Join Ubuntu Workstation to a Windows Domain

By wolffhaven | June 7, 2016 | Active Directory, Linux, Ubuntu 16.04

Here is my guide on how to Join Ubuntu Workstation to a Windows Domain using SSSD and Realmd. There are a few different methods out there on how to do this but from what I've tested and researched, using SSSD and Realmd is the most up to date and easiest way to achieve the desired result at the time of writing this. I've included links to all of the relevant documentation that I used in researching putting together this guide.

I just want to say off the bat, that I'm no Linux expert. I've only recently started to dabble with Linux. I wanted to see if this could be done so I tried it out in my test lab. I created this guide for myself so that I could use it again later when I no doubt forget how I originally done it in the first place. I couldn't really find an up to date step by step guide to join Ubuntu Workstation to a Windows Domain that was easy to follow for beginners so I'm putting this up on my site in the hope that it may help someone else. If you see any glaringly obvious mistakes or if there is a better way of doing something let me know in the comments. This isn't really Ubuntu specific as a lot of the steps from this guide have been adapted from the Redhat and Fedora documentation. If you are here following this guide, I'd say try it out in a test environment first to make sure it does everything that you need.

So in my test lab I went through and tested few different methods on how to go about joining a Ubuntu 16.04 computer to a Windows Domain. The different methods I tried were: –

•Winbind

•SSSD

•RealmD & SSSD

As I said earlier, I found that for a new linux user, the RealmD & SSSD method to Join Ubuntu Workstation to a Windows Domain, was the easiest and most effective. Your mileage may vary.

I'll split this guide up in to separate sections.

1.Configuring the hosts file.

2.Setting up the resolv.conf file.

3.Setting up NTP.

4.Installing the required packages.

5.Configuring the Realmd.conf file.

6.Fixing a bug with the packagekit package.

7.Joining the Active Directory Domain.

8.Configuring the SSSD.conf file.

9.Locking down which Domain Users can login.

10.Granting Sudo access.

11.Configuring home directories.

12.Configuring LightDM.

13.Final Thoughts & Failures

14.Links

1. Configuring the hosts file

To update the hosts file edit the /etc/hosts file. On my workstation, by default the fully qualified domain name wasn't in the hosts file so I had to add it. Note: Coming from
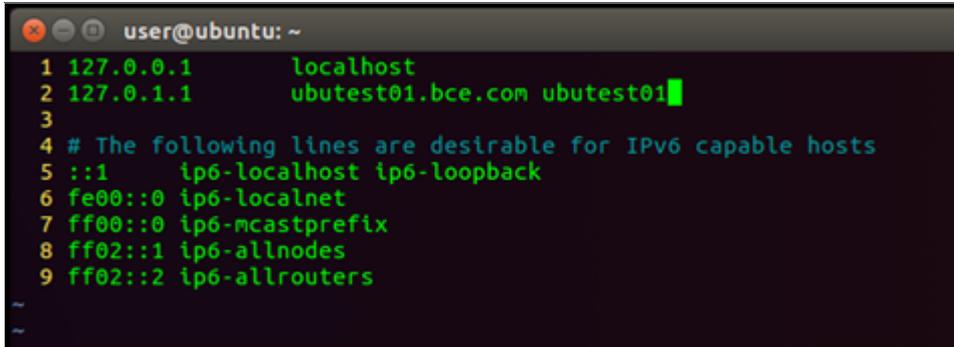
Windows I'd never seen a 127.0.1.1 address used as a loopback address. Seems legit though.

In this example the hostname of the workstation I want to join to the domain is ubutest01.

```
1 sudo vi /etc/hosts
```

Set the 127.0.1.1 address to your new hostname in the following format.
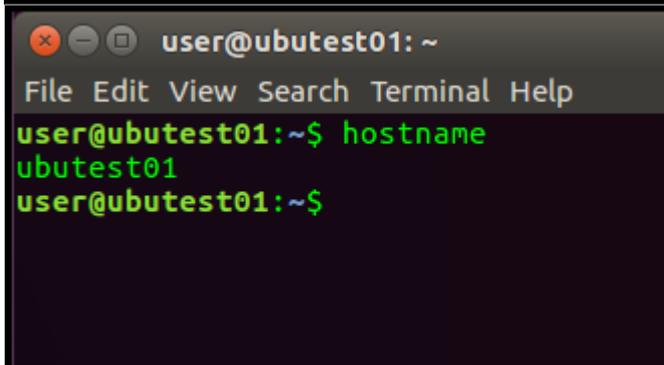
```
127.0.1.1 ubutest01.bce.com ubutest01
```


Reboot the system for the changes to take effect.

```
1 sudo reboot
```

To test if the name has been changed:

```
1 hostname
```


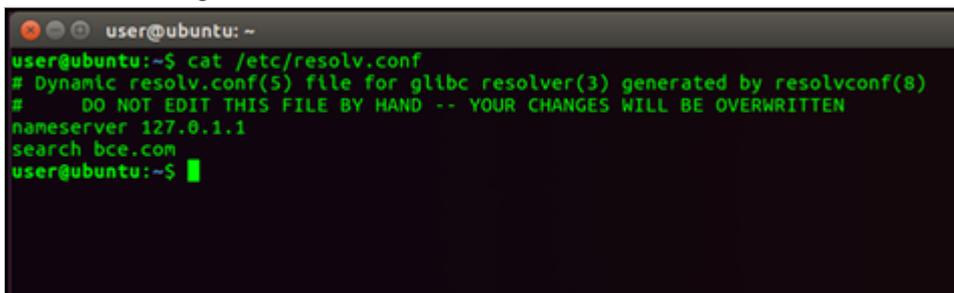## 2. Setting up the resolv.conf file

Make sure you're Ubuntu computer can talk to your DNS Servers. By default, the resolv.conf will be set like the following:

To change it to have the actual DNS servers that you are using do the following:

```
1 sudo vi /etc/Networkmanager/NetworkManager.conf
```

Comment out the dns=dnsmasq line.

```
#dns=dnsmasq
```

```
[main]
plugins=ifupdown,keyfile,ofono
#dns=dnsmasq

[ifupdown]
managed=false
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
-- INSERT --
```

Then restart the network manager.

```
1  sudo systemctl restart network-manager.service
```

If you have set the dns servers via the GUI you should then see them in the resolv.conf file.

```
1  cat /etc/resolv.conf
```



```
user@ubuntu:~$ cat /etc/resolv.conf
# Dynamic resolv.conf(5) file for glibc resolver(3) generated by resolvconf(8)
#       DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN
nameserver 127.0.1.1
search bce.com
user@ubuntu:~$ sudo vi /etc/NetworkManager/NetworkManager.conf
user@ubuntu:~$ sudo systemctl restart network-manager.service
user@ubuntu:~$ cat /etc/resolv.conf
# Dynamic resolv.conf(5) file for glibc resolver(3) generated by resolvconf(8)
#       DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN
nameserver 192.168.1.4
nameserver 192.168.1.200
search bce.com
user@ubuntu:~$
```

Check that you can resolve the SRV records for the domain by running the following:

```
1  dig –t SRV _ldap._tcp.bce.com | grep –A2 "ANSWER SECTION"
```

3. Setting up NTP

It's important to synchronize time with your Domain Controllers so Kerberos works correctly. Install NTP.

```
1  sudo apt -y install ntp
```

Edit the vi ntp.conf file.

```
1 sudo vi /etc/ntp.conf
```

Comment out the ubuntu servers and put your own dc's in there. For example: –

```
server dc.bce.com iburst prefer
```


Restart the ntp service.

```
1 sudo systemctl restart ntp.service
```

Then to check if it's working try running:

```
1 ntpq –p
```


During this process I found this little tip. This is a handy tool to make sure your syncing correctly:

```
1 sudo apt -y install ntpstat
```

Then run:

```
1 ntpstat
```

Should be syncing like a boss.

```
user@ubuntu:~$ ntpstat
synchronised to NTP server (192.168.1.4) at stratum 4
   time correct to within 270 ms
   polling server every 64 s
user@ubuntu:~$ 
```
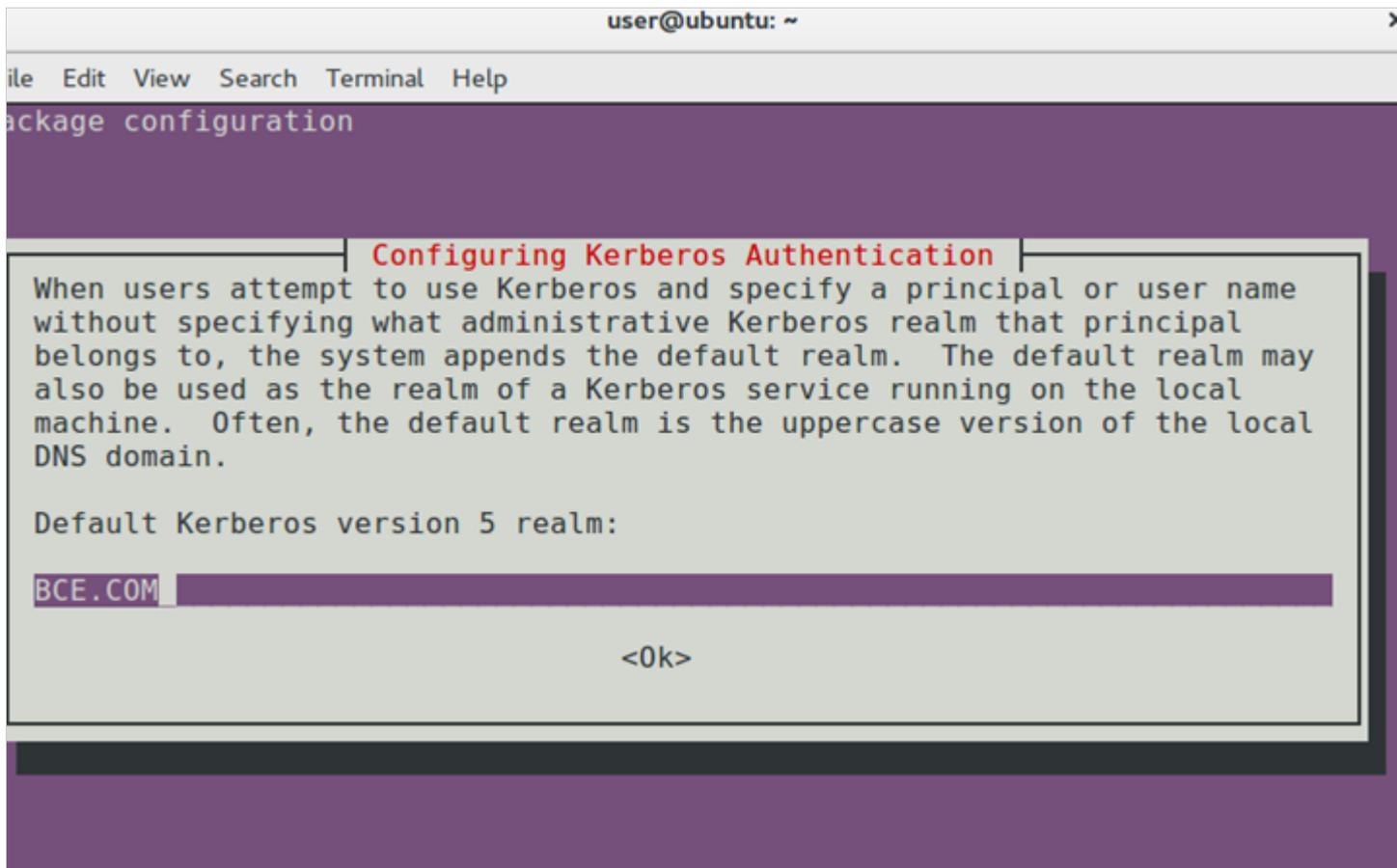
4. Installing the required packages.

Install the necessary packages:

```
1 sudo apt -y install realmd sssd adcli libwbclient-sssd krb5-user sssd-tools
  samba-common packagekit samba-common-bin samba-libs
```
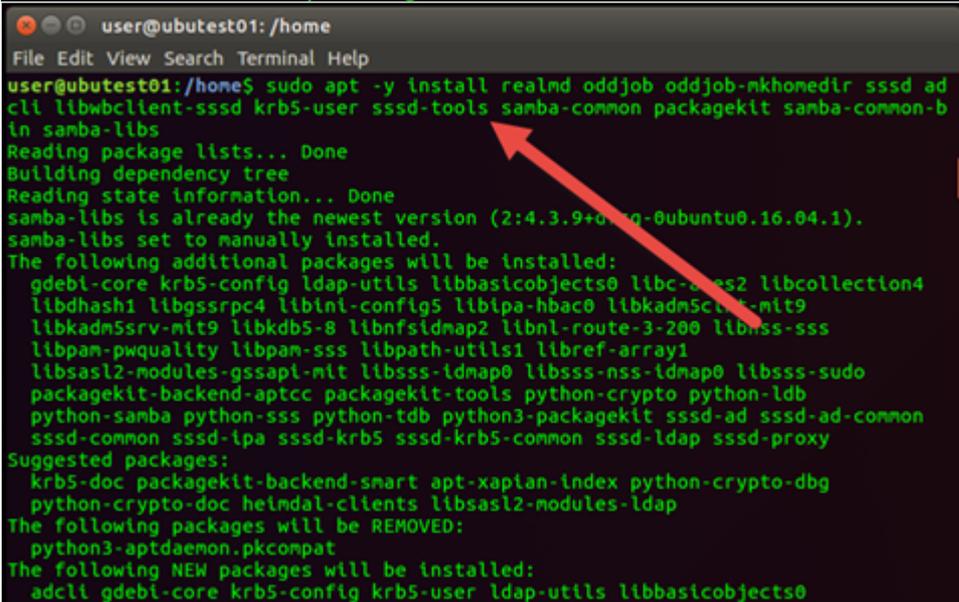
```
user@ubutest01: /home
File  Edit  View  Search  Terminal  Help
user@ubutest01:/home$ sudo apt -y install realmd oddjob oddjob-mkhomedir sssd ad
cli libwbclient-sssd krb5-user sssd-tools samba-common packagekit samba-common-b
in samba-libs
Reading package lists... Done
Building dependency tree
Reading state information... Done
samba-libs is already the newest version (2:4.3.9+dfsg-0ubuntu0.16.04.1).
samba-libs set to manually installed.
The following additional packages will be installed:
   gdebi-core krb5-config ldap-utils libbasicobjects0 libc-ares2 libcollection4
   libdhash1 libgssrpc4 libini-config5 libipa-hbac0 libkadm5clnt-mit9
   libkadm5srv-mit9 libkdb5-8 libnfsidmap2 libnl-route-3-200 libnss-sss
   libpam-pwquality libpam-sss libpath-utils1 libref-array1
   libsasl2-modules-gssapi-mit libsss-idmap0 libsss-nss-idmap0 libsss-sudo
   packagekit-backend-aptcc packagekit-tools python-crypto python-ldb
   python-samba python-sss python-tdb python3-packagekit sssd-ad sssd-ad-common
   sssd-common sssd-ipa sssd-krb5 sssd-krb5-common sssd-ldap sssd-proxy
Suggested packages:
   krb5-doc packagekit-backend-smart apt-xapian-index python-crypto-dbg
   python-crypto-doc heimdal-clients libsasl2-modules-ldap
The following packages will be REMOVED:
   python3-aptdaemon.pkcompat
The following NEW packages will be installed:
   adcli gdebi-core krb5-config krb5-user ldap-utils libbasicobjects0
```

If you are presented with the following screen, put the domain name in CAPITALS.

ile   Edit   View   Search   Terminal   Help

ackage configuration

Configuring Kerberos Authentication

When users attempt to use Kerberos and specify a principal or user name
without specifying what administrative Kerberos realm that principal
belongs to, the system appends the default realm. The default realm may
also be used as the realm of a Kerberos service running on the local
machine. Often, the default realm is the uppercase version of the local
DNS domain.

Default Kerberos version 5 realm:

BCE.COM

<Ok>

5. Configuring the Realmd.conf file

Make the following changes to the realmd.conf file before using realmd to join the domain. This will make domain users have their home directory in the format /home/user. By default it will be /home/domain/user. You might want it like this, I do not. If you want to read more about these options you can do that here.

Note: If you are going to have your domain users not use fully-qualified domain names, then you may run in to issues if you have a local linux user with the same account name as the active directory account name.

```
1 sudo vi /etc/realmd.conf
[active-directory]
os-name = Ubuntu Linux
os-version = 16.04
[service]
automatic-install = yes
 [users]
default-home = /home/%u
default-shell = /bin/bash
[bce.com]
user-principal = yes
fully-qualified-names = no
```

```
1 [active-directory]
2 os-name = Ubuntu Linux
3 os-version = 16.04
4
5 [service]
6 automatic-install = yes
7
8 [users]
9 default-home = /home/%U
10 default-shell = /bin/bash
11
12 [bce.com]
13 user-principal = yes
14 fully-qualified-names = no
```

6. Fix a bug with the Packagekit package.

There is a bug with the packagekit package in Ubuntu 16.04. You will need to do this as a workaround otherwise it will hang when you try to join the domain.

Note: I had to this when I originally wrote this guide in May of 2016. This may have been fixed by the time you are reading this. I thought I'd put it in just in case.

```
1 sudo add-apt-repository ppa:xtrusia/packagekit-fix
2
3 sudo apt update
4
5 sudo apt upgrade packagekit
```


7. Join Ubuntu Workstation to a Windows Domain.

Now, it's time to join the domain. Check that realm can discover the domain you will be joining.

```
1 realm discover
```

Create the kerberos ticket that will be used the domain user that has privileges to join the domain.

```
1 kinit -V a_craig
```


Now you can join the domain using realmd.

```
1 sudo realm --verbose join -U a_craig bce.com
```


To do a quick test to see if it's worked:

```
1 id craig
```

```
user@ubutest01: /home
File Edit View Search Terminal Help
user@ubutest01:/home$ id craig
uid=1562601105(craig) gid=1562600513(domain users) groups=1562600513(domain user
s),1562601615(linuxadmins),1562602106(kodishares),1562601601(downloadredirection
),1562601616(adminshares)
user@ubutest01:/home$
```

This is all the Domain Groups that the domain user Craig belongs to. It's worked HUZZAH!

OK, now that's done. Lets tweak!

8. Configuring the SSSD.conf file.

I'd like to enable Dynamic DNS and some other features that I couldn't set via the realmd.conf file. We now have the opportunity to tweak these settings in the sssd.conf file. I've added the following:

```
1 sudo vi /etc/sssd/sssd.conf
```
auth_provider = ad

chpass_provider = ad

access_provider = ad

ldap_schema = ad

dyndns_update = true

dyndsn_refresh_interval = 43200

dyndns_update_ptr = true

dyndns_ttl = 3600

9. Locking down which Domain Users can login.

Now, let's restrict which domain users can login.

```
1 sudo realm deny -R bce.com -a
```

I want users specified in a specific group to be able to login, as well as the domain admins.

```
1 sudo realm permit -R bce.com -g linuxadmins domain\ admins
```


10. Granting Sudo Access.

Now lets grant some sudo access.

```
1 sudo visudo
```

```
GNU nano 2.5.3                    File: /etc/sudoers.tmp                    Modified

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL

# Members of the admin group may gain root privileges
%admin ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "#include" directives:

#includedir /etc/sudoers.d

%linuxadmins ALL=(ALL:ALL) ALL
%domain\ admins ALL=(ALL:ALL) ALL

^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit       ^R Read File  ^\ Replace    ^U Uncut Text ^T To Spell   ^_ Go To Line
```

11. Configuring home directories.

Lets setup the home directory for domain users logging in.

```
1 sudo vi /etc/pam.d/common-session
```

Add to the bottom:

```
session required                        pam_mkhomedir.so
skel=/etc/skel/ umask=0022
```



```
 8 #
 9 # As of pam 1.0.1-6, this file is managed by pam-auth-update by default.
10 # To take advantage of this, it is recommended that you configure any
11 # local modules either before or after the default block, and use
12 # pam-auth-update to manage selection of other modules.  See
13 # pam-auth-update(8) for details.
14
15 # here are the per-package modules (the "Primary" block)
16 session [default=1]                     pam_permit.so
17 # here's the fallback if no module succeeds
18 session requisite                       pam_deny.so
19 # prime the stack with a positive return value if there isn't one already;
20 # this avoids us returning an error just because nothing sets a success code
21 # since the modules above will each just jump around
22 session required                        pam_permit.so
23 # The pam_umask module will set the umask according to the system default in
24 # /etc/login.defs and user settings, solving the problem of different
25 # umask settings with different shells, display managers, remote sessions et
   c.
26 # See "man pam_umask".
27 session optional                        pam_umask.so
28 # and here are more per-package modules (the "Additional" block)
29 session required        pam_unix.so
30 session optional                        pam_sss.so
31 session optional        pam_systemd.so
32 session required        pam_mkhomedir.so skel=/etc/skel/ umask=0022
33 # end of pam-auth-update config
                                                        33,31        Bot
```
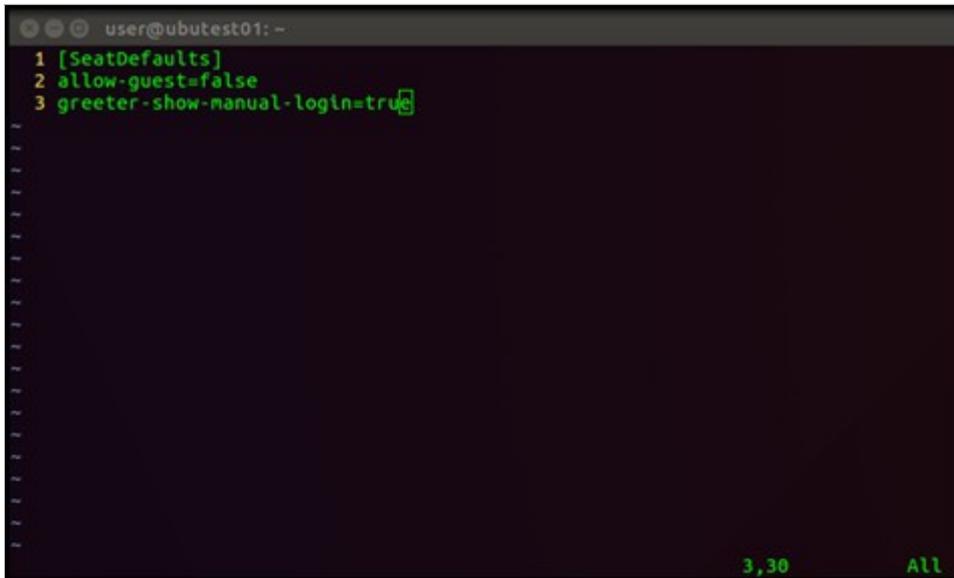
12. Configure Lightdm

The last thing I want to do is edit the lightdm conf file so that I can log in with a domain user at the login prompt.

```
1 sudo vi /etc/lightdm/lightdm.conf
```
[SeatDefaults]
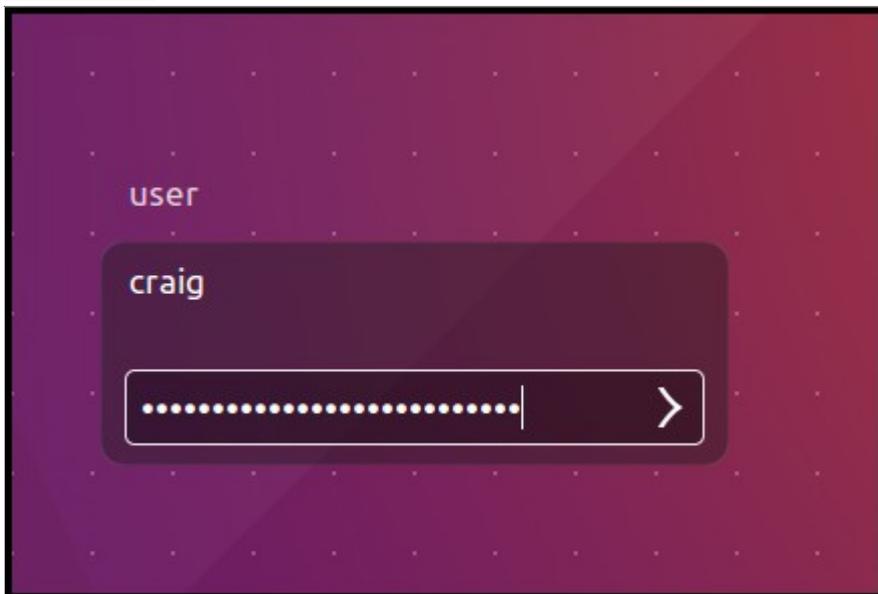allow-guest=false
greeter-show-manual-login=true

I think that's all the tweaking I'm going to do. I'm going to reboot and see if I can login.
```
1 sudo reboot
```
Once the login screen pops up you should be able to manually login. Click login.

I can log in. Huzzah!

13. Final Thoughts & Failures

This was a fun process and I learned a lot about Ubuntu and Linux in creating this guide. There were a few failures however so it wasn't all smooth sailing.

Dynamic DNS

So after all that, I still had issues with Dynamic DNS. I researched this as much as I could but couldn't find a resolution. I manually added the A records on my DNS server but I'd really like to get Dynamic

DNS working. If anyone knows where I have gone wrong or can point out how to get this working please leave a comment.

SAMBA File Sharing

I also had some issues after this with getting SAMBA/CIFS File sharing working with Windows Authentication. I would like to be able to share a folder in Ubuntu to Windows Users and have the Windows Users authtencticate to the Ubuntu share with their Windows credentials. I've spent a fair bit of time trying to find a resolution to this and played a bit with ACLS in Ubuntu as well but couldn't get it working properly. I put this down to being fairly new to Linux and not fully understanding some of the intricacies with SAMBA and Linux authentication. If anyone can point me in the right direction for getting SAMBA File Sharing working please leave me a comment.

14. Links

Below are the links that I used when researching this guide.

SSSD-AD Man Page

http://linux.die.net/man/5/sssd-ad

SSSD.Conf Man Page

http://linux.die.net/man/5/sssd.conf

SSSD-KRB5 Man Page

http://linux.die.net/man/5/sssd-krb5

SSSD-SIMPLE Man Page

http://linux.die.net/man/5/sssd-simple

PAM_SSS Module Man Page

http://linux.die.net/man/8/pam_sss

SSSD – Fedora

https://fedorahosted.org/sssd/

Redhat – Ways to Integrate Active Directory and Linux Environments

https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/Windows_Integration_Guide/introduction.html

Redhat – Using Realmd to Connect to an Active Directory Domain

https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/Windows_Integration_Guide/ch-Configuring_Authentication.html

Realm Man Page

http://manpages.ubuntu.com/manpages/trusty/man8/realm.8.html

Realmd.conf Man Page

http://manpages.ubuntu.com/manpages/trusty/man5/realmd.conf.5.html

Correcting DNS issue by editing Resolv.Conf file

http://askubuntu.com/questions/201603/should-i-edit-my-resolv-conf-file-to-fix-wrong-dns-problem