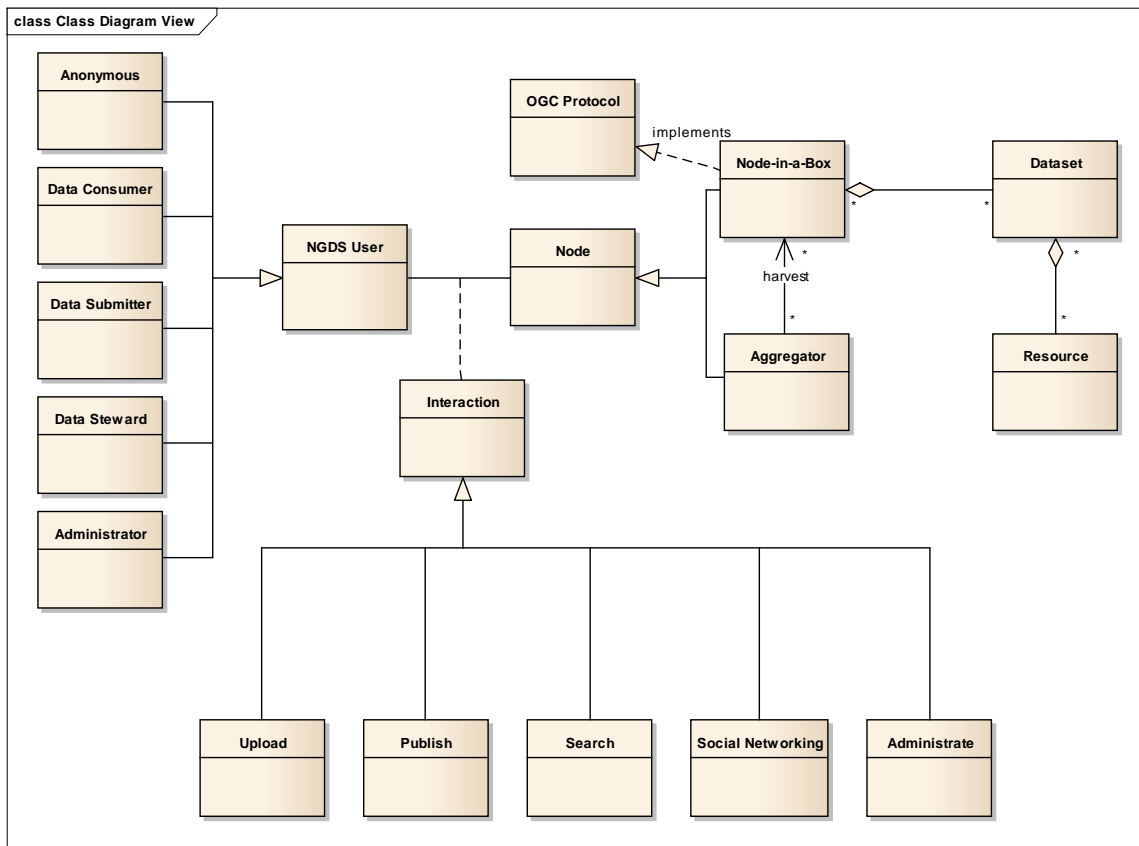


Figure 14: NGDS Domain Model as a Class Diagram

B.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`
- The log file for CKAN is in the following location:
`/var/log/apache2/`

-
- Source code is installed at the following location:

`/opt/ngds/bin/default/`

- 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
```
