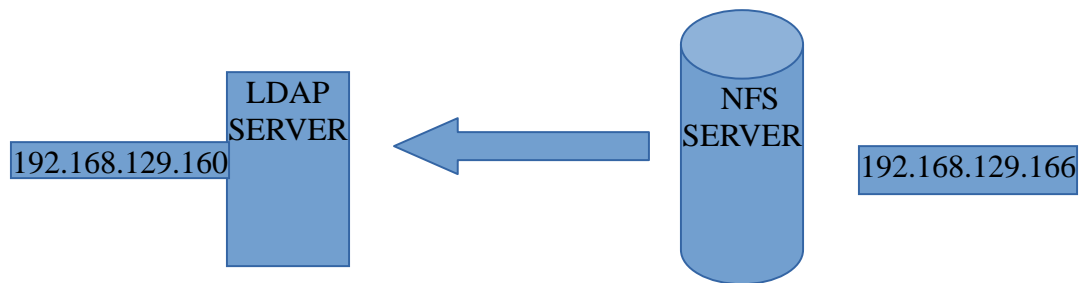


OpenLDAP is a free open source Light Weight Directory Access protocol developed by the OpenLDAP project. It is a platform independent protocol, so that it runs on all Linux/Unix like systems, Windows, AIX, Solaris and Android.



Tasks:

Create a ldap server for centralized login and create a centralized storage where all user logged in via ldap authentication should have common storage.

Pre-requisites:

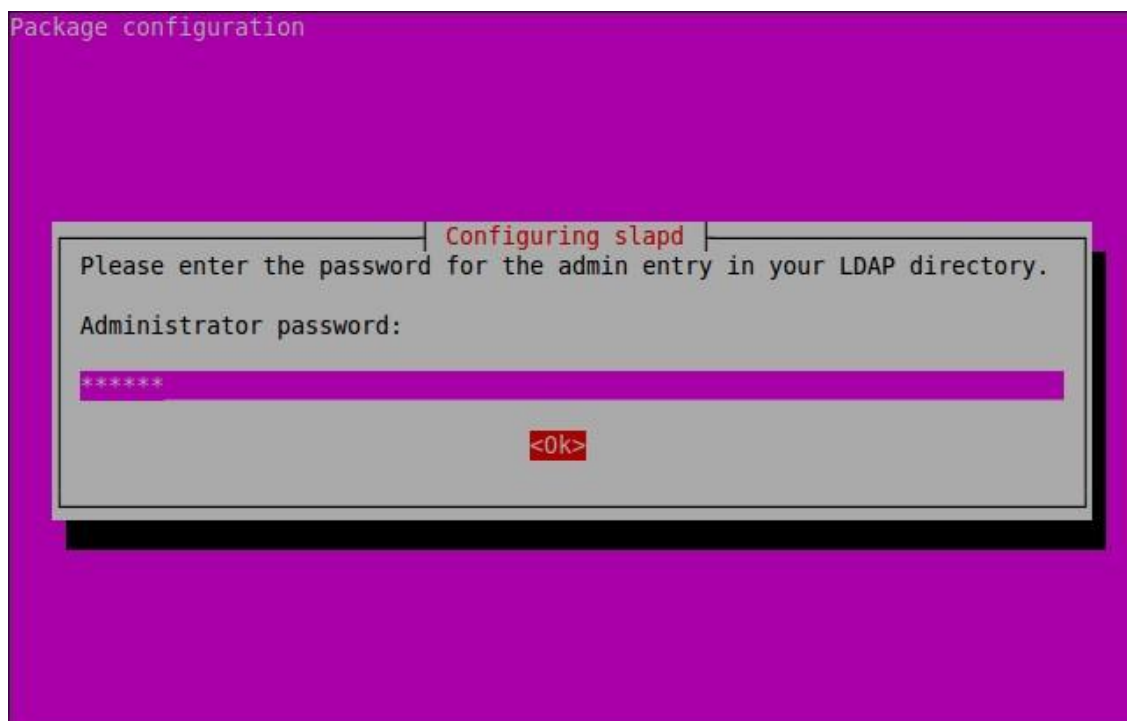
Domain: test.com

IP Address: 192.168.129.160

first install the required package!

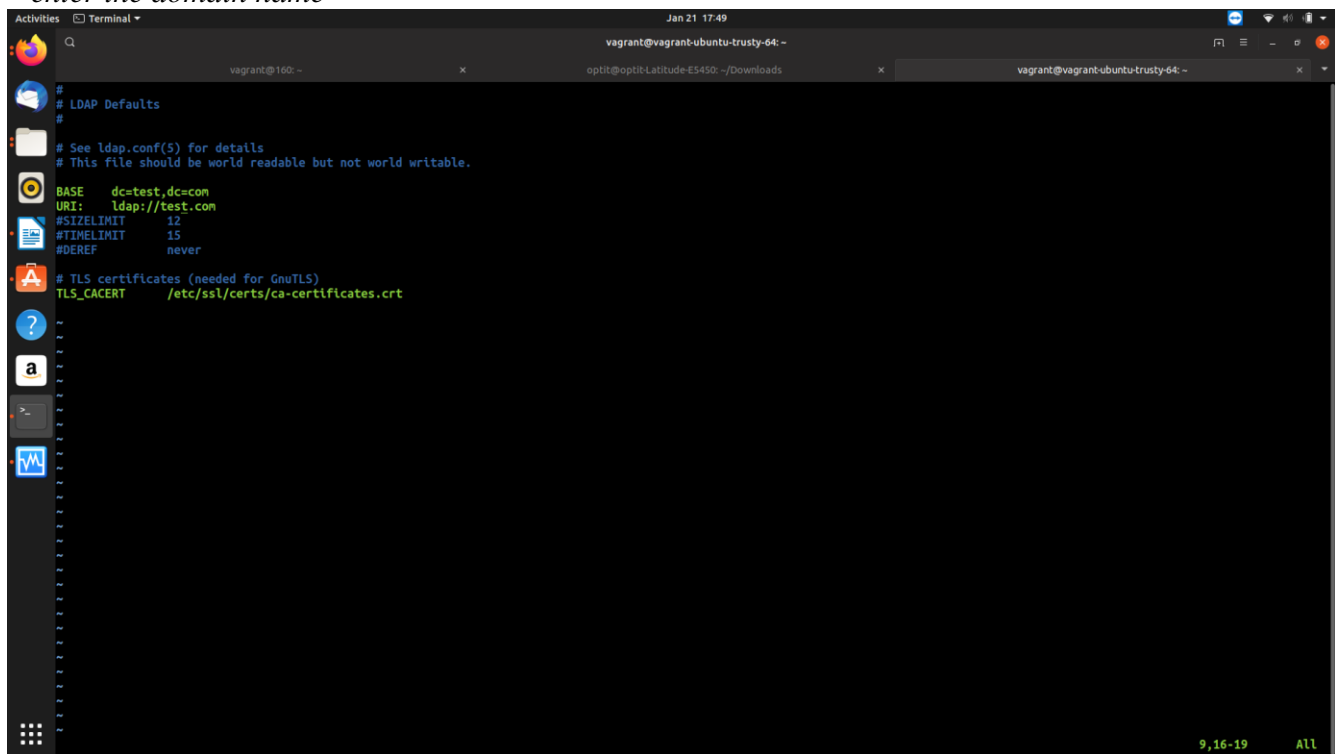
#apt-get install slapd ldap-utils

During installation set the Administrator password



#vi etc/ldap/ldap.conf

enter the domain name



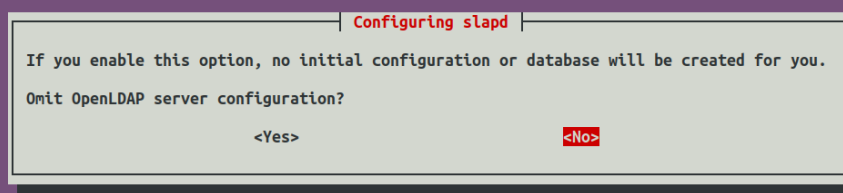
A terminal window titled 'vagrant@vagrant-ubuntu-trusty-64' showing the contents of the file /etc/ldap/ldap.conf. The file contains the following configuration:

```
# LDAP Defaults
#
# See ldap.conf(5) for details
# This file should be world readable but not world writable.
BASE      dc=test,dc=com
URI:      ldap://test.com
#SIZELIMIT 12
#TIMELIMIT 15
#DEREF    never
# TLS certificates (needed for GnuTLS)
TLS_CACERT /etc/ssl/certs/ca-certificates.crt
```

save and close

we need to reconfigure ldap again

#dpkg-reconfigure slap



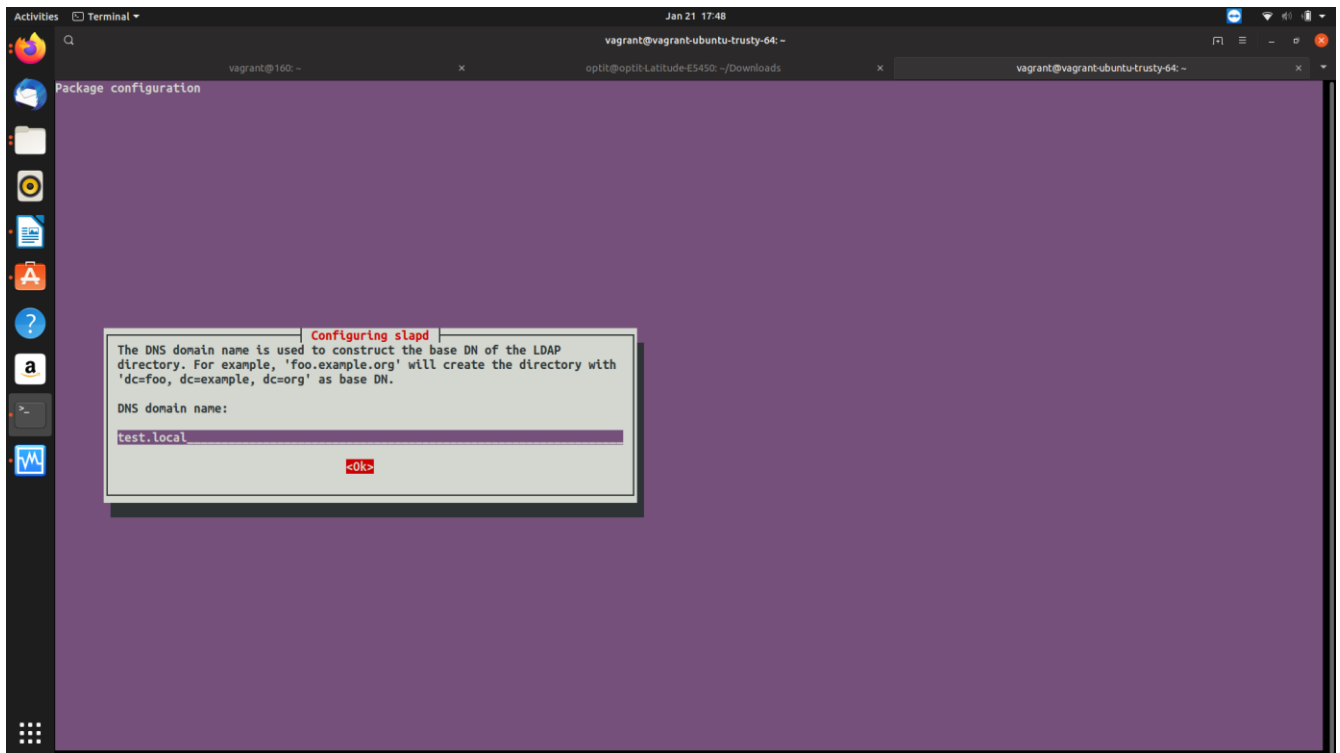
A dialog box titled 'Configuring slapd' with a light green background. It contains the following text:

If you enable this option, no initial configuration or database will be created for you.

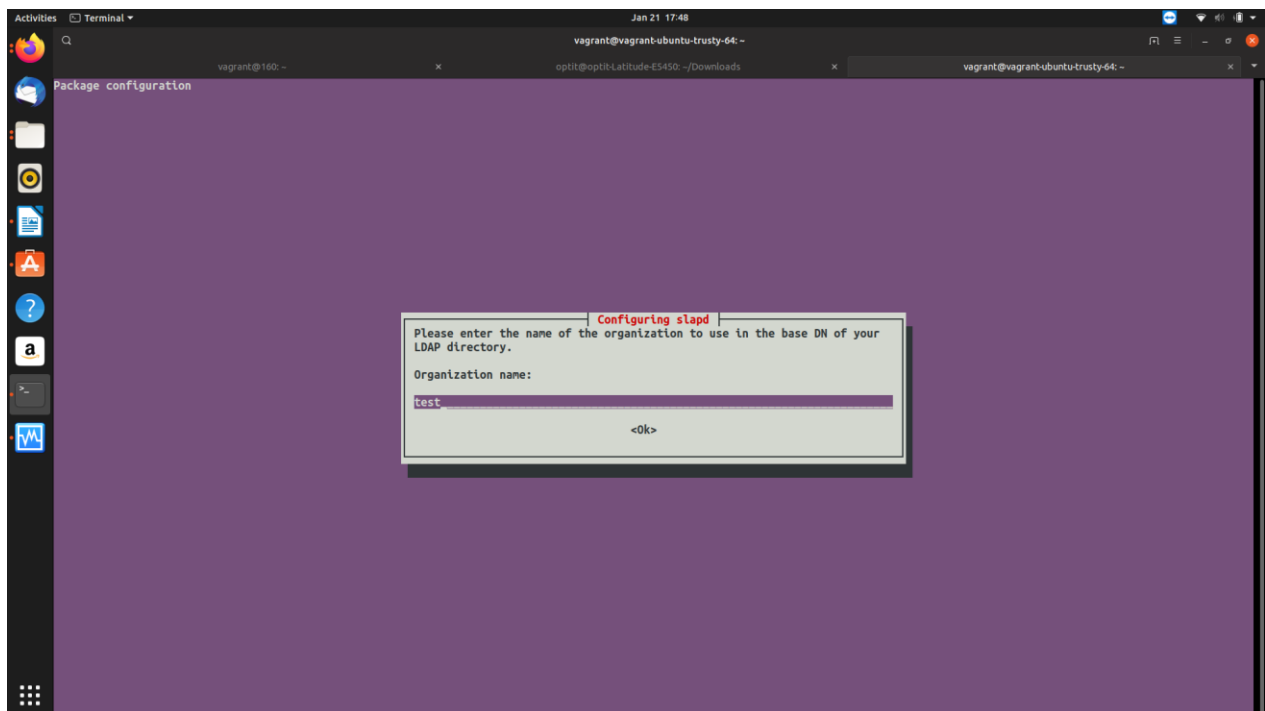
Omit OpenLDAP server configuration?

<Yes> **<No>**

Step 2

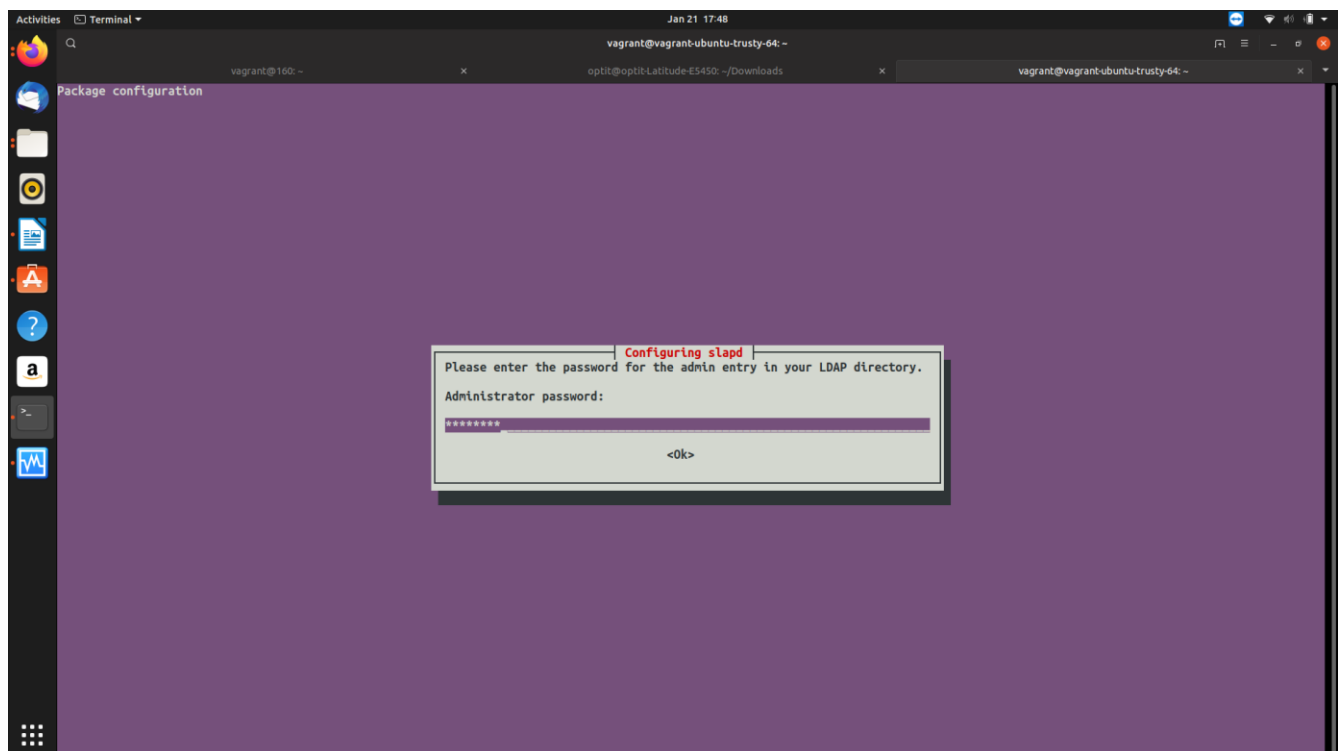


Step 3

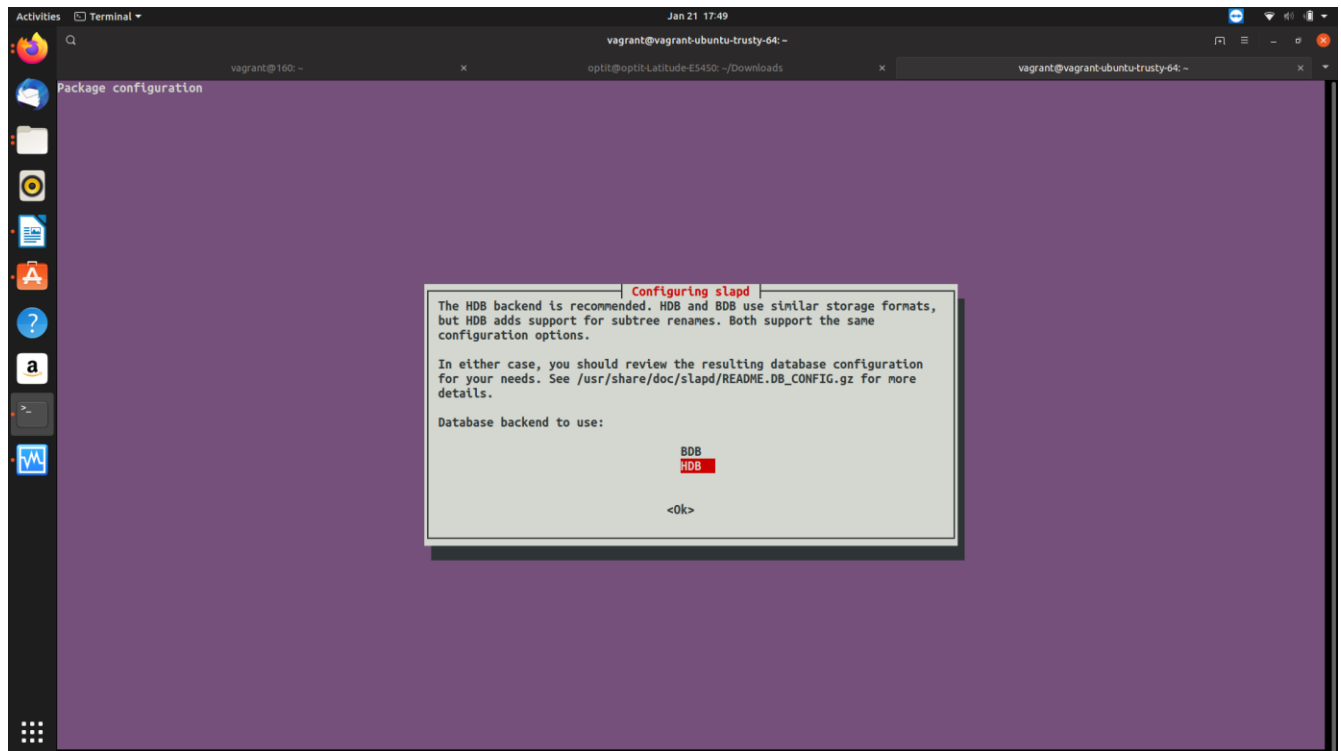


Step 4:

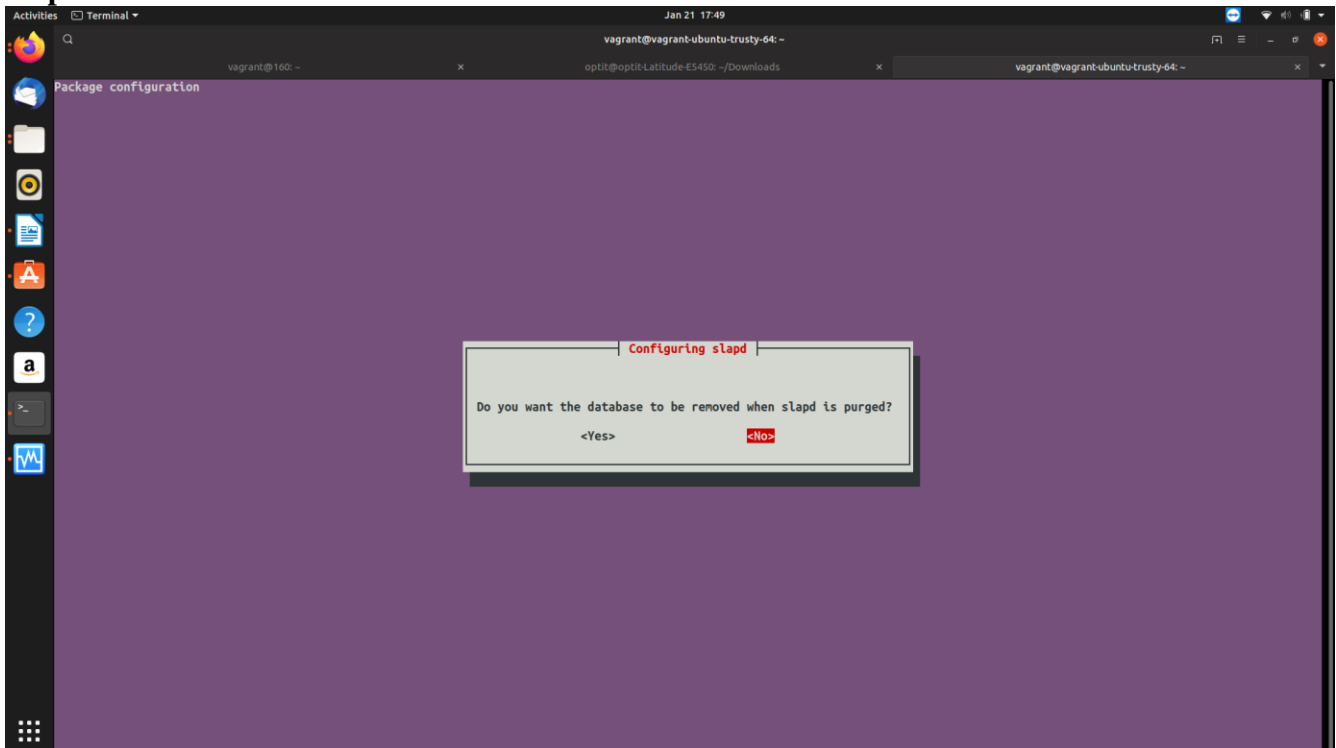
Step 4



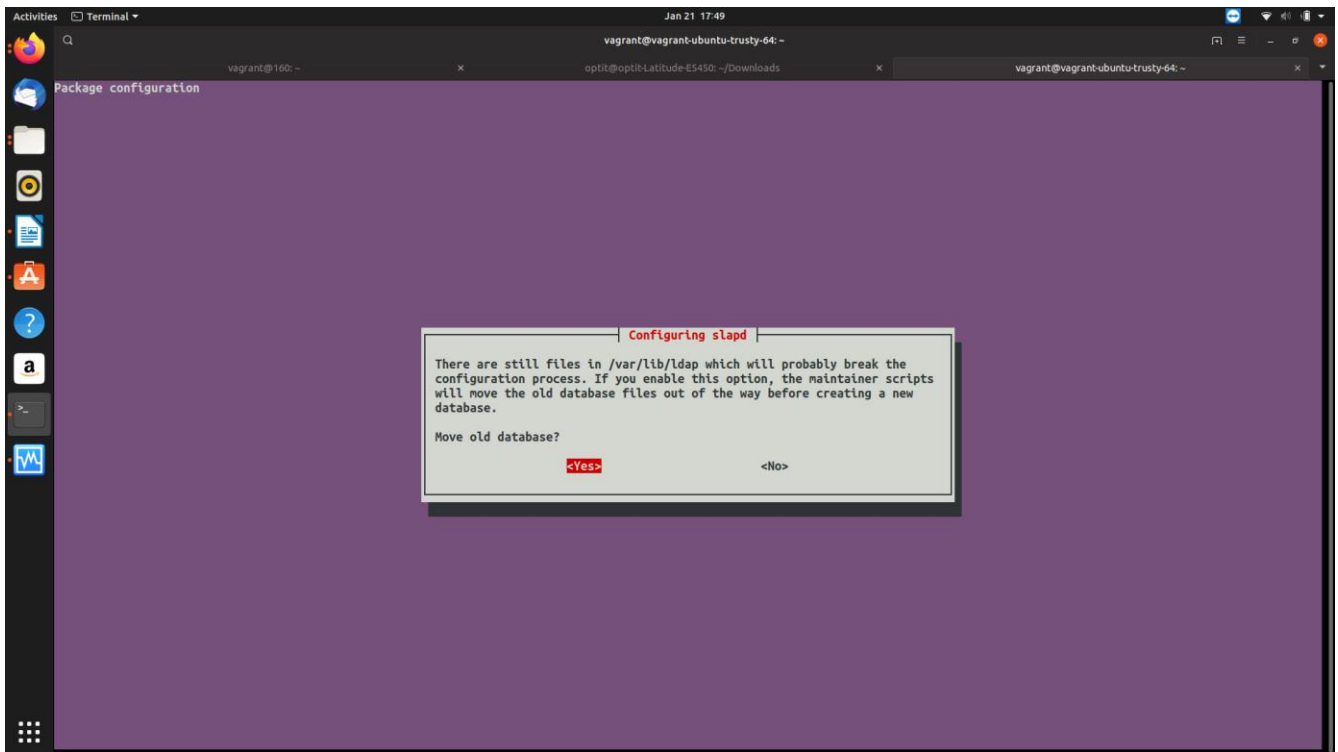
Step 5:



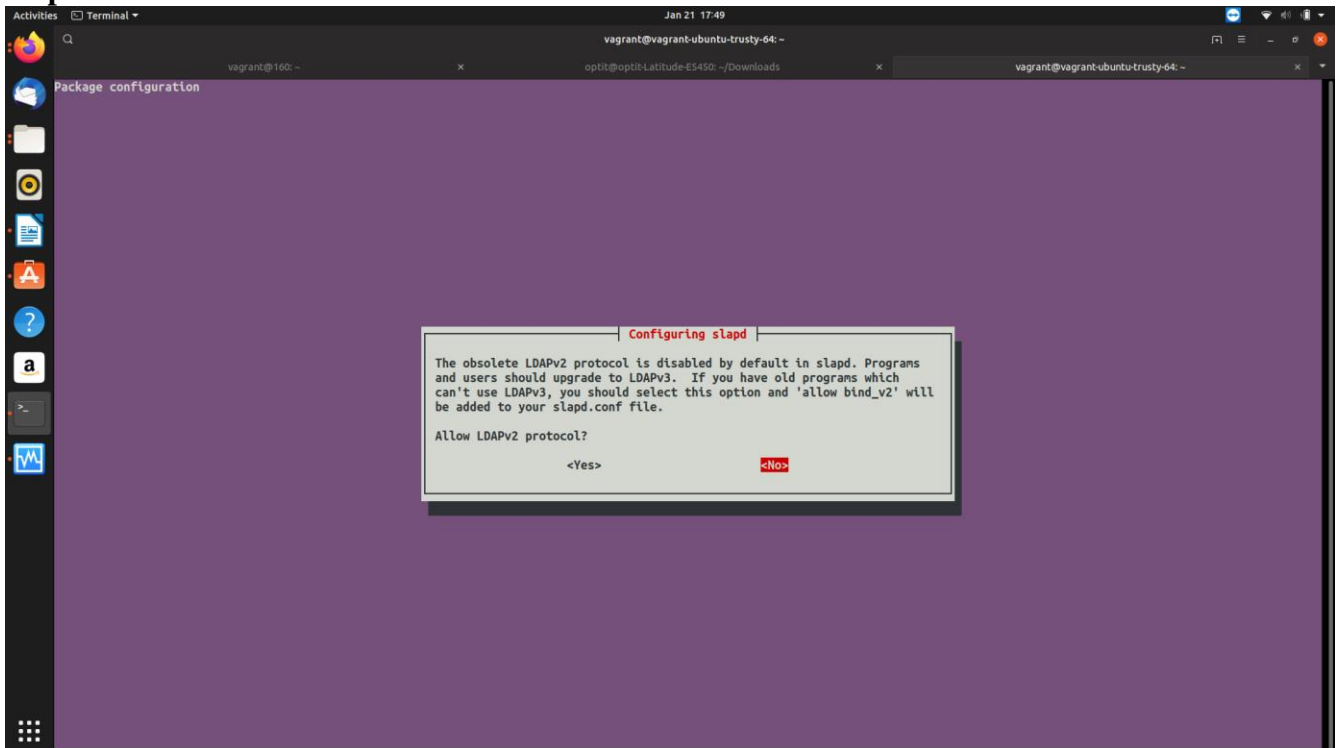
Step 6:



Step 7:



Step 8:



done .

Just Confirm your settings

```
#ldapsearch -x
```

Next we need to install php admin

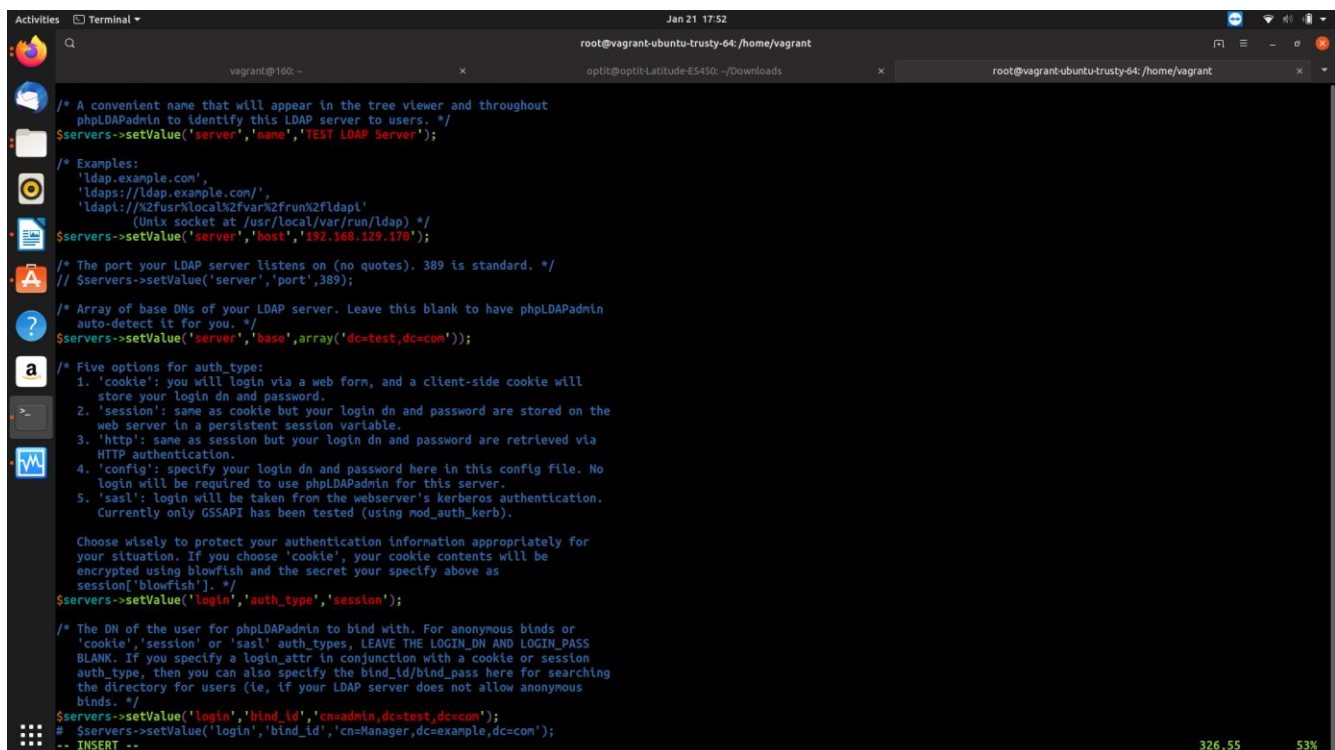
```
# apt-get install phpldapadmin
```

Create a symbolic link for phpldap directory

```
#ln -s /usr/share/phpldapadmin/ /var/www/phpldapadmin
```

Edit the phpldapadmin conf file and do the necessary changes

```
# vi etc/phpldapadmin/config.php
```



```
root@vagrant-ubuntu-trusty-64: /home/vagrant
vagrant@160: ~
opfit@opfit-Latitude-E5450: ~/Downloads
root@vagrant-ubuntu-trusty-64: /home/vagrant

/* A convenient name that will appear in the tree viewer and throughout
   phpldapAdmin to identify this LDAP server to users. */
$servers->setValue('server','name','TEST LDAP Server');

/* Examples:
   'ldap.example.com',
   'ldaps://ldap.example.com/',
   'ldapi://%2fusr%2flocal%2fvar%2frun%2fldapi'
   (Unix socket at /usr/local/var/run/ldap) */
$servers->setValue('server','host','192.168.129.170');

/* The port your LDAP server listens on (no quotes). 389 is standard. */
// $servers->setValue('server','port',389);

/* Array of base DN's of your LDAP server. Leave this blank to have phpldapAdmin
   auto-detect it for you. */
$servers->setValue('server','base',array('dc=test,dc=com'));

/* Five options for auth_type:
   1. 'cookie': you will login via a web form, and a client-side cookie will
      store your login dn and password.
   2. 'session': same as cookie but your login dn and password are stored on the
      web server in a persistent session variable.
   3. 'http': same as session but your login dn and password are retrieved via
      HTTP authentication.
   4. 'config': specify your login dn and password here in this config file. No
      login will be required to use phpldapAdmin for this server.
   5. 'sasl': login will be taken from the webserver's kerberos authentication.
      Currently only GSSAPI has been tested (using mod_auth_kerb).

   Choose wisely to protect your authentication information appropriately for
   your situation. If you choose 'cookie', your cookie contents will be
   encrypted using blowfish and the secret you specify above as
   session['blowfish']. */
$servers->setValue('login','auth_type','session');

/* The DN of the user for phpldapAdmin to bind with. For anonymous binds or
   'cookie', 'session' or 'sasl' auth_types, LEAVE THE LOGIN_DN AND LOGIN_PASS
   BLANK. If you specify a login_attr in conjunction with a cookie or session
   auth_type, then you can also specify the bind_id/bind_pass here for searching
   the directory for users (ie, if your LDAP server does not allow anonymous
   binds. */
$servers->setValue('login','bind_id','cn=admin,dc=test,dc=com');
# $servers->setValue('login','bind_id','cn=Manager,dc=example,dc=com');
-- INSERT --
```

Restart the apache2 service

```
#service apache2 restart
```

update the port in firewall

```
#ufw allow 389
```

```
#ufw allow 80
```

TEST THE PHPLDAP

in browser enter **http://<ip_address_of_server>/phpldapadmin**

Once the Above steps are done, We need to create NFS Server for centralized storage.

SERVER CONFIGURATION

```
#apt-get install nfs-utils
```

```
#mkdir NFS_Storage
```

```
#chmod 777 NFS_Storage
```

```
#vi etc/exports
```

```
/NFS_Storage * (rw,sync,no_root_squash)
```

CLIENT CONFIGURATION (here our client will be ldap server)

```
#apt-get install nfs-common
```

```
#mkdir home/local
```

```
#chmod 777 /home/local
```

```
#mount -t nfs <server_ip:/path> home/local
```

```
#mount -a
```

```
#df -h (to verify the mount point)
```