

Azure AD Connect

Friday, August 4, 2017 7:45 AM

Work with a mock, on-premises Windows 2016 infrastructure connecting it to an Office 365 tenant via AD Connect.

This workshop centers around helping the user better understand the basics of Azure Active Directory, including Office 365. By participating in this workshop, users will learn how to connect and synchronize an on-premises Active Directory with Azure AD. Participants will also gain insight into configuring filtered synchronization and enabling health monitoring for their on-premises AD.

What You Will Learn

- Connecting Office 365 with On-Premises AD
- Azure AD Connect
- Filtering
- Password Synchronization
- Password Writeback
- Azure AD Health

Ideal Audience

- CISOs and VPs of Information Security
- CIOs
- IT Managers
- Active Directory and Network Admins

Overview

This workshop centers around helping the user better understand the basics of Azure Active Directory, including Office 365. By participating in this workshop, users will learn how to connect and synchronize an on-premises Active Directory with Azure AD. Participants will also gain insight into configuring filtered synchronization and enabling health monitoring for their on-premises AD.

Time Estimate: 6.0 hours

Requirements

Setup Requirements

The following workshop assumes that you have used the Azure Workshops CLI to pre-create the necessary lab environment. To use the Azure Workshops CLI, you will need the following applications installed on your local machine:

- [Node.js](#)
- [Git](#)

As stated above, these tools are necessary for downloading and running the CLI locally. Download and install these tools according to the instructions on their respective website.

Additional Requirements

Additionally, you will need a subscription (trial or paid) to both Office 365 and Microsoft Azure. Please see the [next](#) page for how to create trial subscriptions in both.

Office 365 and Azure Registration

Demo Domain

For the purposes of this workshop, you will need a demo domain name - a domain name that you will *not* be required to register with a domain name registrar (DNR), but will be used as your fictitious company. We, of course, do not want to use any domain names associated with production accounts.

The simple way to do this is allow a service to create one for us. So, to create a random domain name, we'll actually use a random username generator.

Open a browser to <http://jimpix.co.uk/words/random-username-generator.asp> and click the green "Go!" button close to the top of the page. Upon doing this, you will be presented with 25 different two-word combinations. Pick one that you like or click the green "Refresh" button until you do.

Once you find a domain name, write it down; you will use it for the remainder of the workshop.

Office 365

Now that we have a domain name, let's create a 1-month trial Office 365 account. This will automatically create a domain in Azure AD which we'll connect to virtual datacenter later in the workshop.

Direct your browser to <https://products.office.com/en-us/business/office-365-affiliate-program-try-business-premium>. In order to take advantage of some of the Azure Active Directory premium features, we will need the Business Premium edition of Office 365.

1. Begin by clicking on the green button "Start your free business trial".
2. Complete the form on the first page:
 - o Choose your country (this cannot be changed later due to data sovereignty and other factors)
 - o Enter your name
 - o Enter an email address (this should be a *legitimate email address* as this will be the administrator's security/reset email)
 - o Enter a phone number (enter a *legitimate cell phone number* in order to test multi-factor authentication)
 - o Enter your company name from above
 - o Choose a company size

3. For the form on the second page:
 - Enter a username for yourself in a format you prefer (e.g. if your name was John Doe, you could enter: john.doe, jdoe, john_doe, etc.)
 - For your company, enter the company name from above (NOTE: you will see here that the initial domain name will be *yourcompany.onmicrosoft.com*. This is the Azure Active Directory domain to which we will connect later in the workshop.) If your domain name has already been used, try another one from the previous list.
 - Enter and confirm your password
4. Prove you are not a robot by entering a telephone number at which you can receive a text or phone call.
5. Enter the code that was text'ed to you or that you received from the auto-attendant.

It should take less than a minute to create your account. After the process is complete, you should see a message stating that you are ready to go. While your account was *created* in less than a minute, it may take up to another 15 minutes or so to finish creating all of the additional services in Office 365. That's fine, as it will be a while before we actually need them.

Finally, remember this trial account is only good for 30 days. While Microsoft will not initially *delete* your account, they will disable functionality.

Azure

Finally, we need to create a trial Azure subscription. Believe it or not, we are already using Azure Active Directory because we just set up Office 365. Office 365 uses Azure AD underneath to manage all of our exchange users. We simply need to create a subscription so that we can leverage Azure's other offerings.

Direct your browser to <https://azure.microsoft.com/en-us/free/> and begin by clicking on the green button that reads **Start free**.

IMPORTANT: On the sign-up form page, you should see your new email address that associated with your new Office 365 account. If not, click on **Sign Out** and re-authenticate using your newly formed credentials (e.g. *username@yourcompany.onmicrosoft.com*).

1. In the first section, complete the form in its entirety. Make sure you use your *real* email address for the important notifications.
2. In the second section, enter a *real* mobile phone number to receive a text verification number. Click send message and re-type the received code.
3. Enter a valid credit card number. **NOTE:** You will *not* be charged. This is for verification of identity only in order to comply with federal regulations. Your account statement may see a temporary hold of \$1.00 from Microsoft, but, again, this is for verification only and will "fall off" your account within 2-3 banking days.
4. Agree to Microsoft's Terms and Conditions and click **Sign Up**.

This may take a minute or two, but you should see a welcome screen informing you that your subscription is ready. Like the Office 365 trial above, the Azure subscription is good for up to \$200 of resources for 30 days. After 30 days, your subscription (and resources) will be suspended unless you convert your trial subscription to a paid one. And, should you choose to do so, you can elect to use a different credit card than the one you just entered.

Congratulations! You've now created an Office 365 tenant; an Azure tenant and subscription; and, have linked the two together.

Setup

Installing the CLI

Once you have the prerequisites installed, you will then need to install the CLI. The CLI can be installed from the command-line or terminal prompt using Node.js.

First, open a command-line window or terminal prompt. Then, type the following command:

```
npm install azworkshops-cli -g
```

Running this command will take a few seconds to complete. But, doing so will download the Azure Workshops CLI, along with its dependencies, into a directory that is located in a globally accessible path.

Azure Subscription

As stated in the requirements section, the workshop requires an active Azure subscription.

Recommendation

It is recommended that you do not use an Azure subscription that is currently being used for production. The CLI will create its own resource groups, but it is not the best practice to utilize production environments for testing and workshops, such as this.

For best results, it is recommended that you setup register for the trial subscription as outlined on the [previous](#) page.

Creating the Lab Environment

Build Time

The automated building of the lab environment can take approximately 30 minutes to complete. It is best to begin this process while you are reviewing the workshop material.

Verify Installation of the CLI

From a prompt, enter the following command:

```
azworkshops --version
```

A successful execution of the command should print the current version of the Azure Workshops CLI which can be found in the right column, slightly down the page, of the Node Package Manager [website](#). If you do not see a version number, return to the requirements [setup](#) and try reinstalling them.

If you successfully see the correct version number, you are ready to begin the lab setup.

Build the Environment

From a prompt, enter the following:

```
azworkshops
```

1. You will be presented with a menu from which to choose a base configuration. Choose the base configuration for **Basic Active Directory**.
2. You will then need to authenticate with Azure. Visit <http://aka.ms/devicelogin> and enter the code provided to you.
3. Choose the subscription that you would like to use for this workshop.
4. Select the location for the created resources. It is best to choose a location that is closest to you in order to reduce latency.
5. You will then be prompted with additional configuration questions.
 1. For the AD domain name, enter your company name from the previous page with '.local' as the TLD (e.g. mycompany.local).
 2. For the NETBIOS name, it should automatically be an ALLCAPS version of the company name that you just entered (without the '.local' TLD extension). If so, just press Enter to accept the default. If not, enter a valid NETBIOS name.

6. After completing the configuration questions, the building of the lab environment will begin. Once completed, you will be presented with all of the lab's configured settings (e.g. resource group, domain, domain admin, password, etc.) It is best to copy this down for future use.

Exploring Azure



Objective

You have just created a lab environment in Azure. The lab environment is intended to mimic a basic, on-premises datacenter. This datacenter, being extremely basic, consists of a single Active Directory domain controller and a utility machine.

The first objective is for you to become familiar with connecting to and navigating the Azure portal. We will also explore the components in our virtual "datacenter" that the CLI created for us. Finally, we will connect to our remote datacenter.




Azure Portal Basics

Let's start by connecting to the Azure portal and becoming familiar with navigation.

1. Open a browser and navigate to <http://www.azure.com>.
2. In the top-right corner of your screen, you will see the menu option **POR TAL**. Click on it.
3. If you have not already, you will be required to authenticate.
4. After authentication is successful, you will be directed to your *Dashboard*. The dashboard is configurable by adding, removing and resizing *tiles*. Additionally, you can have multiple dashboards depending on your preferences. You could have different dashboards for resources dedicated to different functions, lines of business, or for operations.
5. On the left will be your primary navigational menu. You should see a list of favorited services on the menu with descriptions. (NOTE: The size of your menu may differ from that of others depending on the number of services you have selected as a favorite.) If all you see are icons (no descriptions) on your menu, your menu is currently collapsed. Click the "hamburger"  to expand it.
6. Pretty close to the top of your menu, you should see **Resource Groups** . Click this option.
7. Upon clicking the Resource Groups menu item, a *blade* will open revealing your created resource groups. In this list, you should find the resource group that the CLI created for you. It begins with **azworkshops_basicAD_**, followed by a datetime stamp. (NOTE: If you do not

see this listed in your available resources groups, ensure that in the second dropdown box above, you have the correct subscription selected. This should be the same subscription you chose earlier in the CLI.)

8. Clicking on this resource group with expand another blade listing all of the resources created by the CLI. What you should see listed are two storage accounts, two virtual machines, two network interface cards, one public IP address and one virtual network.

 azwdata04190145459710	Storage account
 azwdiags04190145459710	Storage account
 dc1	Virtual machine
 dc1	Network interface
 utility	Virtual machine
 utility	Public IP address
 utility1	Network interface
 vnet	Virtual network

(NOTE: The datetime stamps for your storage accounts will be different.)

Resource Descriptions

As stated in the previous step and indicated by the preceding screen clipping, the CLI created 8 different resources in this group for the workshop. Let's explore these a little bit more detail.


The first two items listed are storage accounts - one for the virtual machine disk drives and another to store diagnostic logs from the VMs. Storage accounts must be globally unique across Azure. Therefore, we've appended datetime stamps to the end of our storage account names in order to prevent collision.


Next, you will see two virtual machines - **dc1** and **utility** - listed. **dc1** is our Active Directory's domain controller. Each machine requires a network interface card for connectivity. Additionally, the **utility** VM has a public IP assigned to it. Exposing our domain controller via a public IP is a very bad practice. Therefore, we will remotely connect to our virtual network via our **utility** VM. All machines in Azure, by default, have connectivity *out* to the Internet. But, only VMs that have public IPs can be accessed from the Internet (e.g. outside of the network).

Finally, our VMs are connected to each other by utilizing a virtual network. With the exception of storage and a few other resources in Azure, a virtual network is required.

Viewing Resource Details



Let's take a moment and view some of the information about the VMs that were created for us. Let's use the **utility** VM as our example.

1. Find the **utility** VM  and click on it. This will expand another blade with our details for the virtual machine.
2. In the **Overview** pane, you'll immediately see three sections:

1. Actions - allows you to perform various actions on the virtual machine (e.g. connect, start, stop, etc.)
 2. Information - displays various information about your virtual machine (e.g. resource group, location, status, IP address, etc.)
 3. Metrics - reports various performance metrics regarding your virtual machine (e.g. CPU, network, etc.)
3. Now, let's look at one more page for some additional details. In the left pane (still on the **utility** blade), approximately half-way down, click on **Properties** . On this blade, you will find additional information like the private IP address and specific resource ID. While there are other places to find this information, this provides a quick-access method.

Connecting to the Network

We will now remotely connect to our virtual network. Remember, exposing our domain controllers via a public IP is unsafe and not recommended. We've, therefore, created a **utility** virtual machine - sometimes known as a *bastion* server - that will allow us an entrypoint into our network.

1. Make sure you have the **utility** VM selected and click on **Overview** .
2. In the *Actions* section, click **Connect** . This will download a Remote Desktop (Protocol) profile to your machine.
3. Open the RDP profile. (NOTE: You may receive a warning that "The publisher of this remote connection can't be identified." Proceed by clicking on **Connect**.)
4. Windows security will prompt you to enter your credentials. Enter the full AD credentials that was reported to you earlier by the CLI (e.g. azurecloud\cloudadmin). Additionally, enter your password. Click **OK**.
5. If the credentials were entered successfully, you should be remotely connected to the **utility** VM.
6. (Optional) If you'd like, once you are connected to the **utility** VM, you can connect remotely to the Active Directory domain controller ("**dc1**") in the virtual network. Simply open up Remote Desktop *in the active, remote session* and use the internal, private IP (e.g. 10.3.1.4) as the address. Use the same credentials to connect to the domain controller as you did with the **utility** virtual machine.

This completes our simple introduction into navigating through Azure. We'll go into more detail as we work through the rest of the workshop, but this is enough to get us started.

Create Connect Server

Objective


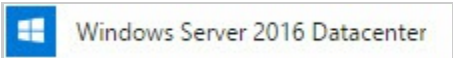
We could use our domain controller for the AD Connect synchronization server, but this is a bad idea. There's typically multiple (primary, secondary, maybe more) domain controllers in an Active Directory environment. We are only allowed to have one active/hot AD Connect synchronization server in our environment. What happens if the domain controller where the synchronization tool is installed fails? We would lose synchronization capabilities.

Let's create a standalone AD Connect synchronization server.

Create the Server in Azure

If you are not currently at dashboard within the Azure portal, go ahead and close all blades.

On the left menu, you should see **Virtual machines** . Click it.

1. In the actions section of the virtual machines blade, click on .
2. In the *Search Compute* search box, type in **Windows Server 2016 Datacenter**. Press Enter.
3. In the returned results, choose the option that simply reads .
4. In the next blade, make sure **Resource Manager** is selected. Then, click *Create*.
5. There are 4 sections to configure the virtual machine.

1. Basics

- Name: **ad-connect**
- VM disk type: **SSD**
- Username: **cloudadmin**
- Password: **Pass@word1234**
- Confirm Password: *<same as above>*
- Subscription: **Free Trial**
- Resource Group: **Use existing** - *<use the same resource group created by the CLI>*
- Location: *<use the same location you chose in the CLI>*

- Save money: **No**
- 2. Size
 - **DS1_V2**
- 3. Settings
 - Use managed disks: **No**
 - Storage account: *<use the same storage account created by the CLI>* (e.g. *azwdata###*)
 - Network: **vnet**
 - Subnet: **default (10.3.1.0/24)**
 - Public IP address: (click on it & *Create new*)
 - Name: **connect-ip**
 - Assignment: **Static**
 - Network security group (firewall): **None**
 - Extensions: **None**
 - Availability set: **None**
 - Boot diagnostics: **Enabled**
 - Guest OS diagnostics: **Disabled**
 - Diagnostics storage account: *<use the same storage account created by the CLI>* (e.g. *azwdiags###*)
- 4. Summary (just click *OK* to continue)

The machine we chose for this workshop is relatively small. After all, we only have 4 identities that we'll be synchronizing with Azure AD. If this was a production environment we would have to take into consideration that password sync's occur approximately every 2 minutes while full synchronization happens every 15-30 minutes. For production, we would need to choose a machine that is more capable of handling the workload.



Keep in mind, that we are treating Azure like our on-premises datacenter. In reality, we would have simply created a new VM in our on-premises hypervisor (Hyper-V, VMware, etc.)


Add Machine to Domain


We need to add the new machine to our Active Directory domain. AD Connect must be installed on an AD-joined machine.

Set the Private IP as Static

Before we add the machine to the domain, we need to set the private IP to static so that Azure's DHCP server doesn't reassign the IP to another machine.



1. If you are not viewing the details on the newly created machine, click on the **Virtual machines**  menu item, then click on the **ad-connect** machine in the list.
2. Once you've clicked on the **ad-connect** machine and are viewing the machine's *Overview* blade, choose **Network interfaces** .
3. In the resulting list of network interfaces, choose the single NIC that is listed (e.g. *ad-connectXXX*).

4. On the network interface menu, click on **IP configurations**  .
5. The resulting list should only contain a single configuration - *ipconfig1*. You'll notice that under the heading *PRIVATE IP ADDRESS*, the configuration is listed as *Dynamic*. Click on this configuration.
6. In the settings for the configuration, under *Private IP address settings*, change the *Assignment* to **Static**. (The IP address should be 10.3.1.6. If it is not, update it, as well.)
7. Click *Save*.

You can now close the two blades (e.g. ad-connectXXX, network interface) to arrive at the main **Network interfaces**  blade for the **ad-connect** virtual machine.

Connect to the Machine via Remote Desktop

To connect to the machine remotely, we need to download the Remote Desktop Protocol (RDP) profile.

1. Click on the **Overview**  to return to the general information for the **ad-connect** virtual machine.
2. In the **Actions** section, click on **Connect**  . This will download the RDP profile to your machine.
3. Open the profile and accept any warnings.
4. For the username, enter **\cloudadmin** (with the backslash). And, for the password, enter **Pass@word1234**. Click *OK*.
5. Again, accept any warnings.

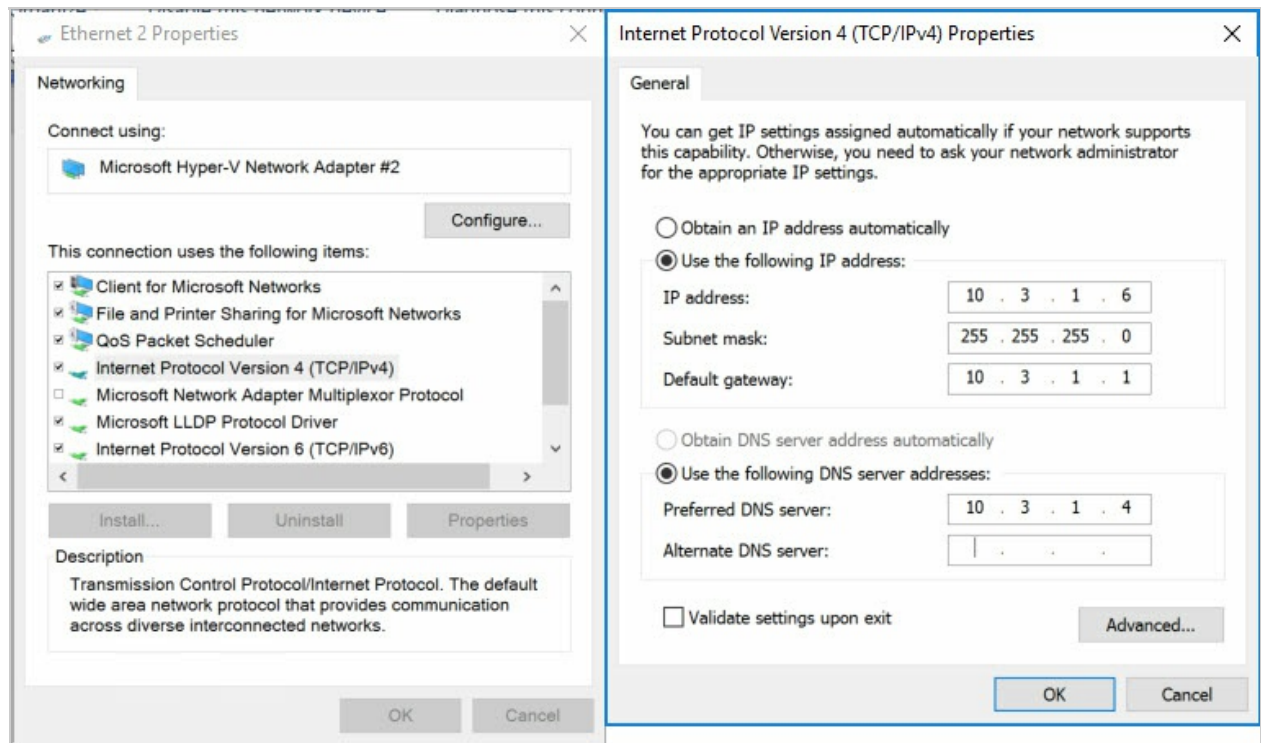
Add the Machine to the Domain

When you initially connect to the machine, you will see the *Server Manager* dashboard.

We've already set the IP on the network interface card (NIC) to be static in Azure. Technically speaking, we've created a *reservation* in Azure's DHCP server for the NIC in our virtual network. However, before we add the machine to the domain, it is best if we set the IP as static within Windows Server's TCP/IP configuration.

1. In the left menu of *Server Manager*, click on **Local Server**.
2. In the resulting page, you'll see a couple of sections. The first section is labeled *Properties*. *Properties* has two columns. Half-way down the left column, you'll see *Ethernet* followed by a number. Beside this, you will see in blue **IPv4 address assigned by DHCP, IPv6 enabled**. Click on this.
3. This will open the *Network Connections* window. Right-click on the single listed adapter and click on **Properties** in the context menu.
4. In the *Properties* window for the NIC, scroll down until you see *Internet Protocol Version 4 (TCP/IPv4)*. Highlight it, then click *Properties*.

5. Enter the values as you see them below.



6. Click *OK*, then *Close*. **NOTE:** Clicking *Close* will cause a brief interruption in your connectivity. That's okay. The connection should be re-established within a couple of seconds.

7. Once the connection has been re-established, you can close the *Network Connections* window.

8. Back in the *Properties* section, in the half-way down the right column, you will see *IE Enhanced Security Configuration*. To the right of that in blue, you probably see **On**. Click on it.

9. In the *Internet Explorer Enhanced Security Configuration* dialog, choose **Off** for both, Administrators and Users. Then, click *OK*.

10. Once more, in the *Properties* section, the second item listed in the left column reads *Workgroup*. To the right of that, you will see in blue **WORKGROUP**. Click on it.

11. In the *System Properties* dialog, half-way down, click on the *Change* button.

12. In the resulting *Computer Name/Domain Changes* dialog:

1. Leave the *Computer name* as it is (e.g. *ad-connect*).
2. Under *Member of*, change the selection to *Domain* and enter the domain name you entered earlier in the CLI (e.g. *mycompany.local*).
3. Click *OK*.
4. For the username and password enter your *Domain Admin* username and *Domain Admin Password*, respectively, as reported previously by the CLI.
5. Click *OK*.

If all goes well, you should be added to the virtual datacenter domain and receive a message stating as much. To complete this will require a reboot, thus disconnecting you from your remote session.



View Azure Domain


Objective

This next objective is very small. We simply want to verify our Azure AD domain settings and enable premium features.

Verify the Domain

If you are not currently at dashboard within the Azure portal, go ahead and close all blades.

1. On the left menu, you should see **Azure Active Directory**  . Click it.
2. On the left, in the newly expanded Azure Active Directory menu, click on **Domain names**  .
3. You should see your Office 365 domain name listed and set as *Primary*.

You may notice, at this point, that if we wanted to add a custom FQDN to Azure AD (e.g. *yourcompany.com*), we could do so here by selecting the **Add domain name**  item from the *Actions* menu at the top.


After we added our custom FQDN, we would be required to verify our ownership of the domain by adding a TXT DNS record. Once we completed the verification process, we could then choose to set our custom domain as *Primary*.



Understand that the *Primary* domain is **not** the only domain we can synchronize with our on-premises domain. In the case that, let's say, we have multiple business units that have their own Accounts Domain, we could have multiple subdomains listed here. Then, each business unit's AD would sync with its respective subdomain in Azure AD.

For our workshop, the Office 365 domain (e.g. *<yourcompany>.onmicrosoft.com*) is sufficient.

Enable Premium Features

Even though we are using a trial of Office 365 Business Premium for our workshop, Azure AD Premium is a different SKU. We, therefore, have to enable the features before we can use them.

1. While still in *Azure Active Directory*, click on the **Licenses**  menu item.

2. In the next menu, click on the **All products**  menu item.
3. On the next page, in the *Actions* section, click on **Try / Buy**  .
4. You will now see two options for enabling premium features - **Azure AD Premium** and **Enterprise Mobility Suite**. For our workshop, **Azure AD Premium** is sufficient. Click on **Free trial** in the **Azure AD Premium** tile.
5. This will initiate a 30-day trial of Azure AD Premium features. Click *Activate*.

You will need to refresh your Internet browser to see the effects of enabling Azure AD Premium. Within the Azure Active Directory blades, you may have noticed a gray bar stating that some of the features were only available in Azure AD Premium. Once you refresh your browser and return to Azure Active Directory, you should no longer see the gray bar and, instead, see all features activated.

Prepare Non-Routable Domain

Objective

In typical on-premises installations of Active Directory, utilized domain name extensions, such as ".local", create what are known as *non-routable domains*. In other words, there's no such top-level domain (TLD) extension. In the words of Microsoft's support:

Synchronization

Azure AD Connect only synchronizes users to domains that are verified by Office 365. This means that the domain also is verified by Azure Active Directory because Office 365 identities are managed by Azure Active Directory. In other words, the domain has to be a valid Internet domain (for example, .com, .org, .net, .us, etc.). If your internal Active Directory only uses a non-routable domain (for example, .local), this can't possibly match the verified domain you have on Office 365.

The objective for this step is to modify our local domain to create a routable domain. We will then update the UPN of our users to take advantage of this new domain.

Add UPN Suffixes

We will need to remotely connect to **dc1** in order to update Active Directory. Because **dc1** is not accessible from outside of the network, we'll need to connect to it through the **utility** virtual machine.

Enable the AD DS Snap-In

By default, the machines do not include the Active Directory management snap-in. For easier management, let's go ahead and enable it.

1. Go ahead and RDP into the **utility** virtual machine. Once connected to the **utility** VM, RDP into **dc1**. You can connect to **dc1** by using its DNS hostname (e.g. "dc1") or its IP address, 10.3.1.4.

2. Once connected to **dc1**, *Server Manager* should automatically open. If it doesn't, go ahead and open it now.
3. In the top-right of *Server Manager*, click on **Manage**. Then, click on **Add Roles and Features**.
4. In the "Before you begin" screen, click "Next."
5. Make sure "Role-based or feature-based installation" is selected, then click "Next."
6. For the destination server, your local domain controller should be highlighted. Click "Next."
7. We don't need to add any additional roles at this point, so just click "Next."
8. For features, we need to add two features. You can install both by selecting: **Remote Server Administration Tools > Role Administration Tools > AD DS and AD LDS Tools > AD DS Tools**. This will add the AD Domain Services snap-in and command-line tools.
9. Click "Next."
10. Finally, click "Install."

This should only take a minute or two to complete. You can click "Close" when the process has completed.

Add Suffix to AD Domains and Trusts

With the snap-in installed, we can easily add the UPN suffix to our Active Directory.

1. If it's not still open, launch *Server Manager*.
2. In the top-right of *Server Manager*, click on **Tools**. Then, click on **Active Directory Domains and Trusts**.
3. In the *Active Directory Domains and Trusts* window, right-click **Active Directory Domains and Trusts** in the left pane, and then choose "Properties."
4. In the *Alternative UPN suffixes* box, enter your full domain of your Office 365 tenant (e.g. <yourcompany>.onmicrosoft.com). Click "Add."
5. Click "OK."

Change the UPN suffix for existing users

Now that we've added an alternative UPN to our domain, we need to update each of our users to use this domain as the *primary* UPN as that is what Azure AD Connect uses to match identities.

1. Again, if it's not still open, launch *Server Manager*.
2. In the top-right of *Server Manager*, click on **Tools**. Then, click on **Active Directory Users and Computers**.
3. In the *Active Directory Users and Computers* window, expand your ".local" domain and click on **Users**.

4. There are 3 user accounts for which we need to update the UPN (NOTE: We do not want to sync the local *cloudadmin* enterprise administrator account to the cloud in order to preserve boundaries. You should utilize a separate account in Azure for administering Azure AD.)
 - Jim Smith
 - Richard Jackson
 - Sally Holly
5. For each of these accounts, right-click on the account and choose **Properties**.
6. Click on the **Account** tab.
7. In the dropdown list next to the username, change the selection from your local domain to the "onmicrosoft.com" domain. Click "OK."

Congratulations! Your local Active Directory is now ready to begin basic synchronization with Azure AD.

One thing to keep in mind is that updating the UPN in the last step now requires these three users to use the FQDN of the *onmicrosoft.com* account rather than the *.local* domain if they use the *first.last@domain.local* format for the username. However, most users don't login using a FQDN. Instead they, like what we've done in this workshop, use the pre-Windows 2000 method of specifying the username (e.g. MYCOMPANY\first.last). Not too big of a deal, but, again, just something to make note of.

Finally, if you have a lot of users in your domain, manually updating the UPN domain for each user can be a tedious task. Luckily for us, here's a PowerShell script for that:

```
$LocalUsers = Get-ADUser -Filter {UserPrincipalName -like '*mycompany.local'} -
Properties userPrincipalName -ResultSetSize $null

$LocalUsers | foreach {$newUpn = $_.UserPrincipalName.Replace("mycompany.local"
,"mycompany.onmicrosoft.com"); $_ | Set-ADUser -UserPrincipalName $newUpn}
```

Installing Azure AD Connect

Objective

We are now finally ready to begin the configuration of our synchronization process. Upon completion of this step, your virtual datacenter will be sync'ing with Azure AD.

Install Azure AD Connect

To have our local domain synchronize with Azure AD we need Azure AD Connect. We will install it on the **ad-connect** virtual machine.

1. As you have previously connected to the **ad-connect** and **utility** VMs already, let's RDP to the **ad-connect** machine once more.
2. Once you've successfully connect to **ad-connect**, you will need to download and install the Azure AD Connect tool. You can download it from <https://www.microsoft.com/en-us/download/details.aspx?id=47594>.
3. Upon installing Azure AD Connect, it will automatically run.
4. Check the box agreeing to the license terms and click **Continue**.
5. For the moment, **Express Settings** are sufficient. We'll customize it later. So, go ahead and click **Use express settings**.
6. Once the basic initialization has completed, you will be asked for your Azure AD credentials. Enter the credentials you use for authenticating against Azure for your trial subscription (e.g. <yourusername>@<yourcompany>.onmicrosoft.com). Click **Next**.
7. For connecting to AD DS, use the *cloudadmin* credentials provided to you by the CLI (you've also used these credentials for connecting remotely into the VMs).
8. The next screen confirms mapping between the local UPN and a *verified* domain in Azure AD. Since we don't have a verified domain in Azure - we're just using the default *.onmicrosoft.com - all local accounts will be "re-mapped" to the onmicrosoft.com domain. For our workshop, we can simply check the box next to **Continue without any verified domains** and click **Next**.
9. **BEFORE YOU CLICK Install**, *uncheck* the box next to **Start the synchronization process when configuration completes**. Otherwise, *all* accounts (including system accounts) will be synchronized creating a lot of bloat in our Azure AD. We're going to create some filters

before we conduct our first sync.

10. Now, you're ready to complete the install for Azure AD Connect. Click **Install**.

After a few minutes, you should receive confirmation that the configuration has completed. It may also give you a couple of house-keeping recommendations. Go ahead and click **Exit** to exit the installer.

Configure Synchronization Filters

We need to create some filters to *only* synchronize our users who's UPNs have been updated to the "new" domain.

In order to do this, we need to create what's called a "Positive Filter." Basically, we're instructing AD Connect to "only sync these." Keep in mind that, by default AD Connect will sync *all* users in our domain (or OU, depending how we have configured the sync scope). So, in order to create a positive filter, we need to create two rules - one that specifies which users to sync; and, another that instructs AD Connect to *not* sync all of the remaining users.

Both of our rules are considered *Incoming Sync Rules (ISR)* because they are determining what data we are allowing *into* the metaverse from our local Active Directory.

First, let's begin by opening up the synchronization rules. In the **Start Menu** of the **ad-connect** VM, click on **Synchronization Rules Editor**. You'll see approximately 15-20 default rules. We're going to add our two rules to the top in order for our rules to take precedence.

Users Match Filter

This filter will instruct which users we *do* want to sync with Azure AD.

1. In the *Synchronization Rules Editor* click on **Add new rule**.
2. **Description:**
 1. Name: **UPN Demo - Users Match Filter**
 2. Description: **Only sync users who match our onmicrosoft.com UPN**
 3. Connected System: **choose your .local domain**
 4. CS Object Type: **user**
 5. Metaverse Object Type: **person**
 6. Link Type: **Join**
 7. Precedence: **50**
 8. Enable Password Sync: **check**
3. **Scoping filter:**
 1. Click **Add group**
 2. Click **Add clause**
 3. In the clause, enter the following values for each column, respectively:
 - Attribute: **userPrincipalName**
 - Operator: **ENDSWITH**
 - Value: **<yourcompany>.onmicrosoft.com**
4. **Join rules:** leave blank
5. **Transformations:**

1. Click **Add transformation**
2. In the transformation, enter the following values for each column, respectively:
 - FlowType: **Constant**
 - Target Attribute: **cloudFiltered**
 - Source: **False**
6. Click **Save**.

Users Catch-All Filter

This filter will instruct which users we *do not* want to sync with Azure AD.

1. In the *Synchronization Rules Editor* click on **Add new rule**.
2. **Description:**
 1. Name: **UPN Demo - Users Catch-All Filter**
 2. Description: **Catch and filter out all other users who do not have the onmicrosoft.com domain.**
 3. Connected System: **choose your .local domain**
 4. CS Object Type: **user**
 5. Metaverse Object Type: **person**
 6. Link Type: **Join**
 7. Precedence: **99**
3. **Scoping filter:** leave blank
4. **Join rules:** leave blank
5. **Transformations:**
 1. Click **Add transformation**
 2. In the transformation, enter the following values for each column, respectively:
 - FlowType: **Constant**
 - Target Attribute: **cloudFiltered**
 - Source: **True**
6. Click **Save**.

Before you close the *Synchronization Rules Editor*, notice that at the bottom of the window, you are able export rules to a PowerShell script. For any custom rules, this should be part of your disaster recovery plan in case the AD Connect synchronization server fails. You may now close the editor.

Enable Password Writeback




One last thing we want to do is configure the Azure AD Connect tool to writeback password changes to our local Active Directory. Additionally, remember that, during installation, we elected to not start the synchronization service. So, we going to do that, as well.

1. On the desktop of your **ad-connect** VM, you should see a new icon for **Azure AD Connect**. Go ahead and open the tool.
2. Immediately, you'll notice that while the connect tool is open, the service is suspended.
3. Click **Configure**.
4. Select **Customize synchronization options** and click **Next**.

5. Type in your credentials for Azure and click **Next**.
6. Type in your credentials for the local Active Directory and click **Next**.
7. In the **Domain and OU Filtering**, we only want to sync our **Users** group. This will keep Azure AD nice and tidy. So:
 1. Select **Sync selected domains and OUs**.
 2. Expand the local domain and uncheck all OUs *except* **Users**.
 3. Click **Next**.
8. Check *both* **Password synchronization** and **Password writeback**. Click **Next**.
9. **BEFORE YOU CLICK Configure**, check the box next to **Start the synchronization process when configuration completes**. This time, we want the synchronization service to begin sync'ing our users.
10. Click **Configure**.
11. Once the configuration has completed, you should receive a confirmation. Click **Exit**.


Confirming a Successful Synchronization

Give the synchronization service a minute to "spin up" and conduct its first sync. Then, let's head over to our Azure portal to confirm that the synchronization was successful. Once you've reached your Azure portal, perform the following steps.

1. On the left menu, click on **Azure Active Directory**  .
2. In the *Azure Active Directory* blade, click on **Users and groups**  .
3. In the *Users and groups* blade, click on **All users**  .

We should now see all 3 users from our local Active Directory listed here. Question... If our Azure AD grows to a huge list of users, how will we know which users originated in the cloud and which ones are sync'ed from our on-premises Active Directory?

While we are still on the same blade (viewing our users list), do the following:



1. In the *Actions* section, click on **Columns**  .
2. Check the box next to **Source of authority**.
3. Click **Apply**.

We now see from where our users are originating, whether that on-premises (e.g. Windows Server AD) or the cloud (e.g. Azure Active Directory).

Remember that any changes made to synchronized users (e.g. Windows Server AD) are replicated back down to our local Active Directory. However cloud users are *not* synchronized.

Completing Password Writeback

In completing the Azure AD Connect configuration, we enabled password writeback. But, by default, users aren't able to update their passwords in Azure. We need to enable users to have the ability to update their passwords.

1. While you are still on the *Users and groups* blade, click on **Password reset**  .
2. You will see here that self-service password is not enabled for anyone. Click on **All** and then click **Save**.
3. Finally, let's confirm that password writebacks are enabled in Azure. Click on **On-premises integration**  .
4. From here, you will see that password writebacks are, indeed, enabled along with restricting users from unlocking their accounts without resetting their passwords.

You now have our local Active Directory sync'ing with our Azure AD.

Additional Notes

Interestingly enough, if you log out of Azure and attempt to login with one of the UPNs that was sync'ed (for example, **jim.smith@<yourcompany>.onmicrosoft.com** with the default password **Pass@word1234**), Azure will require you to set up a secondary authentication method - phone or email - prior to being able to login.

Also, if you login to your Office 365 trial tenant, you'll see the users from your on-premises Active Directory listed. All you would need to do at this point is assign them licenses.

Monitoring Health

Objective

We are going to conclude this workshop with enabling monitoring on our Active Directory Domain Services.

Install the Agent

In order to see reports for our domain services, we need to install the Azure AD Connect Health Agent for AD DS onto our domain controller.

Disable IE ESC

By default, Internet Explorer Enhanced Security Configuration is enabled which will prevent us from downloading anything. We need to disable this. (**NOTE:** In production, you would typically not do this. In production, you would leave IE ESC enabled and copy the downloaded agent via RDP onto the machine. However, since this is a workshop, we'll make some concessions.)

1. If you're not still connected to the **dc1** VM, go ahead and do that now. As a reminder, you will need to do so *through* the **utility** machine.
2. Once you've connected **dc1**, open *Service Manager* if it's not already open.
3. In the left menu of *Server Manager*, click on **Local Server**.
4. In the resulting page, you'll see a couple of sections. The first section is labeled *Properties*. *Properties* has two columns. Half-way down the right column, you will see *IE Enhanced Security Configuration*. To the right of that in blue, you probably see **On**. Click on it.
5. In the *Internet Explorer Enhanced Security Configuration* dialog, choose **Off** for both, Administrators and Users. Then, click *OK*.

Download and Install the Agent



Now we're ready to download and install the agent.

1. On **dc1**, open a web browser and go to <http://go.microsoft.com/fwlink/?LinkID=820540>. This will automatically download the agent.
2. Once the download is complete, run it.

3. In the **Microsoft Azure AD Connect Health agent for AD DS Setup** window, click **Install**.
4. Once it has completed installation and has informed you that the setup was successful, click **Configure Now**.
5. This will run a PowerShell script and require that you authenticate with Azure. Enter your credentials for your *<yourcompany>.onmicrosoft.com* account.
6. After a few seconds of watching the script continue to run, you should see that the **Agent registration completed successfully**. Go ahead and close the PowerShell window.

View Agent Metrics

We're now ready to see how our domain controller is functioning. Let's return to Azure to view the reports.

1. In Azure's left menu, click on **Azure Active Directory**  .
2. In the *Azure Active Directory* blade, click on **Azure AD Connect**  .
3. Under **HEALTH AND ANALYTICS**, click on **Azure AD Connect Health** (I know, it's a little obscure).
4. There are three sections to the health dashboard - AD FS, AD Connect (Sync), and AD DS. Since we don't have Federated Services configured, this tile should be empty. However, you should see both, respective, domains under AD Connect and AD DS. Clicking on these domains will give us details of how they are functioning.

Azure AD Connect Health is still very young in development. As you click around, you may find some features disabled. Keep monitoring this to see how it expands to give you greater visibility into your AD infrastructure.