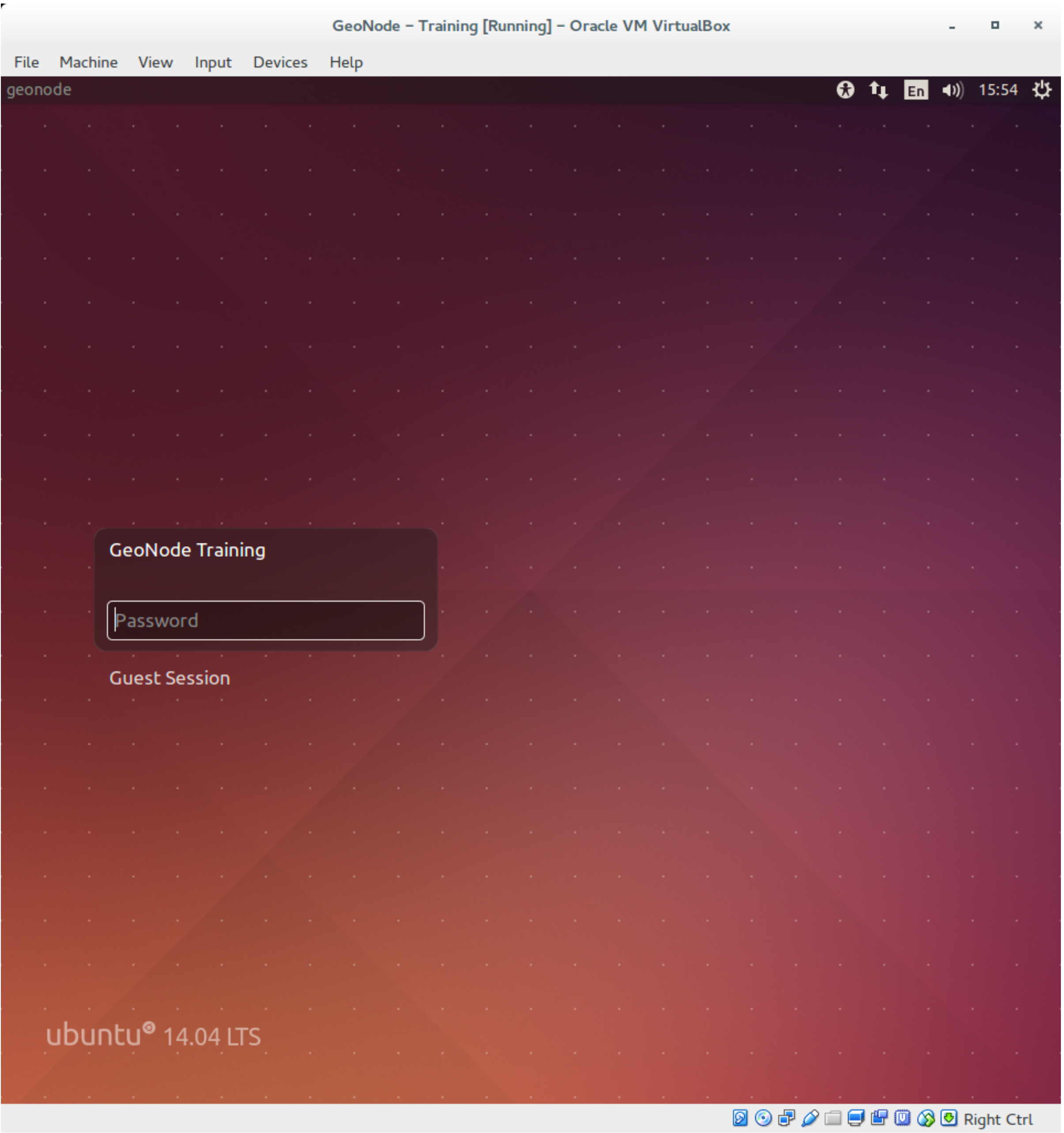
Install GeoNode Application

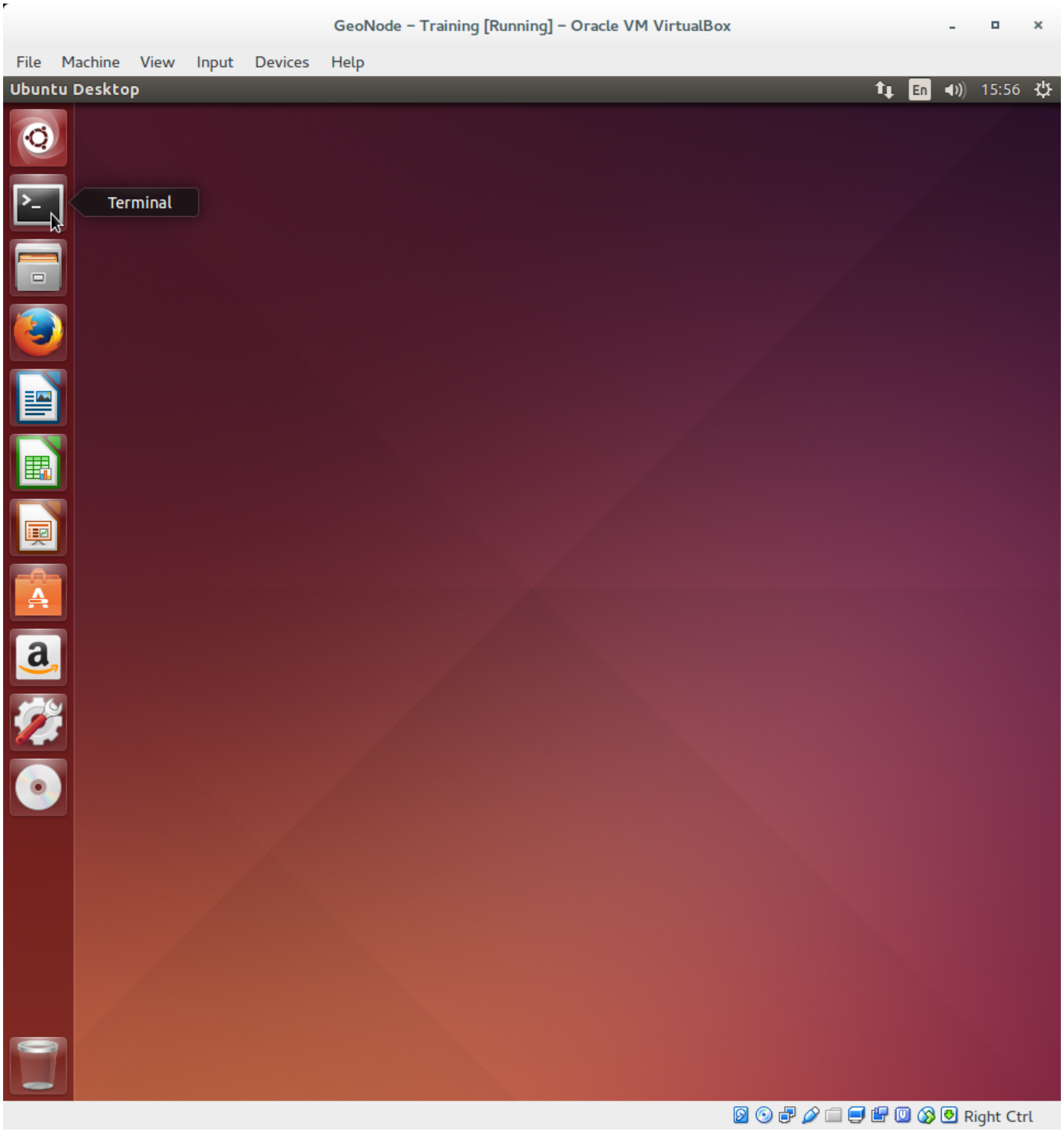
In this section you are going to install all the basic packages and tools needed for a complete GeoNode installation.

Login

When you first start the Virtual Machine at the end of the boot process you will be prompted for the user password to login. Enter *geo* as user password and press *Enter*.



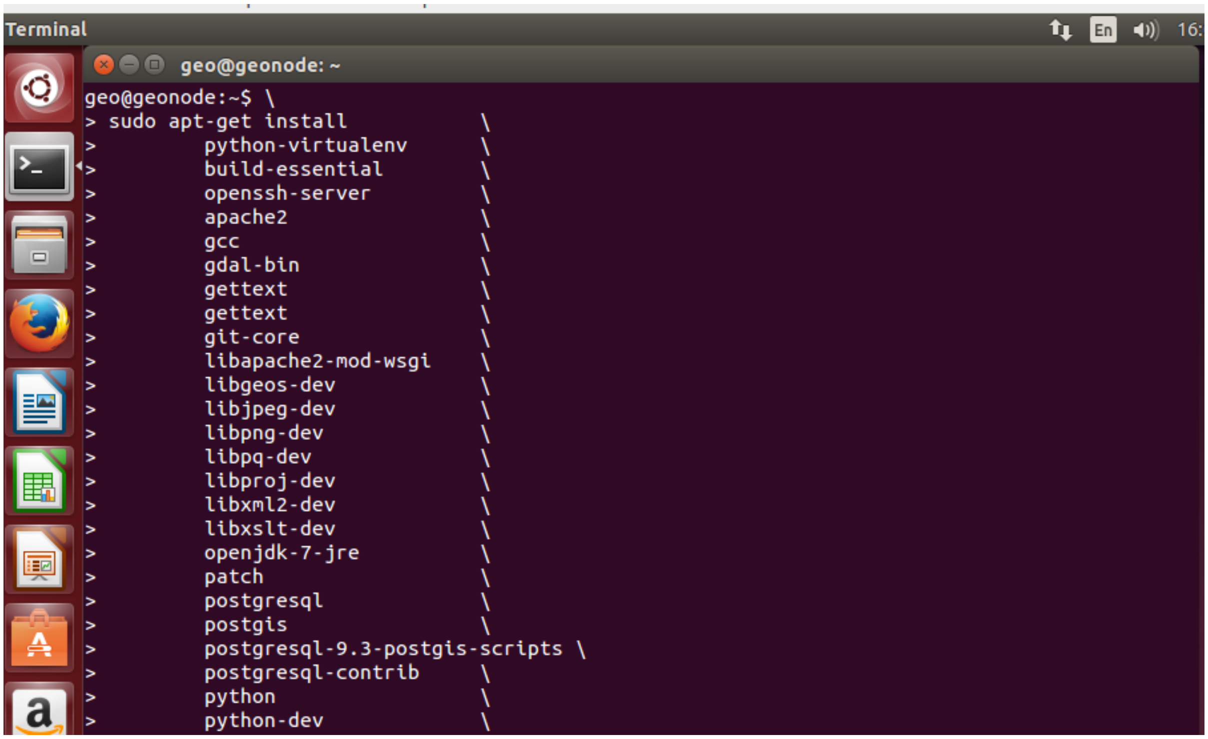
You are now logged in as user ‘geo’. On the left side of the screen there is a panel with shortcuts to common applications, launch a the terminal emulator.



Packages Installation

First we are going to install all the software packages we are going to need for the GeoNode setup. Among others *Tomcat 8*, *PostgreSQL*, *PostGIS*, *Apache HTTP server* and *Git*. Run the following command to install all the packages

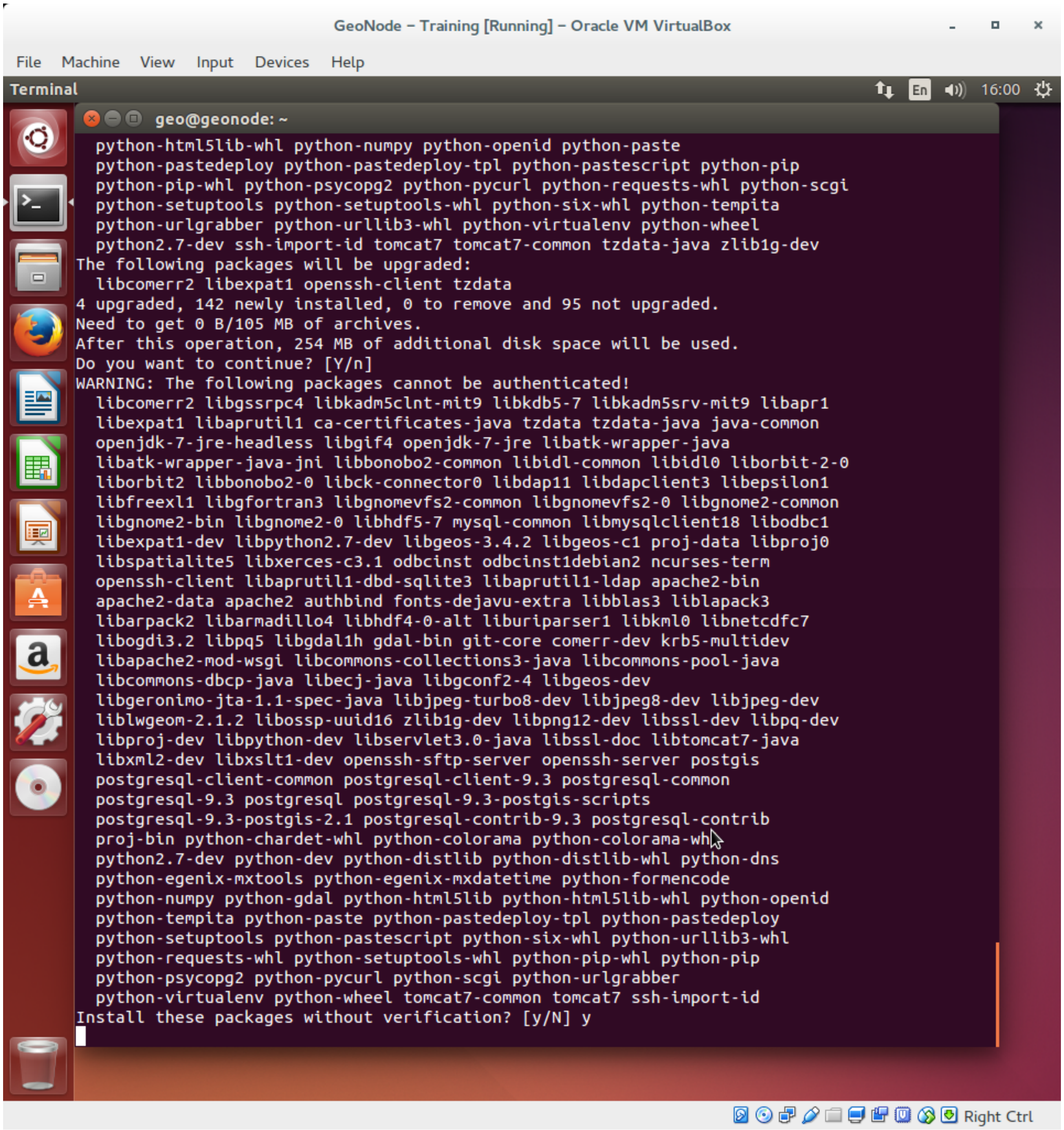
```
$ sudo apt-get update

$ sudo apt-get install python-virtualenv python-dev libxml2 libxml2-dev libxslt1-dev
zlib1g-dev libjpeg-dev libpq-dev git default-jdk
$ sudo apt-get install build-essential openssh-server gettext nano vim unzip zip patch
git-core postfix

$ sudo apt-add-repository ppa:webupd8team/java
$ sudo apt-get update
$ sudo apt-get install oracle-java8-installer

$ sudo apt-add-repository ppa:ubuntugis && sudo apt-get update && sudo apt-get upgrade
$ sudo apt-add-repository ppa:ubuntugis/ppa && sudo apt-get update && sudo apt-get upgrade
$ sudo apt-get install gcc apache2 libapache2-mod-wsgi libgeos-dev libjpeg-dev libpng-dev
libpq-dev libproj-dev libxml2-dev libxslt-dev
$ sudo apt-add-repository ppa:ubuntugis/ubuntugis-testing && sudo apt-get update && sudo
apt-get upgrade
$ sudo apt-get install gdal-bin libgdal20 libgdal-dev
$ sudo apt-get install python-gdal python-pycurl python-imaging python-pastescript python-
psycopg2 python-urlgrabber
$ sudo apt-get install postgresql postgis postgresql-9.5-postgis-scripts postgresql-
contrib
$ sudo apt-get install tomcat8

$ sudo apt-get update && sudo apt-get upgrade && sudo apt-get autoremove && sudo apt-get
autoclean && sudo apt-get purge && sudo apt-get clean
```



```
geo@geonode:~$ \
> sudo apt-get install         \
>         python-virtualenv      \
>         build-essential        \
>         openssh-server         \
>         apache2                \
>         gcc                    \
>         gdal-bin               \
>         gettext                \
>         gettext                \
>         git-core               \
>         libapache2-mod-wsgi    \
>         libgeos-dev            \
>         libjpeg-dev            \
>         libpng-dev             \
>         libpq-dev              \
>         libproj-dev            \
>         libxml2-dev            \
>         libxslt-dev            \
>         openjdk-7-jre          \
>         patch                  \
>         postgresql             \
>         postgis                \
>         postgresql-9.3-postgis-scripts \
>         postgresql-contrib     \
>         python                 \
>         python-dev             \
```

Note

If you will be prompted for *geo* user’s password (*geo*) and for confirmation twice



```
  python-html5lib-whl python-numpy python-openid python-paste
  python-pastedeploy python-pastedeploy-tpl python-pastescript python-pip
  python-pip-whl python-psycopg2 python-pycurl python-requests-whl python-scgi
  python-setuptools python-setuptools-whl python-six-whl python-tempita
  python-urlgrabber python-urllib3-whl python-virtualenv python-wheel
  python2.7-dev ssh-import-id tomcat7 tomcat7-common tzdata-java zlib1g-dev
The following packages will be upgraded:
  libcomerr2 libexpat1 openssh-client tzdata
4 upgraded, 142 newly installed, 0 to remove and 95 not upgraded.
Need to get 0 B/105 MB of archives.
After this operation, 254 MB of additional disk space will be used.
Do you want to continue? [Y/n]
WARNING: The following packages cannot be authenticated!
  libcomerr2 libgssrpc4 libkadm5clnt-mit9 libkdb5-7 libkadm5srv-mit9 libapr1
  libexpat1 libaprutil1 ca-certificates-java tzdata tzdata-java java-common
  openjdk-7-jre-headless libgif4 openjdk-7-jre libatk-wrapper-java
  libatk-wrapper-java-jni libbonobo2-common libidl-common libidl0 liborbit-2-0
  liborbit2 libbonobo2-0 libck-connector0 libdap11 libdapclient3 libepsilon1
  libfreexl1 libgfortran3 libgnomevfs2-common libgnomevfs2-0 libgnome2-common
  libgnome2-bin libgnome2-0 libhdf5-7 mysql-common libmysqlclient18 libodbc1
  libexpat1-dev libpython2.7-dev libgeos-3.4.2 libgeos-c1 proj-data libproj0
  libspatialite5 libxerces-c3.1 odbcinst odbcinst1debian2 ncurses-term
  openssh-client libaprutil1-dbd-sqlite3 libaprutil1-ldap apache2-bin
  apache2-data apache2 authbind fonts-dejavu-extra libblas3 liblapack3
  libarpack2 libarmadillo4 libhdf4-0-alt liburiparser1 libkml0 libnetcdfc7
  libogdi3.2 libpq5 libgdal1h gdal-bin git-core comerr-dev krb5-multidev
  libapache2-mod-wsgi libcommons-collections3-java libcommons-pool-java
  libcommons-dbcp-java libecj-java libgconf2-4 libgeos-dev
  libgeronimo-jta-1.1-spec-java libjpeg-turbo8-dev libjpeg8-dev libjpeg-dev
  liblwgeom-2.1.2 libossp-uuid16 zlib1g-dev libpng12-dev libssl-dev libpq-dev
  libproj-dev libpython-dev libservlet3.0-java libssl-doc libtomcat7-java
  libxml2-dev libxslt1-dev openssh-sftp-server openssh-server postgis
  postgresql-client-common postgresql-client-9.3 postgresql-common
  postgresql-9.3 postgresql postgresql-9.3-postgis-scripts
  postgresql-9.3-postgis-2.1 postgresql-contrib-9.3 postgresql-contrib
  proj-bin python-chardet-whl python-colorama python-colorama-whl
  python2.7-dev python-dev python-distlib python-distlib-whl python-dns
  python-egenix-mxtools python-egenix-mxdatetime python-formencode
  python-numpy python-gdal python-html5lib python-html5lib-whl python-openid
  python-tempita python-paste python-pastedeploy-tpl python-pastedeploy
  python-setuptools python-pastescript python-six-whl python-urllib3-whl
  python-requests-whl python-setuptools-whl python-pip-whl python-pip
  python-psycopg2 python-pycurl python-scgi python-urlgrabber
  python-virtualenv python-wheel tomcat7-common tomcat7 ssh-import-id
Install these packages without verification? [y/N] y
```

Warning

The installation process is going to take several minutes and it will need to download packages from Internet.

At this point we have all the packages we need on the system.

GeoNode Setup

First of all we need to prepare a new Python Virtual Environment:

```
$ sudo apt install python-pip
$ pip install --upgrade pip
$ pip install --user virtualenv
$ pip install --user virtualenvwrapper
# The commands above will install the Python Venv packages

$ export WORKON_HOME=~/Envs
$ mkdir -p $WORKON_HOME
$ source $HOME/.local/bin/virtualenvwrapper.sh
$ printf '\n%s\n%s\n%s' '# virtualenv' 'export WORKON_HOME=~/Envs' 'source
$HOME/.local/bin/virtualenvwrapper.sh' >> ~/.bashrc
$ source ~/.bashrc
# We have now configured the user environment

$ mkvirtualenv --no-site-packages geonode
# Through this command we have created a brand new geonode Virual Environment

$ sudo useradd -m geonode
$ sudo usermod -a -G geonode geo
$ sudo chmod -Rf 775 /home/geonode/
$ sudo su - geo
# The commands above are needed only if geo and geonode users have not been already
defined
```

Let's activate the new *geonode* Python Virtual Environment:

```
$ workon geonode
```

Move into the *geonode* home folder

```
$ cd /home/geonode
```

We are going to install GeoNode as a dependency of a **Customized DJango Project**

Note

A custom project is a DJango application with *ad hoc* configuration and folders, which allows you to extend the original **GeoNode** code without actually dealing or modifying the main source code.

This will allow you to easily customize your GeoNode instance, modify the theme, add new functionalities and so on, and also being able to keep updated with the GeoNode latest source code.

For more deails please check https://github.com/GeoNode/geonode-project/tree/master

```
$ pip install Django==1.8.18
$ django-admin.py startproject --template=https://github.com/GeoNode/geonode-
project/archive/2.8.0.zip -e py,rst,json,yml my_geonode
```

Let's install the GeoNode dependencies and packages into the Python Virtual Environment:

```
$ cd my_geonode

# Find the closest pygdal version.
# Example: 2.2.1 ...  2.2.1.3, ...
$ gdal-config --version && pip install pygdal==

$ vim requirements.txt
# Make sure requirements contains reference to geonode 2.8 branch
# and correct gdal version (see above)
-e git://github.com/GeoNode/geonode.git@2.8.0#egg=geonode
pygdal==2.2.1.3

$ pip install -r requirements.txt --upgrade
$ pip install -e . --upgrade --no-cache
```

In the next section we are going to setup PostgreSQL Databases for GeoNode and finalize the setup