Virtual Lab Setup and Configuration Guide

# Introduction

These instructions are written as a guide to help get your testing environment up and running, and also to assist with the tasks surrounding the practicing of the competencies discussed in these steps. This text is not meant as substitute for classroom instruction, or for detailed vendor implementation documentation, but more as a refresher to help keep you on track. It will also assume that you have a basic understanding of Windows networking and operating system configuration, and that you can perform common tasks, such as reviewing and configuring IP information. Those types of tasks will not be covered in detail.

Also, though typically, administration steps can typically be done remotely, though PowerShell, Server Manager, MMC, RSAT, etc., steps in this document are almost always done on the server, from the console to reduce confusion over which device steps are being executed on.

# Template Creation

1. Install Hyper-V role and containers features:
   1. From Server Manager, click Manage, add roles/features.
   2. Click next twice.
   3. Check Hyper-V.
   4. Select ethernet for virtual networks, and CredSSP for live migrations.
   5. Click next until the install validation screen, and click install.
   6. Restart when prompted.
2. Install Chrome for ease of use via PowerShell.
   1. Note that technically, this is bypassing security features that prevent us from accessing web content on servers, but in the interest of getting you up and running, we are pulling install media directly to the server. This isn’t best practice.
   2. [Run the following command](https://superuser.com/questions/1008307/how-to-click-download-chrome-button-from-windows-server-in-ie) in an elevated PowerShell window. Note that you should always read and understand scripts before running them.

**$LocalTempDir = $env:TEMP; $ChromeInstaller = "ChromeInstaller.exe"; (new-object System.Net.WebClient).DownloadFile('http://dl.google.com/chrome/install/375.126/chrome\_installer.exe', "$LocalTempDir\$ChromeInstaller"); & "$LocalTempDir\$ChromeInstaller" /silent /install; $Process2Monitor = "ChromeInstaller"; Do { $ProcessesFound = Get-Process | ?{$Process2Monitor -contains $\_.Name} | Select-Object -ExpandProperty Name; If ($ProcessesFound) { "Still running: $($ProcessesFound -join ', ')" | Write-Host; Start-Sleep -Seconds 2 } else { rm "$LocalTempDir\$ChromeInstaller" -ErrorAction SilentlyContinue -Verbose } } Until (!$ProcessesFound)**

1. Create new virtual machines in Hyper-V Manager
   1. Launch Hyper-V Manager from the tools menu in Server Manager
   2. Create a new virtual machine from the action menu
   3. Go through the wizard, and use the following settings:
      * Create new virtual machine in hyper-v manager.
      * Name: DC 1
      * Generation 2
      * Memory (RAM): 3072 DYNAMIC MEMORY
      * Network: Virtual Switch
      * VHD: Create a 64 GB dynamic drive
      * Install an operating system later
   4. After reviewing the settings, click finish
2. **IN CHROME**, Download an evaluation copy of Server 2019 **IN ISO FORMAT** at <https://www.microsoft.com/en-us/evalcenter/evaluate-windows-server-2019>. You will need to enter some registration information to be able to download from Microsoft.
3. Add a DVD drive to DC 1 and mount your install media
   1. Right-click on DC 1 and click settings
   2. Click SCSI controller.
   3. Select DVD Drive, and click add.
   4. Select the Image file radio button, click browse, and navigate to where your Server 2019 ISO is stored.
   5. Click apply.
4. Navigate to firmware section of hardware settings, and set your list to boot in the order DVD Drive > Hard Drive > Network Adapter, and then click OK.
5. Repeat steps 3, 5, and 6 (highlighted in green) for a second and third server 2019 install. The only differences should be in the name of the VM. Where the name of the first VM was “DC 1”, the second should be name “DC 2” and the third, “IPAM”.
6. Select DC 1 and connect to it. Click start and press any key at the prompt to boot from DVD, starting the install process. Install server 2019 Datacenter Desktop Experience, and repeat this process for the other two VMs.
   1. Note: You may use the entire drive for the OS, so do a custom install to the entirety of disk 0 when prompted.
7. Download Windows 10 Eval (64-bit) <https://www.microsoft.com/en-us/evalcenter/evaluate-windows-10-enterprise>
8. Create a VM titled Client, gen 2, with 2048 RAM, using the virtual switch. Create a new VHD with 64 gigs of space. Attach the ISO you just downloaded as install media.
9. Boot the Client VM and select install. Do a custom install on the default drive and start the install.
   1. After install finishes, set region and language.
   2. If notified of lack of internet access, skip those setup steps for the moment.
   3. Set the login name and password to the standard credentials, as well as some password reset questions. Decline additional services/telemetry
10. On the Azure machine, [create and configure an internal NAT for the VMs to use](https://docs.microsoft.com/en-us/azure/virtual-machines/windows/nested-virtualization).
    1. In an elevated PS shell: **New-VMSwitch -Name "InternalNAT" -SwitchType Internal**
    2. Run **Get-NetAdapter** and note the ifIndex for the switch you just created (InternalNAT)
    3. Create a new interface using new-netipaddress, substituting your ifindex**. New-NetIPAddress -IPAddress 192.168.0.1 -PrefixLength 24 -InterfaceIndex 13**
    4. Create a new NAT. Our example creates a /24 in the given address range. **New-NetNat -Name "InternalNat" -InternalIPInterfaceAddressPrefix 192.168.0.0/24**
11. If the created VMs are currently running, shut them down, and edit the individual VM's Hyper-V settings to use the InternalNAT as a network adapter instead of its currently configured adapter.
12. Configure each of your machines to use a usable IP address from the range you configured.
    1. Start the nested VMs in Hyper-V.
    2. Open the IPv4 settings for the machine’s interface and configure it with its new IP address.
    3. If you used IPs from the examples given, you could configure your 3 machines to use 192.168.0.2-4, with 255.255.255.0 as a subnet mask, and 192.168.0.1 as the gateway address.
    4. This will give you connectivity between your VMs until network services are configured.
    5. To be able to ping between workstations, you need to set firewall rules to allow this. Note that you will still not be able to ping outside of Azure; all ICMP traffic is blocked at the Azure firewall, so you will not be able to check for connectivity by pinging Google, etc., like you might normally.
       1. Run This command in an elevated PowerShell or command prompt window on each VM, and the host machine. **netsh advfirewall firewall add rule name="ICMP Allow incoming V4 echo request" protocol="icmpv4:8,any" dir=in action=allow**
13. Verify connectivity between all machines with ping
14. Safely shut down any running VMs and the host machine, and publish this template in the Azure Labs interface. This is now the “zero point” for further work.

# Installing Active Directory Domain Services

1. Connect your Azure Labs machine via RDP. Server Manager will automatically launch after you log in.
2. Launch Hyper-V manager from the tools menu, and start both DC 1 and DC 2.
3. Rename them to fit our naming scheme
   1. Right click windows icon > system > rename. Name DC-1 as "DC-1" and DC-2 as "DC-2" and reboot each machine when prompted.
4. On DC-1, add a role in Server Manager (manage > add roles and features). In the wizard, click next > role based (next) > this server (next) > select active directory domain services, and select add features on the spawned window. Select next all the way through until the final screen, and select install.
5. Repeat these steps on DC-2. Once the install completes, close the wizards on both VMs.
6. Update both DCs to use opposite server’s IP addresses as primary DNS, and then own IP as secondary. No other DNS servers need to be configured at the interface level, because the Azure machine's NIC is bound to the Hyper-V virtual switch, and provides access to the outside world.
7. On DC-1, open notifications in server manager, and select "promote this server to domain controller" under post-install config. Select "add a new forest". Call it "lab.local" and click next. Input memorable credentials for the DSRM password and click next. Select next through the rest of the screens until the prerequisite check, and then select install. The VM will restart when completed.
8. On DC-2, open notifications in server manager, and select "promote this server to domain controller" under post-install config. Select "add a domain controller to an existing domain" and type in "lab.local" as the domain name. Enter "lab\Administrator" and your password when prompted for credentials. Click next. Input memorable credentials for the DSRM password and click next. Select next through the rest of the screens until the prerequisite check, and then select install. The VM will restart when completed.

# Configuring Reverse Lookup, and a Secondary DNS Server

1. On DC-1, open DNS management from the tools drop-down in server manager.
2. Expand the console tree, and expand Forward Lookup zones.
3. Right-click on the lab.local zone and click properties.
4. Click on the name servers tab, and if DC-2 is not already listed, and click add.
   1. Type in the FQDN for DC-2 (dc-2.lab.local) and click resolve.
   2. Click OK to add this server once it has resolved and validated.
   3. Accept the defaults for the rest of the options in the wizard and OK back to the DNS Manager.
5. Add a reverse lookup zone by right-clicking on reverse lookup zone under DC-1 in the console tree, and selecting new zone.
   1. Click Next > Primary zone > next> all DNS servers > IPv4 > set the Network ID to 192.168.0 > next > set update to “only secure dynamic” > next > finish.

# Configuring DHCP

1. In DC-1, open the add roles wizard in Server Manager. Select the DHCP server role, and add any required features that are requested by the wizard. Click next until the confirmation screen and click Install. Upon completion, close the wizard, open the notification center, and open the post-install DHCP configuration window.
2. Click next at the description window, and then Commit at the authorization window, using the current user's credentials. Close the wizard once complete.
3. In server manager, launch DHCP manager from the tools menu.
4. In the navigation bar, right-click on the server name, and authorize the server.
5. Expand the server name, and right-click on IPv4 > Create new scope.
6. Name the scope "Clients" and click next.
   1. Configure the range to start at 192.168.0.5 and end at 192.168.0.200 (note that for simplicity, we're taking the network we've been using, and converting it to use DHCP. This could be any network you want to use). The CIDR and mask will autopopulate below.
   2. Click next. Exclusions can be added on this screen for any addresses we want to not be leased out. Click next.
   3. Leave lease durations as default, and click next.
   4. Select "Yes" for configure DHCP options. Add 192.168.0.1 as the default gateway (or the DG of the network you have configured at this step) AND CLICK ADD.
   5. Next through the rest of the options and click finish. *DC-1's IP address is automatically being added as the DNS server to push out to clients (DHCP option 6)*.

Boot the Windows 10 VM, switch the Ethernet interface from a static addressing to automatic, and verify that you are pulling valid information from DHCP, and that you have network access.

# DHCP Failover

1. On DC-2, install DHCP using the steps from above, and authorize the server (steps 1-4), but do no further configuration.
2. On DC-1, open DHCP management in server manager.
3. Expand the server name > IPv4 and right-click on the scope we created above. Click “Configure failover”.
4. Our scope (and any other scopes that may have been configured) will be shown here. Click next.
5. Key in DC-2 as the partner server and click next.
6. Leave these settings as default, but configure a password as the shared secret. Click next, and then Finish.
7. Open DHCP configuration on either server (remember, they're load balanced and the scope config is shared), expand the IPv4 scope for this network, and view the scope options.
8. Double-click on option 6 (DNS servers).
9. Make changes so that only the two servers in the domain are listed here (as in, add the IP of DC-1 or DC-2 if is missing, and remove any extra addresses), and click OK.
   1. As both of our servers provide DNS, and both provide DHCP, an outage of one of them will still allow services to be provided to clients.

This will configure our two servers to load balance DHCP tasks. It will also enable either server to take over DHCP tasks entirely should one of them fail. If you check in DHCP configuration on DC-2, you will see that the scope we have been using on our network now appears under IPv4 without us configuring it.

As a further exercise, you can disable the DHCP service (or pause/shut down the VM completely) on DC-1, do an ipconfig /release and then ipconfig /renew on our client VM, and still pull a valid IP from our configured scope. Prior to the configuration of our failover, this would have failed and an APIPA address would have been generated on our Win10 machine. You can still access the internet as well.

# Joining Computers to the Domain

1. In ADUC, create a standard user to use for logging in to the client workstation in the steps following its domain join.
2. On the Client VM, right click on the Start Menu > click System > click System Info > click Change Settings (next to computer name) > click the “change” button (next to change domain or workgroup)
3. Change name to "Client" and click the "Domain" radio button.
4. Set the domain to "lab.local" and click OK.
5. Use the domain (lab\administrator) administrator credentials at the prompt for credentials, and then click OK.
6. At the "welcome to the domain" pop-up, click OK. You will be prompted to reboot, so do so.

# DFS Namespaces and Replication

1. Add the DFS Namespaces and DFS Replication features (located under file and storage services > file and iSCSI services in the add features portion of the Add Roles wizard in server manager). Also add any dependent roles that are required. A restart my now be pending. If so, reboot.
2. Repeat these tasks on DC-2.
3. On DC-2, open disk management.
4. From the action menu, select “Create VHD”.
   1. Browse to point to the location to save the disk. Key in C:\storage as the location/name for the volume.
   2. Make the volume 1 GB in size. Make the disk dynamic. Click OK.
5. Right-click on the new disk and initialize it. Accept the default settings.
6. Create a new simple volume on the disk, using default values, selecting the E: drive as the drive letter for this volume, and "Storage" as the drive label.
7. Repeat these steps (steps 4-6) on DC-1. For clarification, we want DFS namespaces and replication installed on both servers, as well as identical VHDs created and mounted on both servers.
8. On DC-2, open DFS Management via the tools menu in Server Manager.
   1. In the navigation tree, right-click on name-spaces > click on new namespace.
      1. Type in DC-2 as server. Click next.
      2. Enter a name for the namespace, such as "files". Edit the path of the shared folder to match the volume created above, and modify permissions to "all users have read and write permissions" for this exercise.
      3. Accept the defaults for the remainder of the wizard, and create the namespace.
      4. Expand namespaces and select the new [\\lab.local\files](file:///\\lab.local\files) entry. Click on "add namespace server". Type in DC-1 as the name space server and edit settings. Update the local path of shared folder to match DC-1's E: drive, and permissions to match all users with r/w access, and OK through the dialogs.
   2. Right-click on replication in the left-hand navigation tree, and create a new replication group.
      1. Leave multipurpose replication group selected, and click next.
      2. Title the group "file\_share" and click next.
      3. Add both servers to the members list and click next. Click next twice.
      4. Choose DC-2 as the primary member and click next.
      5. Choose the E: drive as the local path to replicate and click OK.
      6. Edit the local path to replicate to on DC-1. Set its membership status to enabled, and set its local path to E:.
      7. Next through until the creation process completes.

Now, when a user navigates to [\\lab.local\files](file:///\\lab.local\files), they will transparently be redirected to one of the two shares on either DC-1 or DC-2. The contents of those shares will be replicated in the background. Which resource to use normally would be chosen by site or other metrics, but our test environment is rather flat here. We would normally map to a more traditional storage solution, like a SAN/NAS/Storage Spaces/etc., but these are not available to use in this environment, so we are simulating with a VHD. An illustration of DFS is available in the official documentation at <https://docs.microsoft.com/en-us/windows-server/storage/dfs-namespaces/dfs-overview>.

VHDs need to be attached after each reboot, either manually or through a script/scheduled task. Take note of this if you have restarted and you can’t access your DFS share anymore. <https://www.mysysadmintips.com/windows/clients/408-automatically-attach-virtual-hard-disk-vhd-on-system-startup>

# Installing and Configuring IPAM

1. Start and connect to the third install of Server 2019, IPAM.
2. Configure its network interface to use 192.168.0.2 as its IP address, the default gatway of this subnet, DC-1 as DNS 1, and DC-2 as DNS 2.
3. Rename this computer to “IPAM”. Restart when prompted.
4. Join IPAM to the domain. Remember to use the appropriate credentials when prompted.
5. Install IPAM and required additional features from the features section of the Add Roles wizard in Server Manager.
6. Click IPAM in the services list in the left of Server Manager, and click “Provision IPAM” in the task list.
   1. Click next. Accept defaults and click next.
   2. Set a GPO prefix to use for rules created (I use “IPAM” in this example). Click next.
   3. Click apply.
   4. Once provisioning completes, close the wizard.
7. If you haven’t already, log out, and log in as domain admin (lab\administrator). You require DA permissions to invoke GPO provisioning.
8. Run the following command, and accept at each prompt: ***Invoke-IpamGpoProvisioning –Domain lab.local –GpoPrefixName IPAM.*** This is a PowerShell a script to push out the required GPOs to facilitate communication between devices and the IPAM server. If your GPO prefix is different than the one used in this example, make the change accordingly in the script’s switch.
9. Open group policy manager, and locate the three policies that have been created under your domain > group policy objects. You will need to add your IPAM server and the two domain controllers to the security filtering for all IPAM policies, so that these policies are applied to the services that we want to monitor and control. (If they are not included in filtering here, they will be excluded from these policies, and will show up as “blocked” on the IPAM server.”)
10. In the IPAM screen in Server Manager, click on “Configure server discovery” in the task list in server manager.
    1. Click on “get forests” to populate the AD information.
    2. Close the configure server discovery window, and open it in a few seconds once the fetch task completes. The forest and domain drop-downs should be populated with your domain information.
    3. Add “lab.local”, and it will appear in your list of roles to discover. All role boxes will be checked. Click OK.
11. In server manager, click “Start server discovery. This will start the discovery task. Wait for it to complete (up to several minutes).
12. In “select/add servers to manage” (server inventory), right-click on each server and make sure the three roles (DC, DNS, DHCP) are marked, and set the server to managed. Troubleshooting may (will most likely) now be required to get both servers in managed status, and for IPAM status to be “Unblocked”. Here are some points to consider:
    * Run gpupdate /force on all 3 servers, and run gpupdate /r to make sure that the IPAM policies are applying to our servers, allowing data to be shared
    * IPAM will not work when we are configured to manage DHCP and the dhcpaudit share is not present on servers that are running a DHCP server. Verify that a share called “dhcpaudit” is present, and that it is pointing to C:\Windows\System32\dhcp on both DCs. <https://redmondmag.com/articles/2015/11/01/implement-ipam.aspx>
    * Un-tick one of the services (DC, DNS, DHCP) and mark the offending server unmanaged, and click OK, and then set it back to get a full refresh of its status.
    * Reboot the offending server, and then do the above step.
13. Once we have green checks for both DCs, and they are in a managed state, retrieve data from the managed servers by running step 6 from the overview on the IPAM screen in server manager.

Once the data is pulled, we can now view information from our environment in a centralized location. For example, we can see from the IP address blocks section that our private addresses are (significantly) underutilized, which is expected since we have one client in our scope. There is also overlap since we have configured failover. Much information can be edited through IPAM in lieu of connecting to individual servers to edit it there.