

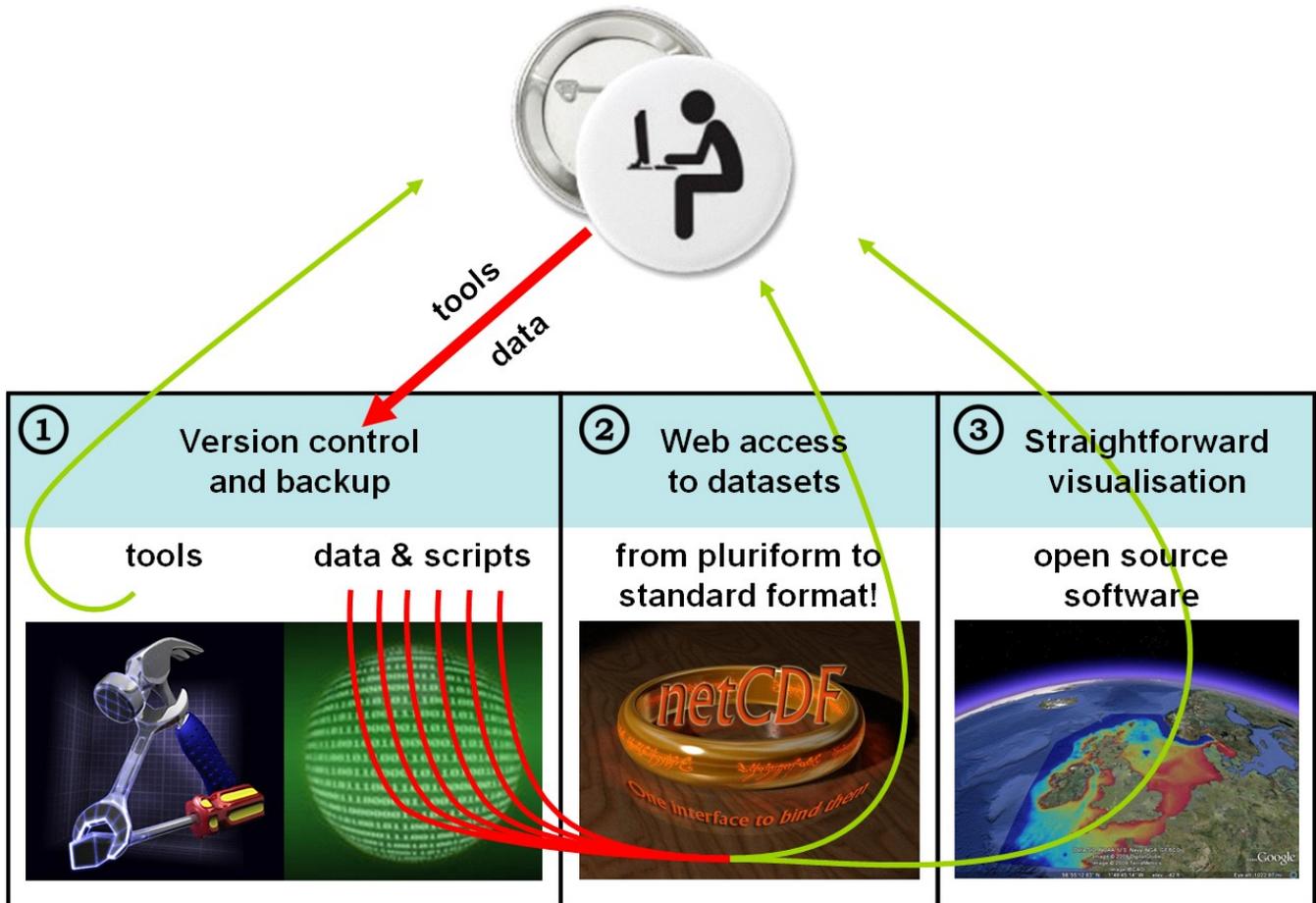
OpenEarth infrastructure - Step by step installation guide

Contents

- 1 Create a virtual machine
- 2 Install LINUX operating system
- 3 Install Subversion and create repositories
- 4 Install Tomcat and Thredds (OpenDap server for netCDF files)
- 5 Install Apache webserver (webserver for kml files)
- 6 Install OpenSSH to enable use of WinSCP
- 7 Overview of useful Linux commands
- 8 Useful links

Acknowledgements

- This OpenEarth Installation Protocol is developed by the Data and Knowledge Management section of the [Building with Nature programme](#) as a **Deliverable of Workpackage DM 1.1**



The OpenEarth infrastructure supports a bottom-up approach for long-term project-transcending collaboration adhering to four basic criteria:

- **open standards & open source** - The infrastructure should be based on open standards, not require non-libre software and be transferable
- **complete transparency** - The collected data should be reproducible, unambiguous and self descriptive. Tools and models should be open source, well documented and tested
- **centralized access** - The collection and dissemination procedure for data, models and tools should be web-based and centralized to promote collaboration and impede divergence
- **clear ownership and responsibility** - Although data, models and tools are collected and disseminated via the OpenEarth infrastructure, the responsibility for the quality of each product remains at its original owner.

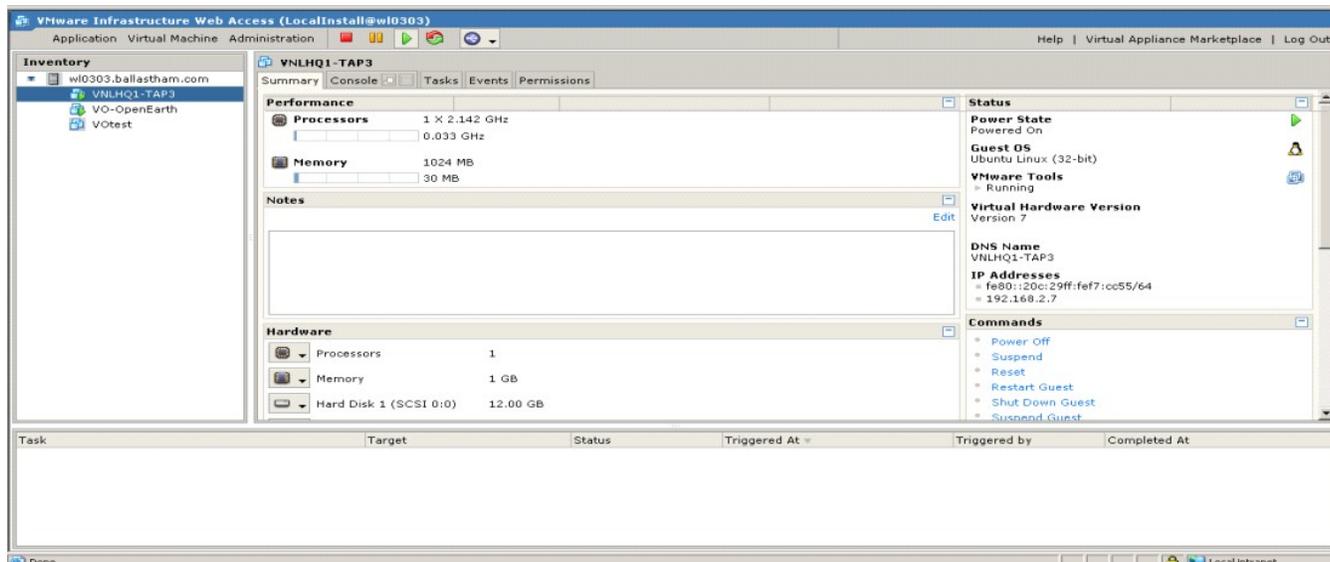
To make the OpenEarth infrastructure work, the following ICT components are required in practice(see figure):

1. a linux server (either virtual or real),

2. a subversion server to enable version control, backup and controlled access,
3. a tomcat6 server running an OPeNDAP protocol to provide netCDF files,
4. an apache2 webserver to provide data visualisation files like *.kml for visualisation in Google Earth, and
5. the OpenSSH protocol to facilitate remote access to the server (for maintenance purposes).

The remainder of this tutorial addresses the steps needed to set up this infrastructure.

Create a virtual machine



Register at VMWARE and download "VMWARE Server" at <https://www.vmware.com/>. Make sure you write down the Windows code as it is needed to install VMWare Server. Once VMWare Server is installed (login with administrator rights if needed) open it and create a virtual machine:

- Use your inlog account with administrator rights when logging into VMWare
- Give name **[next]**
- Select Linux as operating system and choose as version: Ubuntu Linux (32 bits) **[next]**
- Select memory and number of processors to use
 - Memory 512 MB (default)
 - 1 processor (default) **[next]**
- Select hard disk option
 - Now select "Create a New Virtual Disk" this will be replaced by another disk later
 - select a storage capacity (e.g. 100 GB - if remaining disk space is < 100 GB VMWare Server will automatically change the value to the maximum remaining disk space). **TIP:** The selected size will be allocated only actual claim on disk space depends on actual usage. If the data is stored on the virtual server (possibly useful in small scale temporary applications) 100 GB should be enough. If the data is stored on a mounted external drive disk space is only needed for installed services and occasional logfiles (we estimate that 20 GB should be enough in that case)
 - leave rest of the properties at their default values **[next]**
- Add a network adaptor
 - Select "Network Connection" option: "Bridged" (this arranges that the virtual machine has the same internet settings as its 'real' host **[next]**)
- Do connect a CDROM/DVD drive (select defaults) **[next]** **TIP:** If this step presents problems a possible workaround is to make a map ISO in "d:\Virtual Machines", copy and select the UBUNTU ISO image there (see also the guidance in the next section). The virtual CDROM/DVD may require this ISO to proceed to the next steps before finalising the virtual machine setup.
- Do NOT connect floppy **[next]**
- Do connect USB (select defaults) **[next]**

Install LINUX operating system

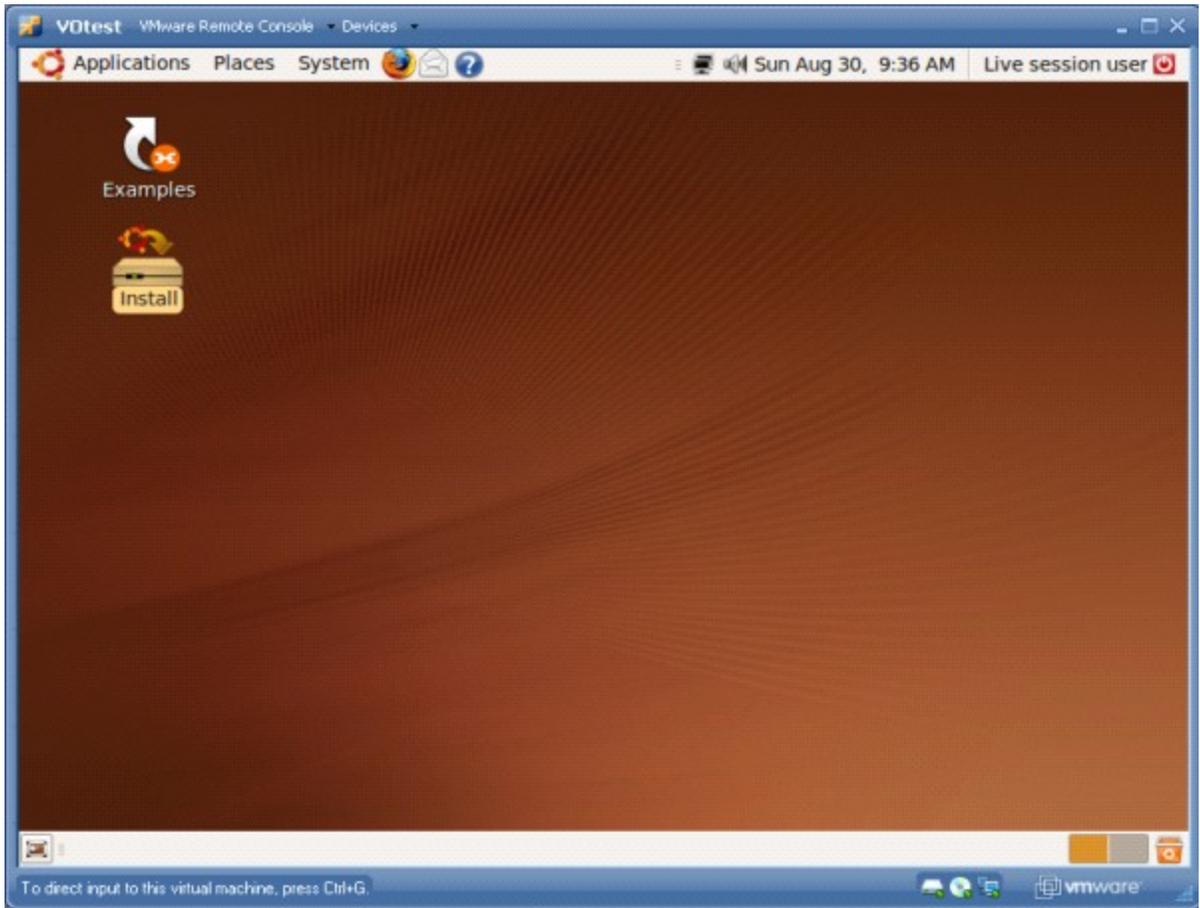
Download

ubuntu Desktop version 9.04 from <http://www.ubuntu.com/GetUbuntu/download> and install it (the download is an ISO-file). Make a map ISO in "d:\Virtual Machines" and copy the UBUNTU ISO image there. In VM Ware select the virtual server you created and select in the hardware list the CDROM/DVD option. Choose edit. Under the section "ISO Image" choose browse and browse to the UBUNTU ISO-image ("standard" then "ISO" then select "Ubuntu") you just copied. Now go to the console and start up the virtual machine (click the screen as indicated, start the machine by pressing the big play button that appears). Now VMWare will open with the virtual machine with the Ubuntu desktop. Select the Install icon. **TIP:** Your computer may require installation of a remote console plugin first.

Now the Ubuntu operating system will install (select all settings to your liking: keyboardtype, timezone etc.). After it is finished you need to reboot the virtual machine. **TIP:** UBUNTU will request to do this automatically, otherwise use the powerbutton in the upper right corner of your virtual desktop. If your virtual machine does not start up properly a possible cause could be that the ISO image is still selected in the CDROM/DVD drive (see previous steps). Unselecting it is likely to fix this problem. The virtual machine will indicate that the virtual CDROM drive is locked (?!). You can select to override. You will be requested to create a useraccount and a password, select something useful like 'admin-oet' or so. **TIP:** write down your username and password, you will need them later!

Once the virtual machine is created select it:

- select in the summary tab to install the VMWare Tools
- the VMWare Tools installation CD is now inserted in the virtual CD Rom drive. It will popup automatically on the desktop of your virtual machine ! Drag the *.tar.gz files to the desktop and extract them (right mouseclick, extract here).



extract here). Open a terminal and install the tools by typing the following command:

```
cd Desktop/vmware-tools-distrib/
sudo ./vmware-install.pl
```

TIP: Select for all options the indicated default value (NB: some steps take some time to complete ... *be patient!*). When prompted for display size option [12] is recommended (1024x768).

Ubuntu will suggest to install security updates. You can install them if you want to (if your internet connection is slow maybe opt to do this later).

Install Subversion and create repositories

To setup the subversion service on your virtual machine, first install subversion using the synaptic package manager (see the section Overview of useful commands for tips on locating the synaptic package manager). You may need to reload for subversion to appear in the package list. Select "subversion" and "subversion-tools" and press 'Apply' to actually install these packages. Once subversion package is installed the instructions on

<https://help.ubuntu.com/community/Subversion>

can be followed to set up subversion repositories.

A user 'subversion' needs to be added by typing the following command in a terminal window:

```
sudo adduser subversion
```

A directory can be created where new repositories can be put:

```
sudo mkdir /home/svn
cd /home/svn
```

Tip: you will probably want to select a more appropriate directory for actual applications (something like `/var/data/repos/`). The appropriate location depends on a companies local preferences.

A new repository is created by issuing the following commands:

```
sudo mkdir myproject
sudo chown -R www-data:subversion myproject
sudo chmod -R g+rws myproject
sudo svnadmin create /home/svn/myproject
sudo chmod -R g+rws myproject
```

In each repository directory uncomment the password keyword in the svn config file "svnserve.conf" and add usernames and passwords to the passwordfile "passwd". Set the authentication settings to your preference: e.g. anon-access = none. **Tip:** Use 'sudo gedit' to start an editor program. You need to run gedit as sudo to allow filesaving.

To start the svnservice and make it remotely accessible start the following svnserve with the following command:

```
sudo svnserve -d --foreground -r /home/svn
```

or

```
sudo -u www-data svnserve -d --foreground -r /home/svn
```

NB: running svnserve in the foreground the cursor will not reappear (until the service is stopped with ctrl-z) so don't wait for it to reappear. If you want to execute another command start a new terminal window. If no error messages appear subversion is running and you can now remotely access the repositories, e.g. with a program like TortoiseSVN.

Besides starting svnserve manually and running it on the foreground, it is also possible to make svnserve start automatically on bootup running it in the background. Follow the instructions on:

<http://odyniec.net/articles/ubuntu-subversion-server/>

under the section "Svnserve Initialisation Script". It basically comes down to copying the info from:

<http://odyniec.net/articles/ubuntu-subversion-server/svnserve>

to a file (e.g. using gedit) and saving it as "/etc/init.d/svnserve". Make sure the info in svnserve is edited to include the directory you selected for the location of repositories. The script is installed by running:

```
sudo update-rc.d svnserve defaults
```

Indicate your repository locations (e.g. `/var/data/repos/`) in the svnserve script:

```
DAEMON_ARGS="-d -r /var/data/repos"
```

To make sure the svnserve script is executed on bootup it needs to be made executable by:

```
sudo chmod +x /etc/init.d/svnserve
```

Subversion should now work and you should be able to commit your first files. For tips see the [five easy steps to becoming an OpenEarth developer](#). The location of your repository will be `svn://[ip-address][repositoryname]`. You can find the IP address of your virtual machine under the summary tab of VMWare. The repository name you selected yourself in one of the previous steps.

Install Tomcat and Thredds (OpenDap server for netCDF files)

With the Synaptic package manager install Tomcat6 (identify all Tomcat6 packages by inserting "Tomcat6*" in the search box). Select them and press Apply to install. Once the packages are installed you can start the Tomcat6 server by issuing the following command:

```
sudo /etc/init.d/tomcat6 start
```

or

```
sudo /etc/init.d/tomcat6 restart
```

Under `/var/lib/tomcat6/` create a directory `content` and set the proper groups and rights:

```
cd /var/lib/tomcat6/  
sudo mkdir content  
sudo chown tomcat6:tomcat6 content  
  
sudo /etc/init.d/tomcat6 restart
```

Now that your Tomcat6 service is running you can install Thredds via a webinterface. However, Thredds will want to write logfiles. The standard security settings on your virtual machine are too strict. An adjusted `.policy` file should be added in `/etc/tomcat6/policy.d/` with the following content:

```
grant codeBase "file:${catalina.base}/-" {  
  permission java.security.AllPermission;  
};
```

Name it either `00openpolicy.policy` or `51openpolicy.policy` so that it is run either first or last (this is indicated by the two digits in the name). Apparently this works. NB: A number that is not first or last still results in problems.

Now [download Thredds](#) and mount the file `thredds.war` file (NB: file should have this exact name) by accessing `http://[ip-number]:8080/manager/html`, where `[ip-number]` is the ip-number of your virtual server. Once you deployed the war file a new option will appear in the Tomcat6 management webpage. Select `'start'` to start the Thredds service.

Edit in `/var/lib/tomcat6/content/thredds/` the file `tomcat-users.xml`. Create a role `admin` and `manager` in the `config.xml` and create a user (e.g. `admin`) and assign it both roles. For more detail see:

<https://help.ubuntu.com/8.10/serverguide/C/tomcat.html>

Now you can configure the data to be served with THREDDDS. There are some manuals on the web ([Signell (USGS)][http://rsignell.tiddlyspot.com/#\[A%20method%20for%20setting%20up%20TDS%20catalogs%20for%20the%20first%20time\]](http://rsignell.tiddlyspot.com/#[A%20method%20for%20setting%20up%20TDS%20catalogs%20for%20the%20first%20time])], [Unidata](#)). Edit in `/var/lib/tomcat6/content/thredds/` the file `catalog.xml`. Place data at:

```
/var/data/
```

Set `datasetRoot`, `datasetScan` to the proper data location and indicate the names you wish to appear on the Thredds server. NB: no spaces are allowed in directory names. Change the wildcard to `"*.nc"`.

For restricted datasets (make internal data available for remote access for your employees only) configure tomcat accordingly: <http://www.unidata.ucar.edu/projects/THREDDDS/tech/reference/RestrictedAccess.html>. We advice to restrict access based on the http rather than based on for instance using ownership properties of the data on the underlying (linux) file server. Note that for remote access the netCDF java library cannot handle http authentication /encryption itself ([<http://user:password@opendap.institution.nl>]...).

Every now and then THREDDDS needs to be updated. Check the [Unidata THREDDDS page](#) for posts. Future releases can be anticipated in the THREDDDS issue tracking system <https://www.unidata.ucar.edu/jira/browse/TDS>. To deploy a new THREDDDS with only a few minutes downtime, do the following. Download the new war `thredds.war` (release 4.2.10 in this example) and prepare it ready for instant copying to the Tomcat webapps directory `/var/lib/tomcat6/content/thredds/`. Please read this [check-list](#) for upgrading.

```
sudo rename thredds.war thredds.war.4.2.9  
sudo cp thredds.war.4.2.10 /var/lib/tomcat6/webapps/  
cd /var/lib/tomcat6/content/webapps/  
cp thredds.war thredds.war.old
```

NB You are not allowed to call it `thredds.old.war` as any file with trailing `*.war` will be unpacked by Tomcat. We advocate the labeling `thredds.war.old` and `thredds.war.4.2.10` to prevent mistakes with releases. After this preparation do the actual re-deployment.

```
cd /var/lib/tomcat6/content/webapps/
sudo /etc/init.d/tomcat6 stop
sudo rm -rf thredds
cp thredds.war.4.2.10 thredds.war
sudo /etc/init.d/tomcat6 restart
```

Install Apache webserver (webserver for kml files)

Installation Package Page



The Apache Software Foundation
http://www.apache.org/



Tomcat Web Application Manager

Message:

Manager

[List Applications](#) [HTML Manager Help](#) [Manager Help](#) [Server Status](#)

Applications				
Path	Display Name	Running	Sessions	Commands
/		true	0	Start Stop Reload Undeploy <input type="button" value="Expire sessions"/> with idle ≥ <input type="text" value="30"/> minutes
/docs	Tomcat Documentation	true	0	Start Stop Reload Undeploy <input type="button" value="Expire sessions"/> with idle ≥ <input type="text" value="30"/> minutes
/examples	Servlet and JSP Examples	true	0	Start Stop Reload Undeploy <input type="button" value="Expire sessions"/> with idle ≥ <input type="text" value="30"/> minutes
				Start Stop Reload Undeploy

manager select:

- Apache2
 - Apache2-doc
- [Apply]**

Create a dir, e.g. *kml*, in `var/data/`. Open `/etc/apache2/sites-enabled/000-default` in a text editor and make sure you point to it as `/var/data/kml`. This is described by Apache's [documentroot](#) documentation. You can also point to files in another directory than `/var/data`. This is described by Apache's [Files Outside the DocumentRoot](#) documentation. On our test server [dtvirt5](#) we made the *kml* folder a subfolder of the data directory for THREDDDS, such that this *data* folder is the only folder where we have to give write access to the group of data-administrators.

```
...
DocumentRoot /var/www

RewriteEngine on
# /kml
Alias /kml /data/kml
<Directory /data/kml>
    Order allow,deny
    Allow from all
</Directory>
...
```

In both cases make sure the `DocumentRoot` contains a representative `index.html` file that contains links to the SubVersion, OPeNDAP, and kml services. You can optionally lay-out the `index.html` with some logos, e.g. <http://http://dtvirt5.deltares.nl/>. You can also delete the default "It works!" `index.html` page, such that the subfolder *kml* will appear (unless you relocated the *kml* folder as described above).

Restart apache service to apply settings

```
sudo /etc/init.d/apache2 restart
```

Now open `/etc/apache2/sites-enabled/000-default` and add (just after `documentroot`):

```
AddType application/vnd.google-earth.kml+xml .kml
AddType application/vnd.google-earth.kmz .kmz
```

Install OpenSSH to enable use of WinSCP

To remotely manage the virtual server install WinSCP. For WinSCP to work you must install OpenSSH-server with the synaptic package manager. For the data directory to be accessible by WinSCP make sure to edit the rights:

```
sudo chmod -R og+w /var/data
sudo chmod +s /var/data
```

Overview of useful Linux commands

- Location of Synaptic Package Manager: select "System" then "Administration" then "Synaptic Package Manager"
- Location of terminal: select "Applications" then "Accessories" then "Terminal"
- In the terminal window you can use tab to let Linux finish the directory or filename you are typing. Tabbing twice provides a list of suggestions in case of ambiguity.
- To find ip-address (see "inet addr" under eth1):

```
ifconfig -a
```

- To make directories (use sudo for proper rights):

```
sudo nautilus
```

- To edit files (use sudo, otherwise you can probably not save the files):

```
sudo gedit
```

- To mount an external disk, check: <http://www.automaticable.com/2008-01-18/how-to-mount-a-network-drive-in-ubuntu/>
- To find out if an external disk is mounted:

```
mount
```

- To see available external disks:

```
ls /dev/sd*
```

This will show externa scsi disks sdb is disk sdb1 is partition1 on disk sdb.

- To mount the external disk

```
mount /dev/sdb1 /var/data -t auto
```

Put in /etc/fstab to do always on bootup (see instruction below)

- To request a manual type eg.:

```
man mount
```

- In case of failing services stop and restart all services (e.g. via Putty):

```
sudo /etc/init.d/tomcat6 restart
sudo /etc/init.d/svnserve start
sudo /etc/init.d/svnserve stop
sudo /etc/init.d/apache2 restart
```

- To restart Thredds go to [http://\[ip-address\]:8080/manager/html](http://[ip-address]:8080/manager/html) and stop and then start the service

Useful links

Apache

<http://httpd.apache.org/docs/2.2/howto/auth.html> Apache authentication and authorization documentation
http://httpd.apache.org/docs/2.2/mod/mod_dav.html Webdav
<http://tomcat.apache.org/tomcat-6.0-doc/index.html> Tomcat documentation

Filesystems

<http://www.drbd.org/> Drbd

NetCDF

<http://www.unidata.ucar.edu/software/netcdf/docs/netcdf-install/> NetCDF downloads

Matlab

<http://mexcdf.sourceforge.net/> Matlab netcdf library

Python

<http://www.scipy.org/> Python numerical package
<http://code.google.com/p/netcdf4-python/> Netcdf python software

Conventions

<http://www.inspire-geoportal.eu/InspireEditor/> Inspire editor
<http://www.gfdl.noaa.gov/~vb/grids/gridspec-tools.html> Gridspec

OpenDAP

<https://help.ubuntu.com/8.10/serverguide/C/tomcat.html> Install tomcat under ubuntu
<http://opendap.org/download/hyrax.html> Hyrax server download
<http://docs.opendap.org/index.php/Hyrax> Hyrax documentation
<http://www.unidata.ucar.edu/projects/THREDDS/> Thredds server

References

http://en.wikipedia.org/wiki/DMZ_%28computing%29 DMZ information
http://en.wikipedia.org/wiki/Secure_copy Secure copy

GUI

<http://www.epic.noaa.gov/java/ncBrowse/> NcBrowse
<http://www.mpimet.mpg.de/fileadmin/software/cdo/> CDO

Tools

<http://nco.sourceforge.net/> NCO
<http://www.unidata.ucar.edu/software/idv/> IDV
<http://www.unidata.ucar.edu/downloads/netcdf/netcdf-java-4/index.jsp> NetCDF-UI
<http://publicwiki.deltares.nl/display/OET/OpenEarth> OpenEarthTools

Subversion

<http://subversion.tigris.org/> Subversion
<https://help.ubuntu.com/community/Subversion> Subversion setup on ubuntu



Unknown macro: 'table-plus'