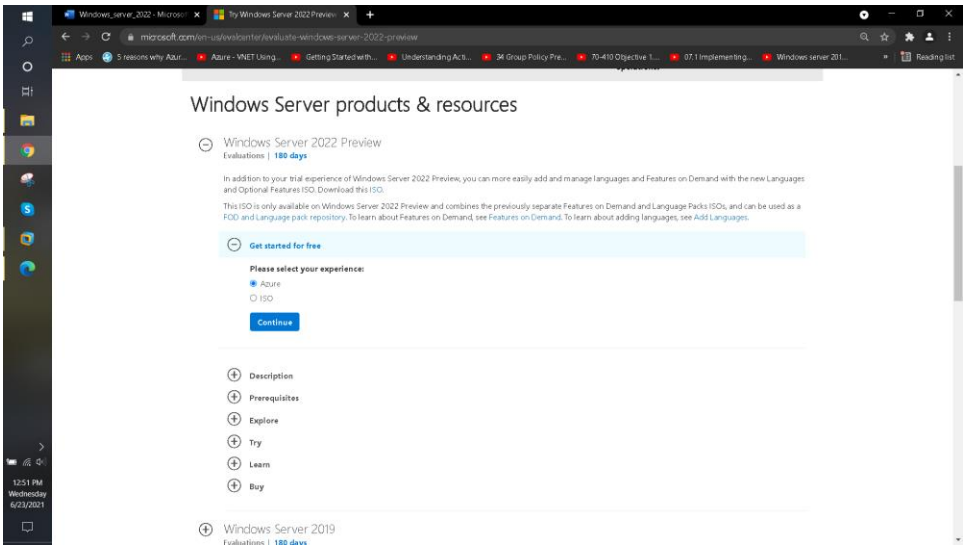
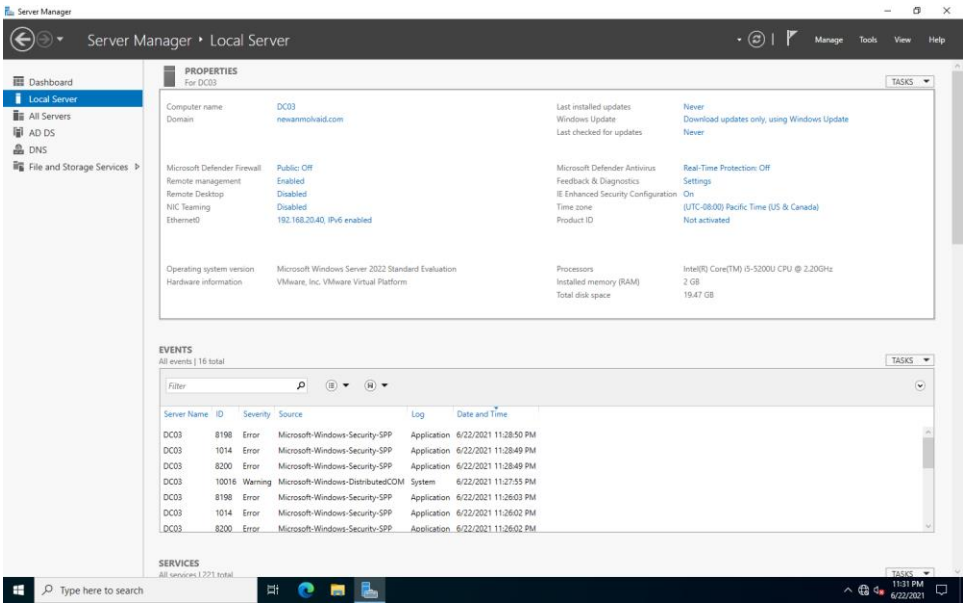
From below screen shot it can be seen that we are on 2022 server which is in preview version till 23/June/2021. However, testing our existing PowerShell commands and promotion of this server as a domain controller can give us fact check on proof of concept (POC). Latest windows server not only help to keep our AD evergreen but also provide new features that add more value and are cloud ready.



Below screen shot help us to see that windows server 2022 can be successfully be promoted as a DC.

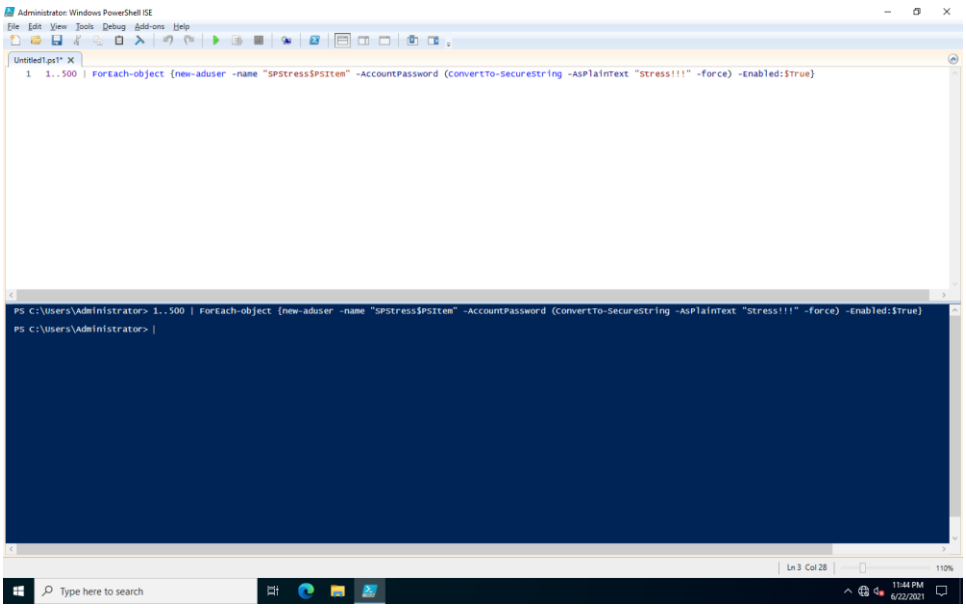
Now we will test how user creation and existing PowerShell command will work on it.



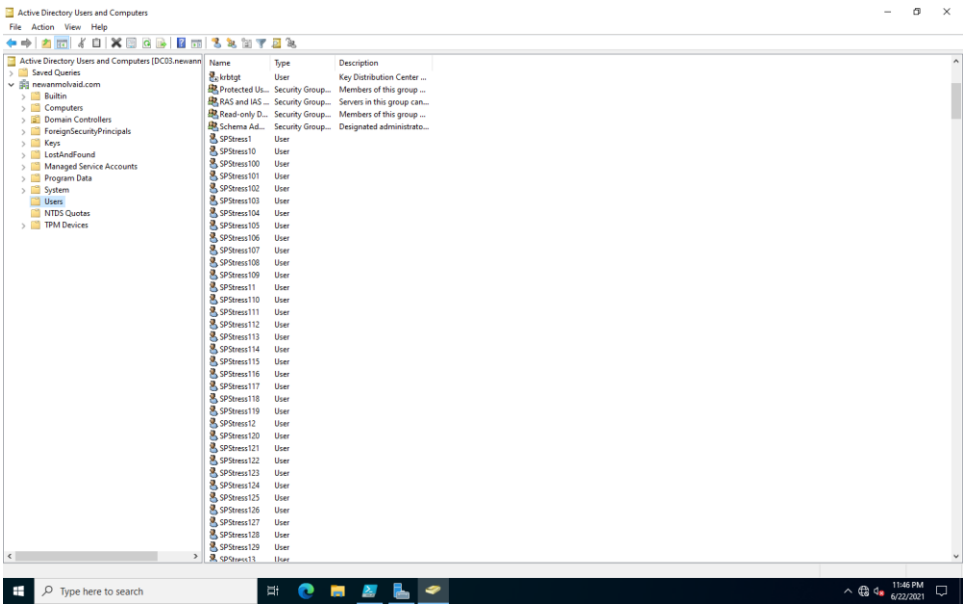
Creating 500 test users with their account enabled.

Command used:

1..500 | ForEach-object {new-aduser -name "SPStress$PSItem" -AccountPassword (ConvertTo-SecureString -AsPlainText "Stress!!!" -force) -Enabled:$True}

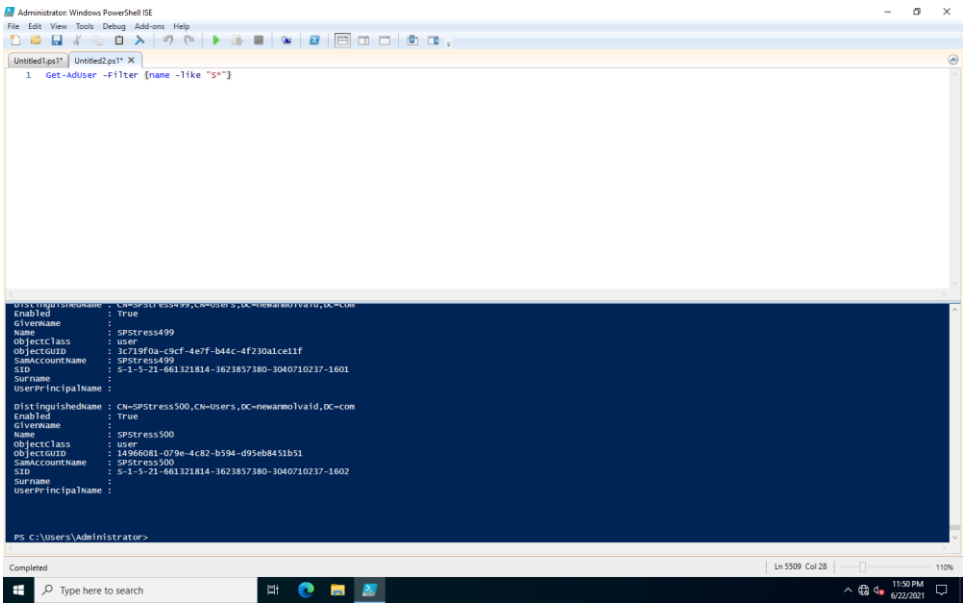


By default, these users are created in container name users. To see them navigate to users container in ADUC as shown below.



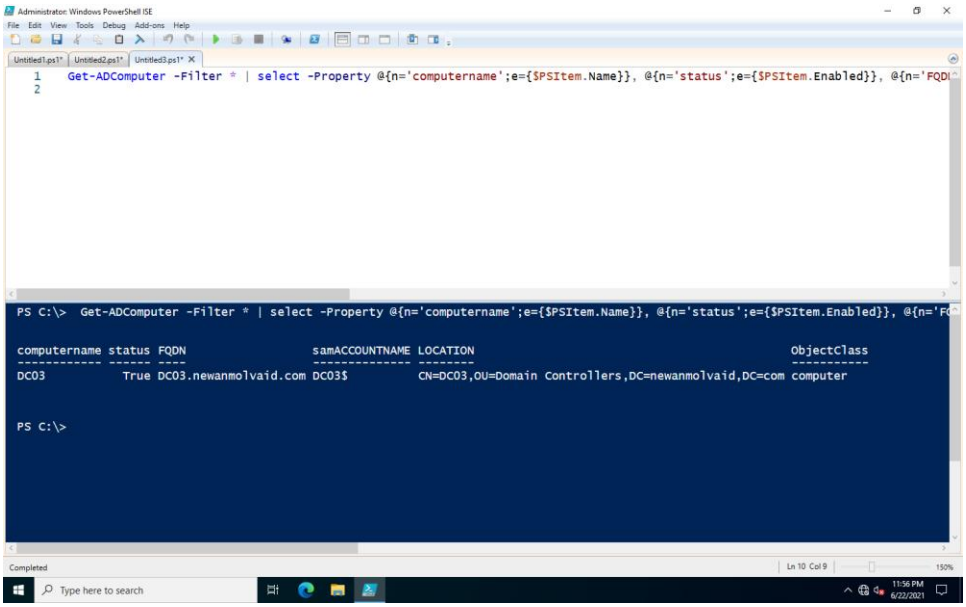
Or you can use below PowerShell command.

Get-AdUser -Filter {name -like "S*"}



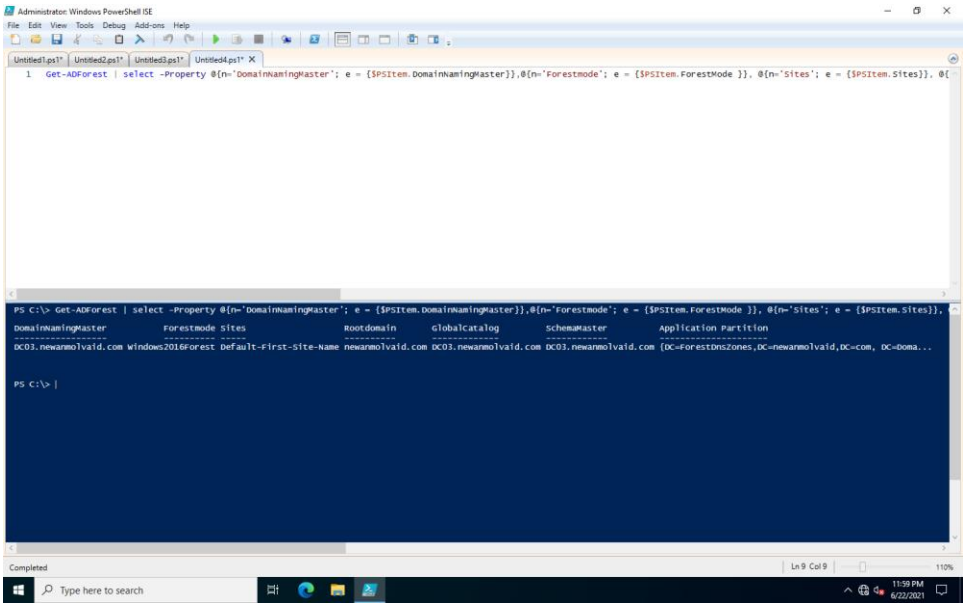
Below command is used to get computer details from AD domain.

Get-ADComputer -Filter * | select -Property @{n='computername';e={$PSItem.Name}}, @{n='status';e={$PSItem.Enabled}}, @{n='FQDN';e={$PSItem.DNSHostName}}, @{n='samACCOUNTNAME';e={$PSItem.SamAccountName}}, @{n='LOCATION';e={$PSItem.DistinguishedName}}, @{n='ObjectClass';e={$PSItem.ObjectClass}} | Format-Table -AutoSize



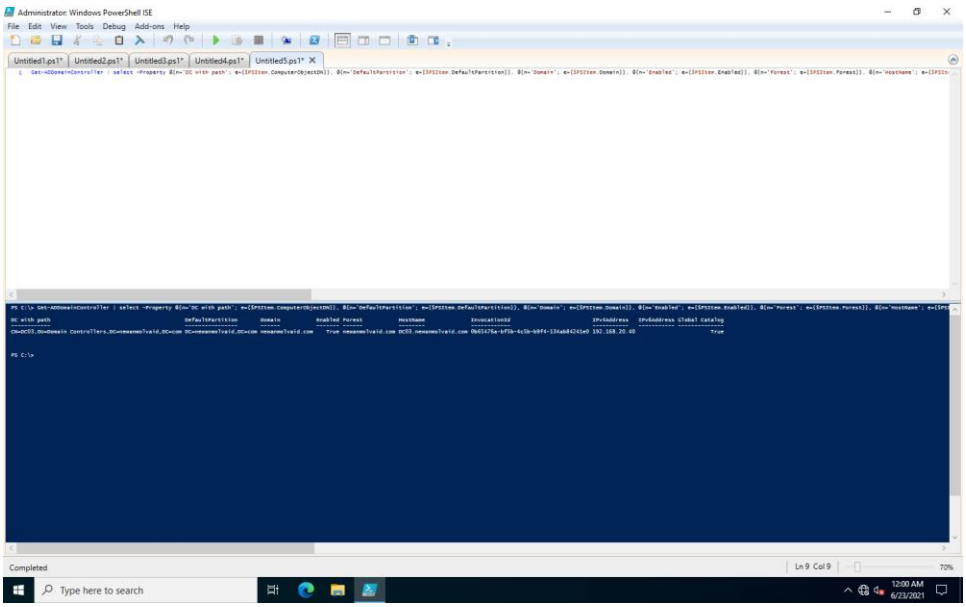
Getting forest details in a tabular format.

```
Get-ADForest | select -Property @{n='DomainNamingMaster'; e = {$PSItem.DomainNamingMaster}},@{n='Forestmode'; e = {$PSItem.ForestMode }}, @{n='Sites'; e = {$PSItem.Sites}}, @{n='Rootdomain'; e = {$PSItem.RootDomain}}, @{n='GlobalCatalog'; e = {$PSItem.GlobalCatalogs}},@{n='SchemaMaster'; e = {$PSItem.SchemaMaster}}, @{n='Application Partition'; e = {$PSItem.ApplicationPartitions}} ,@{n='partitionContainer'; e = {$PSItem.PartitionsContainer}} | Format-Table -AutoSize
```



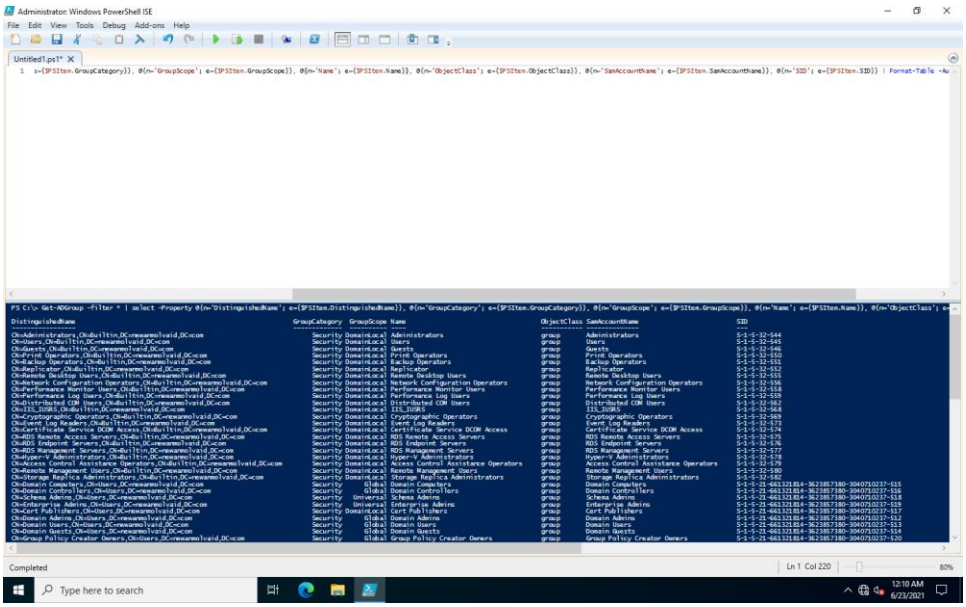
Below command is used to get Dc details in domain.

```
Get-ADDomainController | select -Property @{n='DC with path'; e={$PSItem.ComputerObjectDN}}, @{n='DefaultPartition'; e={$PSItem.DefaultPartition}}, @{n='Domain'; e={$PSItem.Domain}}, @{n='Enabled'; e={$PSItem.Enabled}}, @{n='Forest'; e={$PSItem.Forest}}, @{n='HostName'; e={$PSItem.HostName}}, @{n='InvocationId'; e={$PSItem.InvocationId}}, @{n='IPv4Address'; e={$PSItem.IPv4Address}}, @{n='IPv6Address'; e={$PSItem.IPv6Address}}, @{n='Global Catalog'; e={$PSItem.IsGlobalCatalog}}, @{n='Read Only DC'; e={$PSItem.IsReadOnly}}, @{n='LdapPort'; e={$PSItem.LdapPort}}, @{n='Name'; e={$PSItem.Name}}, @{n='NTDSSettingsObjectDN'; e={$PSItem.NTDSSettingsObjectDN}}, @{n='Operating System'; e={$PSItem.OperatingSystem}}, @{n='OperationMasterRoles'; e={$PSItem.OperationMasterRoles}}, @{n='Partitions'; e={$PSItem.Partitions}} | Format-Table -AutoSize
```



Below PowerShell command is used to get group details in domain:

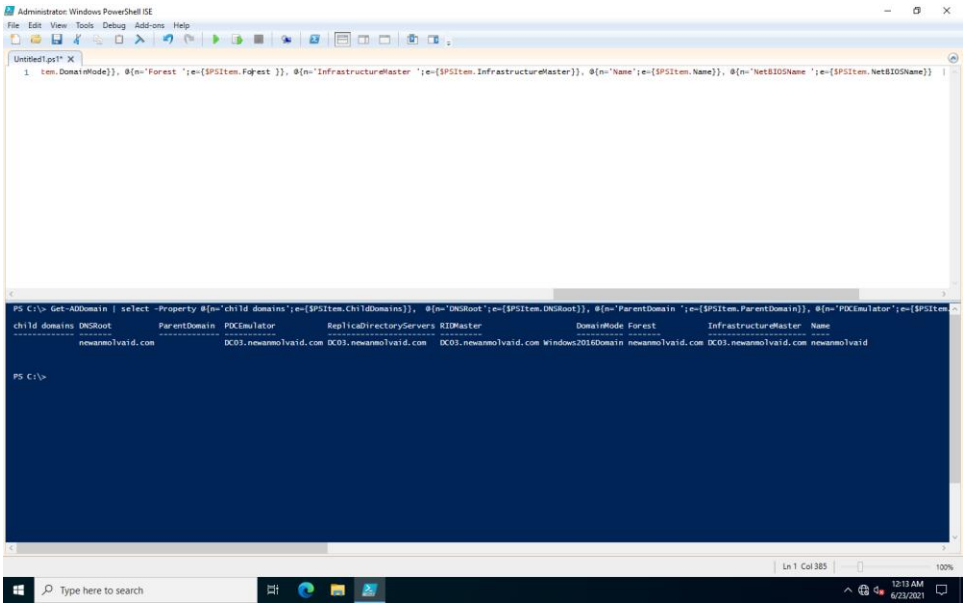
```
Get-ADGroup -filter * | select -Property @{n='DistinguishedName'; e={$PSItem.DistinguishedName}}, @{n='GroupCategory'; e={$PSItem.GroupCategory}}, @{n='GroupScope'; e={$PSItem.GroupScope}}, @{n='Name'; e={$PSItem.Name}}, @{n='ObjectClass'; e={$PSItem.ObjectClass}}, @{n='SamAccountName'; e={$PSItem.SamAccountName}}, @{n='SID'; e={$PSItem.SID}} | Format-Table -AutoSize
```



Below command is used for getting details of the domain:

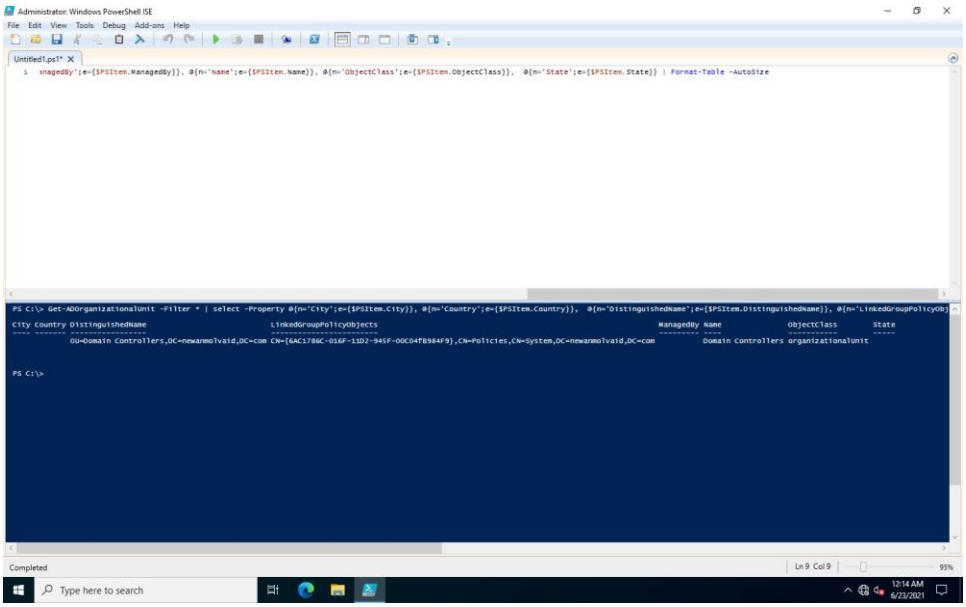
```
Get-ADDomain | select -Property @{n='child domains';e={$PSItem.ChildDomains}}, @{n='DNSRoot';e={$PSItem.DNSRoot}}, @{n='ParentDomain ';e={$PSItem.ParentDomain}}, @{n='PDCEmulator';e={$PSItem.PDCEmulator}}, @{n='ReplicaDirectoryServers';e={$PSItem.ReplicaDirectoryServers}}, @{n='RIDMaster';e={$PSItem.RIDMaster}}, @{n='DomainMode';e={$PSItem.DomainMode}}, @{n='Forest ';e={$PSItem.Forest }}, @{n='InfrastructureMaster
```

';e={$PSItem.InfrastructureMaster}}, @{n='Name';e={$PSItem.Name}}, @{n='NetBIOSName ';e={$PSItem.NetBIOSName}} | Format-Table –AutoSize



Below command is used to get OU structure details in domain:

Get-ADOrganizationalUnit -Filter * | select -Property @{n='City';e={$PSItem.City}}, @{n='Country';e={$PSItem.Country}}, @{n='DistinguishedName';e={$PSItem.DistinguishedName}}, @{n='LinkedGroupPolicyObjects';e={$PSItem.LinkedGroupPolicyObjects}}, @{n='ManagedBy';e={$PSItem.ManagedBy}}, @{n='Name';e={$PSItem.Name}}, @{n='ObjectClass';e={$PSItem.ObjectClass}}, @{n='State';e={$PSItem.State}} | Format-Table -AutoSize



Below command is used for extracting each object from AD:

```
Get-ADObject -Filter * | select -Property @{n='DistinguishedName';e={$psitem.DistinguishedName}},
@{n='Name';e={$psitem.Name}}, @{n='ObjectClass';e={$psitem.ObjectClass}} | Format-Table -AutoSize
```