

LDAP

Task Requirement

- Install and configure Ldap 389 ds
- Create organisation keenable.in
- Create OU [Dev,Support,POC, Document, Observability]
- Create group Admin,Support

List of Tools

- LDAP
- Podman
- Apache Directory Studio

LDAP:- The Lightweight Directory Access Protocol is a communication protocol used to access directory servers and is used to store, update and retrieve data from a directory structure.

Podman:- Podman is an open-source container management tool designed to run on Linux systems. Its main purpose is to create, run, manage, and delete containers.

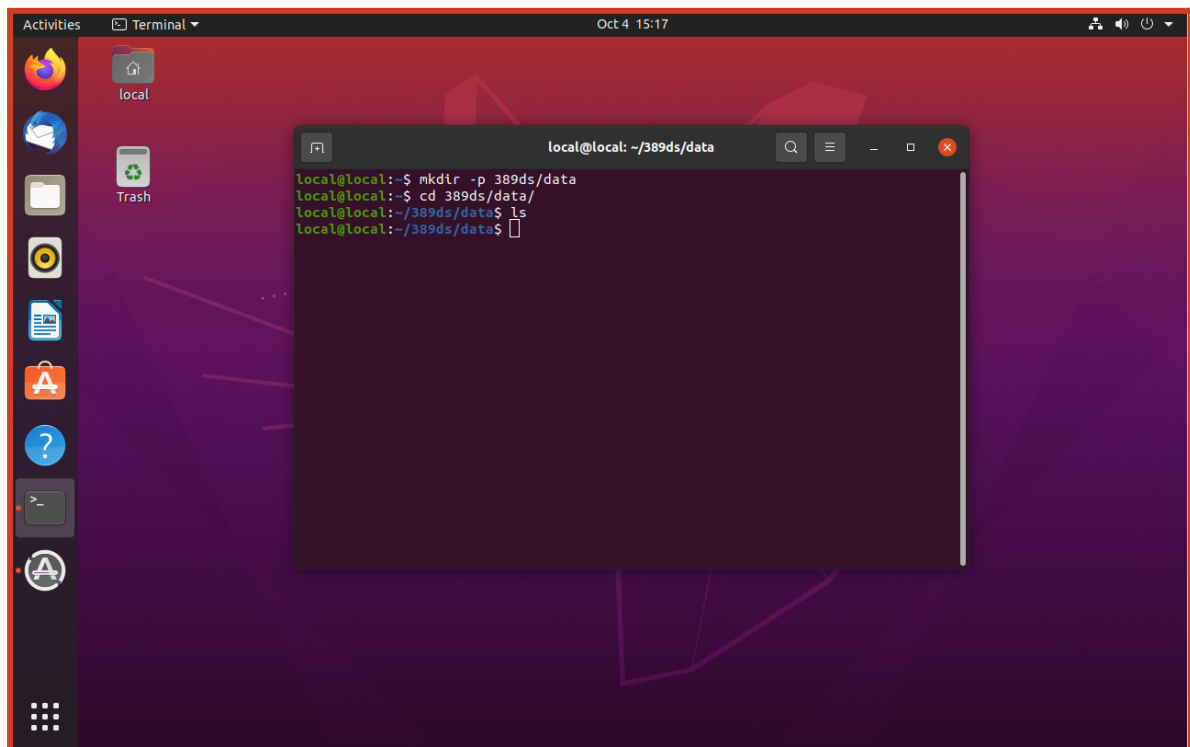
Apache Directory Studio:- Apache Directory Studio is an open-source LDAP (Lightweight Directory Access Protocol) client that helps you communicate with and manage directory servers.

1. Command for the Setup or Configuration

Create directory with name 389ds and also create directory data inside

```
mkdir -p 389ds/data
```

```
cd 389ds/data
```



Create Bash Script for Create Pod and Container

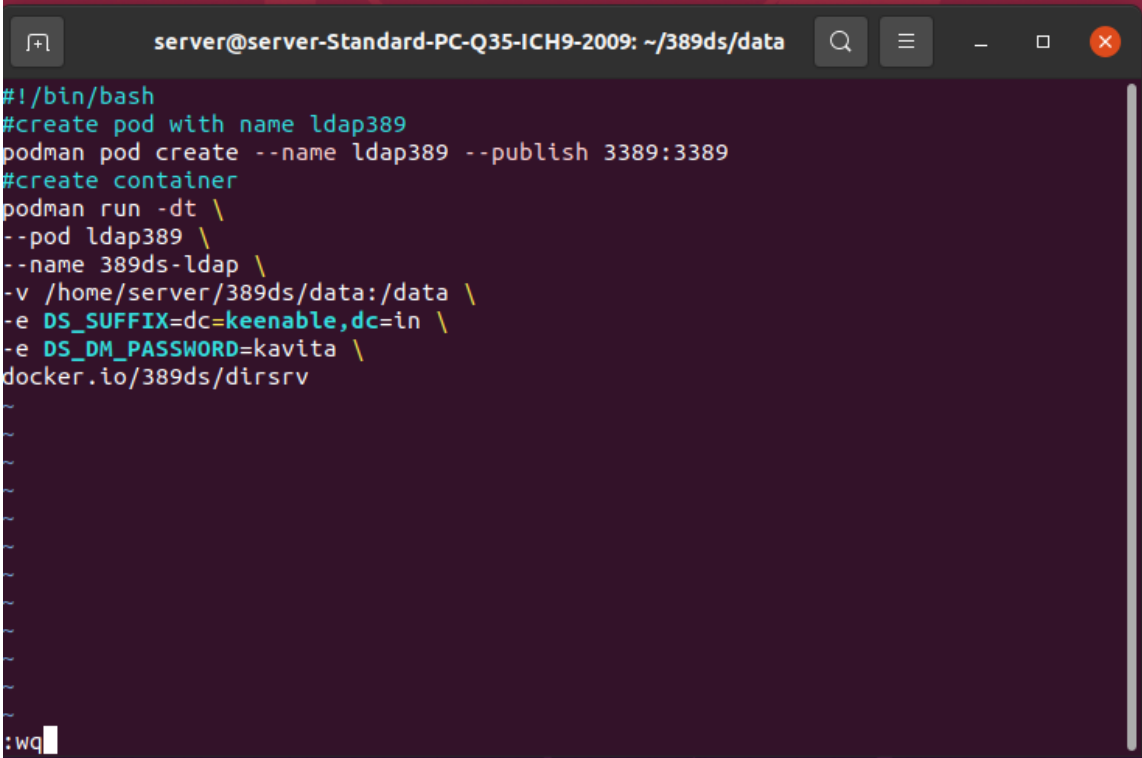
```
vim ldap.sh
```

Run script

Bash script :

```
#!/bin/bash
#create pod with name ldap389
podman pod create --name ldap389 --publish 3389:3389
#create container
podman run -dt \
--pod ldap389 \
--name 389ds-ldap \
-v /home/local/389ds/data:/data \cv
-e DS_SUFFIX=dc=keenable,dc=in \
-e DS_DM_PASSWORD=kavita \
docker.io/389ds/dirsrv
```

/



```
server@server-Standard-PC-Q35-ICH9-2009: ~/389ds/data
#!/bin/bash
#create pod with name ldap389
podman pod create --name ldap389 --publish 3389:3389
#create container
podman run -dt \
--pod ldap389 \
--name 389ds-ldap \
-v /home/server/389ds/data:/data \
-e DS_SUFFIX=dc=keenable,dc=in \
-e DS_DM_PASSWORD=kavita \
docker.io/389ds/dirsrv
:wq
```

Give Permission to Script File

```
sudo chmod 777 ldap.sh
```

```
local@local:~$ sudo chmod 777 ldap.sh
```

changes the permissions of the file and permissions to 777 means that the owner, the group, and everyone else can read, write, and execute the script.

Run the script sh -x ldap.sh

```
sh -x ldap.sh
```

```

local@local:~$ sh -x ldap.sh
+ podman pod create --name ldap389 --publish 3389:3389 --publish 3636:3636
ec17d96f2525f472d5d24311857266a0f9ff824971e9c9e049959636b7d4731b
+ podman run -dt --pod ldap389 --name 389ds-ldap -v /home/local/389ds/data:/data -e DS_SUFFIX=dc=ke
le,dc=in -e DS_DM_PASSWORD=kavita docker.io/389ds/dirsrv
Trying to pull docker.io/389ds/dirsrv:latest...
Getting image source signatures
Copying blob b84593d8c7f9 done
Copying blob c373c35465bf done
Copying config 9731286686 done
Writing manifest to image destination
Storing signatures
b9e3f2f50176e845f9d1588771c702326c00bcc912135e6f4df9fbdd0aae3913

```

Check container list

```
podman ps -a --pod
```

```

local@local:~$ podman ps -a --pod
CONTAINER ID   IMAGE                                COMMAND                  CREATED          STATUS          PORTS
0799e8154130   k8s.gcr.io/pause:3.5               /pause                  59 seconds ago   Up 14 seconds   0
.0.0.0:3389->3389/tcp, 0.0.0.0:3636->3636/tcp   ec17d96f2525-infra     ec17d96f2525     ldap389
b9e3f2f50176   docker.io/389ds/dirsrv:latest      /usr/lib/dirsrv/d...    14 seconds ago   Up 14 seconds   0
.0.0.0:3389->3389/tcp, 0.0.0.0:3636->3636/tcp   389ds-ldap             ec17d96f2525     ldap389
local@local:~$ podman exec -it 389ds-ldap bash

```

Go inside container

```
Podman exec -it 389ds-ldap bash
```

```

local@local:~$ podman exec -it 389ds-ldap bash
ldap389:/ #

```

Create backend suffix list

```
dsconf -D "cn=Directory Manager" ldap://localhost:3389 backend create
--suffix="dc=keenable,dc=in" --be-name="keenable"
```

```

ldap389:/ # dsconf -D "cn=Directory Manager" ldap://localhost:3389 backend create --suffix="dc=keen
able,dc=in" --be-name="keenable"
Enter password for cn=Directory Manager on ldap://localhost:3389:

```

Check backend suffix list

```
dsconf -D "cn=Directory Manager" ldap://localhost:3389 backend create  
--suffix="dc=keenable,dc=in" --be-name="keenable"
```

```
The database was successfully created  
ldap389:/ # dsconf -D "cn=Directory Manager" ldap://localhost:3389 backend suffix list  
Enter password for cn=Directory Manager on ldap://localhost:3389:  
dc=keenable,dc=in (keenable)  
ldap389:/ #
```

Install Ldap utility on bash machine for run ldap command

```
sudo apt install ldap-utils
```

```
exit  
local@local:~/389ds/data$ sudo apt install ldap-utils  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
Suggested packages:  
  libsasl2-modules-gssapi-mit | libsasl2-modules-gssapi-heimdal  
The following NEW packages will be installed:  
  ldap-utils  
0 upgraded, 1 newly installed, 0 to remove and 322 not upgraded.  
Need to get 121 kB of archives.  
After this operation, 745 kB of additional disk space will be used.  
Get:1 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 ldap-utils amd64 2.4.49+dfsg-2ub  
untu1.9 [121 kB]  
57% [1 ldap-utils 41.7 kB/121 kB 34%]
```

Install OpenJDK (java Development kit)

```
sudo apt install openjdk-11-jdk
```

```
localuser@local:~$ sudo apt install openjdk-11-jdk
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  ca-certificates-java fonts-dejavu-extra java-common
  libatk-wrapper-java-jni libice-dev libpthread-stub0
  libx11-dev libxau-dev libxcb1-dev libxdmcp-dev libxext-dev
  openjdk-11-jdk-headless openjdk-11-jre openjdk-11-jre-headless
  x11proto-core-dev x11proto-dev xorg-sgml-doctools
Suggested packages:
  default-jre libice-doc libsm-doc libx11-doc libxcb-doc
  openjdk-11-demo openjdk-11-source visualvm fonts-ipafont-mincho
  fonts-wqy-microhei | fonts-wqy-zenfa
The following NEW packages will be installed:
```

Check java version

```
java -version
```

```
localuser@local:~$ java -version
openjdk version "11.0.20.1" 2023-08-24
OpenJDK Runtime Environment (build 11.0.20.1+1-post-Ubuntu-0ubuntu120.04)
```

2. Setup ApacheDirectory studio for ldap db UI

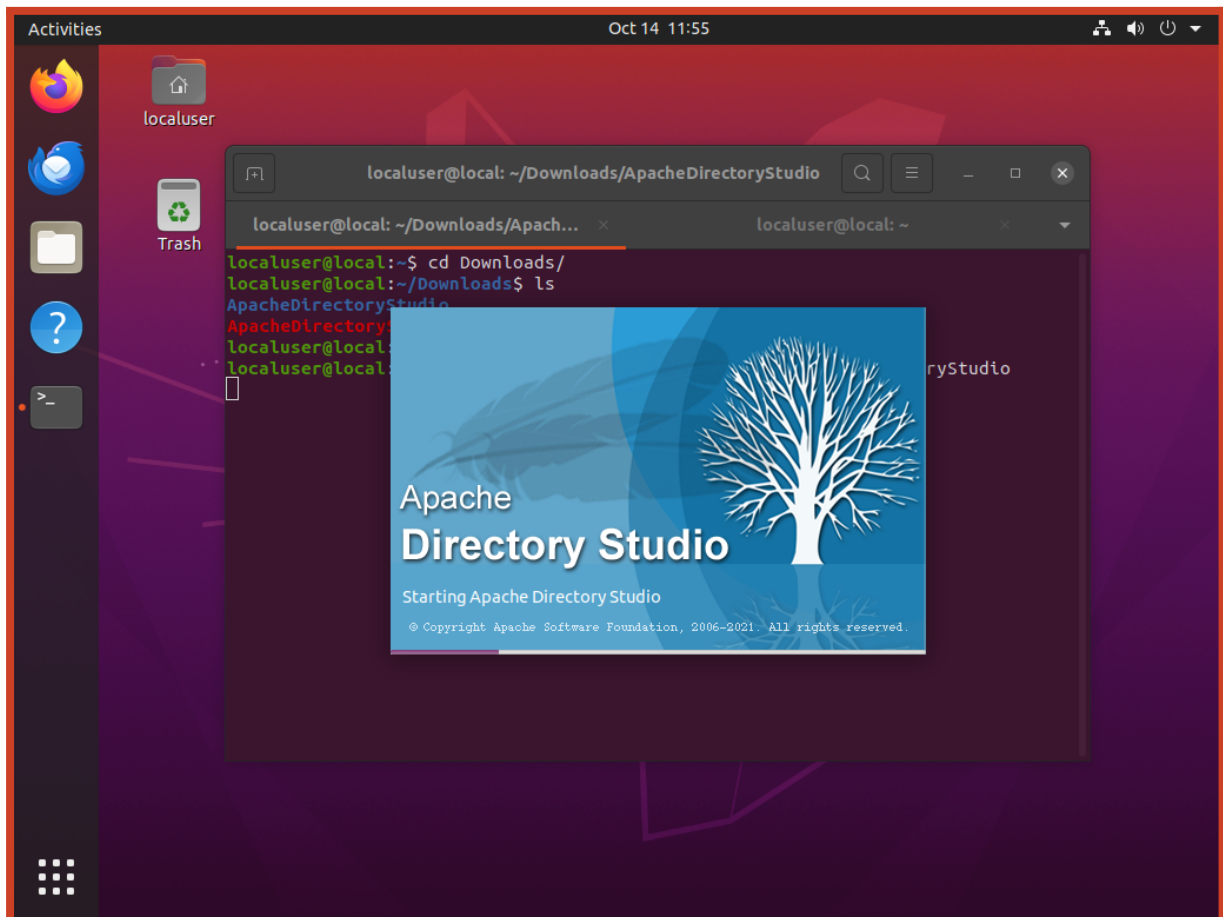
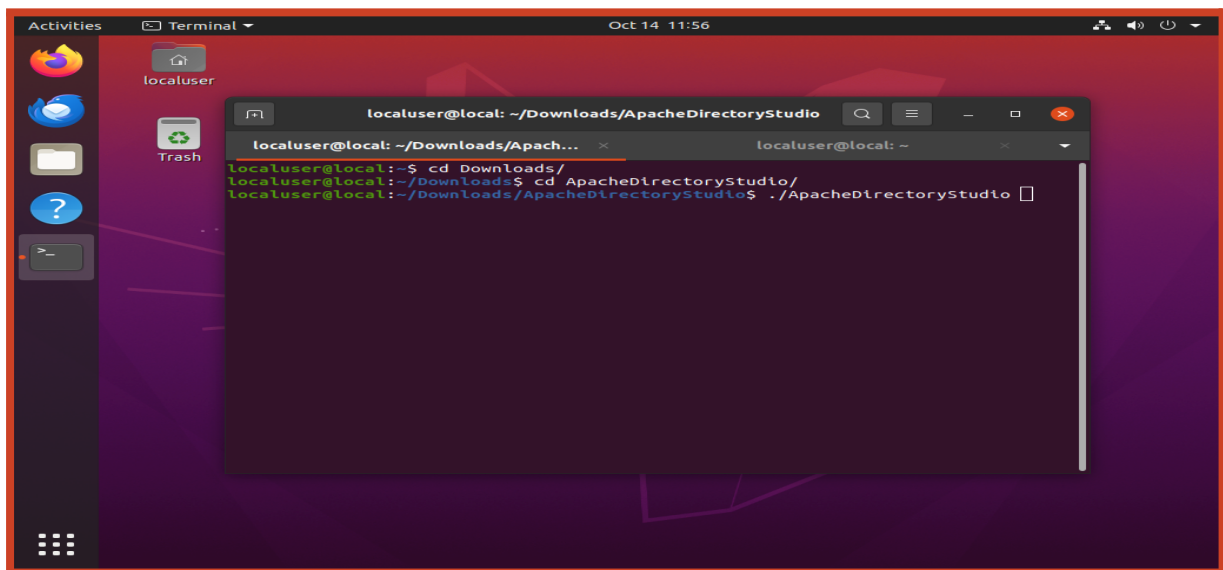
Install [ApacheDirectory studio tar](#) file and extract in directory

Untar the download tar file

```
tar -xvf
ApacheDirectoryStudio-2.0.0.v20210717-M17-linux.gtk.x86_64.tar.gz
```

Open apache directory

```
localuser@local:~$ cd Downloads/
Downloads$ cd ApacheDirectoryStudio/
Downloads/ApacheDirectoryStudio$ ./ApacheDirectoryStudio
```

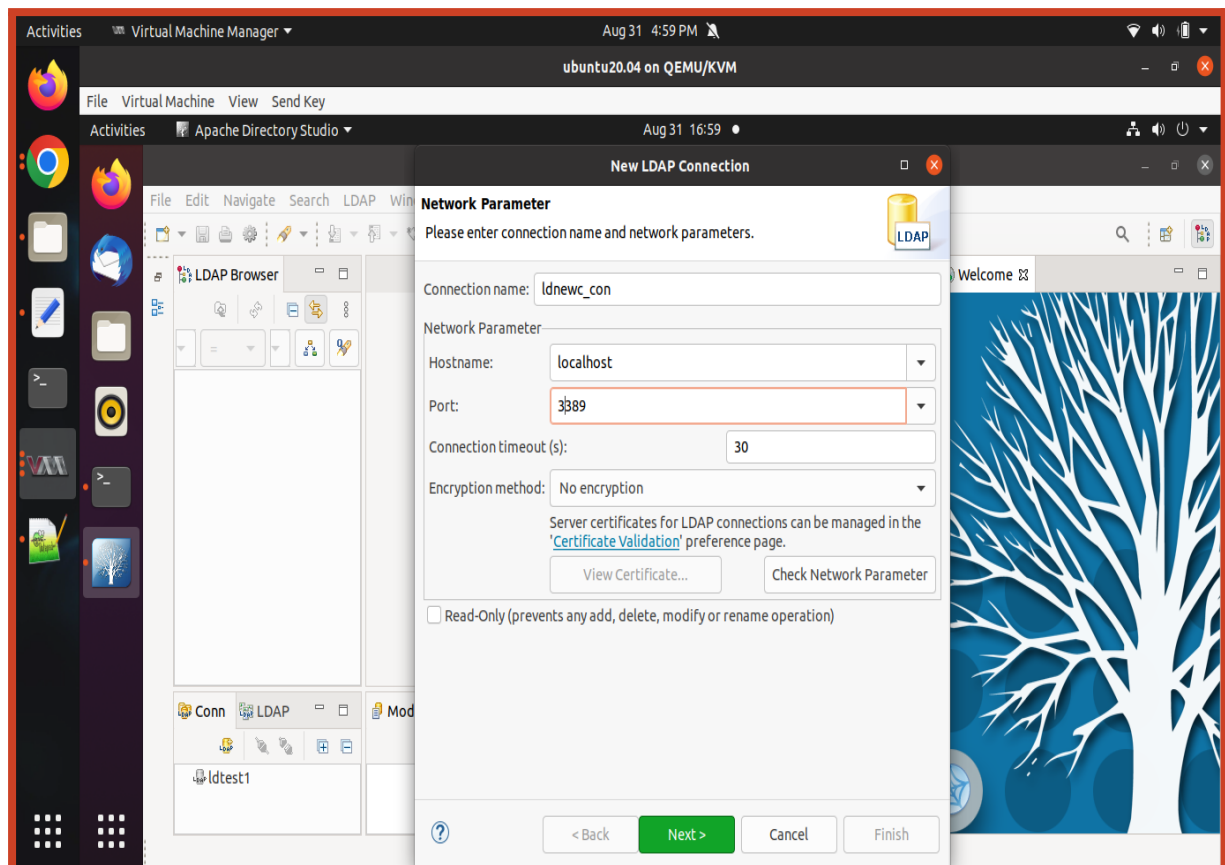


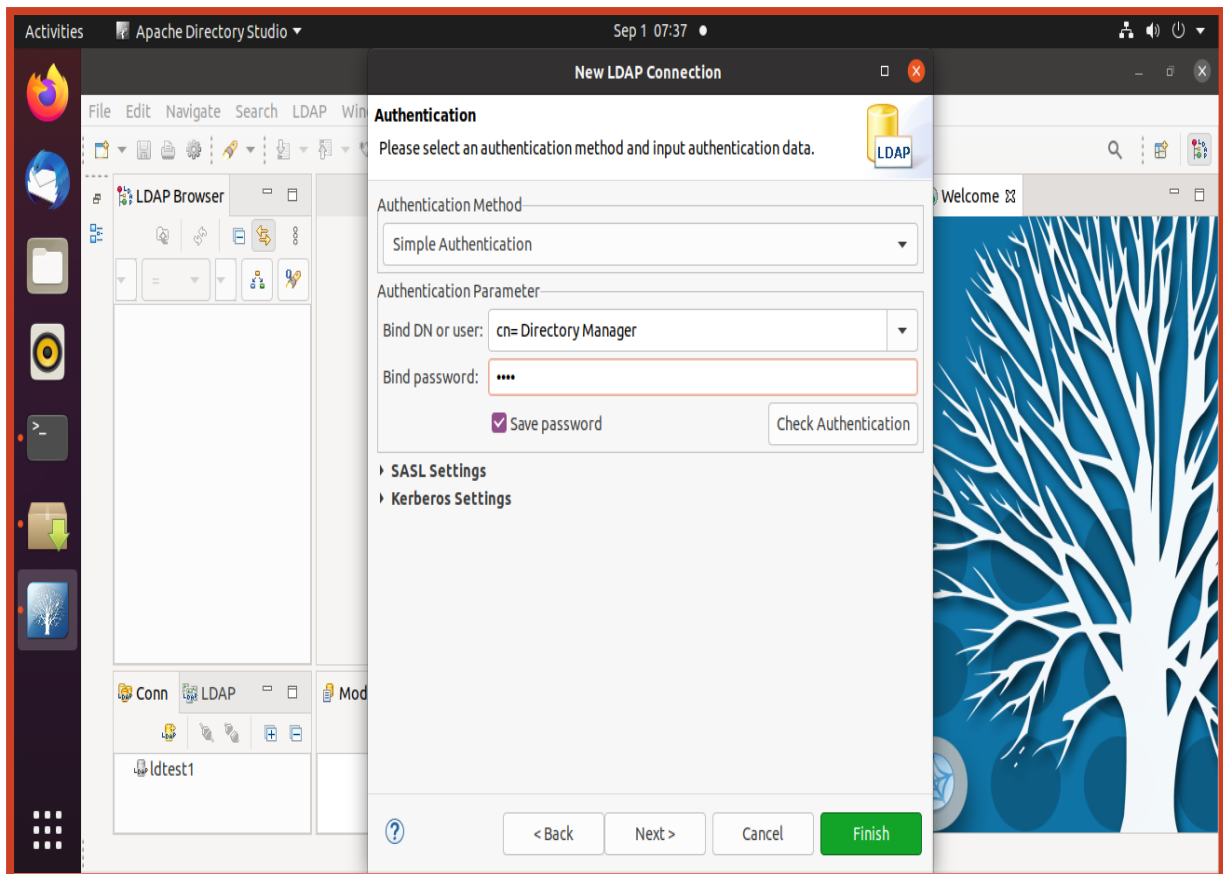
Create a New LDAP Connection

- In Apache Directory Studio, go to **File > New > LDAP Connection**.
- In the **"New LDAP Connection"** wizard, provide the connection details to your LDAP server:

Go to new connection and enter some details

```
ldap connection name -> ldap
Hostname -> localhost
ldap port -> Press next (3389)
Enter Bind DN name -> (cn= Directory Manager)
Enter Bind DN password -> (kavita)
Enter Bind DN password -> finish
```





3. We have create 5 organisation (Dev, Support, POC, Document, Observability)

Create Organization_unit Idif file (file extension name .ldif)

```
vim organisation.ldif
```

```
dn: dc=keenable,dc=in
objectClass: top
objectClass: domain
dc: keenable

dn: ou=Dev,dc=keenable,dc=in
objectClass: top
objectClass: organizationalUnit
ou: Dev

dn: ou=Support,dc=keenable,dc=in
objectClass: top
objectClass: organizationalUnit
ou: Support
dn: ou=POC,dc=keenable,dc=in
objectClass: top
objectClass: organizationalUnit
ou: POC

dn: ou=Document,dc=keenable,dc=in
objectClass: top
objectClass: organizationalUnit
ou: Document

dn: ou=Observability,dc=keenable,dc=in
objectClass: top
objectClass: organizationalUnit
ou: Observability
```

Run this command to add organisation.ldif

```
lovekavita@love:~/389ds/data/ldif$ ldapadd -a -c -xH
ldap://localhost:3389 -D "cn=Directory Manager" -w kavita -f
organisation.ldif
```

Create 2 group inside Support

```
vim group.ldif
```

```
dn: uid=001,ou=dev,dc=keenable,dc=
objectClass: top
objectClass: inetOrgPerson
```

```
objectClass: customEmployee
cn: kavita
sn: yadav
uid: 001
EmployeeCode: 101
userPassword: 12345@
personalemail-id: kavita.x.kyadav@fosteringlinux.com
mobileNo: 1213141500
documentssubmitted: yes
DateofJoining: 05-01-2022
Gender: Female
DateofBirth: 06-01-1998
Panno: ABCDH7654P
Qualification: MCA
YearsofQualification: 2021
ProfessionalStartYEARS: 2022
```

Run this command to add group.ldif

```
ldapadd -a -c -xH ldap://localhost:3389 -D "cn=Directory Manager" -w
kavita -f group.ldif
```

Run Some Command of ldap

First check how many default object class created

```
ldapsearch -o ldif-wrap=no -x -H ldap://localhost:3389 -D "cn=Directory
Manager" -w "kavita" -b "cn=schema" "(objectClass=subSchema)" -s sub
"objectClasses"
```

Check how many default attributes created

```
vim custom_attribute.ldif
```

```
dn: cn=schema
changetype: modify
add: attributeTypes
attributetypes: (emp_code-oid NAME 'EmployeeCode' DESC 'EmployeeCode')
```

```
EQUALITY caseIgnoreMatch SUBSTR caseExactSubstringsMatch SYNTAX
1.3.6.1.4.1.1466.115.121.1.15{10} SINGLE-VALUE X-ORIGIN 'user defined' )
attributetypes: (gender-oid NAME 'Gender' DESC 'Gender' EQUALITY
caseIgnoreMatch SUBSTR caseExactSubstringsMatch SYNTAX
1.3.6.1.4.1.1466.115.121.1.15{8} SINGLE-VALUE X-ORIGIN 'user defined')
attributetypes: (certifications-oid NAME 'Certifications' DESC
'Certifications' EQUALITY caseIgnoreMatch SUBSTR caseExactSubstringsMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{10} X-ORIGIN 'user defined')
attributetypes: (passport-oid NAME 'PassportNo' DESC 'PassportNo.'
EQUALITY caseIgnoreMatch SUBSTR caseExactSubstringsMatch SYNTAX
1.3.6.1.4.1.1466.115.121.1.15{10} SINGLE-VALUE X-ORIGIN 'user defined')
attributetypes: (pan_no-oid NAME 'Panno' DESC 'Panno.' EQUALITY
caseIgnoreMatch SUBSTR caseExactSubstringsMatch SYNTAX
1.3.6.1.4.1.1466.115.121.1.15{10} SINGLE-VALUE X-ORIGIN 'user defined')
attributetypes: (qualification-oid NAME 'Qualification' DESC
'Qualification' EQUALITY caseIgnoreMatch SUBSTR caseExactSubstringsMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{10} X-ORIGIN 'user defined')
attributetypes: (correspondence_address-oid NAME 'CorrespondenceAddress'
DESC 'CorrespondenceAddress' EQUALITY caseIgnoreMatch SUBSTR
caseExactSubstringsMatch SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{100}
SINGLE-VALUE X-ORIGIN 'user defined')
attributetypes: (personalemail-id-oid NAME 'personalemail-id' DESC
'personalemail-id' EQUALITY caseIgnoreMatch SUBSTR
caseExactSubstringsMatch SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{20}
SINGLE-VALUE X-ORIGIN 'user defined')
attributetypes: (facebookaccount-oid NAME 'facebookaccount' DESC
'facebookaccount' EQUALITY caseIgnoreMatch SUBSTR
caseExactSubstringsMatch SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{20}
SINGLE-VALUE X-ORIGIN 'user defined')
attributetypes: (twitteraccount-oid NAME 'twitteraccount' DESC
'twitteraccount' EQUALITY caseIgnoreMatch SUBSTR caseExactSubstringsMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{15} SINGLE-VALUE X-ORIGIN 'user
defined')
attributetypes: (MaritalStatus-oid NAME 'MaritalStatus' DESC
'MaritalStatus' EQUALITY caseIgnoreMatch SUBSTR caseExactSubstringsMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{10} SINGLE-VALUE X-ORIGIN 'user
defined')
attributetypes: (Childinfo-oid NAME 'Childinfo' DESC 'Childinfo' EQUALITY
caseIgnoreMatch SUBSTR caseExactSubstringsMatch SYNTAX
1.3.6.1.4.1.1466.115.121.1.15{20} X-ORIGIN 'user defined')
attributetypes: (pfno-oid NAME 'pfno' DESC 'pfno' EQUALITY
caseIgnoreMatch SUBSTR caseExactSubstringsMatch SYNTAX
1.3.6.1.4.1.1466.115.121.1.15{20} SINGLE-VALUE X-ORIGIN 'user defined')
attributetypes: (bankname-oid NAME 'BankName' DESC 'BankName' EQUALITY
caseIgnoreMatch SUBSTR caseExactSubstringsMatch SYNTAX
1.3.6.1.4.1.1466.115.121.1.15{20} SINGLE-VALUE X-ORIGIN 'user defined')
```

```
attributetypes: (AccountNo-oid NAME 'AccountNo' DESC 'AccountNo' EQUALITY
caseIgnoreMatch SUBSTR caseExactSubstringsMatch SYNTAX
1.3.6.1.4.1.1466.115.121.1.15{20} SINGLE-VALUE X-ORIGIN 'user defined')
attributetypes: (IFSCCode-oid NAME 'IFSCCode' DESC 'IFSCCode' EQUALITY
caseIgnoreMatch SUBSTR caseExactSubstringsMatch SYNTAX
1.3.6.1.4.1.1466.115.121.1.15{10} SINGLE-VALUE X-ORIGIN 'user defined')
attributetypes: (ESICCardNo-oid NAME 'ESICCardNo' DESC 'ESICCardNo'
EQUALITY caseIgnoreMatch SUBSTR caseExactSubstringsMatch SYNTAX
1.3.6.1.4.1.1466.115.121.1.15{20} SINGLE-VALUE X-ORIGIN 'user defined')
attributetypes: (FamilyMembersInsured-oid NAME 'FamilyMembersInsured'
DESC 'FamilyMembersInsured' EQUALITY caseIgnoreMatch SUBSTR
caseExactSubstringsMatch SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{10}
X-ORIGIN 'user defined')
attributetypes: (documentssubmitted-oid NAME 'documentssubmitted' DESC
'documentssubmitted' EQUALITY caseIgnoreMatch SUBSTR
caseExactSubstringsMatch SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{60}
X-ORIGIN 'user defined')
attributetypes: (doj-oid NAME 'DateofJoining' DESC 'DateofJoining'
EQUALITY caseIgnoreIA5Match SUBSTR caseIgnoreIA5SubstringsMatch SYNTAX
1.3.6.1.4.1.1466.115.121.1.26{8} SINGLE-VALUE X-ORIGIN 'user defined')
attributetypes: (doreg-oid NAME 'DateOfResignation' DESC
'DateOfResignation' EQUALITY caseIgnoreIA5Match SUBSTR
caseIgnoreIA5SubstringsMatch SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{8}
SINGLE-VALUE X-ORIGIN 'user defined')
attributetypes: (dob-oid NAME 'DateofBirth' DESC 'DateofBirth' EQUALITY
caseIgnoreIA5Match SUBSTR caseIgnoreIA5SubstringsMatch SYNTAX
1.3.6.1.4.1.1466.115.121.1.26{8} SINGLE-VALUE X-ORIGIN 'user defined')
attributetypes: (passport_valid_upto-oid NAME 'PassportValidupto' DESC
'PassportValidupto' EQUALITY caseIgnoreIA5Match SUBSTR
caseIgnoreIA5SubstringsMatch SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{8}
SINGLE-VALUE X-ORIGIN 'user defined')
attributetypes: (ProfessionalStartYEARS-oid NAME 'ProfessionalStartYEARS'
DESC 'ProfessionalStartYEARS' EQUALITY caseIgnoreIA5Match SUBSTR
caseIgnoreIA5SubstringsMatch SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{8}
SINGLE-VALUE X-ORIGIN 'user defined')
attributetypes: (YEARSOfExperience-oid NAME 'YEARSOfExperience' DESC
'YEARSOfExperience' EQUALITY caseIgnoreIA5Match SUBSTR
caseIgnoreIA5SubstringsMatch SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{8}
SINGLE-VALUE X-ORIGIN 'user defined')
attributetypes: (dateofjoiningasintern-oid NAME 'dateofjoiningasintern'
DESC 'dateofjoiningasintern' EQUALITY caseIgnoreIA5Match SUBSTR
caseIgnoreIA5SubstringsMatch SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{8}
SINGLE-VALUE X-ORIGIN 'user defined')
attributetypes: (AadharNo-oid NAME 'AadhaarNo' DESC 'AadhaarNo' EQUALITY
integerMatch SUBSTR caseIgnoreIA5SubstringsMatch SYNTAX
1.3.6.1.4.1.1466.115.121.1.27{20} SINGLE-VALUE X-ORIGIN 'user defined')
```

```

attributetypes: (YearsofQualification NAME 'YearsofQualification' DESC
'YearsofQualification' EQUALITY integerMatch SYNTAX
1.3.6.1.4.1.1466.115.121.1.27{10} SINGLE-VALUE X-ORIGIN 'user defined')
attributetypes: (mobilenoid NAME 'mobilenoid' DESC 'mobilenoid' EQUALITY
integerMatch SYNTAX 1.3.6.1.4.1.1466.115.121.1.27{20} X-ORIGIN 'user
defined')
attributetypes: (UANnooid NAME 'UANno' DESC 'UANno' EQUALITY
integerMatch SYNTAX 1.3.6.1.4.1.1466.115.121.1.27{20} SINGLE-VALUE
X-ORIGIN 'user defined')
attributetypes: (InsuranceMonthlyAmountDeductionINRoid NAME
'InsuranceMonthlyAmountDeductionINR' DESC
'InsuranceMonthlyAmountDeductionINR' EQUALITY integerMatch SYNTAX
1.3.6.1.4.1.1466.115.121.1.27{10} SINGLE-VALUE X-ORIGIN 'user defined')
attributetypes: ( projectnameoid NAME 'ProjectName' DESC 'ProjectName'
EQUALITY caseIgnoreMatch SUBSTR caseExactSubstringsMatch SYNTAX
1.3.6.1.4.1.1466.115.121.1.15{50} SINGLE-VALUE X-ORIGIN 'user defined')

```

Add this file to ldap db

```

ldapadd -a -c -xH ldap://localhost:3389 -D "cn=Directory Manager" -w
kavita -f custom_attribute.ldif

```

Create Object Class file for add attribute to Object Class

```

vim object_class.ldif

```

```

dn: cn=schema
changetype: modify
add: objectClasses
objectClasses: ( customEmployeeoid NAME 'customEmployee' SUP top STRUCTURAL
MUST ( EmployeeCode $ DateofJoining $ Gender $ DateofBirth $ Panno $
Qualification $ YearsofQualification $ ProfessionalStartYEARS $
YEARSOfExperience $ CorrespondenceAddress $ personalemail-id $ mobilenoid $
MaritalStatus $ BankName $ AccountNo $ IFSCCode $ documentssubmitted) MAY (
DateOfResignation $ Certifications $ PassportNo $ PassportValidupto $ AadhaarNo
$ facebookaccount $ twitteraccount $ Childinfo $ pfno $ UANno $ ESICCardNo $
InsuranceMonthlyAmountDeductionINR $ FamilyMembersInsured $
dateofjoiningasintern $ ProjectName ) X-ORIGIN 'user defined')

```

Add object class ldif file

```
ldapadd -a -c -x -H ldap://localhost:3389 -D "cn=Directory Manager" -W  
-f object_class.ldif
```

Create user with custom attribute

```
vim user1.ldif
```

```
dn: cn=Admins,ou=Support,dc=keenable,dc=in  
objectClass: posixGroup  
cn: Test  
gidNumber: 4000  
  
dn: uid=user1,ou=dev,dc=keenable,dc=in  
objectClass: inetOrgPerson  
objectClass: posixAccount  
objectClass: shadowAccount  
uid: user1  
sn: kavi  
givenName: user1  
cn: user1  
uidNumber: 5001  
gidNumber: 5000  
userPassword: kavita  
loginShell: /bin/bash  
homeDirectory: /home/user1
```

In this file we have gave object class name and must custom attribute

After that add user to db

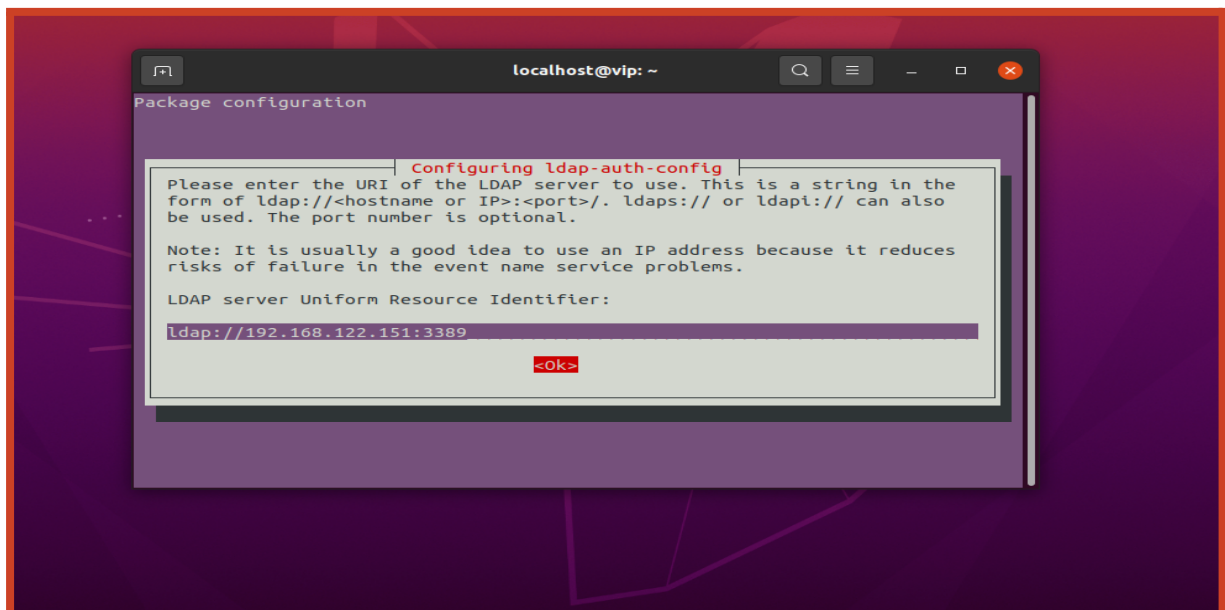
```
ldapadd -a -c -x -H ldap://localhost:3389 -D "cn=Directory Manager" -W  
-f user1.ldif
```

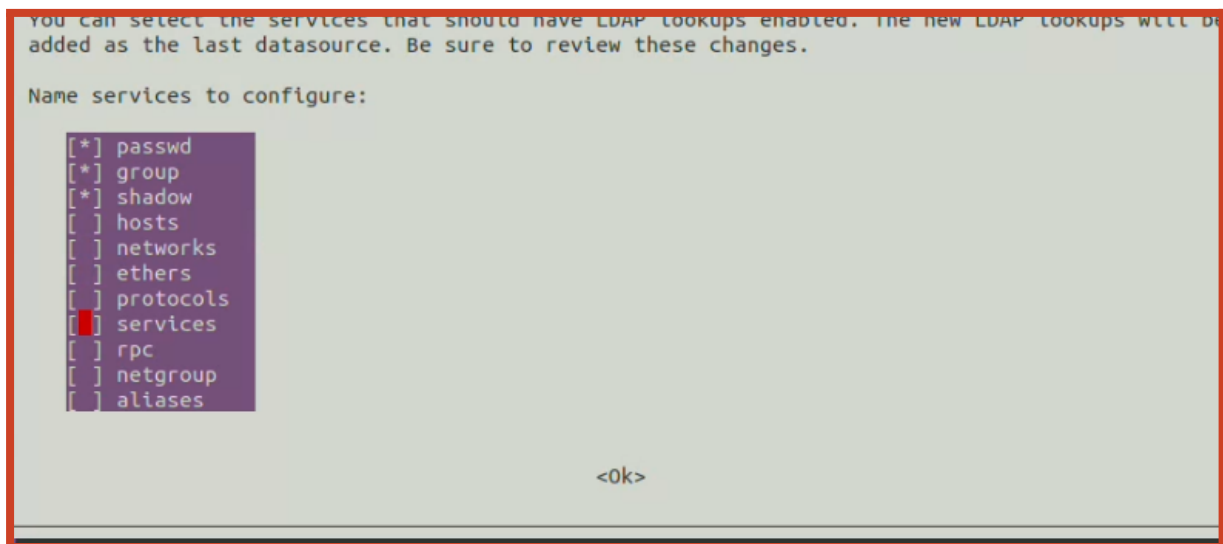
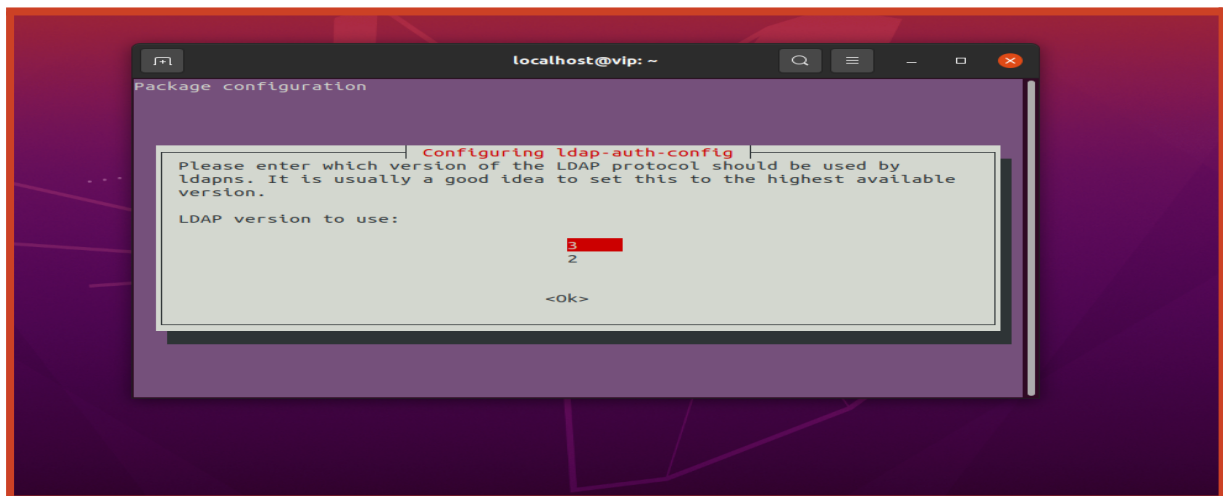
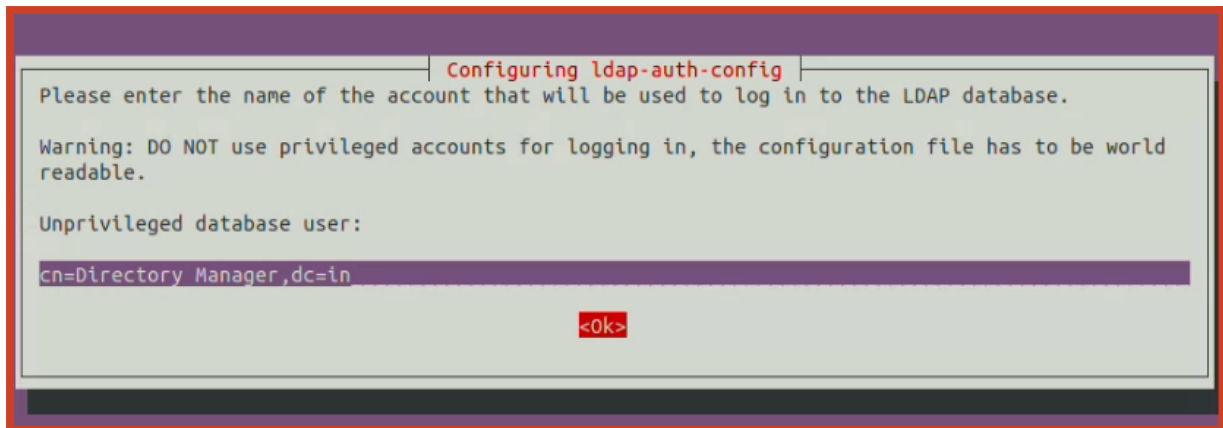
Check user reflected or not through ldapsearch command


```
ldapsearch -x -D "cn=Directory Manager" -W -H ldap://localhost:3389 -b  
"ou=dev,dc=keenable,dc=in" -s sub "(uid=user1)"
```

4. Setup ldap Client on another VM

```
sudo apt install libnss-ldapd libpam-ldapd ldap-utils
```





After run this command we get one pop up screen

Enter LDAP URI: IP address or hostname

Enter Set a Distinguished name(dn) of the search base

Go inside file **/etc/nslcd.conf** and check

Open **/etc/nslcd.conf** and check configuration

Output

```
# /etc/nslcd.conf
# nslcd configuration file. See nslcd.conf(5)
# for details.

# The user and group nslcd should run as.
uid nslcd
gid nslcd

# The location at which the LDAP server(s) should be reachable.
uri ldap://192.168.122.109:3389

# The search base that will be used for all queries.
base dc=keenable,dc=in

# The LDAP protocol version to use.
#ldap_version 3

# The DN to bind with for normal lookups.
binddn cn=Directory Manager
bindpw kavita

# The DN used for password modifications by root.
#rootpwmoddn cn=admin,dc=example,dc=com

# SSL options
#ssl off
#tls_reqcert never
tls_cacertfile /etc/ssl/certs/ca-certificates.crt

# The search scope.
#scope sub
```

Restart nslcd and nscd service

```
sudo systemctl restart nslcd
```

```
sudo systemctl restart nscd
```

```
reboot
```

Run getent passwd command to check server user reflect or not

```
getent passwd
```

```
client@client:~$ getent passwd
```

Output

```
user1:x:5000:5000:user1:/home/user1:/bin/bash
```

```
nslcd:x:127:134:nslcd name service LDAP connection daemon,,,:/
user1:x:5001:5000:user1:/home/user1:/bin/bash
client@client:~$
```

Reference Link:-

For Understand Ldap:-

<https://www.windows-active-directory.com/active-directory-ldap.html>

For Attribute Syntax:- <https://ldap.com/attribute-syntaxes/>

For Client Setup:-

https://computingforgeeks.com/how-to-configure-ubuntu-as-ldap-client/?expand_article=1&expand_article=1

