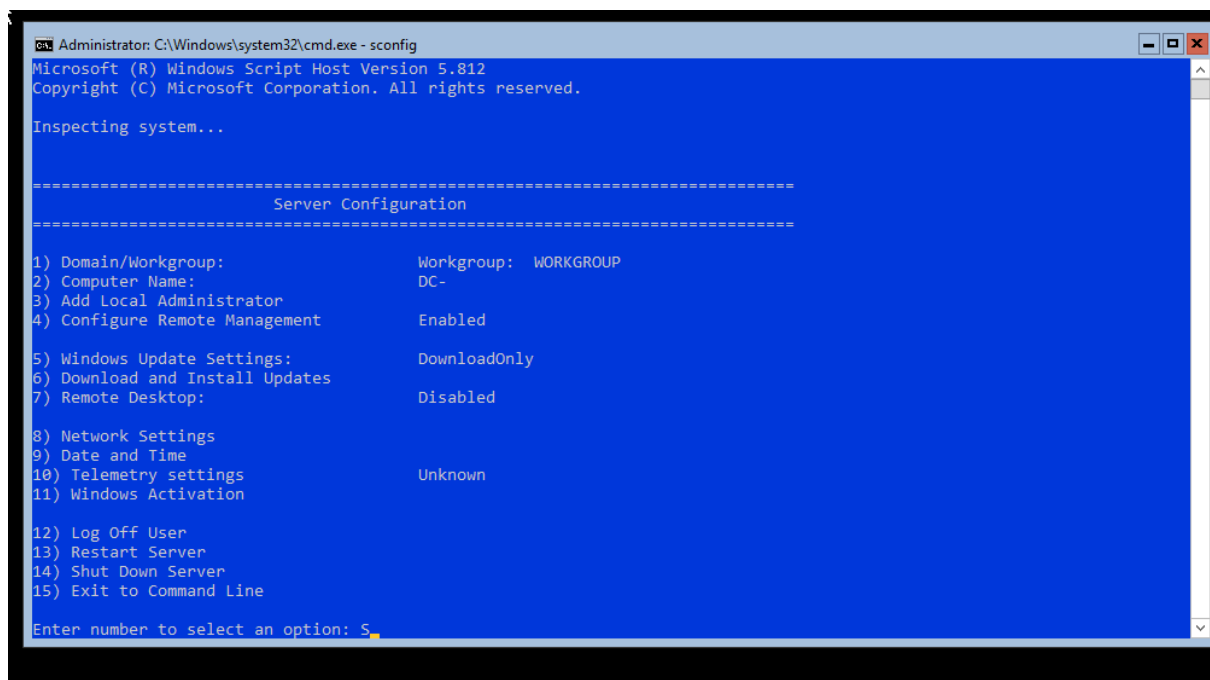


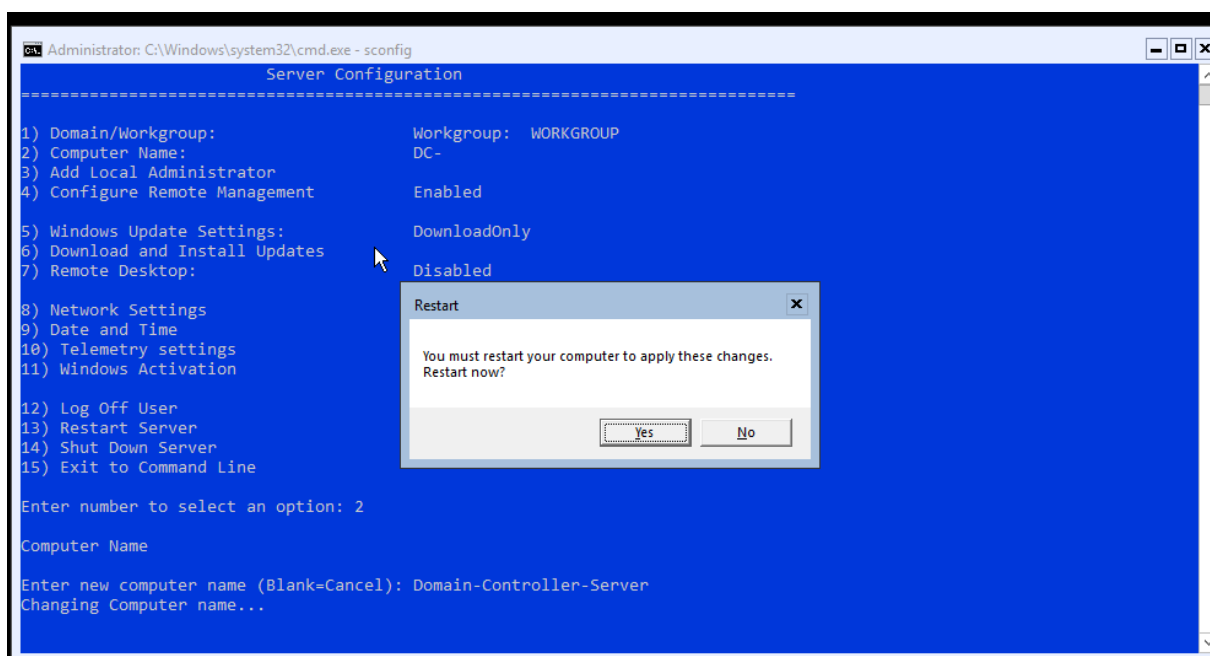
# Windows Server From Nothing

Edition 3

Upon installing the Windows Server 2019 ISO from Microsoft, I've used *sconfig* and opened the server configuration menu.



It looks simple in theory but makes me feel like a genius. I have started tinkering with the settings and changed the computer name too.



I realised, more or less after installing it, that this is the Windows Server *Core*, and not the desktop experience. I did some online research, and learned that it should, in theory, be possible to administer active directory through the command line interface alone. It should be a fun challenge, most definitely difficult, but I should learn a great deal.

I have gotten to the PowerShell interface by simply typing **powershell** into the command module, which makes it easy.

```
C:\Users\vboxuser>powershell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\vboxuser> _
```

AI suggests that I install Active Directory Domain Services now using some cryptic powershell command, but since I do not know how to use that command yet, I've used the **Get** command to understand what features I can actually add to the Windows Server. This'll make it handy since it is basically a catalogue of all windows features that I might ever need in administering or developing this Domain Controller.

```
PS C:\Users\vboxuser> Get-WindowsFeatures
Get-WindowsFeatures : The term 'Get-WindowsFeatures' is not recognized as the name of a cmdlet, function, script file,
or operable program. Check the spelling of the name, or if a path was included, verify that the path is correct and
try again.
At line:1 char:1
+ Get-WindowsFeatures
+ ~~~~~
+ CategoryInfo          : ObjectNotFound: (Get-WindowsFeatures:String) [], CommandNotFoundException
+ FullyQualifiedErrorId : CommandNotFoundException

PS C:\Users\vboxuser> Get-WindowsFeature

Display Name                                         Name                               Install State
-----
[ ] Active Directory Certificate Services            AD-Certificate                    Available
[ ] Certification Authority                         ADCS-Cert-Authority              Available
[ ] Certificate Enrollment Policy Web Service        ADCS-Enroll-Web-Pol              Available
[ ] Certificate Enrollment Web Service               ADCS-Enroll-Web-Svc              Available
[ ] Certification Authority Web Enrollment           ADCS-Web-Enrollment              Available
[ ] Network Device Enrollment Service               ADCS-Device-Enrollment           Available
[ ] Online Responder                                ADCS-Online-Cert                  Available
[ ] Active Directory Domain Services                AD-Domain-Services               Available
```

By the way, what is a domain controller anyway? It always sounded so fancy and authoritative whenever I heard it. Here's a nice online source that talks about it: *"A domain controller is the server responsible for managing network and identity security requests. It acts as a gatekeeper and authenticates whether the user is authorized to access the IT resources in the domain"* (SolarWinds 2025). Isn't that neat?

I never realized it was a security thing, but it makes sense, you wouldn't want random users on your network accessing your resources.

Now, I need to install active directory domain services on my Windows Server Core, Microsoft provides some documentation on how to do this, as well as the powershell lines themselves, which is handy.

<https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/deploy/install-active-directory-domain-services--level-100>

It's amusing how easy that was, especially considering I didn't have to craft the command myself. Microsoft describes this as "AD DS server role and installs the AD DS and Active Directory Lightweight Directory Services (AD LDS) server administration tools, including GUI-based tools such as Active Directory Users and Computers and command-line tools such as dcdia.exe". They note how server administration tools are not installed by default, but I don't understand the use case of installing ADDS, but not server admin tools.

```
Collecting data...
10%
[oooooooooooo]

10.0.2.2

Ethernet adapter Ethernet 2:

    Connection-specific DNS Suffix  . : 
    Link-local IPv6 Address . . . . . : fe80::f71e:a061:7589:5435%4
    Autoconfiguration IPv4 Address. . . : 169.254.51.43
    Subnet Mask . . . . . : 255.255.0.0
    Default Gateway . . . . . : 

C:\Users\vboxuser>adprep
'adprep' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\vboxuser>powershell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\vboxuser> Install-WindowsFeature AD-Domain-Services -IncludeManagementTools
```

The command used above was:

***Install-WindowsFeature -name AD-Domain-Services -IncludeManagementTools***

```
PS C:\Users\vboxuser> Get-Command -Module ADDSDeployment
```

CommandType	Name	Version	Source
Cmdlet	Add-ADDSDomainControllerAccount	1.0.0.0	ADDSDeployment
Cmdlet	Install-ADDSDomain	1.0.0.0	ADDSDeployment
Cmdlet	Install-ADDSDomainController	1.0.0.0	ADDSDeployment
Cmdlet	Install-ADDSDomainForest	1.0.0.0	ADDSDeployment
Cmdlet	Test-ADDSDomainControllerInstallation	1.0.0.0	ADDSDeployment
Cmdlet	Test-ADDSDomainControllerUninstallation	1.0.0.0	ADDSDeployment
Cmdlet	Test-ADDSDomainInstallation	1.0.0.0	ADDSDeployment
Cmdlet	Test-ADDSDomainForestInstallation	1.0.0.0	ADDSDeployment
Cmdlet	Test-ADDSDomainControllerAccountCreation	1.0.0.0	ADDSDeployment
Cmdlet	Uninstall-ADDSDomainController	1.0.0.0	ADDSDeployment

This command:

***Get-Command -Module ADDSDeployment***

Which is listed by Microsoft, gets the available cmdlets in the ADDSDeployment module, which will be handy if I need them. Which they are, now that I need to setup a domain.

```
CommandType      Name                                     Version          Source
-----
Cmdlet            Add-ADDSDomainControllerAccount         1.0.0.0          ADDSDeployment
Cmdlet            Install-ADDSDomain                      1.0.0.0          ADDSDeployment
Cmdlet            Install-ADDSDomainController            1.0.0.0          ADDSDeployment
Cmdlet            Install-ADDSDomainForest                1.0.0.0          ADDSDeployment
Cmdlet            Test-ADDSDomainControllerInstallation    1.0.0.0          ADDSDeployment
Cmdlet            Test-ADDSDomainControllerUninstallation 1.0.0.0          ADDSDeployment
Cmdlet            Test-ADDSDomainInstallation             1.0.0.0          ADDSDeployment
Cmdlet            Test-ADDSDomainForestInstallation        1.0.0.0          ADDSDeployment
Cmdlet            Test-ADDSDomainControllerAccountCreation 1.0.0.0          ADDSDeployment
Cmdlet            Uninstall-ADDSDomainController          1.0.0.0          ADDSDeployment
```

```
PS C:\Users\vboxuser> Install-ADDSDomain
```

cmdlet Install-ADDSDomain at command pipeline position 1  
Supply values for the following parameters:  
NewDomainName: farDomain  
ParentDomainName: farDomain  
SafeModeAdministratorPassword: \*\*\*\*\*

In my heart, I truly feel that setting up a domain before setting up a domain controller is wise.

```

PS C:\Users\vboxuser> Install-ADDSDomain

cmdlet Install-ADDSDomain at command pipeline position 1
Supply values for the following parameters:
NewDomainName: farDomain
ParentDomainName: farDomain
SafeModeAdministratorPassword: *****
Confirm SafeModeAdministratorPassword: *****

The target server will be configured as a domain controller and restarted when this operation is complete.
Do you want to continue with this operation?
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is "Y"): _

```

Entirely confused by this statement, it wants to turn my domain into a domain controller.

```

Install-ADDSDomain : Verification of user credential permissions failed. An Active Directory domain controller for the
in "farDomain" could not be contacted.
re that you supplied the correct DNS domain name.
ine:1 char:1
Install-ADDSDomain
+ CategoryInfo          : NotSpecified: (:) [Install-ADDSDomain], TestFailedException
+ FullyQualifiedErrorId : Test.VerifyUserCredentialPermissions.DCPromo.General.25,Microsoft.DirectoryServices.Deplo
yment.PowerShell.Commands.InstallADDSDomainCommand

age
---
Verification of user credential permissions failed. An Active Directory domain controller for the domain "farDomain"...

C:\Users\vboxuser> _

```

Ah yes, big old red text. I really enjoy seeing this whenever I'm learning something, because it means I've hit my first roadblock. Clearly, whatever I did was wrong, or did not make sense, but in this case maybe I didn't do things in the right order. The error message describes that a domain controller for the domain 'farDomain' (which is what I wanted to name my domain), could not be contacted.

ChatGPT asks that I set my IP address on the machine to static first, and describes its reasoning as ADDS needing to rely on DNS to locate domain controllers and other services. Basically, if the IP of the DC changes, other machines and the DC itself wouldn't be able to resolve domain names in the domain, you'd think the DC could at least find itself, but I guess not.

But it makes sense for other services, DCs are the cornerstone of ADDS from what I've seen, they're needed for every directory-wide policy or anything such, and to authenticate users. It's clearly important, so it makes sense it should be easily found.

```
Select (D)HCP, (S)tatic IP (Blank=Cancel): S
Set Static IP
Enter static IP address: 192.168.1.50
Enter subnet mask (Blank = Default 255.255.255.0):
Enter default gateway: 192.168.1.1
Setting NIC to static IP...
```

Guess this means I've set it up correctly, at least I've done everything it's asked of me. I understand the need for a static IP address, and I understand the subnet too. The AI describes the use of the default gateway, which is used for handling traffic outside the local network. It needs it for things like software updates, time sync, and external DNS lookups (which I don't plan to make my server do much of). This is a pretty cool summary of those things:

**IP Address:** "Where am I?"

**Subnet Mask:** "Who is local?"

**Gateway:** "Where do I send traffic outside?"

**DNS:** "Who answers name lookups for the domain?"

Also, since it needs to be able to find itself, I've set the DNS to itself.

```
Select option: 2
DNS Servers
Enter new preferred DNS server (Blank=Cancel): 192.168.1.50
Enter alternate DNS server (Blank = none):
```

**Sconfig** is proving invaluable honestly, it's not really a wizard, but it is quite magical indeed.

Hopefully it works this time.

```
PS C:\Users\vboxuser> install-addsforest -domainname far.domain.com
SafeModeAdministratorPassword: *****
Confirm SafeModeAdministratorPassword: *****

The target server will be configured as a domain controller and restarted when this operation is complete.
Do you want to continue with this operation?
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is "Y"): y
```

It worked, I now have my very own forest, and domain.

```
PS C:\Users\vboxuser> Get-ADDomain

AllowedDNSSuffixes      : {}
ChildDomains            : {}
ComputersContainer      : CN=Computers,DC=far,DC=domain,DC=com
DeletedObjectsContainer : CN=Deleted Objects,DC=far,DC=domain,DC=com
DistinguishedName       : DC=far,DC=domain,DC=com
DNSRoot                 : far.domain.com
DomainControllersContainer : OU=Domain Controllers,DC=far,DC=domain,DC=com
DomainMode              : Windows2016Domain
DomainSID               : S-1-5-21-2005061878-36205755-715187568
ForeignSecurityPrincipalsContainer : CN=ForeignSecurityPrincipals,DC=far,DC=domain,DC=com
Forest                  : far.domain.com
InfrastructureMaster     : Domain-Controller-Server.far.domain.com
LastLogonReplicationInterval : 
LinkedGroupPolicyObjects : {CN={31B2F340-016D-11D2-945F-00C04FB984F9},CN=Policies,CN=System,DC=far,DC=domain,DC=com}
LostAndFoundContainer    : CN=LostAndFound,DC=far,DC=domain,DC=com
ManagedBy               : 
Name                     : far
NetBIOSName              : FAR
ObjectClass              : domainDNS
ObjectGUID               : 093268fa-4b76-4b08-83a7-e0d4b7c72ec8
ParentDomain             : 
PDCEmulator              : Domain-Controller-Server.far.domain.com
PublicKeyRequiredPasswordRolling : True
QuotasContainer          : CN=NTDS Quotas,DC=far,DC=domain,DC=com
ReadOnlyReplicaDirectoryServers : {}
```

```
PS C:\Users\vboxuser> Get-AdForest

ApplicationPartitions : {DC=ForestDnsZones,DC=far,DC=domain,DC=com, DC=DomainDnsZones,DC=far,DC=domain,DC=com}
CrossForestReferences : {}
DomainNamingMaster    : Domain-Controller-Server.far.domain.com
Domains               : {far.domain.com}
ForestMode            : Windows2016Forest
GlobalCatalogs        : {Domain-Controller-Server.far.domain.com}
Name                  : far.domain.com
PartitionsContainer    : CN=Partitions,CN=Configuration,DC=far,DC=domain,DC=com
RootDomain            : far.domain.com
SchemaMaster          : Domain-Controller-Server.far.domain.com
Sites                 : {Default-First-Site-Name}
SPNSuffixes           : {}
UPNSuffixes           : {}
```

I found the syntax for creating users using the New-ADer cmdlet.

```

PS C:\Users\vboxuser> Get-Help New-AdUser

NAME
    New-AdUser

SYNTAX
    New-AdUser [-Name] <string> [-WhatIf] [-Confirm] [-AccountExpirationDate <datetime>] [-AccountNotDelegated <bool>]
    [-AccountPassword <securestring>] [-AllowReversiblePasswordEncryption <bool>] [-AuthenticationPolicy
    <ADAuthenticationPolicy>] [-AuthenticationPolicySilo <ADAuthenticationPolicySilo>] [-AuthType {Negotiate | Basic}]
    [-CannotChangePassword <bool>] [-Certificates <X509Certificate[]>] [-ChangePasswordAtLogon <bool>] [-City
    <string>] [-Company <string>] [-CompoundIdentitySupported <bool>] [-Country <string>] [-Credential <pscredential>]
    [-Department <string>] [-Description <string>] [-DisplayName <string>] [-Division <string>] [-EmailAddress
    <string>] [-EmployeeID <string>] [-EmployeeNumber <string>] [-Enabled <bool>] [-Fax <string>] [-GivenName
    <string>] [-HomeDirectory <string>] [-HomeDrive <string>] [-HomePage <string>] [-HomePhone <string>] [-Initials
    <string>] [-Instance <ADUser>] [-KerberosEncryptionType {None | DES | RC4 | AES128 | AES256}] [-LogonWorkstations
    <string>] [-Manager <ADUser>] [-MobilePhone <string>] [-Office <string>] [-OfficePhone <string>] [-Organization
    <string>] [-OtherAttributes <hashtable>] [-OtherName <string>] [-PassThru] [-PasswordNeverExpires <bool>]
    [-PasswordNotRequired <bool>] [-Path <string>] [-POBox <string>] [-PostalCode <string>]
    [-PrincipalsAllowedToDelegateToAccount <ADPrincipal[]>] [-ProfilePath <string>] [-SamAccountName <string>]
    [-ScriptPath <string>] [-Server <string>] [-ServicePrincipalNames <string[]>] [-SmartcardLogonRequired <bool>]
    [-State <string>] [-StreetAddress <string>] [-Surname <string>] [-Title <string>] [-TrustedForDelegation <bool>]
    [-Type <string>] [-UserPrincipalName <string>] [<CommonParameters>]

```

This is an extremely important command, it basically informs me on everything I can do within AD, to my knowledge.

```

PS C:\Users\vboxuser> Get-Command -Module ActiveDirectory

CommandType      Name                                     Version      Source
-----
Cmdlet           Add-ADCentralAccessPolicyMember        1.0.1.0      ActiveDirectory
Cmdlet           Add-ADComputerServiceAccount           1.0.1.0      ActiveDirectory
Cmdlet           Add-ADDomainControllerPasswordReplicationPolicy 1.0.1.0      ActiveDirectory
Cmdlet           Add-ADFineGrainedPasswordPolicySubject 1.0.1.0      ActiveDirectory
Cmdlet           Add-ADGroupMember                      1.0.1.0      ActiveDirectory
Cmdlet           Add-ADPrincipalGroupMembership          1.0.1.0      ActiveDirectory
Cmdlet           Add-ADResourcePropertyListMember        1.0.1.0      ActiveDirectory
Cmdlet           Clear-ADAccountExpiration               1.0.1.0      ActiveDirectory
Cmdlet           Clear-ADClaimTransformLink              1.0.1.0      ActiveDirectory
Cmdlet           Disable-ADAccount                      1.0.1.0      ActiveDirectory
Cmdlet           Disable-ADOptionalFeature               1.0.1.0      ActiveDirectory

```

It is quite crucial that I am able to view all my users, and I can do this with the following:



```
Administrator: C:\Windows\system32\cmd.exe - powershell - Powershell
PS C:\Users\vboxuser> Get-ADUser -Filter *
```

DistinguishedName : CN=Administrator,CN=Users,DC=far,DC=domain,DC=com  
Enabled : True  
GivenName :  
Name : Administrator  
ObjectClass : user  
ObjectGUID : 0486f57a-c886-4569-9a7e-1c837730f86a  
SamAccountName : Administrator  
SID : S-1-5-21-2005061878-36205755-715187568-500  
Surname :  
UserPrincipalName :

## Edition 2

Today, I have started tinkering with the Active Directory users, and I've learned how to set userPrincipalNames and other ADUser properties using the Set-ADUser command.

```
Administrator: C:\Windows\system32\cmd.exe - powershell
```

Surname :  
UserPrincipalName :  
DistinguishedName : CN=krbtgt,CN=Users,DC=far,DC=domain,DC=com  
Enabled : False  
GivenName :  
Name : krbtgt  
ObjectClass : user  
ObjectGUID : ab341a87-293d-4945-ad70-8914091cb1da  
SamAccountName : krbtgt  
SID : S-1-5-21-2005061878-36205755-715187568-502  
Surname :  
UserPrincipalName :  
DistinguishedName : CN=JohnDoe,CN=Users,DC=far,DC=domain,DC=com  
Enabled : False  
GivenName :  
Name : JohnDoe  
ObjectClass : user  
ObjectGUID : d2744e88-cf95-441d-a74d-ee5af2341440  
SamAccountName : JohnDoe  
SID : S-1-5-21-2005061878-36205755-715187568-1104  
Surname :  
UserPrincipalName :

```
PS C:\Users\vboxuser> Set-ADUser -Identity JohnDoe -UserPrincipalName jdoe@far.domain.cmo
PS C:\Users\vboxuser> Set-ADUser -Identity JohnDoe -UserPrincipalName jdoe@far.domain.com
PS C:\Users\vboxuser>
```

We can check if the change was successful by doing the following:

```
PS C:\Users\vboxuser> Get-ADUser -Filter *
```

Since we only have a few users, we can get away with doing this for now, but as the userbase expands, we will need to start using filters to locate particular users.

```
DistinguishedName : CN=JohnDoe,CN=Users,DC=far,DC=domain,DC=com
Enabled           : False
GivenName        :
Name             : JohnDoe
ObjectClass       : user
ObjectGUID        : d2744e88-cf95-441d-a74d-ee5af2341440
SamAccountName    : JohnDoe
SID              : S-1-5-21-2005061878-36205755-715187568-1104
Surname          :
UserPrincipalName : jdoe@far.domain.com
```

The change went through, and the userprincipalname has now been set. It is important to note that this user account is still disabled, because it doesn't have a password set, so as a system administrator I need to set this account password and enable it. I will attempt to do this using the following command:

```
PS C:\Users\vboxuser> Set-ADAccountPassword -Identity "JohnDoe" -Reset -NewPassword (ConvertTo-SecureString Password!#@1012" -AsPlainText -Force)
PS C:\Users\vboxuser>
```

Without the 'reset' parameter, PS assumes that we want to change the password, rather than set the password. Apparently, this is what admins normally do.

AD cmdlets enforce security, so passwords cannot be passed as plain text, in this case it is wrapped in ConvertTo-SecureString, and we force it to go through, since AD will warn us that using plain text is not secure. In a circumstance where we have an established key vault, Windows Credential Manager, SecretManagement modules, etc. For this test, this is okay for our fictional user.

There is another, more secure way to do it, where it doesn't show up in powershell history. It apparently doesn't even show up in powershell, which I'm going to test now. We can actually store things in memory securely, to use them in the powershell command later. In Java or Python, this would basically be using variables to accomplish this task. Not sure what it's called in PS.

```
PS C:\Users\vboxuser> $User = Read-Host "Enter the username"
Enter the username: JohnDoe
PS C:\Users\vboxuser> $Password = Read-Host "Enter new password" -AsSecureString
Enter new password: *****
```

It looks like using 'variables', or whatever they're called in PS, has actually worked, not only was I able to type in the password securely, but I was also able to input those into the Set-ADAccountPassword fields.

```
PS C:\Users\vboxuser> Set-ADAccountPassword -Identity $User -Reset -NewPassword $Password
PS C:\Users\vboxuser> _
```

Now, we can go ahead and enable the ADAccount.

```
PS C:\Users\vboxuser> Set-ADAccountPassword -Identity $User -Reset -NewPassword $Password
PS C:\Users\vboxuser> Enable-ADAccount -Identity $User
PS C:\Users\vboxuser> Get-ADUser -Filter *
```

```
DistinguishedName : CN=JohnDoe,CN=Users,DC=far,DC=domain,DC=com
Enabled           : True
GivenName        :
Name             : JohnDoe
ObjectClass       : user
ObjectGUID        : d2744e88-cf95-441d-a74d-ee5af2341440
SamAccountName    : JohnDoe
SID              : S-1-5-21-2005061878-36205755-715187568-1104
Surname          :
UserPrincipalName : jdoe@far.domain.com
```

The next thing to do is create groups, using groups we can administer users as a group, rather than individually. This is simple to do, thanks to Microsoft's documentation.

```
PS C:\users\vboxuser> New-ADGroup -Name "DCAdmin" -SamAccountName DCAdmin -GroupCategory Security

cmdlet New-ADGroup at command pipeline position 1
Supply values for the following parameters:
GroupScope: Global
PS C:\users\vboxuser> Set-ADGroup -Identity DCAdmin -description DC Administrators belong to this
group.
Set-ADGroup : A positional parameter cannot be found that accepts argument 'Administrators'.
At line:1 char:1
+ Set-ADGroup -Identity DCAdmin -description DC Administrators belong t ...
+ ~~~~~
    + CategoryInfo          : InvalidArgument: (:) [Set-ADGroup], ParameterBindingException
    + FullyQualifiedErrorId : PositionalParameterNotFound,Microsoft.ActiveDirectory.Management.
    Commands.SetADGroup

PS C:\users\vboxuser> Set-ADGroup -Identity DCAdmin -description "DC Administrators belong to thi
s group."
PS C:\users\vboxuser>
```

<https://learn.microsoft.com/en-us/powershell/module/activedirectory/new-adgroup?view=windowsserver2025-ps>

At this stage, we have a domain, domain controller, users, and a group. I have also gone ahead and replicated this and made multiple groups.

```

DistinguishedName : CN=DCAdmin,CN=Users,DC=far,DC=domain,DC=com
GroupCategory      : Security
GroupScope         : Global
Name               : DCAdmin
ObjectClass        : group
ObjectGUID         : 400a9b9c-e1b7-424d-9556-9feac3479c0d
SamAccountName     : DCAdmin
SID                : S-1-5-21-2005061878-36205755-715187568-1106

DistinguishedName : CN=HR_Users,CN=Users,DC=far,DC=domain,DC=com
GroupCategory      : Security
GroupScope         : Global
Name               : HR_Users
ObjectClass        : group
ObjectGUID         : 2a240d99-ed46-409d-a777-2de45fa4e1a6
SamAccountName     : HRUser
SID                : S-1-5-21-2005061878-36205755-715187568-1107

DistinguishedName : CN=Finance_Admns,CN=Users,DC=far,DC=domain,DC=com
GroupCategory      : Security
GroupScope         : Global
Name               : Finance_Admns
ObjectClass        : group
ObjectGUID         : 427f4859-2ad6-4b43-afe1-ed1520d3c1ae
SamAccountName     : FinanceAdmin
SID                : S-1-5-21-2005061878-36205755-715187568-1108

```

At this stage, we should add users to groups, which will make it easier to administer them together rather than must administer them individually. I've gone ahead and added a member to the Finance Admin group created.

```

PS C:\users\vboxuser> Add-ADGroupMember -Identity Finance_Admns -Members janeDoe
Add-ADGroupMember : Cannot find an object with identity: 'Finance_Admns' under:
'DC=far,DC=domain,DC=com'.
At line:1 char:1
+ Add-ADGroupMember -Identity Finance_Admns -Members janeDoe
+ ~~~~~
+ CategoryInfo          : ObjectNotFound: (Finance_Admns:ADGroup) [Add-ADGroupMember], ADI
  identityNotFoundException
+ FullyQualifiedErrorId : ActiveDirectoryCmdlet:Microsoft.ActiveDirectory.Management.ADIde
  ntityNotFoundException,Microsoft.ActiveDirectory.Management.Commands.AddADGroupMember

PS C:\users\vboxuser> Add-ADGroupMember -Identity FinanceAdmin -Members janeDoe
PS C:\users\vboxuser> Get-ADGroupMember -Identity FinanceAdmin

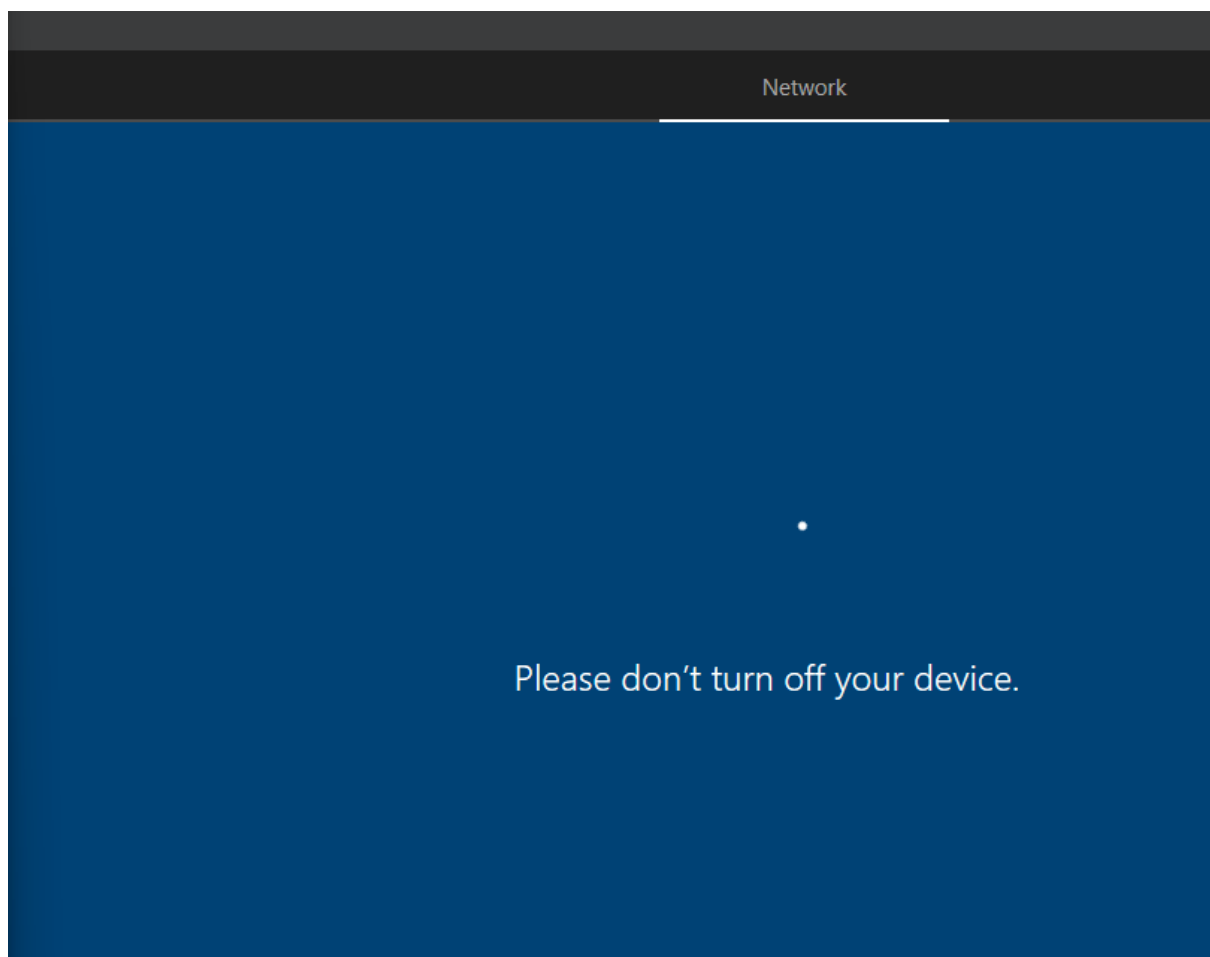
distinguishedName : CN=Jane Doe,CN=Users,DC=far,DC=domain,DC=com
name              : Jane Doe
objectClass       : user
objectGUID        : 91499cec-2251-4977-849a-23d074feb26d
SamAccountName    : janeDoe
SID               : S-1-5-21-2005061878-36205755-715187568-1109

```

I realise now, through experimenting, that the SamAccountName is basically an object identifier, rather than the actual name, when dealing with putting users into groups and stuff like that. Something like the barcode of a product, as opposed to the actual name of the product.

## Edition 3

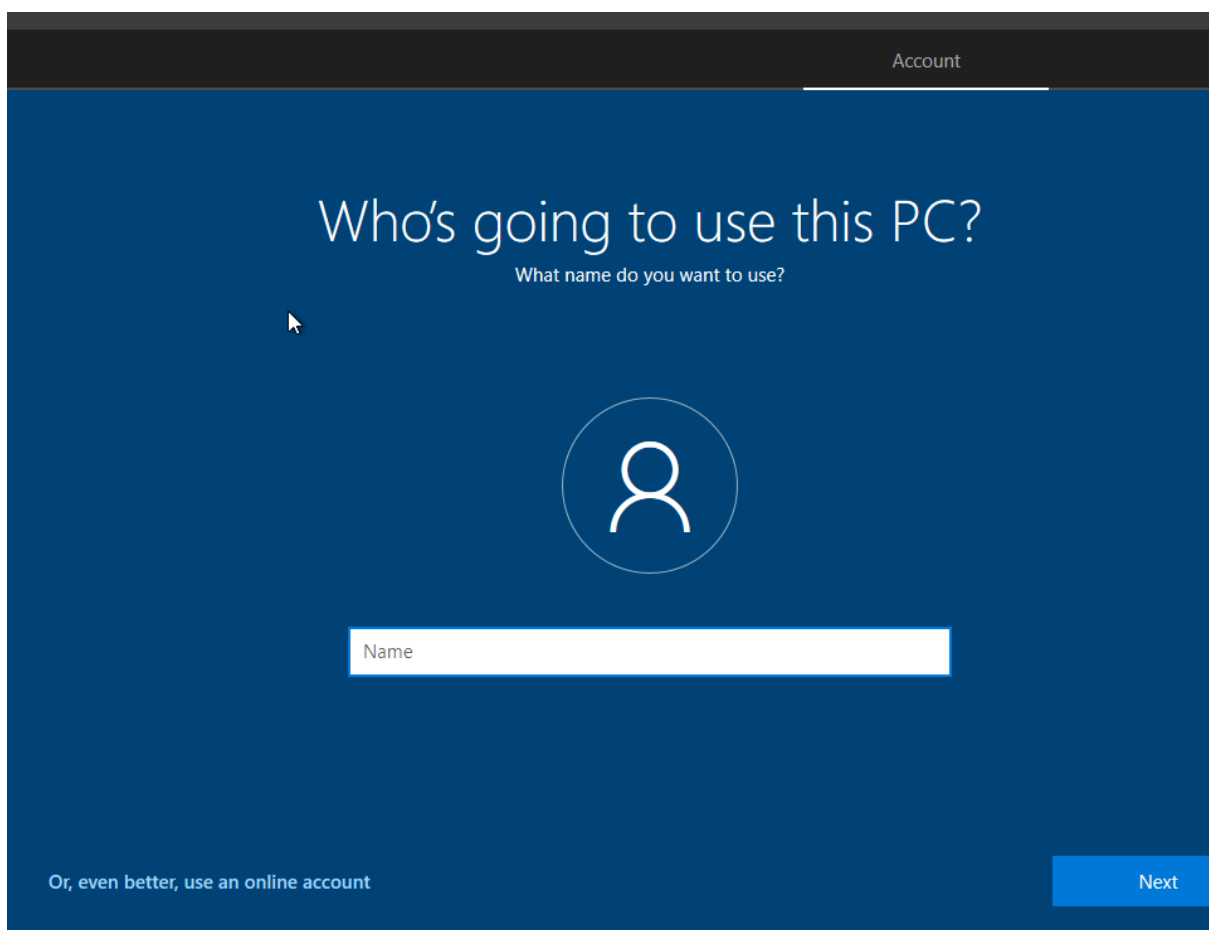
Now that we have the basic components of an active directory: domain, domain controller, groups, and users. We can start making it useful. The next step is to add some devices, like computers (which I can spin up virtually in VirtualBox), and connect them to the AD. Using the GPO's and AD, I (the administrator), can apply security policies, and other device policies that control the virtualized machines. For example, a standardized background screen, or app and access control.



Here, we are setting up a new computer to later add to the AD. In the meantime, we will make a new organizational unit called “UserComputers” to prepare for the new computer.

```
C:\Users\ vboxuser> New-ADOrganizationalUnit -Name "UserComputers"  
C:\Users\ vboxuser> Get-ADOrganizationalUnit -filter *
```

On the new windows 10 installation, we need to set it up, and wire its networking so it is able to connect to our AD.



To do this, we need to ensure they are on the same network, and in a virtual environment, they need to be on the same adapter. We can see the current, fresh networking configuration of the machine below.

```

C:\Users\finSyd-Comp1>ipconfig /all

Windows IP Configuration

Host Name . . . . . : DESKTOP-RLDSLRG
Primary Dns Suffix . . . . . :
Node Type . . . . . : Hybrid
IP Routing Enabled. . . . . : No
WINS Proxy Enabled. . . . . : No

Ethernet adapter Ethernet:

Connection-specific DNS Suffix . :
Description . . . . . : Intel(R) PRO/1000 MT Desktop Adapter
Physical Address. . . . . : 08-00-27-70-96-1E
DHCP Enabled. . . . . : Yes
Autoconfiguration Enabled . . . . : Yes
Link-local IPv6 Address . . . . . : fe80::1404:d5ec:ac9d:16be%4(Preferred)
Autoconfiguration IPv4 Address. . : 169.254.0.138(Preferred)
Subnet Mask . . . . . : 255.255.0.0
Default Gateway . . . . . :
DHCPv6 IAID . . . . . : 101187623
DHCPv6 Client DUID. . . . . : 00-01-00-01-30-70-7E-D9-08-00-27-70-96-1E
DNS Servers . . . . . : fec0:0:0:ffff::1%1
                       fec0:0:0:ffff::2%1
                       fec0:0:0:ffff::3%1
NetBIOS over Tcpip. . . . . : Enabled

```

Now, we need to set the DNS to the DC. The following netsh commands will help us accomplish this.

```

C:\Users\finSyd-Comp1>netsh
netsh>interface show interface

Admin State   State        Type          Interface Name
-----
Enabled       Connected    Dedicated     Ethernet

netsh>interface ip set dns name="Ethernet" source="static" address="192.168.1.50"

```

This is the first network error we've encountered.



```
netsh>interface ip set dns name="Ethernet" source="static" address=192.168.1.50"
The parameter is incorrect.

netsh>interface ip set dns name="Ethernet" source="static" address="192.168.1.50"
The configured DNS server is incorrect or does not exist.

netsh>interface ip set dns name ="Ethernet" source="static" address="192.168.1.50"
The configured DNS server is incorrect or does not exist.
```

We need to figure out why our DNS server / DC is not being detected. One main problem is since we have set the computer to internal network, and the DC was on NAT, they cannot see each other. I have set both of them to internal network.

Still no luck, I have even set the IP of the w10 machine to a static IP. Perhaps it is the firewall on the DC. I have disabled the firewalls of both machines, but still no luck. I think it is the virtualbox network configuration. I will change the promiscuous mode setting on both VMs, which enables them to see each other's network traffic.

Promiscuous Mode: Allow VMs

I've decided to try and use the other NIC on the DC as the DNS server, this one has an APIPA range.

```
NIC Index          2
Description        Intel(R) PRO/1000 MT Desktop Adapter #2
IP Address          169.254.51.43    fe80::f71e:a061:7589:5435
Subnet Mask         255.255.0.0
DHCP enabled       True
Default Gateway
Preferred DNS Server 169.254.51.43
Alternate DNS Server
```

It worked, I've set the NIC IP that's on the internal network to a static IP, and the DC and Machine can now see each other.

```

PS C:\Windows\system32> netsh interface show interface

Admin State      State           Type           Interface Name
-----
Enabled          Connected      Dedicated      Ethernet

PS C:\Windows\system32> netsh interface ip set dns name ="Ethernet" static 192.168.52.52

PS C:\Windows\system32> nslookup far.domain.com
DNS request timed out.
    timeout was 2 seconds.
Server:  UnKnown
Address:  192.168.52.52

Name:     far.domain.com
Addresses: 192.168.52.52
           192.168.1.50

PS C:\Windows\system32>

```

DC [Running] - Oracle VirtualBox

File Machine View Input Devices Help

Select Administrator: C:\Windows\system32\cmd.exe - powershell

Select Network Adapter Index# (Blank=Cancel): 1

-----  
Network Adapter Settings  
-----

NIC Index	1
Description	Intel(R) PRO/1000 MT Desktop Adapter
IP Address	192.168.1.50 fe80::7af9:6c22:357a:4b19
Subnet Mask	255.255.255.0
DHCP enabled	False
Default Gateway	192.167.1.1
Preferred DNS Server	192.168.1.50
Alternate DNS Server	

- 1) Set Network Adapter Address
- 2) Set DNS Servers
- 3) Clear DNS Server Settings
- 4) Return to Main Menu

Select option:

Successfully added the first computer to the AD.

```
Select Administrator: C:\Windows\system32\cmd.exe - powershell

cmdlet Get-ADComputer at command pipeline position 1
Supply values for the following parameters:
(Type !? for Help.)
Filter: *

DistinguishedName : CN=DOMAIN-CONTROLL,OU=Domain Controllers,DC=far,DC=domain,DC=com
DNSHostName       : Domain-Controller-Server.far.domain.com
Enabled           : True
Name              : DOMAIN-CONTROLL
ObjectClass       : computer
ObjectGUID        : 4f9eb0bf-9f38-46f5-8b40-90b23b345bac
SamAccountName    : DOMAIN-CONTROLL$
SID               : S-1-5-21-2005061878-36205755-715187568-1001
UserPrincipalName :

DistinguishedName : CN=DESKTOP-FINSYD2,CN=Computers,DC=far,DC=domain,DC=com
DNSHostName       : DESKTOP-FINSYD2.far.domain.com
Enabled           : True
Name              : DESKTOP-FINSYD2
ObjectClass       : computer
ObjectGUID        : ea41a3c0-01e1-49bc-814e-6aa3ead8bb07
SamAccountName    : DESKTOP-FINSYD2$
SID               : S-1-5-21-2005061878-36205755-715187568-1110
UserPrincipalName :

PS C:\Users\vboxuser>
```

I added this computer to the AD from the W10 Wizard, now I will do the same but using the PS on Windows Server Core.

```
PS C:\Users\vboxuser> Get-ADComputer -Identity "DESKTOP-FINSYD2"

DistinguishedName : CN=DESKTOP-FINSYD2,OU=UserComputers,DC=far,DC=domain,DC=com
DNSHostName       :
Enabled           : True
Name              : DESKTOP-FINSYD2
ObjectClass       : computer
ObjectGUID        : fc10e8df-fa0c-474d-bd81-041d97f1f485
SamAccountName    : DESKTOP-FINSYD2$
SID               : S-1-5-21-2005061878-36205755-715187568-1112
UserPrincipalName :
```

Now that this object has been added, I need to connect the computer to the AD.



← Join a Domain or Workgroup

Type your user name, password, and domain name for your domain account

User name:	<input type="text" value="administrator"/>
Password:	<input type="password" value="••••••••••••••"/>
Domain name:	<input type="text" value="SAR.DOMAIN.COM"/>

Next

Cancel

It should find that a computer object in the AD already exists. Which it has done so.

---

#### User Account and Domain Information

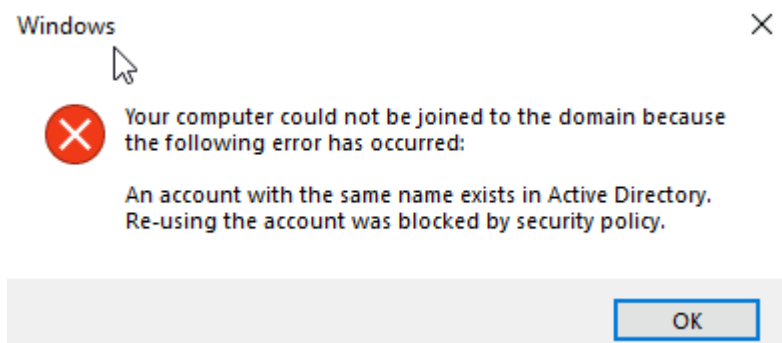


An account for this computer ("DESKTOP-FINSYD2") has been found in the domain "FAR".

Would you like to use this?

Yes

No



The computer is not using the existing computer object unfortunately, so I'm going to remove the existing computer object, and let the machine create a new one on its own.

```
PS C:\Users\vboxuser> Remove-ADComputer -Identity "DESKTOP-FINSYD2"
Confirm
Are you sure you want to perform this action?
Performing the operation "Remove" on target "CN=DESKTOP-FINSYD2,OU=UserComputers,DC=far,DC=domain,DC=com".
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is "Y"): y
PS C:\Users\vboxuser>
```

Join a Domain or Workgroup (Not Responding)

Join a Domain or Workgroup

Type the computer name and computer domain name

Windows cannot find an account for your computer in the DESKTOP-FINSYD2 domain.  
(Your computer domain can be different from your user account domain.)

Computer name:

DESKTOP-FINSYD2

Computer domain:

FAR.DOMAIN.COM

Next

Cancel

We are now able to login with AD users on the machine.

<https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/deploy/install-active-directory-domain-services--level-100-#install-ad-ds-by-using-windows-powershell>