LAB 4: INSTALL AN AD DS DOMAIN CONTROLLER TO CREATE A SINGLE DOMAIN FOREST

PRAMOD NEPAL

02/22/16

CSCI 5417

EAST TENNESSEE STATE UNIVERSITY

PURPOSE

The main focus of this lab was to learn to install an Active Directory DS Controller to create a single Domain Forest. We also installed an application called Webmin in Ubuntu, that can be used to configure the OS through a web interface.

MATERIALS

- Lab Instructions
- Web browser
- PuTTY
- AWS console

PROCEDURE

Setup Windows Server 2012 Forest

From AWS console we launched both of the Windows server instances that was created in the previous labs. I selected the **NepalWindowsDC01** instance to be the domain controller and **NepalWindowsMS01** instance to be the first member server. After the AWS console shows the public IP address we can RDP into the Domain Controller. If it prompted to install updates we can do so while proceeding to do the remaining tasks. The first task we did was change the server's name. We used the metro interface to browse to the System window from lower left corner. We then clicked the 'Change settings' link. We again used the naming convention we have been using throughout the labs. I choose the server name as **NepalWinDC01**. We also made sure that the the length of the name was no longer than 15 characters. We adhered to this restriction because of Linux limitations. Once we changed the name in System Properties we clicked 'OK' to apply the new name.

We had to restart the server for the name to take effect. We used the metro interface's Shutdown button (using reason 'Other (planned)') to shutdown the server. We also made sure the server was properly shutdown by stopping it from the AWS console. We then restarted the domain server.

Similarly, we RDP into the member server **NepalWindowsMS01**and renamed it as **NepalWinMS01**. We then restarted it as well.

Next task in the lab was to install Active Directory Domain Services. We RDP into the DC01. We then launched the Server Manager by clicking 'Server Manager' button (button right of Metro button). If there are roles it takes some time to populate. At this moment we did not have any. To add a role we clicked on '2 Add roles and features' link. We clicked 'Next' and followed along the wizard. During the process we selected role as 'Active Directory Domain Services' and when it asked to add features, we did so. We then completed the wizard by pressing 'Finish' button at the end.

After the Active Directory Domain Service was installed, we noticed an exclamation mark hovering by a flag icon in the upper left of the display. We clicked on that and clicked on the link that said 'Promote this server to domain controller'. It was also mentioned in the class that in Windows Server 2012 we cannot run **dcpromo.exe** application from the command line as in the past.

We then added Domain Name Server role using a wizard. In the wizard we selected a forest and I named the root domain as **nepal.loc**. We did not change the default functional levels ('Windows Server 2012 R2') and choose the password as '**Passw0rd!**'. On the next screen we saw a message: "**A delegation for this DNS server cannot be created**". We got this error because this was the first server we were creating in the forest. Once such a server is created we won't get that error. We clicked through and finished the installation process.

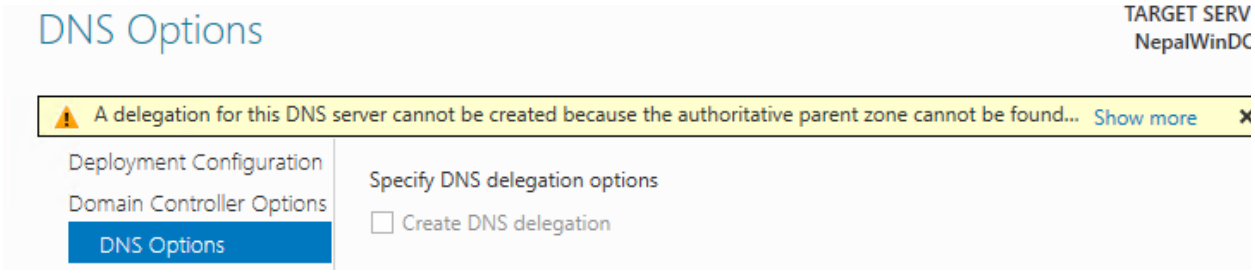

Figure 1: DNS Options

Once the AD DS was installed we added a user and made him a Domain Administrator. In the Server Manager menu we clicked on 'Tools' and selected 'Active Directory Users and Computers'. We found the

domain name (**nepal.loc**) by expanding the tree on the left pane and right-clicking on 'Users' folder to select New → User.

As a test name we used 'Bob Smith' for the first and last names and 'bsmith' as the email address name. We entered '**Passw0rd!**' as the password and unchecked 'User must change password at next logon' to make the process easier for this lab. We clicked the 'Finish' button to complete the process. If we went to the 'Users' folder we could see 'Bob Smith' name. We right clicked on Bob and selected properties. We could click 'Member Of' tab and click 'Add' button. In this window we entered 'Domain' in the search box and clicked the 'Check Names' button. We selected the 'Domain Admins' group and clicked 'OK'. We also selected the 'Domain Admins' group in the properties window and clicked the 'Set Primary Group' button. This process made Bob as a domain administrator. Finally, we exited out of the AD DS Users and Computers MMC (Microsoft Management Console).

After installing AD DS and setting up Bob Smith as our first admin next we needed to make the member server join this domain server. In order to make sure that all the machines on the private network could communicate with one another we added another rule to the inbound rules in our security group in the AWS console. We selected inbound tab and clicked Edit in the security group tab. We selected 'All traffic' for the type, 'All' for the Protocol, 'All' (or default range) for the Port Range and '173.1.0.0/16' for the Source. We RDP in to the member server from our DC01 using the private IP assigned to the MS01 instance (The IP address probably should be 173.1.1.30). We used '**Administrator**' as the username and '**Passw0rd!**' as the password. We then navigated to the Network Adapter settings from Control Panel and selected 'Network connections'. We choose 'Internet Protocol version 4 (TCP/IPV4)' option and selected 'Properties'. We left the IP address assignment to DHCP, but changed the Preferred and Alternate DNS server to 173.1.1.20 (IP address of our DC) and 8.8.8.8 respectively.

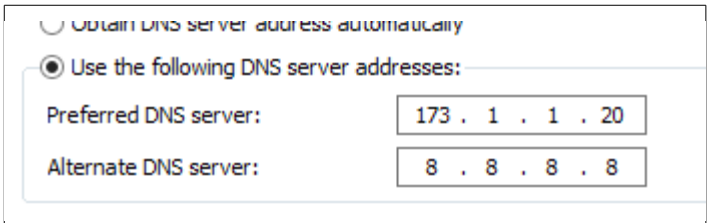

Figure2: DNS Server Setup

Next we launched the PowerShell in the MS01 instance (should run in Administrator Mode by default) and typed '**nslookup nepal.loc**' command. This should confirm that the member server can find the domain server.

```
PS C:\Users\Administrator> nslookup nepal.loc
Server:  173.1.1.20.reverse.gogrid.com
Address:  173.1.1.20

Name:    nepal.loc
Addresses:  2002:ad01:114::ad01:114
          173.1.1.20
```

Figure 3: NS Lookup

We then added the domain server to the member server and restarted the member server with '**add-computer -domainname nepal.loc –restart**' command. While typing this command we also learned about the tab completion feature of PowerShell. In this lab I had to go through few odd issues like AWS not reflecting the MS01 being restarted and MS01 disconnecting itself and throwing me into DS01 and not reconnecting. In any case we login into the MS01 with new credentials of Bob. We used '**bsmith/Passw0rd!**' to connect. Once logged in we confirmed that Bob is a Domain Admin by launching the Server Manager. If Bob did not have administrative privileges we would not have been able to launch the Server Manager. We then sign out of the MS01 instance by selection Shut down from the Metro interface. We could also confirm that MS01 has been added as a resource to active directory by going into 'Active Directory Users and Computers MMC' in the DS01 and clicking on Computers tab. Here we saw the MS01 instance listed. We could also see more details about the server by right clicking the name and selecting properties.

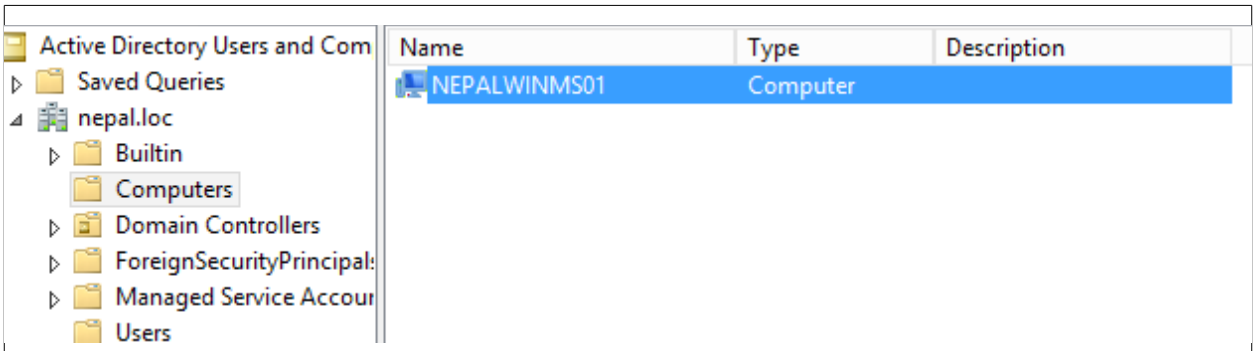

Figure 4: Member Server in list of computers

Install Webmin in Ubuntu

In this part of the lab we learned to install a web based server management tool called Webmin. During our lab we downloaded the Webmin package from webmin's website using following command.

$ wget wget http://www.webmin.com/download/deb/webmin-current.deb

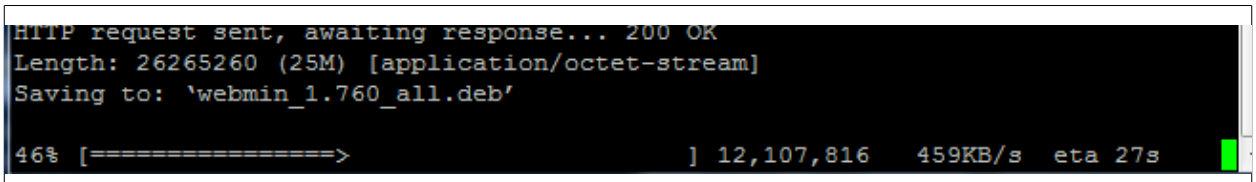

Figure 5: Download Webmin

We then installed the dependencies required to install above downloaded package with following command.

$ sudo apt-get install perl libnet-ssleay-**perl** openssl libauthen-pam-**perl** libpam-runtime libio-pty-**perl** apt-show-versions python

Since Webmin runs in port 10000 over ssh, we opened that port using our security group in AWS console. We navigated to the security group and added a new rule by selecting 'Custom TCP Rule' for the Type, 'TCP' for the Protocol, '10000' for the Port Range and '0.0.0.0' for the Source. To login to Webmin as a

normal user (and not as a root), we added a new user called '**webmin**' using **'sudo adduser webmin**' command. For the password we typed '**Passw0rd!**'. For other prompts we just pressed 'Enter'. We also gave **webmin** user **root** privileges by adding it to **sudo** group using '**sudo usermod -a -G sudo webmin**' command. After executing these commands we are able to visit **https://<Public_IP_Of_Ubuntu_Instance>:1000**. The browser showed a security warning, because the ssl certificate was not certified. We accepted it as an exception and proceeded to see the following interface.

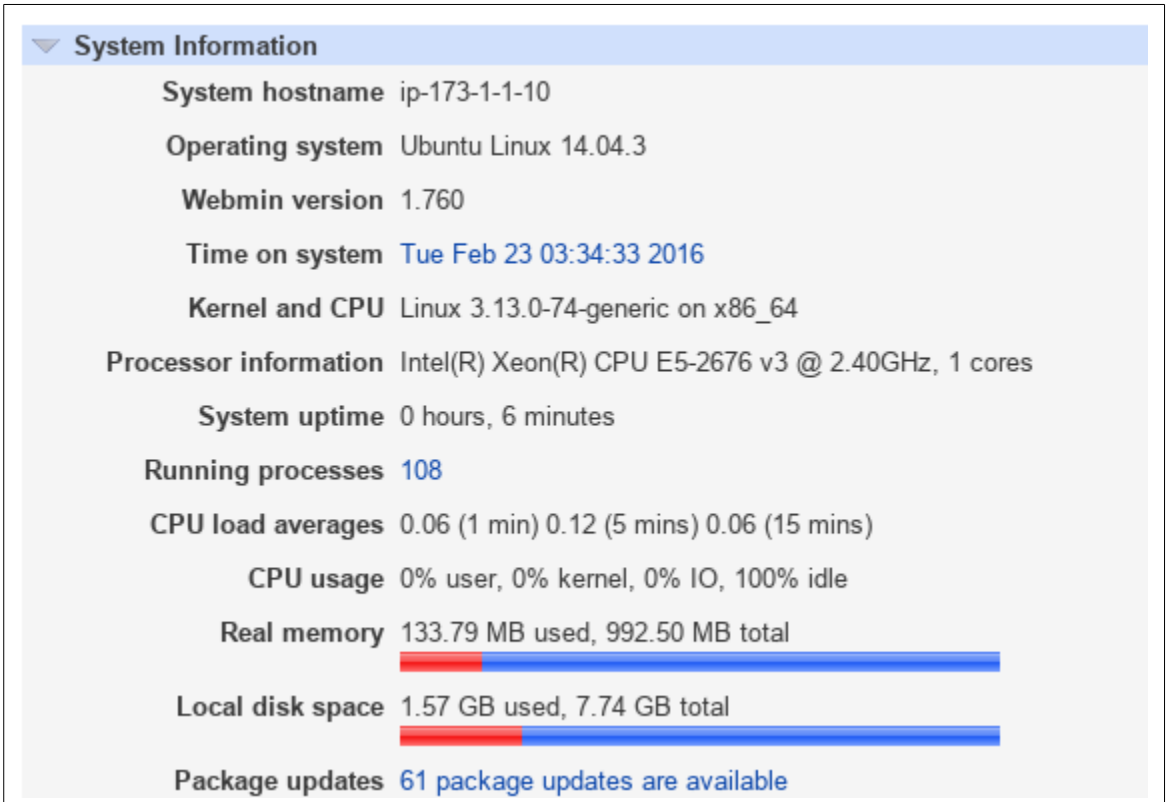

Figure 6: Webmin Console

OBSERVATIONS

In this Lab we learned about setting up Active Directory Domain Server. We used the DC01 instance as our Active Directory Server. We first changed the domain controller's name using **lastnamewindc01** format. In my case the name was **NepalWinDC01**. We also made sure the name was no longer than 15 characters, because it conflicts with how Linux looks at domain names. I also renamed the member server

using similar procedure and named it as **NepalWinMS01**. We restarted the servers for the name to take effect.

We then connected into the domain controller server and launched 'Server Manager'. From here we selected 'Add roles and features' to add 'Active Directory Domain Services. During the process we also had to click an exclamation mark hovering in upper left corner and click 'Promote this server to domain controller'. We used a wizard to create a new forest and named our root domain as lastname.loc (**nepal.loc** in my case). We got an error message saying 'A delegation for this DNS server cannot be created' This is because we were creating our first server and it had no other DNS to delegate to.

After DNS was installed next we added a new user called Bob Smith using 'Active Directory Users and Computer's option from Server Manager → Tools menu. To create a new user we right-clicked on the 'Users' folder in the AD DS Users and Computers MMC. For the email address we selected 'bsmith' and 'Passw0rd!' for the password. We also made Bob as an administrator by making Bob a domain admin. From Bob's properties window we selected 'Member Of' tab and searched for 'Domain Admins' and clicked the 'Set Primary Group' button.

Next, we setup our member server to join to the domain server. In order to ensure that all the machines on our private network communicated with each other we added 173.1.0.0/16 into our security group with all Protocol, Port and All traffic options. We then connected into the member server, and added the IP address of the DC (173.1.1.20) as Preferred DNS server and 8.8.8.8 as alternative DNS. My guess is, 8.8.8.8 is the DNS server of Google. We also searched the DNS server using '**dnslookup nepal.loc**' command in Windows's PowerShell. Once the domain controller was identified, we added it to our member server using '**add-computer -domainname nepal.doc –restart**' command. The restart option restarts the server once the domain name is set. When we re-connected to the member server this time we connected using Bob's credentials with username '**bsmith**' and password '**Passw0rd!**'. If we checked the Computers in the Active Directory Users and Computers MMC we were able to see MS01 as one of the instances. In this way we were able to setup Active Directory Domain Server and connect to it from a member server.

Next, we installed a web based server management tool called Webmin in Ubuntu. We used **wget** to download the package. Since the package was not installed from the repository manager we had to install the dependencies separately. We used **apt-get** to install the dependencies and **dpkg** to install Webmin deb package. During the lab I had few issues installing the dependencies. I used "**apt-get -f install**" command to resolve the dependency issues. After the package was successfully installed we made sure our server was browsable by adding a TCP custom rule with 10000 as the port and 0.0.0.0/0 as the source IP. In Ubuntu, since we did not want users to connect to Webmin interface using root credentials we created a user called **webmin**. We added this user to the **sudo** group. This made sure that the user was able to make changes to the system. We were then able to browse using http://Ubuntu_Public_IP_Address:10000 address in the browser. Since the ssl certificate used in this web site was not signed, we were prompted with a warning message which we ignored and proceeded. We used **webmin** user's credentials to login to the **Webmin** console. In this way, we successfully installed Webmin in Ubuntu.