**COMP 3962 – Lab 9**

**OpenVPN and Windows Active Directory Domain on AWS**

Introduction

In this lab, you will do two exercises. First, you will create an OpenVPN environment. You will create a secure tunnel between your AWS VPC and your laptop. This allows you to create and connect to EC2 instances using their *private* IP addresses. Both two instances in this lab will be created *without public* IP addresses. In the second part of the lab, you will build a very small Windows Active Directory Domain in your VPC.

Please do this lab in Oregon region.

Part 1 – OpenVPN, Client and Connection

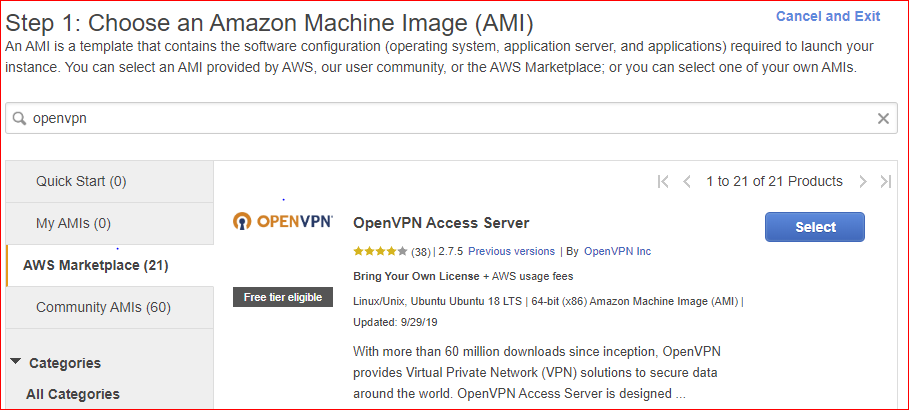
In this lab, you are going to launch an OpenVPN Server from the AWS Marketplace available on AWS.

1. Download and install a 30-day trial version of Viscosity (OpenVPN client) from SparkLabs:

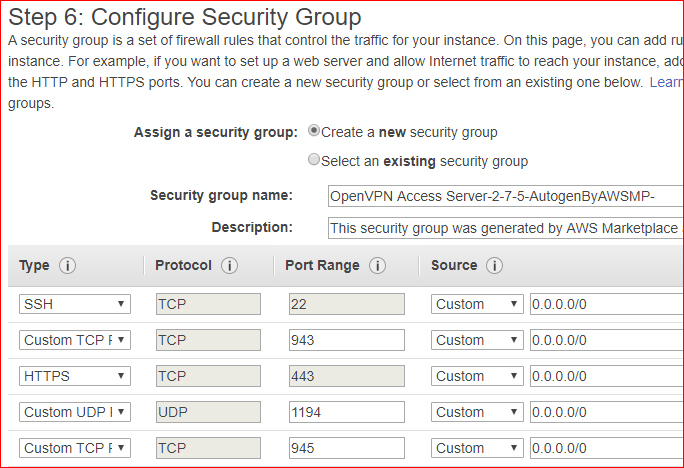
<https://www.sparklabs.com/viscosity/>

If you are using the lab workstation, you should be able to install this on a temporary basis. It will not be here next week. You can reinstall it next week again, if you need to.

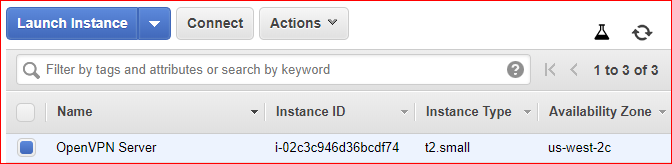
1. Log into your AWS console.
2. Go to EC2 and click Launch Instance.
3. On the left-hand side, under Quick Start, click “AWS Marketplace.”
4. Enter “openvpn” in the search window. (See screenshot on the next page.)
5. Select the top one in the list. See screen shot below. Click Select on the right.



1. In Step 2, select the t2.micro (not t2.small) free tier eligible AMI.
2. In Step 3, accept the default for Auto-assign Public IP.
3. In Step 4, the Volume Type should be “General Purpose SSD.”
4. In Step 6; Configure Security Group, accept the default to “Create a new security group.”



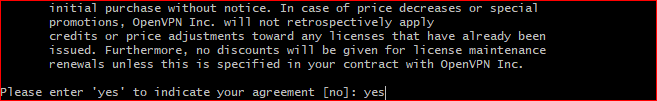
1. Review and launch with your existing key pair.
2. Go to your EC2 Console.
3. Modify the name of the instance to be OpenVPN Server.



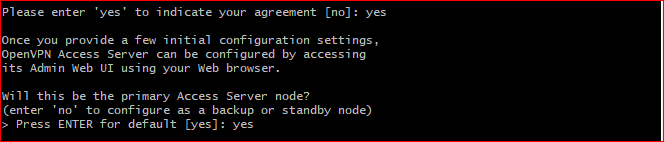
1. **Important:** If you are going to use AWS for your Windows Server Administration course (ACIT 3420), then attach an Elastic IP address to this instance.
2. After the instance has been created, SSH into the instance. The username is “openvpnas”

“ssh -i <myKeyPair>.pem **openvpnas**@<Public IP Address>

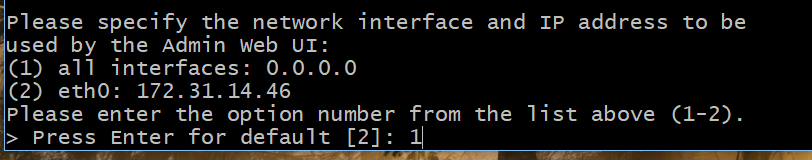
1. You are now going to configure the VPN server.
2. Enter ‘yes’ at the first prompt regarding the license agreement.



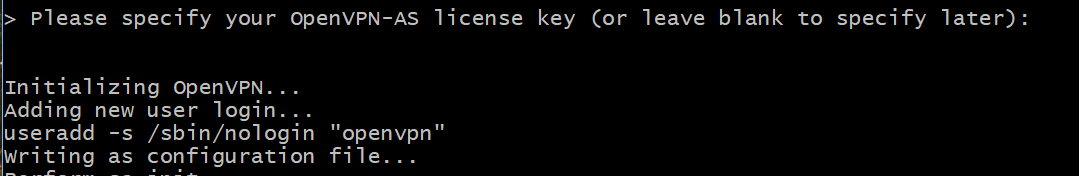
1. Accept the next default.



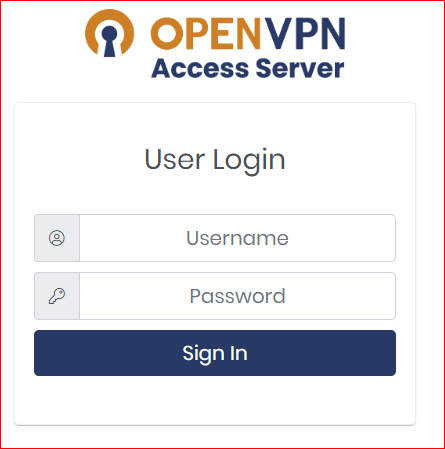
1. On the question with the two choices, select ‘1’ for “all interfaces” (See screenshot, below.)



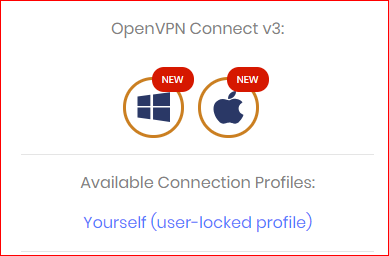
1. Accept the defaults for the remaining prompts.
2. Notice a new user called “openvpn” has been added:



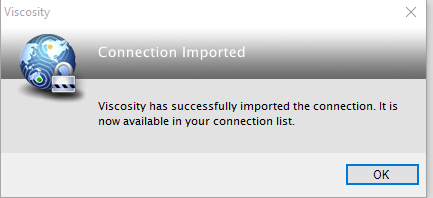
1. Set the password for the user ‘openvpn.” (Not ‘as’ at the end.)
   1. Type “sudo passwd openvpn”
   2. Enter a password and tattoo it on your arm. I used “admin.”
2. Exit/close/terminate/end the *connection* to the OpenVPN instance. (Close the Git Bash terminal window.)
3. Launch a web browser, like Google Chrome.
4. Enter the following URL: http**s**://<OpenVPNInstance Public IP address>:943
5. Ignore the error message about privacy and proceed to the website.
6. Login with the username (openvpn) and the password you created earlier.



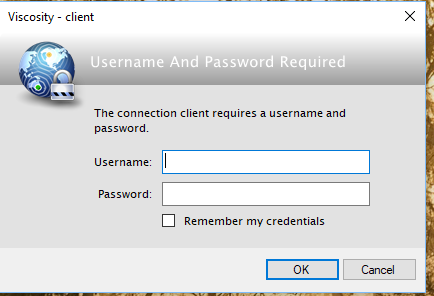
1. On the next screen, click on the last connection profile “Yourself (user-locked profile).” See screen shot below:



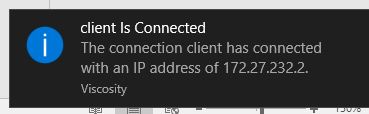
1. This will download a file called “client.ovpn”
2. Once it is downloaded, double click on the “client.ovpn” file.
3. This should launch Viscosity. (See screen shot.) Click OK.



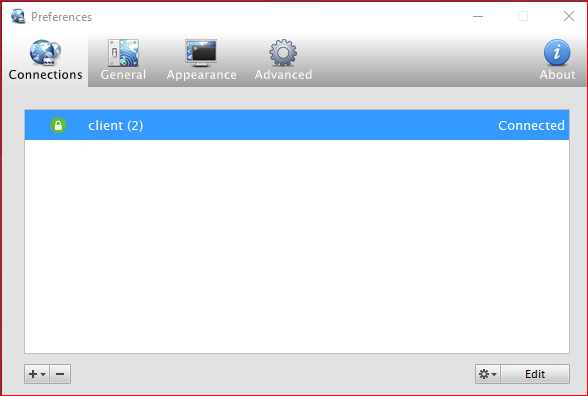
1. Access the system tray in the lower right hand corner of the Windows 10 menu bar.
2. Right click on the Viscosity icon and click on “client.” You should see this screen:



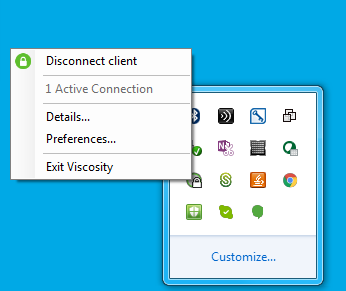
1. Enter your username (openvpn) and password. You should now have a VPN connection into your AWS virtual private cloud (VPC). You should see a connection like this:



1. Open the Viscosity application. You should see a connection:



1. Hover over the Viscosity icon. You should see that you have one active connection. You also have the option to disconnect the connection. Leave the connection open for the next step. Your screen shot may look different to mine, but you should be able to tell if you have an active connection.

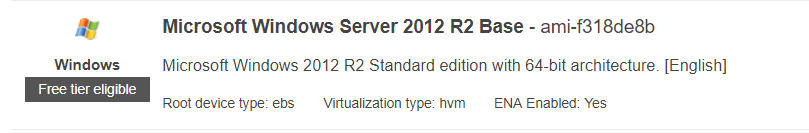


1. Do *not* close your VPN connection. Leave it open for the next part of the lab.

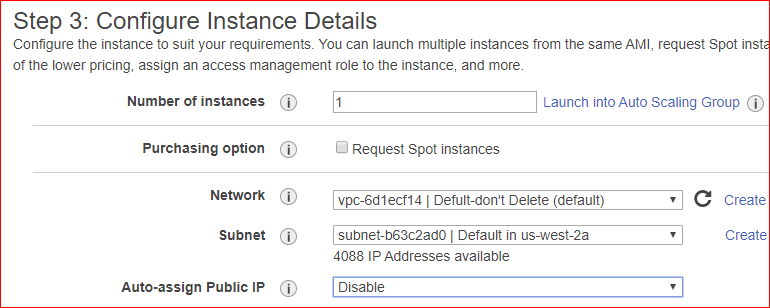
Part 2

In this part of the lab, you will create a Windows domain consisting of a Windows Server, a Windows 10 client. The windows 10 client machines will be connected to the Windows domain. The entire domain will be created on AWS.

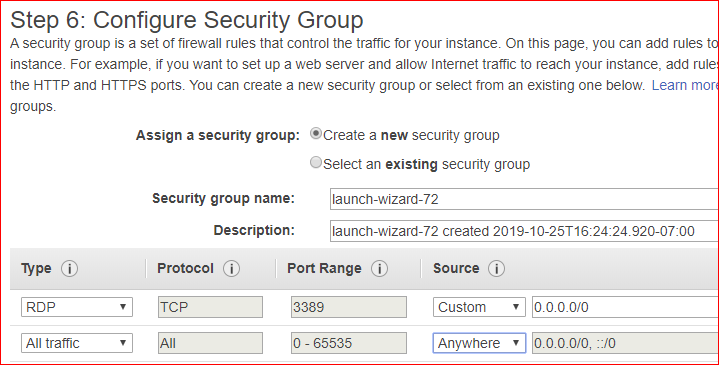
1. From within the EC2 AWS console, launch a Windows Server 2012 R2 Base (free tier eligible) instance.



1. In “Step 3: Configure Instance Details,” do two things:
   1. For subnet select a subnet as the same availability zone of your OpenVPN server. (To check availability zone of your open VPN, look at the description tab.)
   2. **Disable** “auto-assign public IP.”



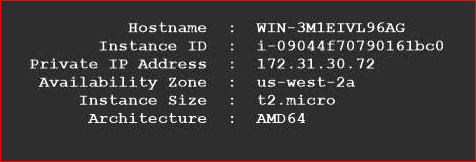
1. Create a new security group or use an existing security group that allows “All Inbound” traffic. This is never a good idea, but this will save you a lot of time configuring security groups and firewalls for this lab.



1. Select your key pair and launch the instance.
2. Name your instance as Windows Server 2012.



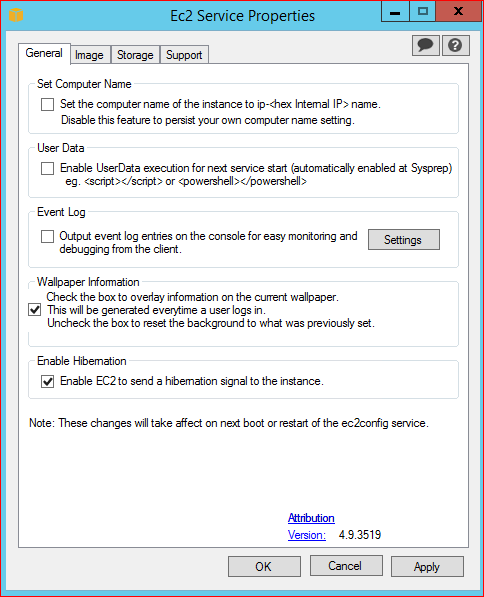
1. Connect (RDP) into the Windows instance. (Remember, this is a Windows machine. You use RDP, not SSH, to connect.)
2. Using your key pair \*.pem file, generate a password. Save the password, although you will change it later in the lab.
3. Once you are logged in, you may be asked to make your instance discoverable on the right. Click “Yes.”
4. In the top right corner of your desktop is networking information about your instance. The default hostname, assigned by AWS, is not verify meaningful. You want to change this.



1. Run “EC2ConfigureService Settings.” Click on the Windows icon in the lower left corner. See screen shot containing “EC2ConfigureService Settings” on the left.



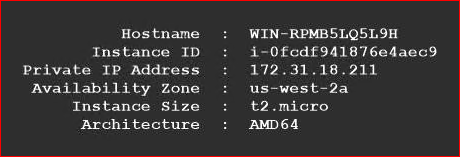
1. Verify that “Set Computer Name” is not enabled. No check mark. (See screen shot below. It should be disabled by default. ) This will allow us to change the computer name to something meaningful.



1. You can change the hostname through Server Manager (described below) or from the Windows command line. Open a Windows command prompt. To change the hostname from the command, type the following:

**NETDOM renamecomputer [Current Hostname] /NewName:[NewName]**

Remember, your current hostname is in the top right corner:

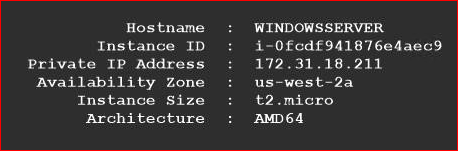


This is what I entered for my instance:

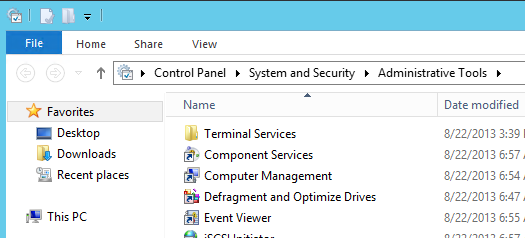
**NETDOM renamecomputer WIN-3M1EIVL96AG /NewName:WindowsServer**

The NewName can be whatever you like. It does not have to be “WindowsServer.”

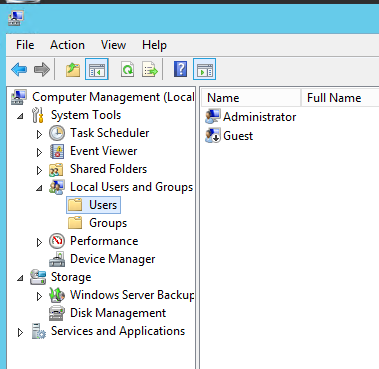
1. Do you have your password? You will need it now…
2. You will have to reboot/restart the instance for this name change to take effect. (Note: You can also wait until you install Active Directory Domain Services to change the computer name.)
3. Reconnect/login to your Windows Server instance. You do not have to download a new RDP file! Just double click on the RDP file that was downloaded before. You should see your new Hostname:



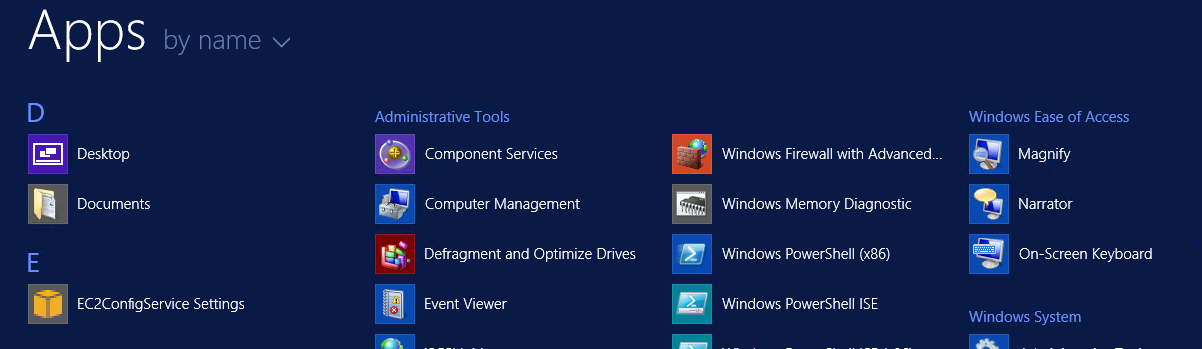
1. Set a new Administrative password. To do this, go to the Administrative Tools menu and select Computer Management.



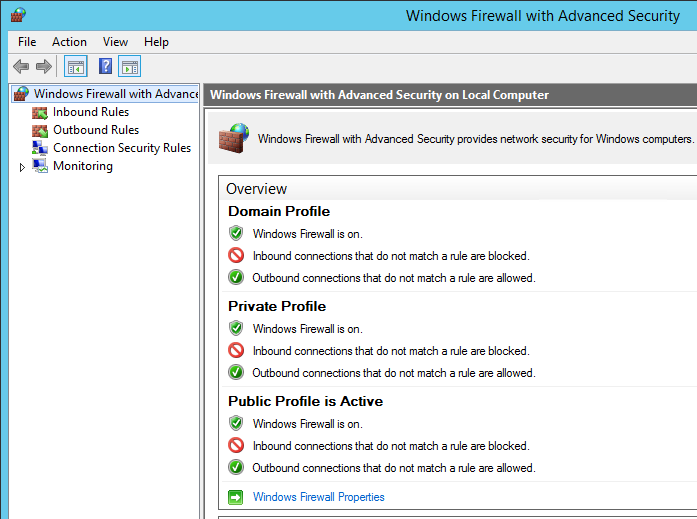
1. On the left side, expand Local Users and Groups. Click on Users. Right click on the Administrator and select Set Password. Write down this password! (Admin1234) You will now use this password when you RDP into your instance.



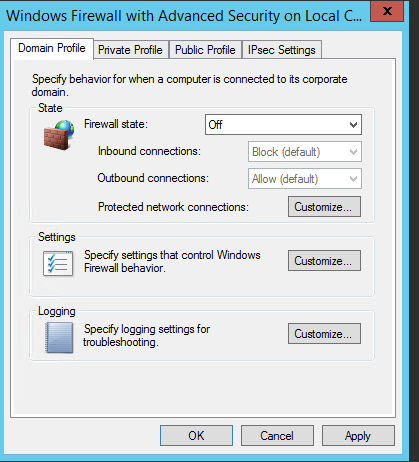
1. From the Administrative Tools menu, select “Windows Firewall Advanced Security.”



1. In the middle panel, click on the blue hyperlink “Window Firewall Properties.” See bottom of screen shot.



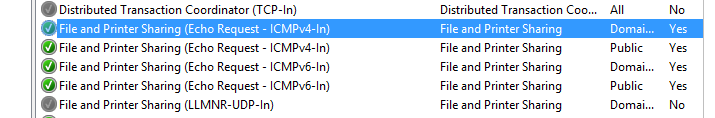
1. For *both* the Domain Profile and Private Profile, set the Firewall state to “Off.” (Don’t forget to click the Apply button.) This will make it easier for us to ping other computers on the subnet.



1. On the left, click on the Inbound Rules and enable the following two rules (in the middle column):
   1. File and Printer Sharing (Echo Request – ICMPv4-In)
   2. File and Printer Sharing (Echo Request – ICMPv6-In)

(Right click on the rule and select Enable Rule. The green check means enabled.)

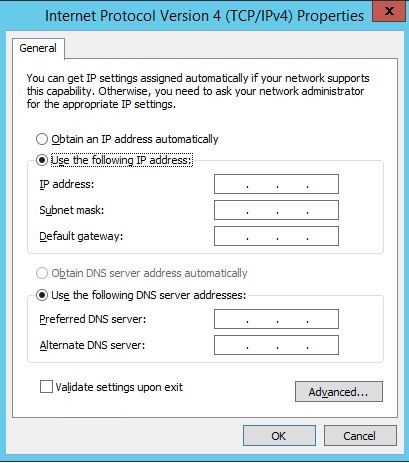




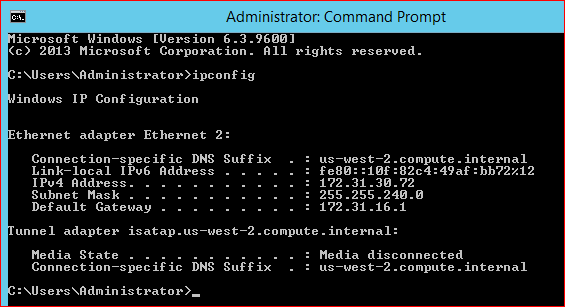
1. Follow the steps in the YouTube video “Active Directory (ADDS) Installation & Configuration – Windows Server 2012.” (<https://www.youtube.com/watch?v=4fVi0_iFEJ4>) You will change a few of the steps along the way, but for the most part the video does a good job. (I have not put any screen shots in this section of the lab. Hopefully, you can follow the video.)
   1. Launch Server Manager. The blue computer with tool box in the lower left corner.



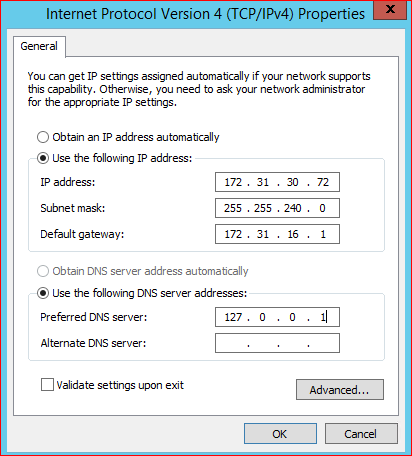
* 1. On the left, click Local Server. In the video, he changes the server name. If you did that above, then don’t do it again.
  2. As shown in the video, click “Ethernet” to change the properties of the network adaptor. Disable IPv6 by unchecking “Internet Protocol Version 6.” When you click on the Properties button for “Internet Protocol Version 4 (TCP/IPv4,” you will see the screen below. Where do I get the IP address?



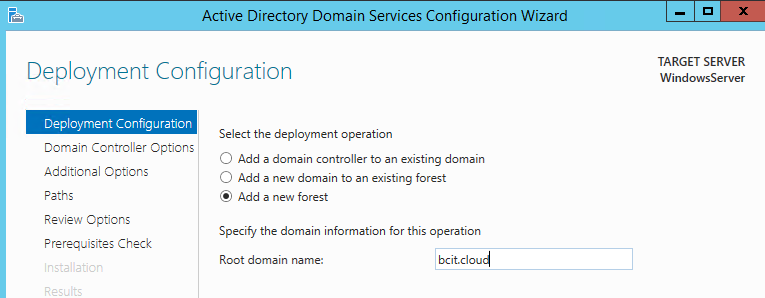
* 1. Open a Windows command prompt. Enter the command “ipconfig.”



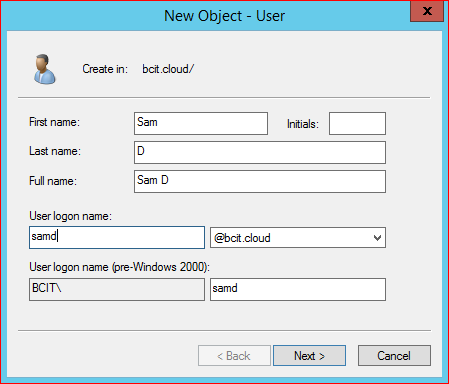
* 1. Set the properties for IPv4 similar to this screen shot below, using the information returned by the “ipconfig” command. The “IP address” is the private IP address of your instance. The “subnet mask” is returned by the ipconfig command. The “default gateway” IP address is also returned by ipconfig. Since this instance is the DNS server, simply enter IP address of the loopback adaptor or private IP address of your instance for the Preferred DNS Server.



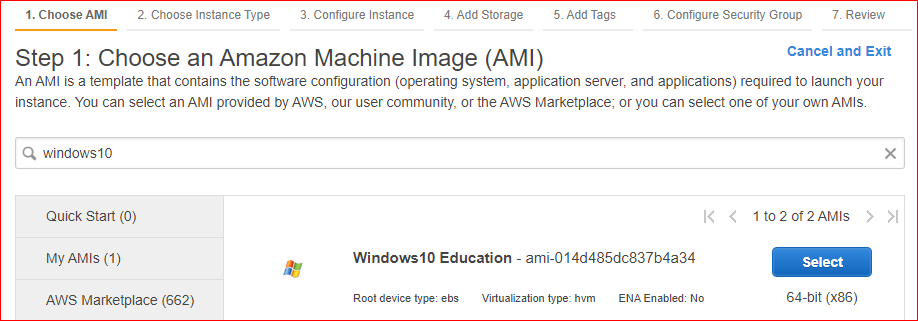
1. Install ADDS according to the video. Enter the domain name shown: bcit.cloud. Do *not* put “www” in front of the domain name. See screen shot below.



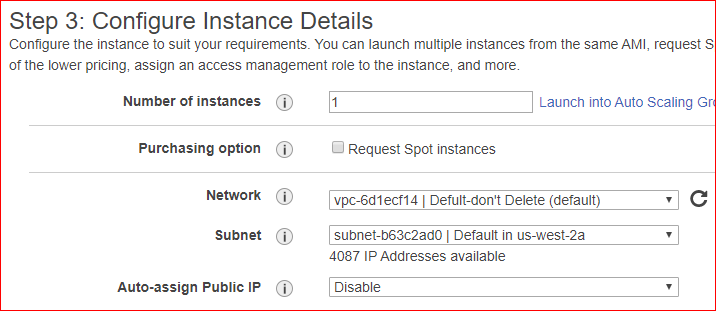
1. After the ADDS has been installed, you will have to *wait several minutes* for your instance to reboot. Your instance will reboot automatically.
2. After the instance reboots, reconnect into your instance. You don’t have to create another RDP file. Simply double-click on the RDP file that was downloaded previously and enter the password you created and save (!!!) above.
3. Go back into Server Manager.
4. Verify that your Windows Server is a domain controller as shown in the YouTube video. You are now done with this video.
5. Add a User to the domain. To do this, watch the first **3:30** minutes of this video: <https://www.youtube.com/watch?v=yZ6Te5TDvhg>.
6. When creating the user, email address should be added by default to user logon name. If the domain name you entered above was “bcit.cloud,” then the email address should be [name@bcit.cloud.](mailto:name@bcit.cloud.) See screen shot below.



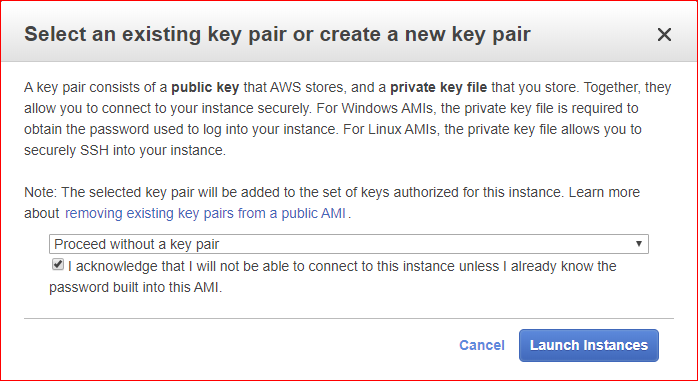
1. Go the EC2 Dashboard. Click on Launch Instance and select Community AMIs.
2. In step1: choose an AMI, type Windows10.



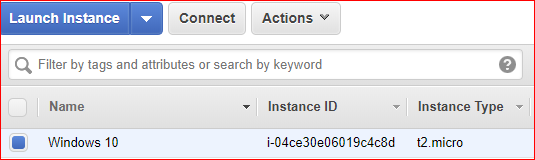
1. This is displaying a Windows 10 Education instance. Select and launch this image.
2. In “Step 3: Configure Instance Details,” do two things:
   1. Select the same subnet as the OpenVPN and Windows Server and
   2. **Disable** “auto-assign public IP.



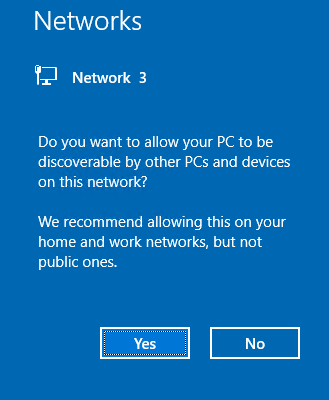
1. Select the same security group you used for the Windows Server instance (All traffic).
2. IMPORTANT: On the screen where you select an existing key pair, select “Proceed without a key pair.” This instance has a password.



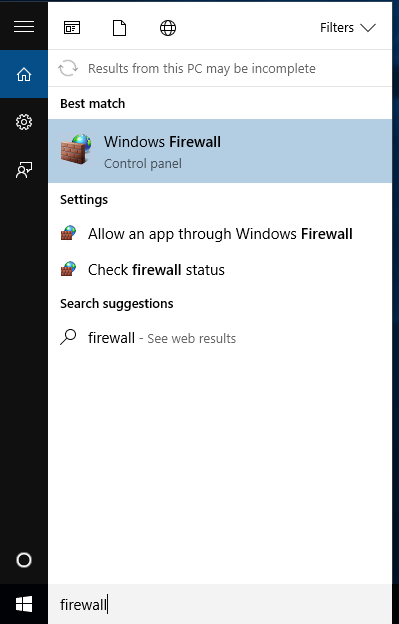
1. Name your instance as Windows 10.



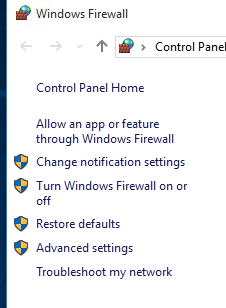
1. After the Windows 10 instance is running, connect and login with the password “admin” (all lowercase). Yes, this will download another RDP file to your computer. You now have two RDP files, one for connecting to the Windows Server and anther for connecting to Windows 10.
2. When you see this screen, click “Yes.”



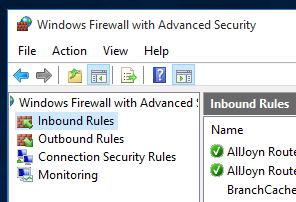
1. If your window prompting you to update OneDrive, click “Close.”
2. In the Windows 10, you are going to set the firewall with the same settings you did for your Window Server instance. To do this, enter “firewall” in the search window and select “Windows Firewall.”



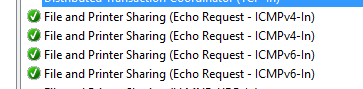
1. On the next screen, click “Advanced Settings.”



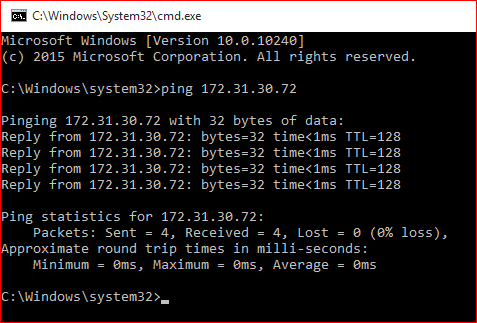
1. You will see this screen:



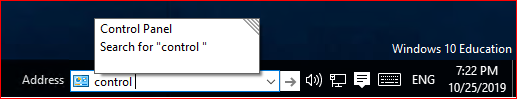
1. Do the same steps you did in Step 21, above.



1. Open a command prompt and ping the *private* IP address of the Windows Server 2012 R2 instance. What does this mean? (You can tell me in class.)



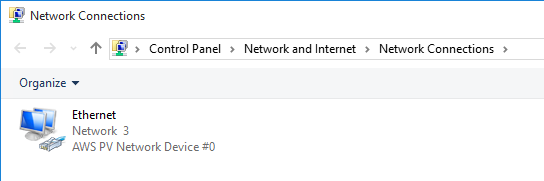
1. Set the properties on the network card. To do this, enter “Control Panel” in the search window.



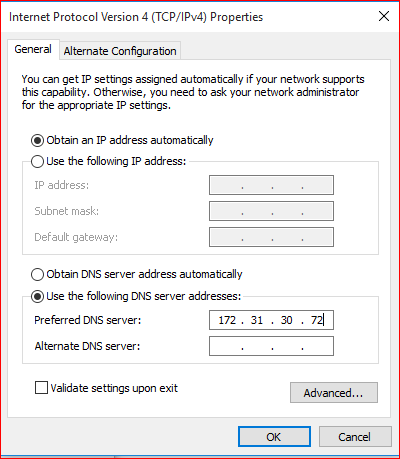
1. You should see this screen:



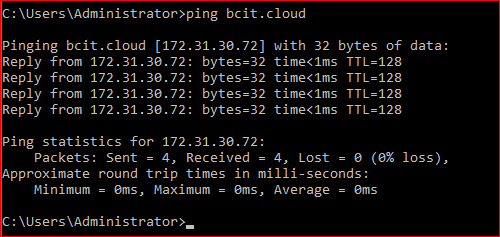
1. Click “Network and Internet.” Then click “Network and Sharing Center.” On the left side, click “Change adapter settings.” You should now be here:



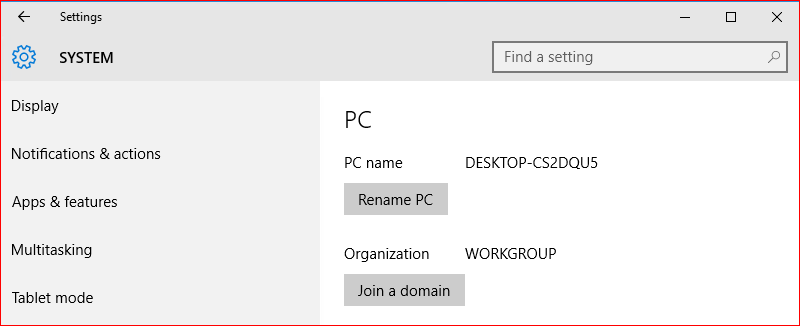
1. Right click on the Network card and select Properties. Disable (un-check) Internet Protocol version 6 (IPV6). Select the Properties for Internet Protocol IPv4.
2. Select “Obtain an IP address automatically. IMPORTANT: Set the Preferred DNS Server to the *private* IP address of the Windows *Server* running ADDS!



1. Open a command prompt again. You should now be able to ping the domain name of the Windows Server. Example: “ping bcit.cloud” How is this possible? (Tell me in class.)



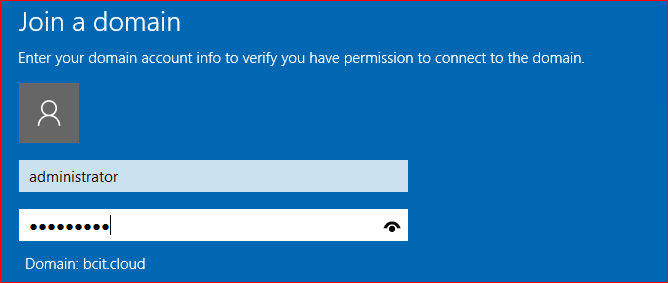
1. Next, you are going to join the Windows 10 machine to the Windows Server domain. To do this, in your windows 10 go to Settings > System > About then click Join a domain.



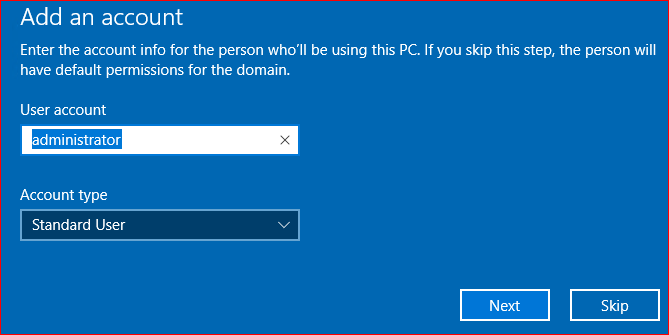
1. Enter the Domain name and click Next.



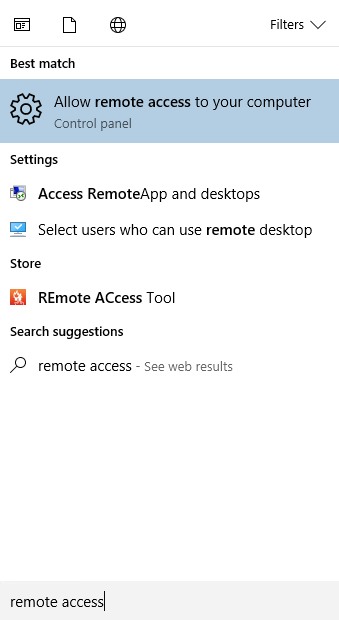
1. Enter account information which is used to authenticate on the windows server 2012. User name: administrator, password: Admin1234, click OK.



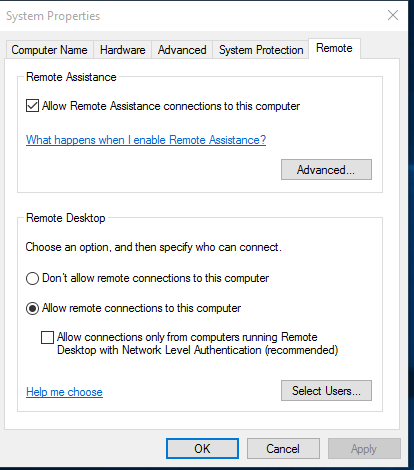
1. Accept the default and click Next. IMPORTANT:DO not restart now.



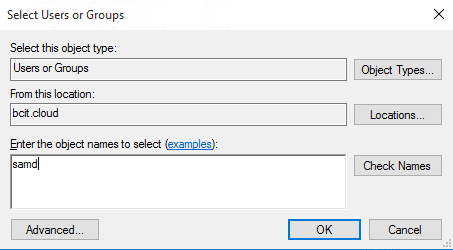
1. Click on Restart later.
2. Before you restart the instance, change the PC Name to “Windows10” or something similar. (If you didn’t do this, no big deal.)
3. Set up Windows 10 to allow remote user access. To do this, enter “remote access” in the search window and select “Allow remote access to your computer.”



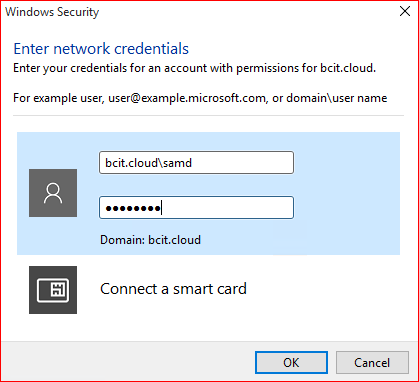
1. You should see this screen:



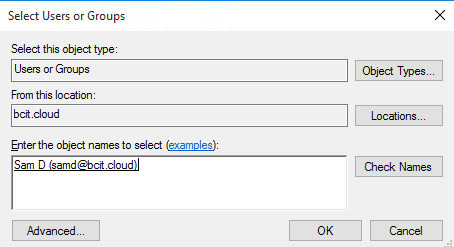
1. At the bottom of the screen, enable “Allow remote connections to this computer.” Click “Select Users” button in the lower right. Click the “Add…” In the field “Enter the object names to select,” enter the username of the user you created on the Window Server. In the video, the user was “SamD.”



1. Click Check Names. You will asked to enter the user name and password of the user.



You should see this screen;

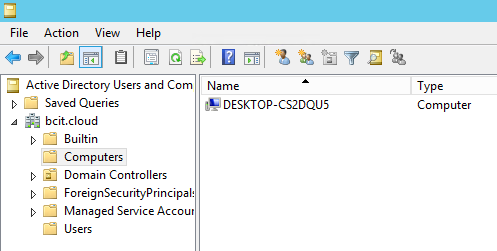


1. Click OK. Note: For both Windows 10 and Window Server, you can also access the remote desktop settings by going Control Panel -> System and Security -> System -> Remote Settings.

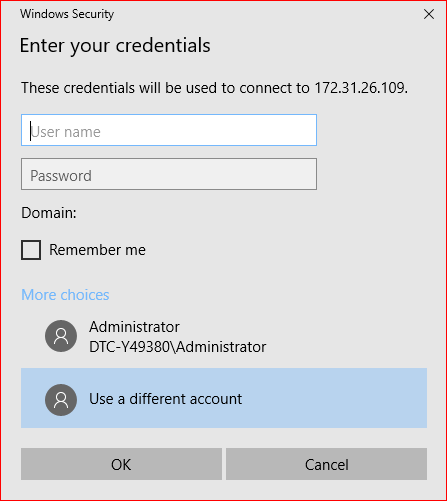
(Reference for Windows 10: http://www.groovypost.com/howto/setup-use-remote-desktop-windows-10/)

(Reference for Windows Server, [https://technet.microsoft.com/en us/library/cc794832(v=ws.10).aspx](https://technet.microsoft.com/en%20us/library/cc794832(v=ws.10).aspx))

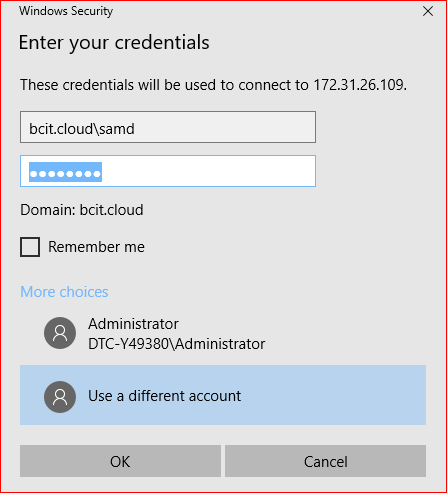
1. Disconnect (close the connection) from your Windows 10 instance. Reboot the instance from the EC2 Dashboard. Action-> reboot. (DON’T terminate!)
2. Go back to your Windows Server instance to verify that the Windows 10 instance is part of the domain. To do this, go to the Tools menu and select “Active Directory Users and Computer.” On the left side, expand your domain name. Click on Computers. On the right side, you should see your Windows 10 computer listed. My Windows 10 is called “DESKTOP-CS2DQU5.”



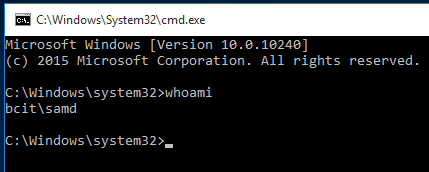
1. Reconnect to your Windows 10 instance. When you see the login screen, click More Choices and select “Use a different account.”



1. Login using the domain name (bcit.cloud) and the user (samd). Your domain name and user will be different than mine. (To do this, the Windows Server instance must be running.) See screen shot below. It is important to use a backslash (\) to separate the domain name from the username. See screenshot below.



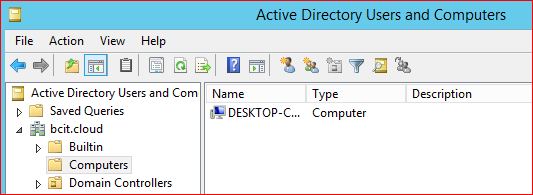
1. To confirm you are logged in as “samd,” open a command prompt and type “whoami.”



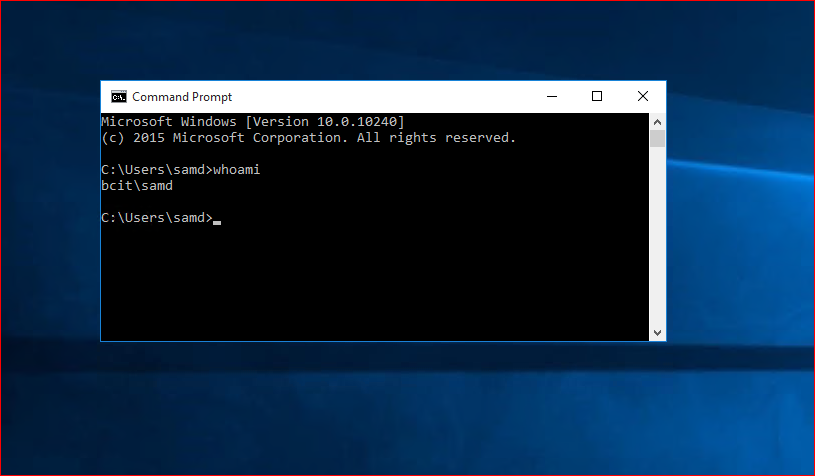
Now you have a Windows Server Domain Controller and one Windows 10 machine that is part of the domain.

Lab Deliverables

1. You should have three running instances: Windows Server 2012 R2, Windows 10 and Openvpn.
2. Show your instructor that the Windows 10 is part of the domain. Show your instructor this screen on the Windows Server:



1. Show your instructor that you have logged into the Windows 10 instance using the domain user that was created on the Windows Server



Clean up:

Terminate all your three instances.

OpenVPN Troubleshooting

1. On the OpenVPN instance, you might have to disable Source/Destination check.
   1. In EC2, select the OpenVPN instance.
   2. Select Actions -> Networking -> Change Source/Dest Check.
   3. Disable.
2. Your anti-virus software might prevent Viscosity from connecting to the OpenVPN instance.
   1. Make sure the anti-virus software allows the two Viscosity applications through the firewall to run:
      1. Viscosity.exe
      2. ViscosityService.exe
   2. Make sure the firewall allows any traffic to VPNs.