AVD SETUP DOCUMENTATION

Create an AVD setup from scratch through azure portal using PowerShell script

Azure VM setup requirement:

1.Management group- (tenant)

2.Subscription

3.Resource group

4.Vnet/subnet

5.hostpool (workspace/Appgroup-RAG/DAG)

6.Virtual machine

IMP links to refer-
https://learnthecontent.com/exam/azure/az-140-configuring-and-operating-microsoft-azure-virtual-desktop/s/automate-creation-of-azure-virtual-desktop-hosts-and-host-pools-by-using-powershell-azure-cli-azure-resource-manager-templates-arm-templates-and-bicep

Domain Name - harshsonar2151gmail.onmicrosoft.com
Subscription ID - 0c3bb37f-98c6-42f6-96c7-7bb5d7e34f15
To check power shell version - $PSVersionTable.PSVersion

Create Resource group
 # Variables

$resourceGroupName = "YourResourceGroupName"

$location = "YourRegion" # e.g., "East US", "West Europe"

$hostPoolName = "YourHostPoolName"

$workspaceName = "YourWorkspaceName"

$vmName = "YourVMName"

$vmSize = "Standard_D2s_v3" # Adjust as needed

$adminUsername = "AdminUsername"

$adminPassword = ConvertTo-SecureString "YourSecurePassword" -AsPlainText -Force

$virtualNetworkName = "YourVNetName"

$subnetName = "YourSubnetName"

Create Resource Group (if not already created)

New-AzResourceGroup -harshrg $resourceGroupName -Eastasia $location

e.g-New-AzResourceGroup -Name harshrg -Location "eastasia"

```
PS /home/harsh> New-AzResourceGroup -Name harshrgg -Location "Centralindia"

ResourceGroupName : harshrgg
Location          : centralindia
ProvisioningState : Succeeded
Tags              :
ResourceId        : /subscriptions/0c3bb37f-98c6-42f6-96c7-7bb5d7e34f15/resourceGroups/harshrgg
```

 # Create V net-

$virtualNetwork = New-AzVirtualNetwork -ResourceGroupName harshrg -Location eastasia -Name Harshvnet -AddressPrefix 10.0.0.0/16

$subnetConfig = Add-AzVirtualNetworkSubnetConfig -Name default -AddressPrefix 10.0.0.0/24 -Harshvnet $virtualNetwork

Main Script - New-AzVirtualNetwork -Name Harshvnet -ResourceGroupName harshrg -Location eastasia -AddressPrefix 10.0.0.0/16 -Subnet $subnetConfig

```
PS /home/harsh> New-AzVirtualNetwork -Name Harshvnett -ResourceGroupName harshrgg -Location Centralindia -AddressPrefix 10.0.0.0/16 -Subnet $subnetConfig

ResourceGroupName Name       Location     ProvisioningState EnableDdosProtection
----------------- ----       --------     ----------------- --------------------
harshrgg          Harshvnett centralindia Succeeded         False
```

 # Define subnet prefix

$subnetPrefix = "10.0.0.0/24"

Step 1: Get the existing virtual network object

$virtualNetwork = Get-AzVirtualNetwork -Name "Harshvnet" -ResourceGroupName "harshrg"

Step 2: Add the subnet configuration to the virtual network object

$virtualNetwork = Add-AzVirtualNetworkSubnetConfig -Name "default" -AddressPrefix $subnetPrefix -VirtualNetwork $virtualNetwork

Step 3: Update the virtual network with the new subnet

$virtualNetwork | Set-AzVirtualNetwork

```
PS /home/harsh> $subnetPrefix = "10.0.0.0/24"
PS /home/harsh> $virtualNetwork = Get-AzVirtualNetwork -Name "Harshvnett" -ResourceGroupName "harshrgg"
PS /home/harsh> $virtualNetwork = Add-AzVirtualNetworkSubnetConfig -Name "default" -AddressPrefix $subnetPrefix -VirtualNetwork $virtualNetwork
PS /home/harsh> $virtualNetwork | Set-AzVirtualNetwork

ResourceGroupName Name       Location     ProvisioningState EnableDdosProtection
----------------- ----       --------     ----------------- --------------------
harshrgg          Harshvnett centralindia Succeeded         False

PS /home/harsh>
```

 # Create Host Pool

Define variables

$resourceGroupName = "harshrg"

```
$location = "EastAsia"

$hostPoolName = "HarshHostPool"

$hostPoolFriendlyName = "Harsh AVD Host Pool"

$hostPoolType = "Pooled"  # or "Personal"

$loadBalancerType = "BreadthFirst"  # Or "DepthFirst"

$preferredAppGroupType = "Desktop"

$customRdpProperty = ""


# Make sure resource group exists (Refer this if resoucegroup is not created)

if (-not (Get-AzResourceGroup -Name $resourceGroupName -ErrorAction SilentlyContinue)) {

    New-AzResourceGroup -Name $resourceGroupName -Location $location

}
```

#**Actual script for host pool creation**-

New-AzWvdHostPool -ResourceGroupName "harshrg" -Name "HarshHP" -Location "Eastasia" -FriendlyName "Harsh AVD desktop" -HostPoolType "pooled" -LoadBalancerType "BreadthFirst" -PreferredAppGroupType "Desktop"

```
PS /home/harsh> New-AzWvdHostPool -ResourceGroupName "harshrgg" -Name "HarshHPP" -Location "Centralindia" -FriendlyName "Harsh AVD desktop" -HostPoolType "pooled" -LoadBal
ancerType "BreadthFirst" -PreferredAppGroupType "Desktop"

AgentUpdateMaintenanceWindow               :
AgentUpdateMaintenanceWindowTimeZone       :
AgentUpdateType                            :
AgentUpdateUseSessionHostLocalTime         :
AppAttachPackageReference                  : {}
ApplicationGroupReference                  : {}
CloudPcResource                            : False
CustomRdpProperty                          : drivestoredirect:s:*;usbdevicestoredirect:s:*;redirectclipboard:i:1;redirectprinters:i:1;audiomode:i:0;videoplaybackmode:i:1;
                                             devicestoredirect:s:*;redirectcomports:i:1;redirectsmartcards:i:1;enablecredsspsupport:i:1;redirectwebauthn:i:1;use
                                             multimon:i:1;
Description                                :
Etag                                       :
FriendlyName                               : Harsh AVD desktop
HostPoolType                               : Pooled
Id                                         : /subscriptions/0c3bb37f-98c6-42f6-96c7-7bb5d7e34f15/resourcegroups/harshrgg/providers/Microsoft.DesktopVirtualization/hostpoo
                                             ls/HarshHPP
IdentityPrincipalId                        :
IdentityTenantId                           :
IdentityType                               :
Kind                                       :
LoadBalancerType                           : BreadthFirst
Location                                   : centralindia
ManagedBy                                  :
MaxSessionLimit                            : 999999
Name                                       : HarshHPP
ObjectId                                   : e045f2ee-e7a2-4d40-886d-6b891e159ae8
PersonalDesktopAssignmentType              :
```

```
PlanProduct                                :
PlanPromotionCode                          :
PlanPublisher                              :
PlanVersion                                :
PreferredAppGroupType                      : Desktop
PrivateEndpointConnection                  :
PublicNetworkAccess                        : Enabled
RegistrationInfoExpirationTime             :
RegistrationInfoRegistrationTokenOperation : None
RegistrationInfoToken                      :
ResourceGroupName                          : harshrgg
Ring                                       : 1
SkuCapacity                                :
SkuFamily                                  :
SkuName                                    :
SkuSize                                    :
SkuTier                                    :
SsoClientId                                :
SsoClientSecretKeyVaultPath                :
SsoSecretType                              :
SsoadfsAuthority                           :
StartVMOnConnect                           : False
SystemDataCreatedAt                        : 5/25/2025 10:54:21 AM
SystemDataCreatedBy                        : harshsonar21510@gmail.com
SystemDataCreatedByType                    : User
SystemDataLastModifiedAt                   : 5/25/2025 10:54:21 AM
SystemDataLastModifiedBy                   : harshsonar21510@gmail.com
SystemDataLastModifiedByType               : User
Tag                                        : {
                                             }
Type                                       : Microsoft.DesktopVirtualization/hostpools
VMTemplate                                 :
ValidationEnvironment                      : False
```

Error-Register-AzResourceProvider -ProviderNamespace "Microsoft.DesktopVirtualization"

Run this command if you get above error

Register-AzResourceProvider -ProviderNamespace "Microsoft.DesktopVirtualization"

Wait for the registration to complete (you can check status using the next command):

Get-AzResourceProvider -ProviderNamespace "Microsoft.DesktopVirtualization"

Make sure the RegistrationState is Registered.

 # Create Workspace

Variables

$resourceGroupName = "harshrgg"

$location = "EastAsia"

$workspaceName = "HarshWorkspace"

$appGroupName = "HarshAppGroup"

$hostPoolName = "HarshHostPool"

#Actual script-

New-AzWvdWorkspace -Name "HarshWS" -ResourceGroupName "harshrg" -Location "EastAsia" -FriendlyName "Harsh AVD Workspace" -Description "Workspace for Harsh AVD environment"

```
PS /home/harsh> New-AzWvdWorkspace -Name "HarshWSS" -ResourceGroupName "harshrgg" -Location "Centralindia" -FriendlyName "Harsh AVD Workspace" -Description "Workspace for
Harsh AVD environment"

ApplicationGroupReference    : {}
CloudPcResource              : False
Description                  : Workspace for Harsh AVD environment
Etag                         :
FriendlyName                 : Harsh AVD Workspace
Id                           : /subscriptions/0c3bb37f-98c6-42f6-96c7-7bb5d7e34f15/resourcegroups/harshrgg/providers/Microsoft.DesktopVirtualization/workspaces/HarshWSS
IdentityPrincipalId          :
IdentityTenantId             :
IdentityType                 :
Kind                         :
Location                     : centralindia
ManagedBy                    :
Name                         : HarshWSS
ObjectId                     : 0024a94d-0c94-4a25-9d90-33d5d9c260c0
PlanName                     :
PlanProduct                  :
PlanPromotionCode            :
PlanPublisher                :
PlanVersion                  :
PrivateEndpointConnection    :
PublicNetworkAccess          : Enabled
ResourceGroupName            : harshrgg
SkuCapacity                  :
SkuFamily                    :
SkuName                      :
SkuSize                      :
SkuTier                      :
SystemDataCreatedAt          : 5/25/2025 10:58:45 AM
SystemDataCreatedBy          : harshsonar2151@gmail.com
SystemDataCreatedByType      : User
SystemDataLastModifiedAt     : 5/25/2025 10:58:45 AM
```

 # Actual powershell script for app group creation

Define variables

$resourceGroupName = "harshrgg"

$location = "Centralindia"

$hostPoolName = "HarshHPP"

$appGroupName = "HarshAGG"

$friendlyName = "Harsh Desktop App Group"

Get the subscription ID properly

$subscriptionId = (Get-AzContext).Subscription.Id)

Construct the full Host Pool ARM path

$hostPoolArmPath = "/subscriptions/0c3bb37f-98c6-42f6-96c7-7bb5d7e34f15/resourceGroups/harshrgg/providers/Microsoft.DesktopVirtualization/hostPools/HarshHPP"

Create the Application Group (now with valid HostPoolArmPath)

New-AzWvdApplicationGroup -ResourceGroupName harshrgg -Location Centralindia -Name HarshAGG -FriendlyName $friendlyName -ApplicationGroupType "Desktop" -HostPoolArmPath $hostPoolArmPath

 # Actual Script: Associate Application Group with Workspace

Actual Script: Associate Application Group with Workspace

Step 1: Install or Update the Az.DesktopVirtualization Module

First, ensure that the Az.DesktopVirtualization module is installed and updated to the latest version.

Install-Module -Name Az.DesktopVirtualization -Force -AllowClobber

Step 2: Import the Module into the Current Session

After installing or updating the module, import it into your current PowerShell session:

Import-Module Az.DesktopVirtualization

Step 3: Verify the Cmdlet Availability

Check if the New-AzWvdWorkspaceAssociation cmdlet is available:

Get-Command -Module Az.DesktopVirtualization -Name *WorkspaceAssociation*

If the cmdlet is listed, you can proceed to use it. If not, ensure that the module is correctly installed and imported.

Get workspace ARM path.

$resourceGroupName = "harshrgg"

$workspaceName = "HarshWSS"

Get the workspace object

$workspace = Get-AzWvdWorkspace -ResourceGroupName $resourceGroupName -Name $workspaceName

Display the ARM path (resource ID)

$workspace.Id

ARM PATH

/subscriptions/0c3bb37f-98c6-42f6-96c7-7bb5d7e34f15/resourcegroups/harshrgg/providers/Microsoft.DesktopVirtualization/workspaces/HarshWSS

Set variables

$resourceGroupName = "harshrgg"

$appGroupName = "HarshAGG"

$workspaceName = "HarshWSS"

Get existing objects

$appGroup = Get-AzWvdApplicationGroup -ResourceGroupName $resourceGroupName -Name $appGroupName

$workspace = Get-AzWvdWorkspace -ResourceGroupName $resourceGroupName -Name $workspaceName

Update the application group to associate it with the workspace

Update-AzWvdApplicationGroup -ResourceGroupName $resourceGroupName -Name $appGroupName -FriendlyName $appGroup.FriendlyName -Description $appGroup.Description

```
PS /home/harsh> $resourceGroupName = "harshrgg"
PS /home/harsh> $appGroupName = "HarshAGG"
PS /home/harsh> $workspaceName = "HarshWSS"
PS /home/harsh>
PS /home/harsh> # Get existing objects
PS /home/harsh> $appGroup = Get-AzWvdApplicationGroup -ResourceGroupName $resourceGroupName -Name $appGroupName
PS /home/harsh> $workspace = Get-AzWvdWorkspace -ResourceGroupName $resourceGroupName -Name $workspaceName
PS /home/harsh>
PS /home/harsh> Update-AzWvdApplicationGroup -ResourceGroupName $resourceGroupName -Name $appGroupName -FriendlyName $appGroup.FriendlyName -Description $appGroup.Description

ApplicationGroupType          : Desktop
CloudPcResource               : False
Description                   :
Etag                          :
FriendlyName                  :
HostPoolArmPath               : /subscriptions/0c3bb37f-98c6-42f6-96c7-7bb5d7e34f15/resourcegroups/harshrgg/providers/Microsoft.DesktopVirtualization/hostpools/HarshHPP
Id                            : /subscriptions/0c3bb37f-98c6-42f6-96c7-7bb5d7e34f15/resourcegroups/harshrgg/providers/Microsoft.DesktopVirtualization/applicationgroups/HarshAGG
IdentityPrincipalId           :
IdentityTenantId              :
IdentityType                  :
Kind                          : Desktop
Location                      : centralindia
ManagedBy                     :
Name                          : HarshAGG
ObjectId                      : 34e101fe-50e1-46bb-b1cc-d457572ea69b
PlanName                      :
PlanProduct                   :
PlanPromotionCode             :
PlanPublisher                 :
PlanVersion                   :
ResourceGroupName             : harshrgg
ShowInFeed                    :
```

```
PlanVersion                   :
ResourceGroupName             : harshrgg
ShowInFeed                    :
SkuCapacity                   :
SkuFamily                     :
SkuName                       :
SkuSize                       :
SkuTier                       :
SystemDataCreatedAt           : 5/25/2025 11:10:12 AM
SystemDataCreatedBy           : harshsonar2151@gmail.com
SystemDataCreatedByType       : User
SystemDataLastModifiedAt      : 5/25/2025 11:40:49 AM
SystemDataLastModifiedBy      : harshsonar2151@gmail.com
SystemDataLastModifiedByType  : User
Tag                           : {
                                }
Type                          : Microsoft.DesktopVirtualization/applicationgroups
WorkspaceArmPath              :
```

Create VM

PowerShell Script to Create a General-Purpose VM

module need to create/install

Potential Causes and Solutions

1. Verify Active Subscription Context

Ensure that the correct subscription is set as the active context:

Get-AzContextAutosaveSetting

```
PS /home/harsh>  Get-AzContextAutosaveSetting

Mode             : CurrentUser
ContextDirectory : /home/harsh/.Azure
ContextFile      : AzureRmContext.json
CacheDirectory   : /home/harsh/.local/share/.IdentityService
CacheFile        : msal.cache.cae
KeyStoreFile     : keystore.cache
Settings         : {[InstallationId, 4267292f-2313-4ac6-9f13-df3278075363]}
```

If the output does not display your intended subscription, set it explicitly:

Set-AzContext -SubscriptionId "0c3bb37f-98c6-42f6-96c7-7bb5d7e34f15"

Replace "your-subscription-id" with the actual Subscription ID.

2. Check Resource Group Existence

Confirm that the specified resource group exists:

Get-AzResourceGroup -Name "harshrgg"

```
PS /home/harsh> Get-AzResourceGroup -Name "harshrgg"

ResourceGroupName : harshrgg
Location          : centralindia
ProvisioningState : Succeeded
Tags              :
ResourceId        : /subscriptions/0c3bb37f-98c6-42f6-96c7-7bb5d7e34f15/resourceGroups/harshrgg

PS /home/harsh>
```

If the resource group is missing, create it:

E.g. -> New-AzResourceGroup -Name "harshrg" -Location "EastAsia"

3. Ensure Resource Provider Registration

Verify that the Microsoft.Compute resource provider is registered:

Get-AzResourceProvider -ProviderNamespace Microsoft.Compute

```
PS /home/harsh> Get-AzResourceProvider -ProviderNamespace Microsoft.Compute

ProviderNamespace : Microsoft.Compute
RegistrationState : Registered
ResourceTypes     : {availabilitySets}
Locations         : {East US, East US 2, West US, Central US...}

ProviderNamespace : Microsoft.Compute
RegistrationState : Registered
ResourceTypes     : {virtualMachines}
Locations         : {East US, East US 2, West US, Central US...}

ProviderNamespace : Microsoft.Compute
RegistrationState : Registered
ResourceTypes     : {virtualMachines/extensions}
Locations         : {East US, East US 2, West US, Central US...}

ProviderNamespace : Microsoft.Compute
RegistrationState : Registered
ResourceTypes     : {virtualMachineScaleSets}
Locations         : {East US, East US 2, West US, Central US...}

ProviderNamespace : Microsoft.Compute
RegistrationState : Registered
ResourceTypes     : {virtualMachineScaleSets/extensions}
Locations         : {East US, East US 2, West US, Central US...}

ProviderNamespace : Microsoft.Compute
RegistrationState : Registered
ResourceTypes     : {virtualMachineScaleSets/virtualMachines}
Locations         : {East US, East US 2, West US, Central US...}
```

If it's not registered, register it:

Register-AzResourceProvider -ProviderNamespace Microsoft.Compute

4. Update Azure PowerShell Modules

Outdated modules can cause issues. Update the Az module:

Update-Module -Name Az

If the issue persists, consider installing a specific version known to be stable:

Install-Module -Name Az.Compute -RequiredVersion 4.23.0 -Force

Then, restart your PowerShell session.

5. Check Azure Subscription Permissions

Ensure your account has sufficient permissions to create resources in the subscription. If you're unsure, contact your Azure administrator

Variables

$resourceGroupName = "harshrgg"

$location = "Centralindia"

$vmName = "HarshVMMM"

$virtualNetworkName = "Harshvnett"

$subnetName = "default"

$ipName = "$vmName-ip"

$nicName = "$vmName-nic"

$size = "Standard_D4s_v3"

$adminUsername = "azadmin"

$adminPassword = ConvertTo-SecureString "Password@1234!" -AsPlainText -Force

```
PS /home/harsh> $resourceGroupName = "harshrgg"
PS /home/harsh> $location = "Centralindia"
PS /home/harsh> $vmName = "HarshVMMM"
PS /home/harsh> $virtualNetworkName = "Harshvnett"
PS /home/harsh> $subnetName = "default"
PS /home/harsh> $ipName = "$vmName-ip"
PS /home/harsh> $nicName = "$vmName-nic"
PS /home/harsh> $size = "Standard_D4s_v3"
PS /home/harsh> $adminUsername = "azadmin"
PS /home/harsh> $adminPassword = ConvertTo-SecureString "Password@1234!" -AsPlainText -Force
```

Login to Azure if not already logged in

Connect-AzAccount

If above command showing error then run below command (Interactive authentication is not supported in this session, please run cmdlet)

Connect-AzAccount -UseDeviceAuthentication

```
PS /home/harsh> Connect-AzAccount
WARNING: Interactive authentication is not supported in this session, please run cmdlet 'Connect-AzAccount -UseDeviceAuthentication'.
PS /home/harsh> Connect-AzAccount -UseDeviceAuthentication
WARNING: You may need to login again after updating "EnableLoginByWam".
Please select the account you want to login with.

[Login to Azure] To sign in, use a web browser to open the page https://microsoft.com/devicelogin and enter the code NDC884SF6 to authenticate.
Retrieving subscriptions for the selection...

[Announcements]
With the new Azure PowerShell login experience, you can select the subscription you want to use more easily. Learn more about it and its configuration at https://go.microsoft.com/fwlink/?
inkid=2271909.

If you encounter any problem, please open an issue at: https://aka.ms/azpsissue

Subscription name     Tenant
-----------------     ------
Azure subscription 1 Default Directory
```

1. Get Virtual Network and Subnet

$vnet = Get-AzVirtualNetwork -Name $virtualNetworkName -ResourceGroupName $resourceGroupName

$subnet = $vnet | Select-Object -ExpandProperty subnets | Where-Object { $_.Name -eq $subnetName }

```
PS /home/harsh> $vnet = Get-AzVirtualNetwork -Name $virtualNetworkName -ResourceGroupName $resourceGroupName
PS /home/harsh> $subnet = $vnet | Select-Object -ExpandProperty subnets | Where-Object { $_.Name -eq $subnetName }
```

2. Create a Public IP Address

$publicIp = New-AzPublicIpAddress -Name $ipName -ResourceGroupName $resourceGroupName -Location $location -AllocationMethod static

```
PS /home/harsh> $publicIp = New-AzPublicIpAddress -Name $ipName -ResourceGroupName $resourceGroupName -Location $location -AllocationMethod static
PS /home/harsh>
```

3. Create a Network Interface Card (NIC)

$nic = New-AzNetworkInterface -Name $nicName -ResourceGroupName $resourceGroupName -Location $location -SubnetId $subnet.Id -PublicIpAddressId $publicIp.Id

```
PS /home/harsh> $nic = New-AzNetworkInterface -Name $nicName -ResourceGroupName $resourceGroupName -Location $location -SubnetId $subnet.Id -PublicIpAddressId $publicIp.Id
```

4. Specify VM Config

$vmConfig = New-AzVMConfig -VMName $vmName -VMSize $size |

Set-AzVMOperatingSystem -Windows -ComputerName $vmName -Credential (New-Object System.Management.Automation.PSCredential($adminUsername, $adminPassword)) -ProvisionVMAgent -EnableAutoUpdate |

Set-AzVMSourceImage -PublisherName MicrosoftWindowsServer -Offer WindowsServer -Skus 2019-Datacenter -Version latest |

Add-AzVMNetworkInterface -Id $nic.Id

```
PS /home/harsh> $publicIp = New-AzPublicIpAddress -Name $ipName -ResourceGroupName $resourceGroupName -Location $location -AllocationMethod static
PS /home/harsh> $nic = New-AzNetworkInterface -Name $nicName -ResourceGroupName $resourceGroupName -Location $location -SubnetId $subnet.Id -PublicIpAddressId $publicIp.Id
PS /home/harsh> $vmConfig = New-AzVMConfig -VMName $vmName -VMSize $size |
>>     Set-AzVMOperatingSystem -Windows -ComputerName $vmName -Credential (New-Object System.Management.Automation.PSCredential($adminUsername, $adminPassword)) -ProvisionVMAgent -EnableAu
toUpdate |
>>     Set-AzVMSourceImage -PublisherName MicrosoftWindowsServer -Offer WindowsServer -Skus 2019-Datacenter -Version latest |
>>     Add-AzVMNetworkInterface -Id $nic.Id
PS /home/harsh>
```

5. Create the VM

New-AzVM -ResourceGroupName $resourceGroupName -Location $location -VM $vmConfig

#If above command not work then run below command

New-AzVM -ResourceGroupName "harshrg" -Location "EastAsia"

hit enter

1. enter VM name

2. adminuser name

3. admin password

Write-Host "`n✅ Virtual Machine '$vmName' created successfully in '$location'." -ForegroundColor Green

```
PS /home/harsh> New-AzVM -ResourceGroupName "harshrg" -Location "EastAsia"

cmdlet New-AzVM at command pipeline position 1
Supply values for the following parameters:
Name: HarshVMMM
Credential
User: azadmin
Password for user azadmin: **************
```

 # AVD AGENT INSTALLATION & REGISTRATION

--

1.# Prepare script to run on VM

$installAVDScript = @"

Invoke-WebRequest -Uri https://aka.ms/avdagent -OutFile AVD-Agent.msi

Invoke-WebRequest -Uri https://aka.ms/avdbootloader -OutFile AVD-Bootloader.msi

Start-Process msiexec.exe -ArgumentList '/i AVD-Agent.msi /quiet /qn /norestart' -Wait

Start-Process msiexec.exe -ArgumentList '/i AVD-Bootloader.msi /quiet /qn /norestart REGISTRATIONTOKEN=$avdRegistrationToken' -Wait

"@

2.

Encode the script to Base64

$encodedScript = [Convert]::ToBase64String([System.Text.Encoding]::Unicode.GetBytes($installAVDScript))

3.

Set-AzVMExtension -ResourceGroupName "harshrg" `

-VMName "HarshVMM" `

-Location "EastAsia" `

-Name "InstallAVDAgent" `

-Publisher "Microsoft.Compute" `

-ExtensionType "CustomScriptExtension" `

-TypeHandlerVersion "1.10" `

-SettingString $settings

AVD environment setup Manually from azure portal

1.Resource group

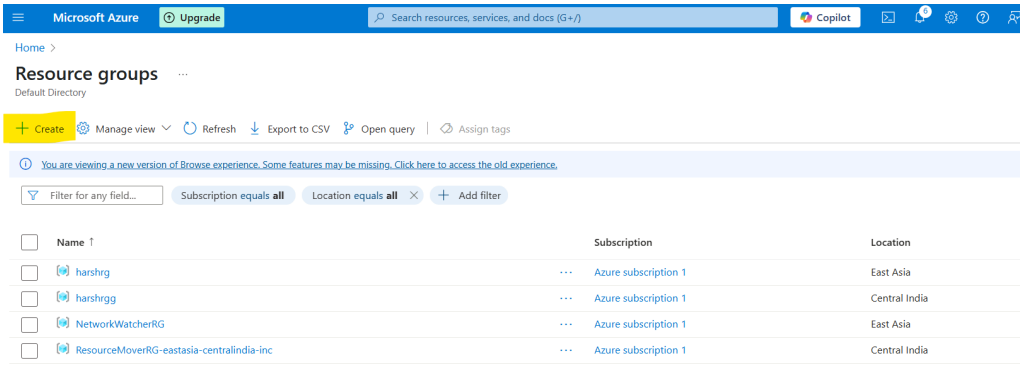
2.Vnet/subnet

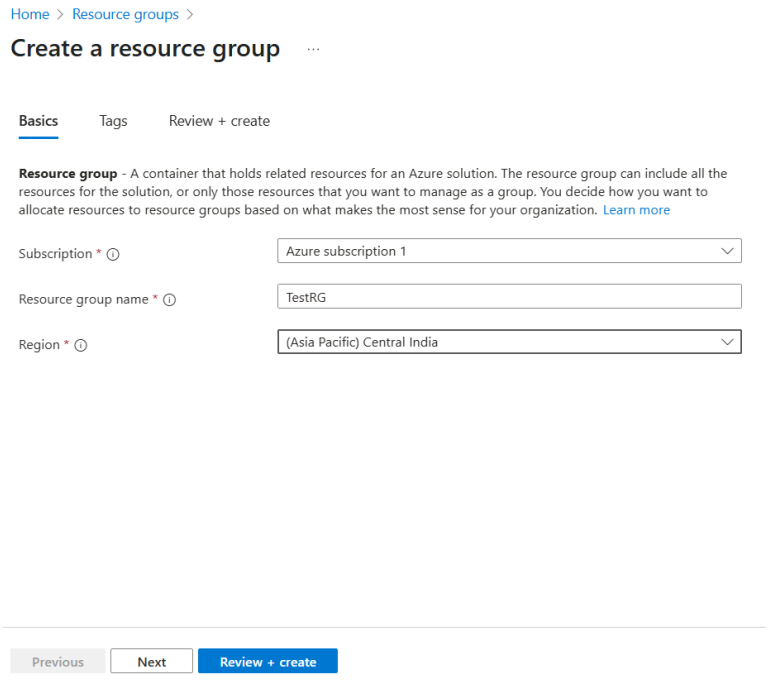
3.hostpool (workspace/Appgroup-RAG/DAG)

4.Virtual machine

1.Resource group

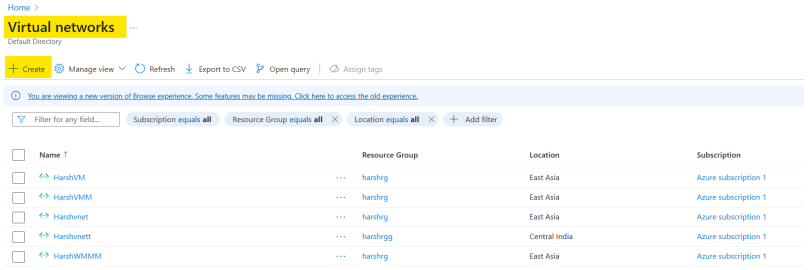
Type resource group in search bar and click on create

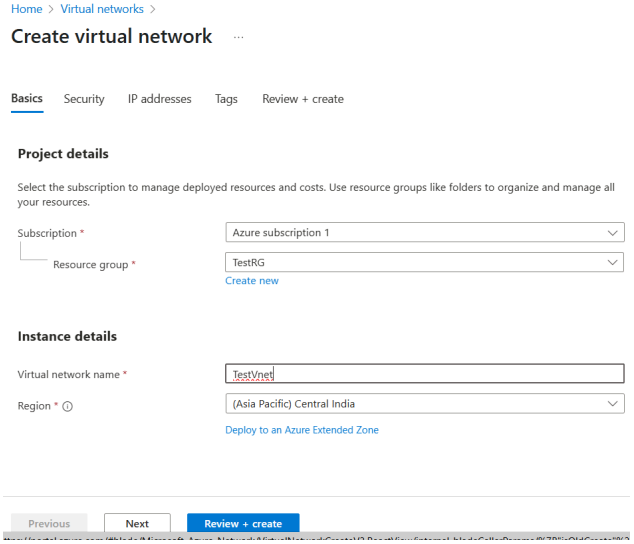




2.Vnet/subnet

Search for Virtual network in search bar then click on create





Home > Virtual networks >

Create virtual network ...

Basics Security **IP addresses** Tags Review + create

Add a subnet

10.0.0.0/16 Delete address space

This address prefix overlaps with virtual network 'Harshvnet'. If you intend to peer these virtual networks, change the address space. Learn more

10.0.0.0 /16

10.0.0.0 - 10.0.255.255 65,536 addresses

Subnets	IP address range	Size	NAT gateway
default	10.0.0.0 - 10.0.0.255	/24 (256 addresses)	-
AzureBastionSubnet	10.0.1.0 - 10.0.1.63	/26 (64 addresses)	-

Add IPv4 address space

Previous Next **Review + create**

Home > Virtual networks >

Create virtual network ...

Basics Security IP addresses Tags **Review + create**

View automation template

Basics

Subscription	Azure subscription 1
Resource Group	TestRG
Name	TestVnet
Region	Central India

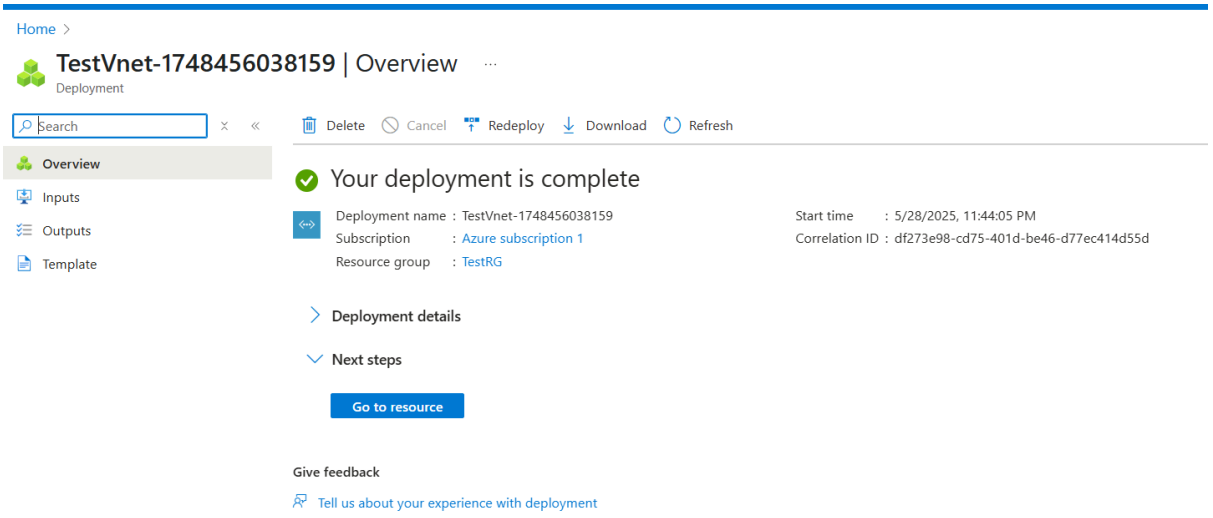
Security

Azure Bastion	Disabled
Azure Firewall	Disabled
Azure DDoS Network Protection	Disabled

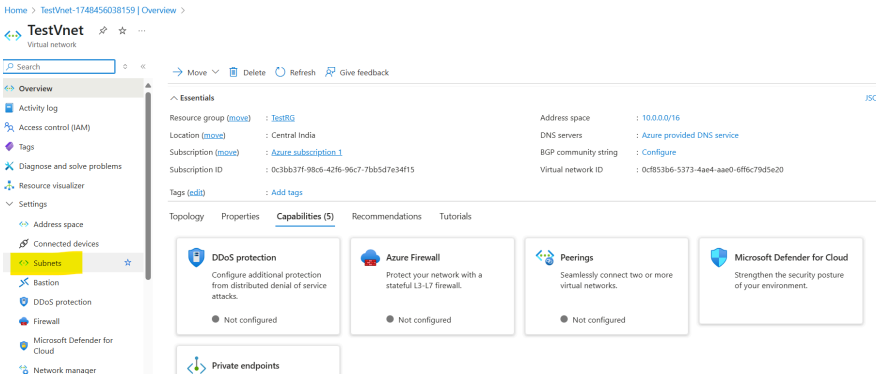
IP addresses

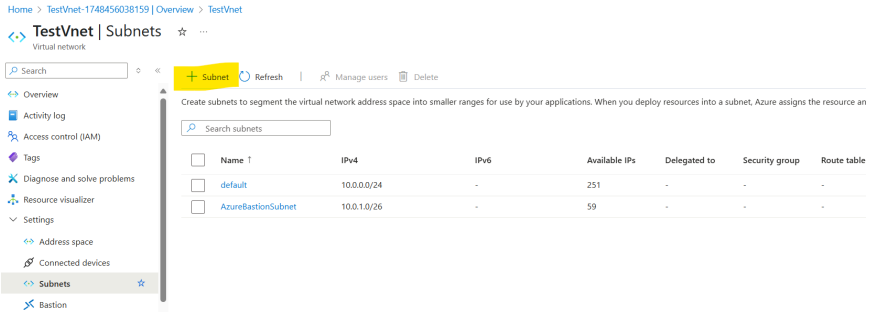
Address space	10.0.0.0/16 (65,536 addresses)

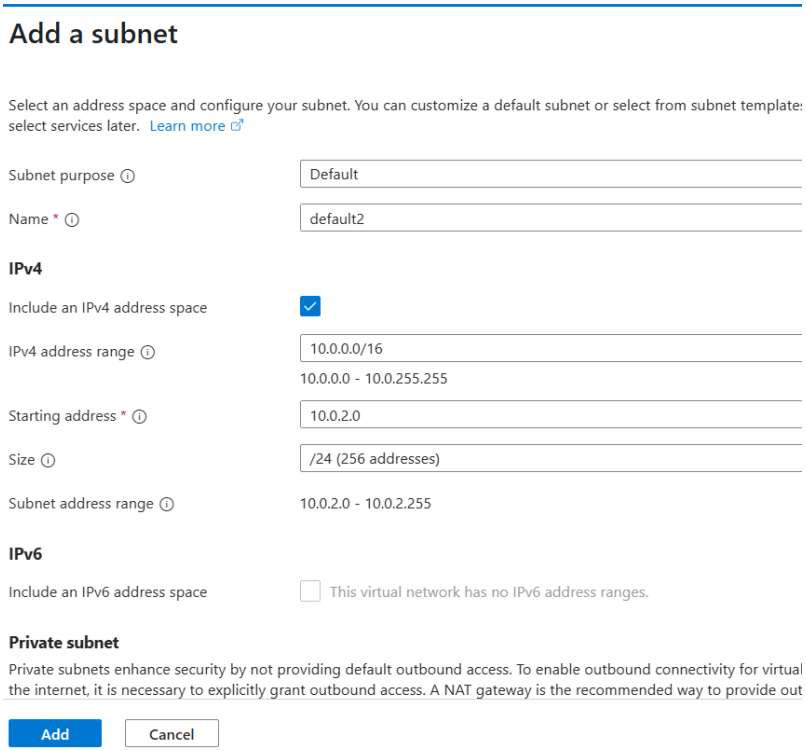
Previous Next **Create**



To create subnet with in the Virtual network

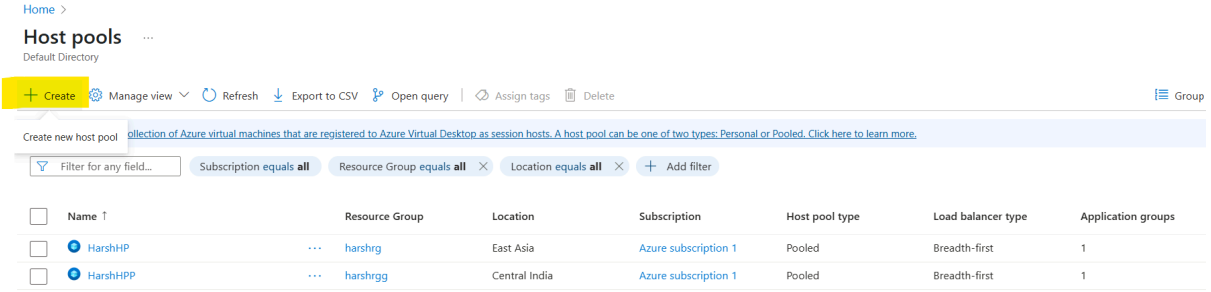






3.hostpool (workspace/Appgroup-RAG/DAG)

Search for host pool in search bar.



Home > Host pools >

Create a host pool ...

Basics | Session hosts | Workspace | Advanced | Tags | Review + create

Project details

A host pool is a collection of one or more identical virtual machines within an Azure Virtual Desktop environment. Here you can give details to create Azure virtual machines for your host pool now, or you can create and add them later, for example if you plan to add virtual machines from Azure Local. Learn more

Subscription * : Azure subscription 1

Resource group * : TestRG
Create new

Host pool name * : TestHP

Location * : Central India
Metadata will be stored in Azure geography associated with (Asia Pacific) Central India. Learn more

Validation environment : No / Yes

Preferred app group type * : Desktop

Host pool details

Define how session hosts in this host pool will be created, managed, and assigned.

Review + create | < Previous | Next: Session hosts >

Home > Host pools >

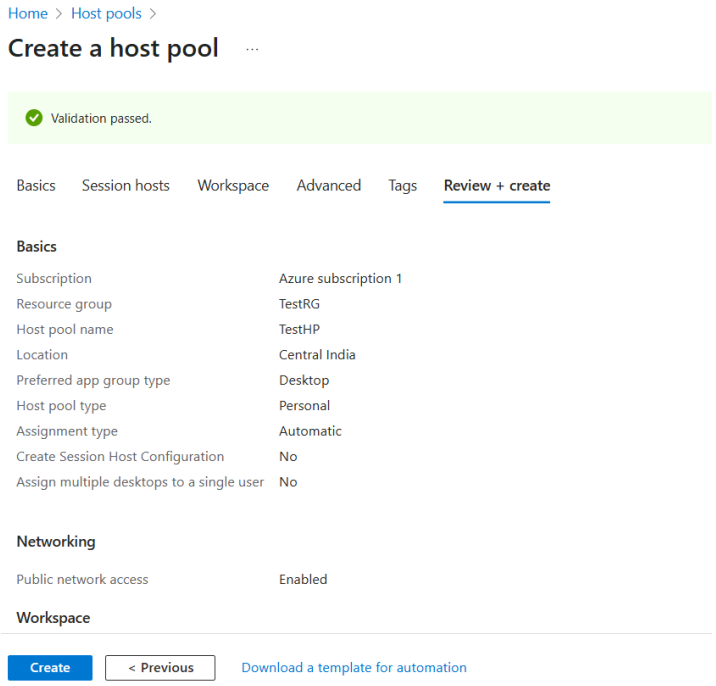
Create a host pool ...

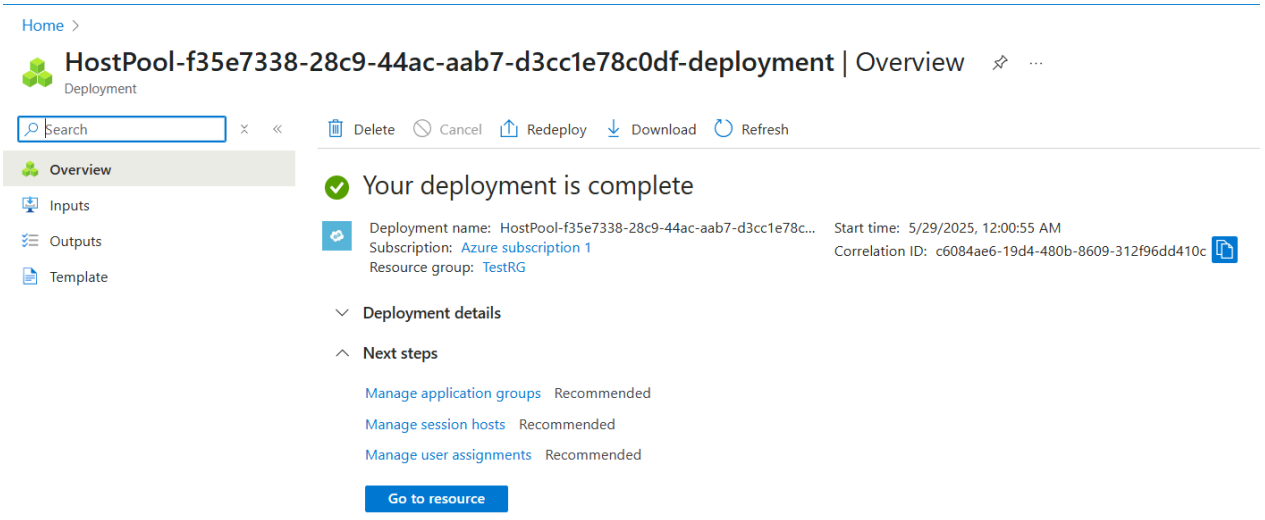
Basics | Session hosts | Workspace | Advanced | Tags | Review + create

To save some time, you can register the default desktop application group from this host pool, with a new or pre-existing workspace.

Register desktop app group : No / Yes

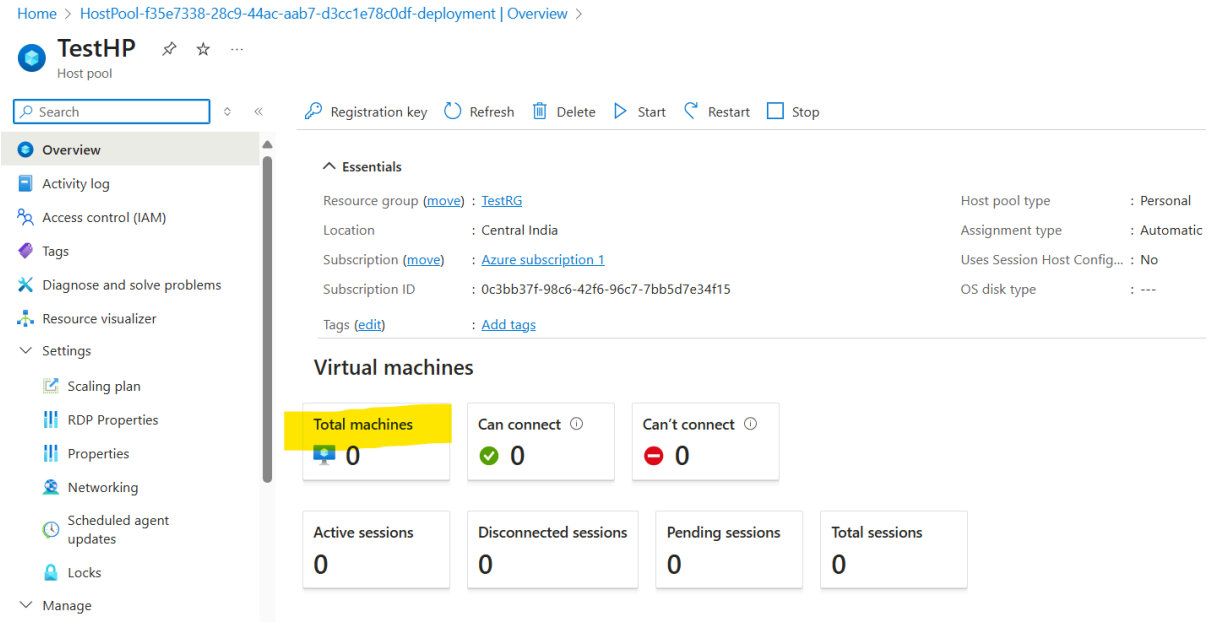
To this workspace * : -
Create new

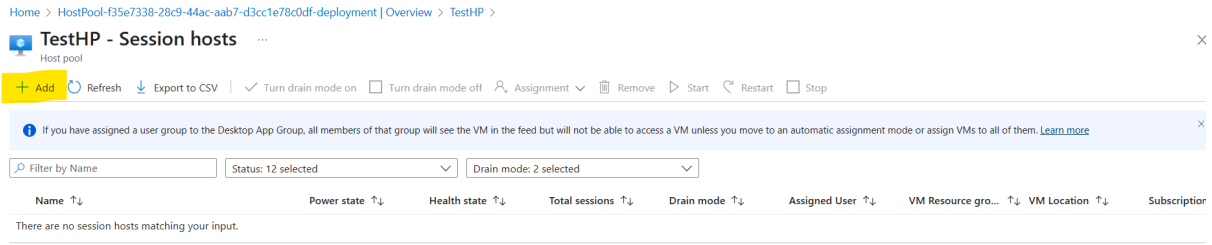




4.Virtual machine

To create virtual machine within the host pool.


Home > HostPool-f35e7338-28c9-44ac-aab7-d3cc1e78c0df-deployment | Overview >
TestHP
Host pool
Search
Registration key
Refresh
Delete
Start
Restart
Stop
Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Resource visualizer
Settings
Scaling plan
RDP Properties
Properties
Networking
Scheduled agent updates
Locks
Manage
Essentials
Resource group (move) : TestRG
Location : Central India
Subscription (move) : Azure subscription 1
Subscription ID : 0c3bb37f-98c6-42f6-96c7-7bb5d7e34f15
Tags (edit) : Add tags
Host pool type : Personal
Assignment type : Automatic
Uses Session Host Config... : No
OS disk type : ---
Virtual machines
Total machines 0
Can connect 0
Can't connect 0
Active sessions 0
Disconnected sessions 0
Pending sessions 0
Total sessions 0


Home > HostPool-f35e7338-28c9-44ac-aab7-d3cc1e78c0df-deployment | Overview > TestHP >
TestHP - Session hosts
Host pool
Add
Refresh
Export to CSV
Turn drain mode on
Turn drain mode off
Assignment
Remove
Start
Restart
Stop
If you have assigned a user group to the Desktop App Group, all members of that group will see the VM in the feed but will not be able to access a VM unless you move to an automatic assignment mode or assign VMs to all of them. Learn more
Filter by Name
Status: 12 selected
Drain mode: 2 selected
Name
Power state
Health state
Total sessions
Drain mode
Assigned User
VM Resource gro...
VM Location
Subscription
There are no session hosts matching your input.

Home > HostPool-f35e7338-28c9-44ac-aab7-d3cc1e78c0df-deployment | Overview > TestHP > TestHP - Session hosts >

Add virtual machines to a host pool ...

Basics | Virtual Machines | Tags | Review + create

Project details

A host pool is a collection of one or more identical virtual machines within an Azure Virtual Desktop environment. Here you can give details to create Azure virtual machines for your host pool now, or you can create and add them later, for example if you plan to add virtual machines from Azure Local. Learn more

Subscription	Azure subscription 1
Resource group	TestRG
Host pool name	TestHP
Location	Central India
Validation environment	No / Yes
Preferred app group type	Desktop

Host pool details

Define how session hosts in this host pool will be created, managed, and assigned.

Host pool type	Personal

If you select pooled, users can maintain their personalization and user data

Review + create | < Previous | Next: Virtual Machines >

Home > HostPool-f35e7338-28c9-44ac-aab7-d3cc1e78c0df-deployment | Overview > TestHP > TestHP - Session hosts >

Add virtual machines to a host pool ...

Basics | Virtual Machines | Tags | Review + create

Add virtual machines: (○) No (●) Yes

Host pools are a collection of one or more identical virtual machines within Azure Virtual Desktop environments. Here you provide a common set of properties to update the Session hosts within your host pool.

Resource group: TestRG

Name prefix *: TestVM

Session host name must be unique within the Resource Group.

Virtual machine type: (●) Azure virtual machine (○) Azure Local virtual machine

Virtual machine location: Central India

Availability options: Availability zones

Availability zones *: Zone 1

Security type *: Trusted launch virtual machines

Enable secure boot: ☑

Enable vTPM: ☑

Integrity monitoring: ☑

Image *: Windows 11 Enterprise, Version 24H2

See all images

Virtual machine size *: **Standard D2as v5**
2 vCPU's, 8 GiB memory
Change size

Hibernate: ☐

Number of VMs *: 1

OS disk type *: Standard SSD

OS disk size *: Default size (128GiB)

Boot diagnostics: (●) Enable with managed storage account (recommended) (○) Enable with custom storage account (○) Disable

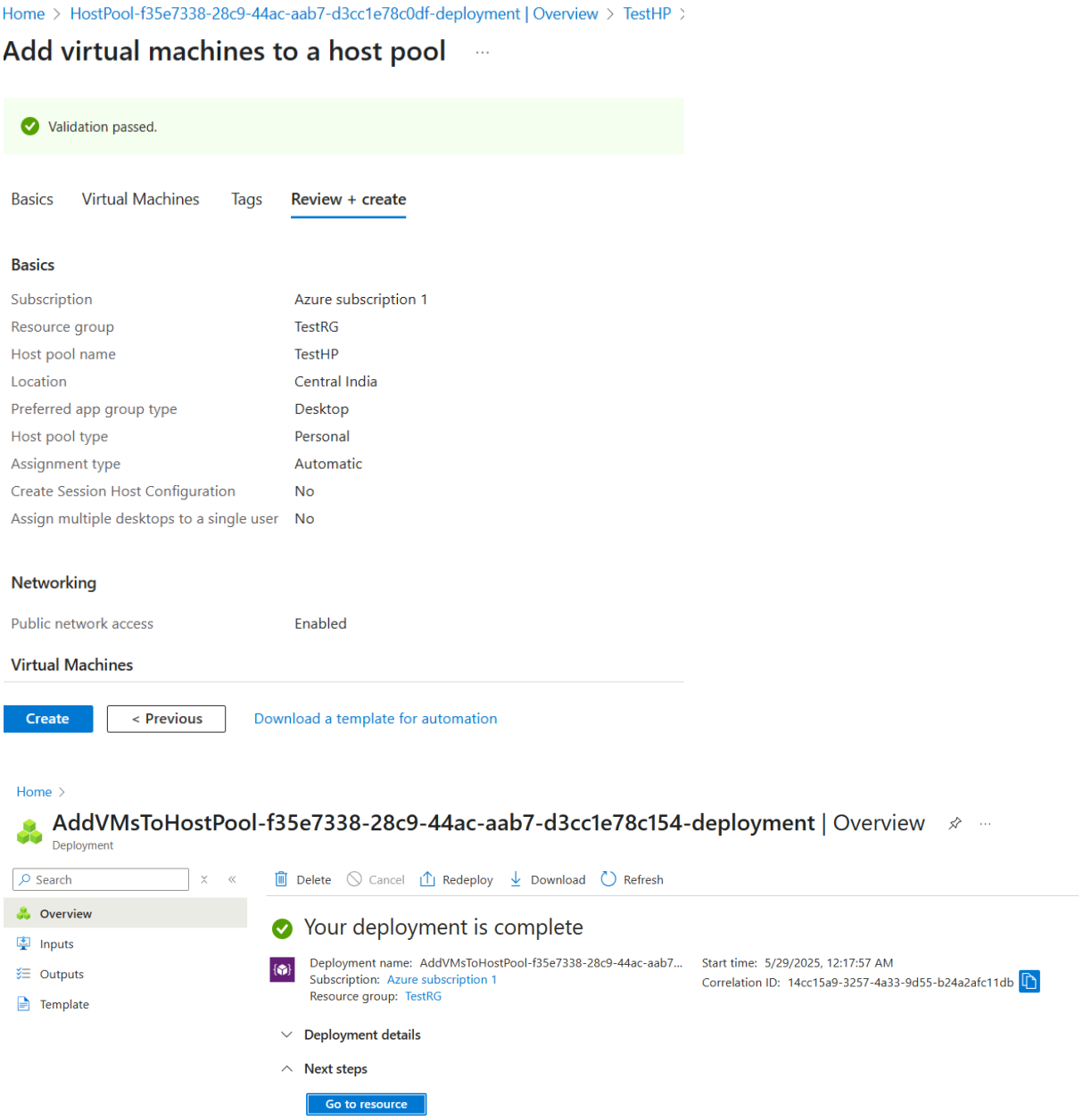
Network and security

Use Azure Firewall to secure your VNET and host pool resources. Learn more

Virtual network *: TestVnet

Subnet: default2

Review + create | < Previous | Next: Tags >

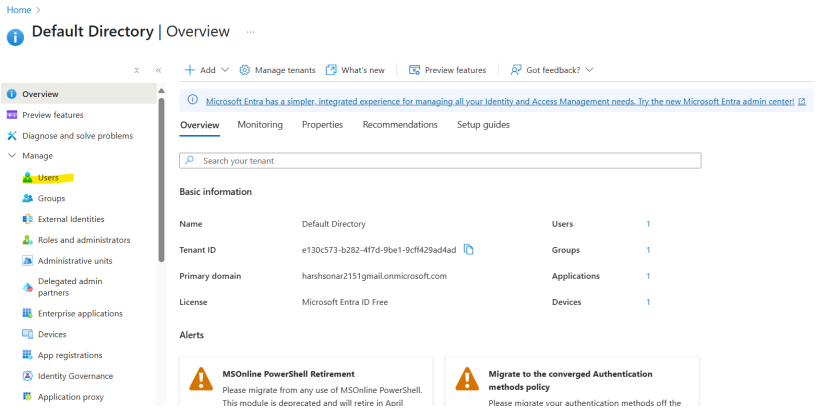

Home > HostPool-f35e7338-28c9-44ac-aab7-d3cc1e78c0df-deployment | Overview > TestHP >
Add virtual machines to a host pool
Validation passed.
Basics
Virtual Machines
Tags
Review + create
Basics
Subscription
Azure subscription 1
Resource group
TestRG
Host pool name
TestHP
Location
Central India
Preferred app group type
Desktop
Host pool type
Personal
Assignment type
Automatic
Create Session Host Configuration
No
Assign multiple desktops to a single user
No
Networking
Public network access
Enabled
Virtual Machines
Create
< Previous
Download a template for automation
Home >
AddVMsToHostPool-f35e7338-28c9-44ac-aab7-d3cc1e78c154-deployment | Overview
Deployment
Search
Overview
Inputs
Outputs
Template
Delete
Cancel
Redeploy
Download
Refresh
Your deployment is complete
Deployment name: AddVMsToHostPool-f35e7338-28c9-44ac-aab7...
Subscription: Azure subscription 1
Resource group: TestRG
Start time: 5/29/2025, 12:17:57 AM
Correlation ID: 14cc15a9-3257-4a33-9d55-b24a2afc11db
Deployment details
Next steps
Go to resource

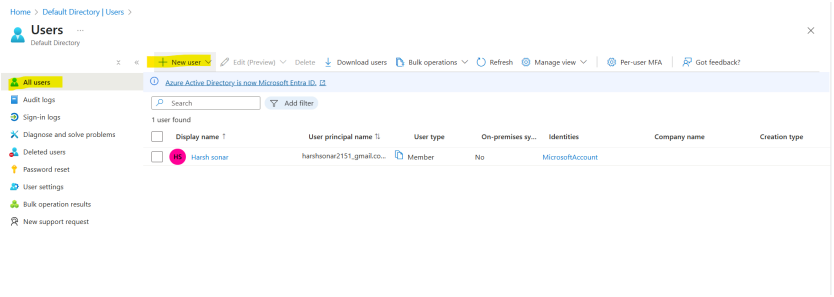
Add users and assign to Host pool

1. Add User to AD
2. Create group.
3. Add user in group
4. Assign user group to host pool.

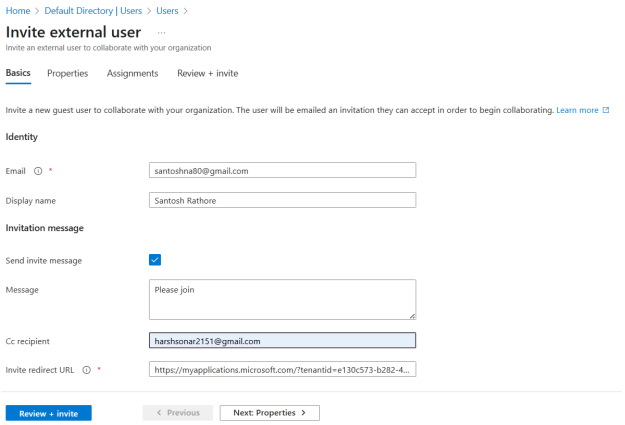
ADD Users to AD

Go to the Microsoft Entra ID and click on the users

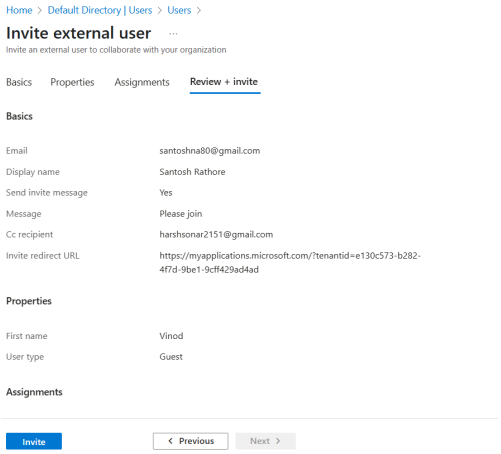




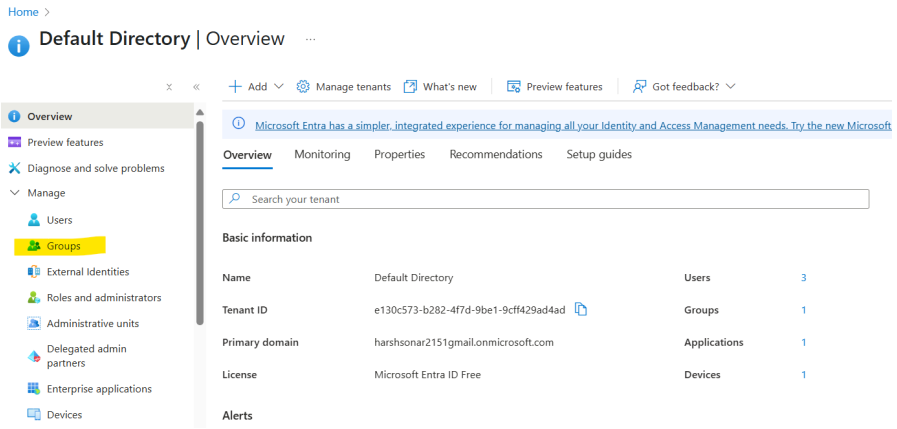
For inviting external user

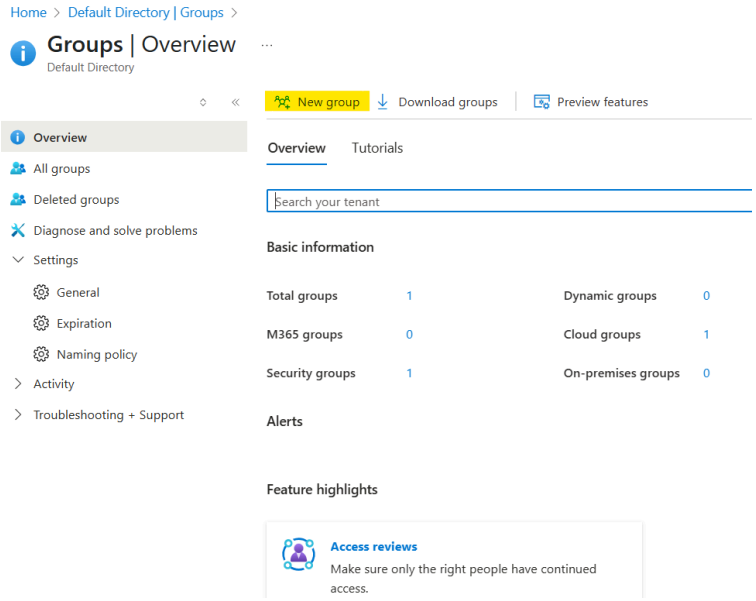


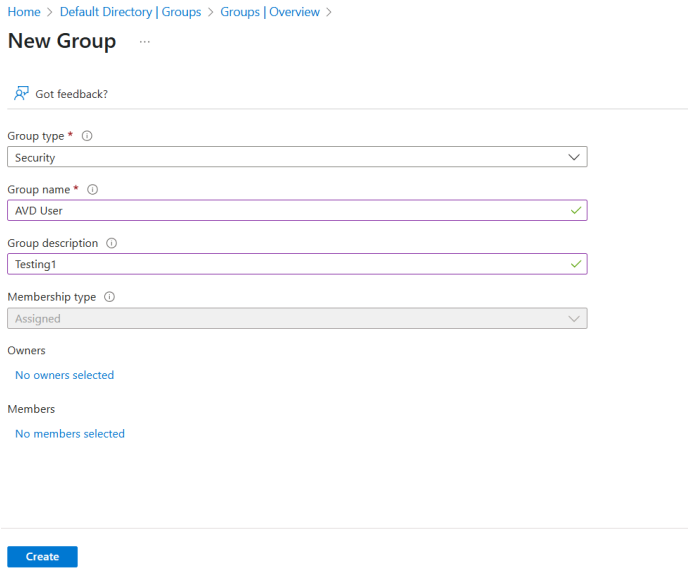
In next step u can fill basic details as per required in properties.



Create group



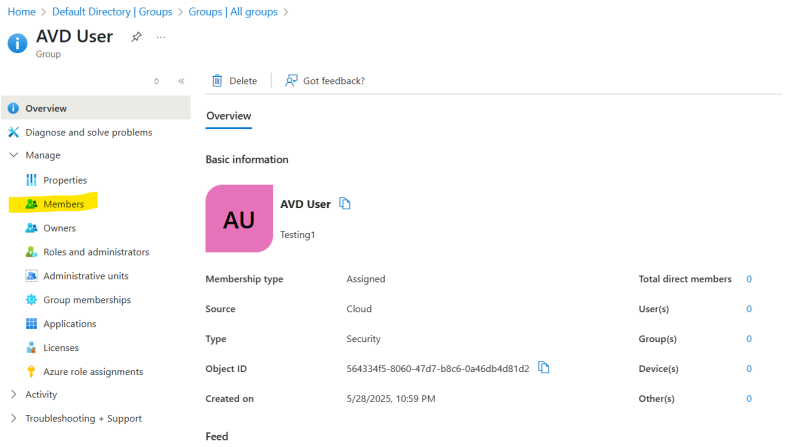


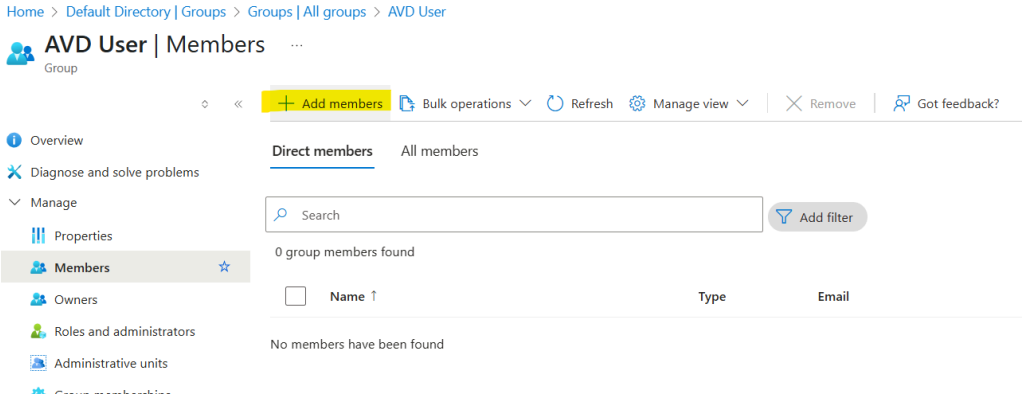


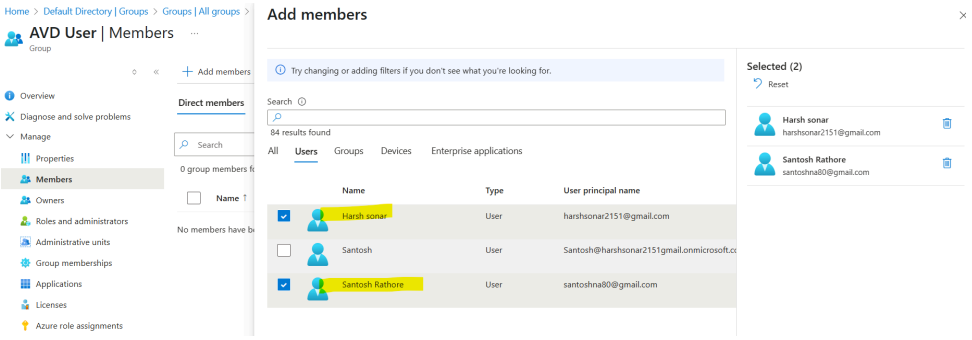
You can add owner in the group.

Add user in group

Select the group that you already created and add the member.

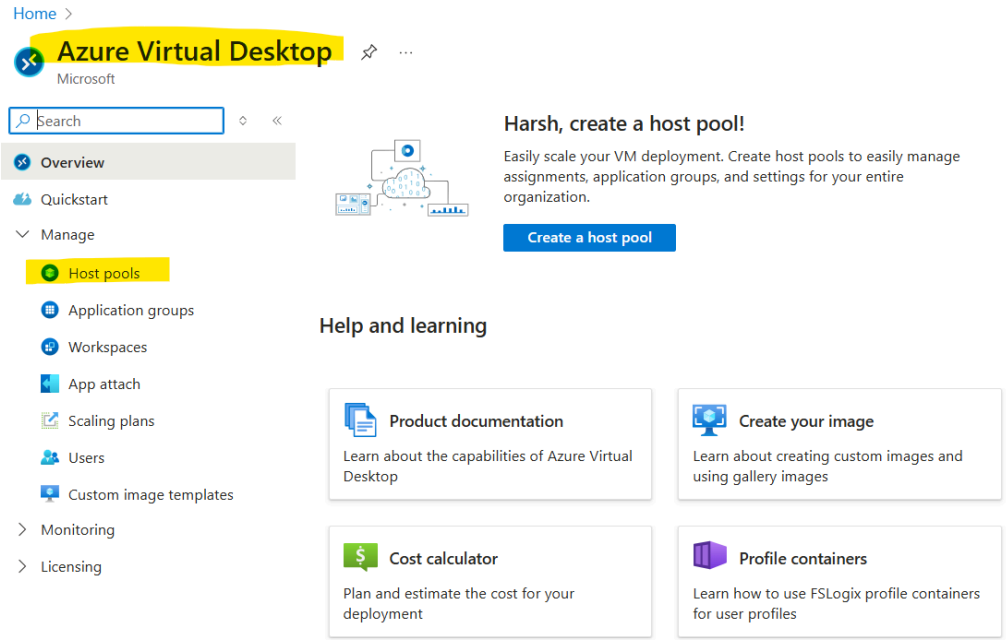


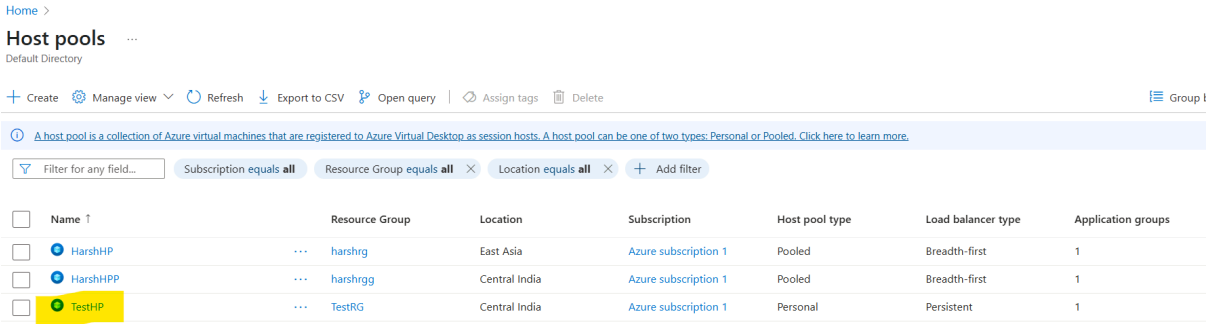


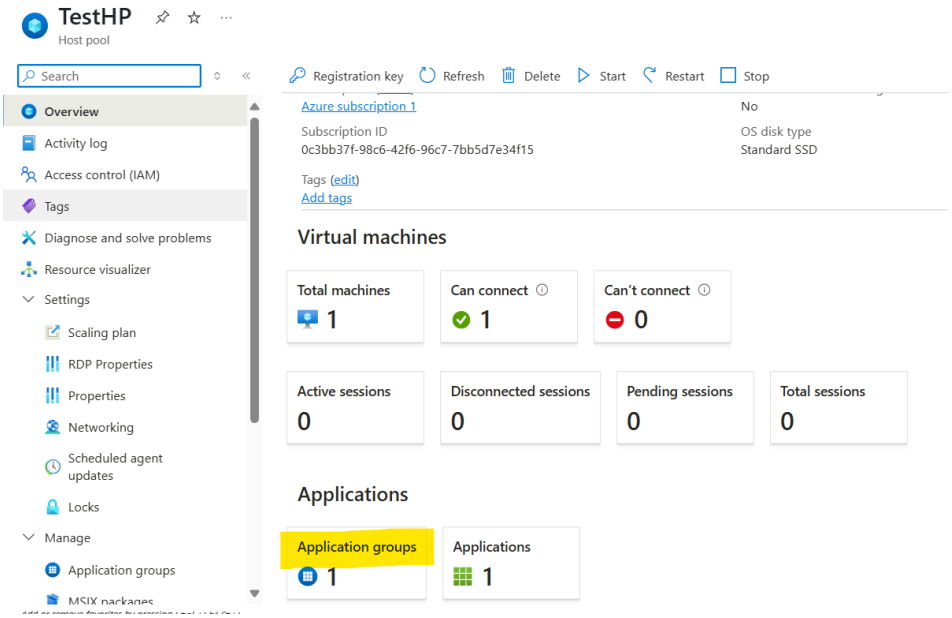


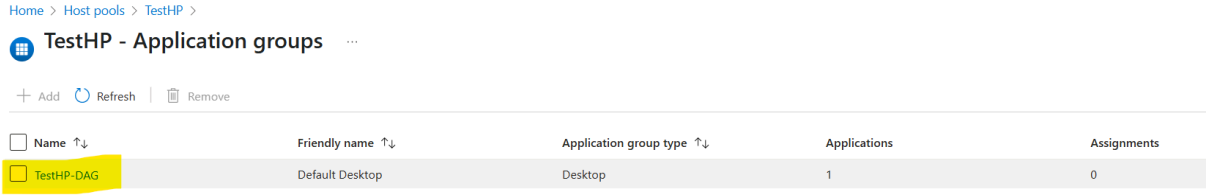
4. Assign user group to host pool

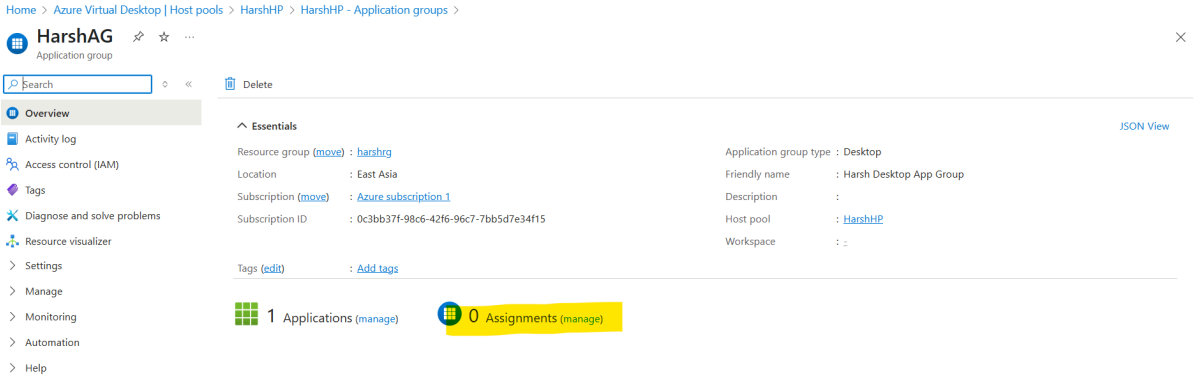
Go to AVD then click on the host Pool.

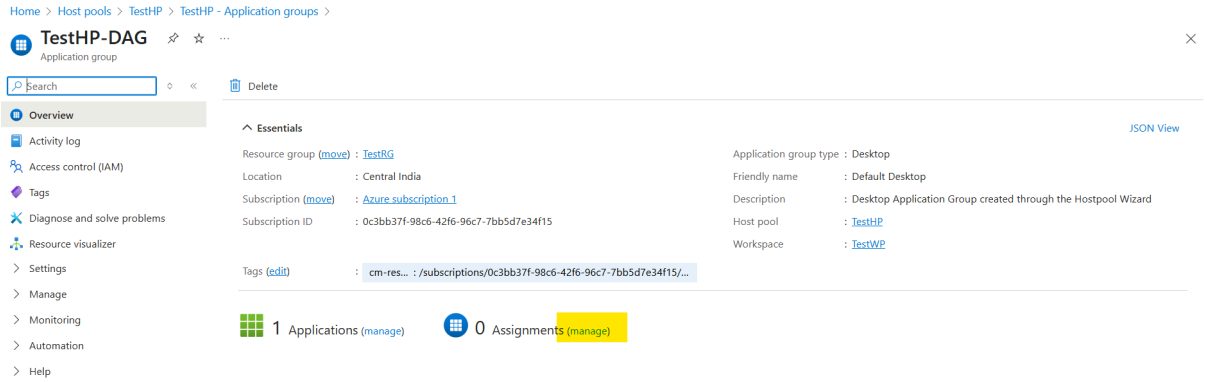

Home
Azure Virtual Desktop
Microsoft
Search
Overview
Quickstart
Manage
Host pools
Application groups
Workspaces
App attach
Scaling plans
Users
Custom image templates
Monitoring
Licensing
Harsh, create a host pool!
Easily scale your VM deployment. Create host pools to easily manage assignments, application groups, and settings for your entire organization.
Create a host pool
Help and learning
Product documentation
Learn about the capabilities of Azure Virtual Desktop
Create your image
Learn about creating custom images and using gallery images
Cost calculator
Plan and estimate the cost for your deployment
Profile containers
Learn how to use FSLogix profile containers for user profiles

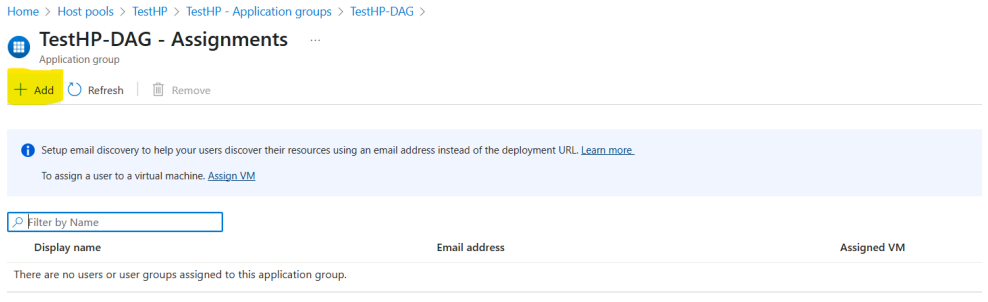

Home
Host pools
Default Directory
Create
Manage view
Refresh
Export to CSV
Open query
Assign tags
Delete
A host pool is a collection of Azure virtual machines that are registered to Azure Virtual Desktop as session hosts. A host pool can be one of two types: Personal or Pooled. Click here to learn more.
Filter for any field...
Subscription equals all
Resource Group equals all
Location equals all
Add filter
Name
Resource Group
Location
Subscription
Host pool type
Load balancer type
Application groups
HarshHP harshrg East Asia Azure subscription 1 Pooled Breadth-first 1
HarshHPP harshrgg Central India Azure subscription 1 Pooled Breadth-first 1
TestHP TestRG Central India Azure subscription 1 Personal Persistent 1

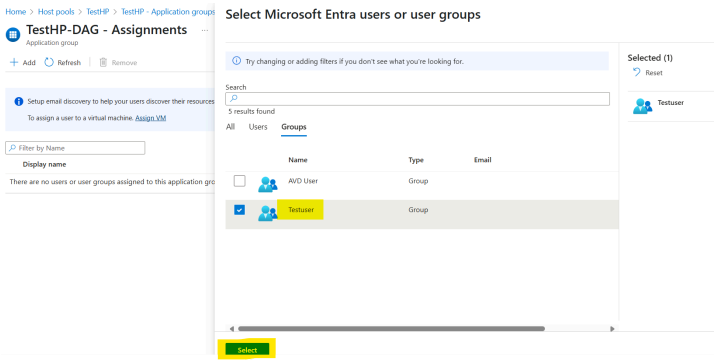

TestHP
Host pool
Search
Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Resource visualizer
Settings
Scaling plan
RDP Properties
Properties
Networking
Scheduled agent updates
Locks
Manage
Application groups
Registration key
Refresh
Delete
Start
Restart
Stop
Azure subscription 1
Subscription ID
0c3bb37f-98c6-42f6-96c7-7bb5d7e34f15
No
OS disk type
Standard SSD
Tags (edit)
Add tags
Virtual machines
Total machines 1
Can connect 1
Can't connect 0
Active sessions 0
Disconnected sessions 0
Pending sessions 0
Total sessions 0
Applications
Application groups 1
Applications 1


Home > Host pools > TestHP >
TestHP - Application groups
Add
Refresh
Remove
Name
Friendly name
Application group type
Applications
Assignments
TestHP-DAG
Default Desktop
Desktop
1
0


Home > Azure Virtual Desktop | Host pools > HarshHP > HarshHP - Application groups >
HarshAG
Application group
Search
Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Resource visualizer
Settings
Manage
Monitoring
Automation
Help
Delete
Essentials
JSON View
Resource group (move) : harshrg
Location : East Asia
Subscription (move) : Azure subscription 1
Subscription ID : 0c3bb37f-98c6-42f6-96c7-7bb5d7e34f15
Tags (edit) : Add tags
Application group type : Desktop
Friendly name : Harsh Desktop App Group
Description :
Host pool : HarshHP
Workspace :
1 Applications (manage)
0 Assignments (manage)

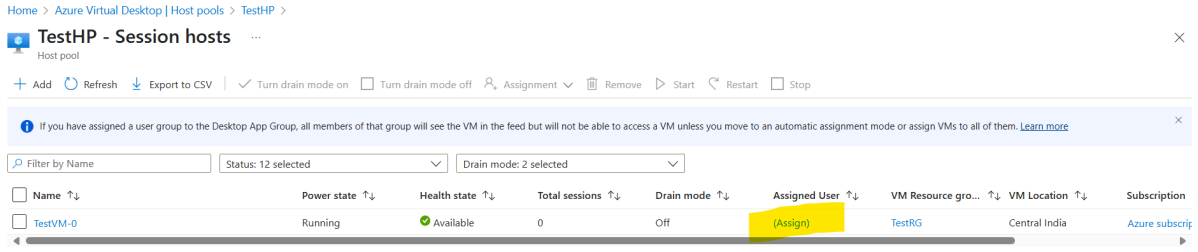

Home > Host pools > TestHP > TestHP - Application groups >
TestHP-DAG
Application group
Search
Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Resource visualizer
Settings
Manage
Monitoring
Automation
Help
Delete
Essentials
JSON View
Resource group (move) : TestRG
Location : Central India
Subscription (move) : Azure subscription 1
Subscription ID : 0c3bb37f-98c6-42f6-96c7-7bb5d7e34f15
Tags (edit) : cm-res... : /subscriptions/0c3bb37f-98c6-42f6-96c7-7bb5d7e34f15/...
Application group type : Desktop
Friendly name : Default Desktop
Description : Desktop Application Group created through the Hostpool Wizard
Host pool : TestHP
Workspace : TestWP
1 Applications (manage)
0 Assignments (manage)

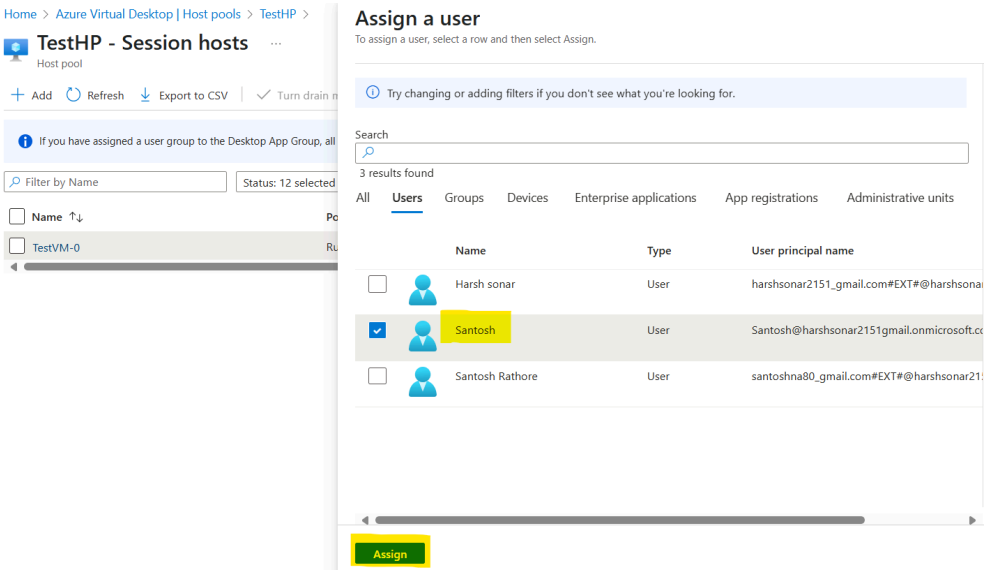

Home > Host pools > TestHP > TestHP - Application groups > TestHP-DAG >
TestHP-DAG - Assignments
Application group
Add
Refresh
Remove
Setup email discovery to help your users discover their resources using an email address instead of the deployment URL. Learn more
To assign a user to a virtual machine. Assign VM
Filter by Name
Display name
Email address
Assigned VM
There are no users or user groups assigned to this application group.



Assign personal virtual machine to user

Go to host pool and select the virtual machine click on assign





Now, check your virtual machine (VDI) access by clicking below link-

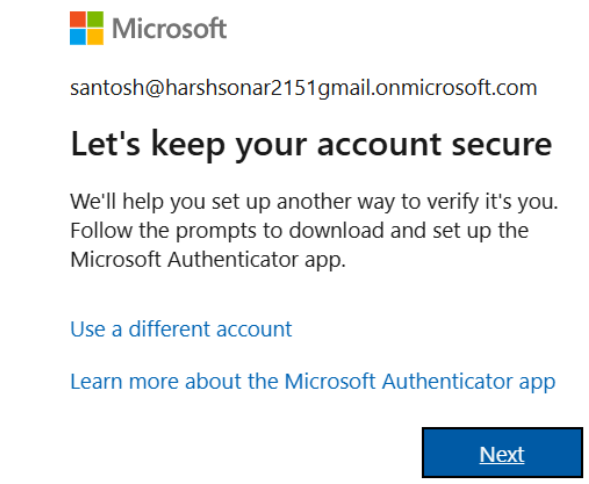
https://windows365.microsoft.com/ent#/devices

OR

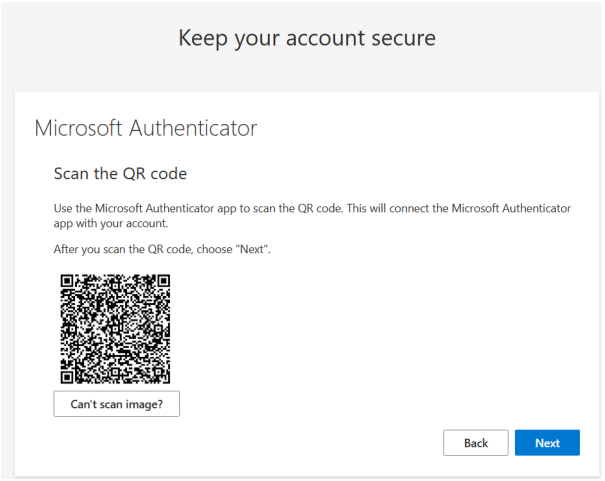
You can download remote desktop app from below link-

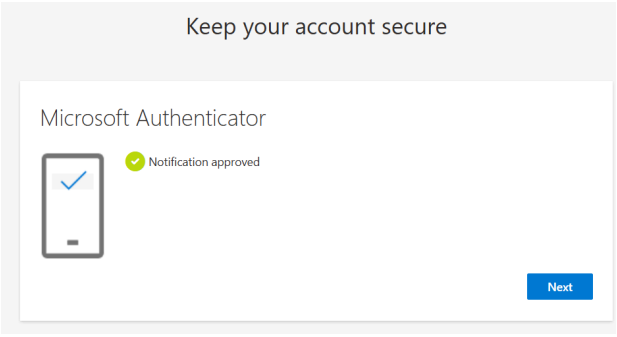
https://learn.microsoft.com/en-us/previous-versions/remote-desktop-client/connect-windows-cloud-services?tabs=windows-msrdc-msi

Authenticate with your credentials .

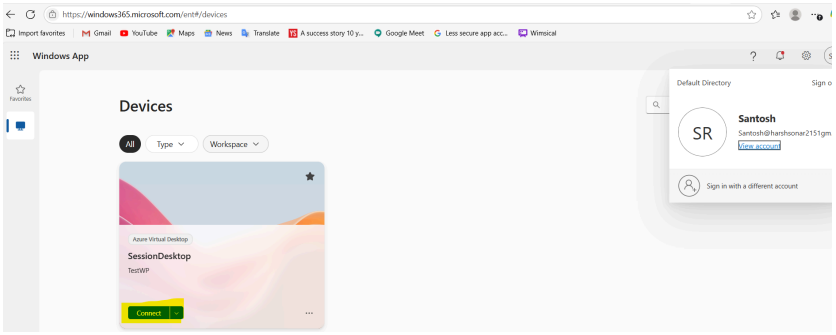


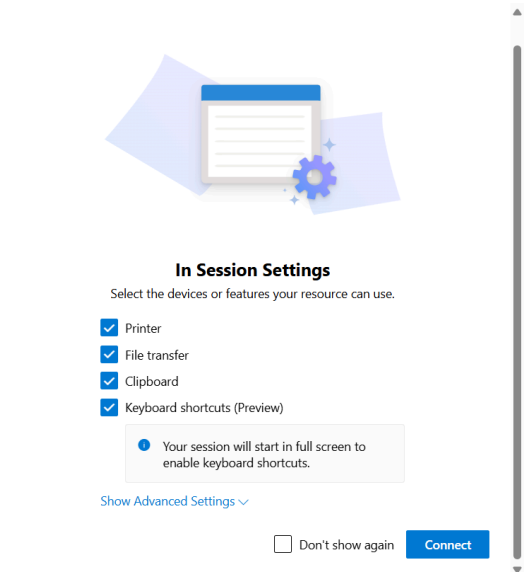
after completing above step you will get Microsoft authenticator QR code to authenticator (you can download Microsoft authenticator app from play store).

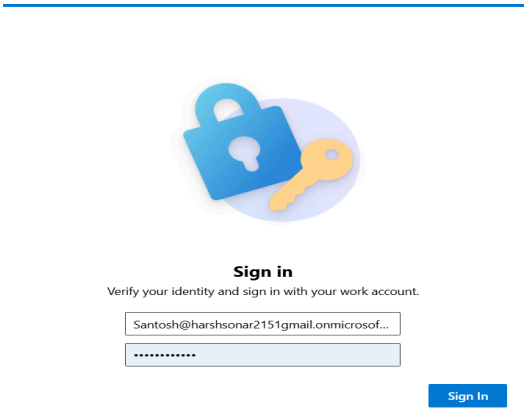




after successful authentication you can able to see your VDI to connect.


Windows App
Devices
All
Type
Workspace
SessionDesktop
TestWP
Connect
Default Directory
SR
Santosh
Sign in with a different account


In Session Settings
Select the devices or features your resource can use.
Printer
File transfer
Clipboard
Keyboard shortcuts (Preview)
Your session will start in full screen to enable keyboard shortcuts.
Show Advanced Settings
Don't show again
Connect


Sign in
Verify your identity and sign in with your work account.
Santosh@harshsonar2151gmail.onmicrosof...
Sign In

Tasks: we have to implement below mentioned tasks to make environment more scalable

1.Setup scaling plan for pooled host pool (upscale and downscale virtual machines)

Setting up a **scaling plan for a pooled host pool** in **Azure Virtual Desktop (AVD)** using the **Azure Portal** — especially with a **free subscription** — has some limitations. Free subscriptions (like the Azure Free Account) often have constraints in terms of available resources, quotas, and cost-free limits. However, I'll walk you through the steps to set up a **scaling plan**, and I'll highlight what's possible within a free subscription.

✅ Prerequisites

1. **Azure Free Subscription**: Make sure it's active and you've not exhausted the $200 credit or 12-month limits.
2. **Host Pool**: A **pooled host pool** should already be deployed.
3. **Session Hosts**: At least one VM joined to the host pool and domain (can be Azure AD Join).
4. **Azure Virtual Desktop Insight Workspace**: Required for scaling plan monitoring.

⚙️ Step-by-Step: Create Scaling Plan for Pooled Host Pool

1. Log in to Azure Portal

- Navigate to https://portal.azure.com

---

2. Go to "Azure Virtual Desktop"

- In the search bar, type **"Azure Virtual Desktop"**
- Select **"Scaling plans"** from the left menu.

---

3. Create a Scaling Plan

Click **+ Create** and follow these steps:

Basics

- **Subscription**: Select your free subscription
- **Resource Group**: Use existing or create new (e.g., `AVD-RG`)
- **Name**: e.g., `PooledScalingPlan`
- **Location**: Choose region where your host pool is deployed
- **Time zone**: Set appropriate time zone for scaling triggers

Schedule

You define working hours and off-hours logic:

1. Click **+ Add schedule**:
    - **Name**: e.g., `Weekday Schedule`
    - **Days**: Mon-Fri
    - **Start time**: 8:00 AM
    - **Ramp-up load balancing**: Breadth-first / Depth-first
    - **Minimum host percentage**: e.g., 25%
    - **Capacity threshold**: e.g., 60% usage
    - **Ramp-down time**: 6:00 PM
    - **Log off users after**: e.g., 15 minutes
    - **Force logoff**: Choose Yes/No
2. Repeat for **weekend** or non-working hours.

Assignments

- **Assign to host pools**: Select your **pooled host pool**
- Choose **custom or predefined schedules**

---

4. Review + Create

- Review all settings and click **Create**

⚠️ Important Notes for Free Subscription

1. **Free tier VM limits**:
    - You may only be able to run small VMs (like B1s, which aren't ideal for AVD).
    - Auto-scaling requires **at least 2 VMs** to show effect.
    - Free subscription limits make scaling more of a *demo* than production.
2. **Log Analytics**:
    - Scaling plans use **Azure Monitor Logs**.
    - Enabling diagnostics may incur cost after the free limit (5 GB/month).
3. **Azure AD Join + Intune**: Use for simplified session host domain join to avoid setting up full AD or domain controllers.

🧪 Testing the Scaling Plan

- Log into one session host, simulate load by connecting multiple sessions.
- Verify host pool scaling activity in Azure Monitor > Logs.
- Try reducing session host usage outside schedule to see auto-shutdown.

📘 Tips

- Use **Auto-shutdown** feature in VM settings to avoid cost if scaling is not fully functional.
- Monitor usage regularly via **Cost Management** to stay within free limits.

Refer- https://youtu.be/L9j_pSBVGy4?si=PALsd46c3K7m34H2

🧰 1. PowerShell Script to Create a Scaling Plan

🔧 Prerequisites:

- Az PowerShell Module

Azure Virtual Desktop PowerShell module:

powershell
CopyEdit

```
Install-Module -Name Az.DesktopVirtualization -Force
```

-

✅ PowerShell Script:

powershell

CopyEdit

```
# Login to Azure

Connect-AzAccount


# Variables

$subscriptionId = "<your-subscription-id>"

$resourceGroupName = "AVD-RG"

$location = "eastus"

$hostPoolName = "MyPooledHostPool"

$scalingPlanName = "MyScalingPlan"


# Select subscription

Set-AzContext -SubscriptionId $subscriptionId


# Create Scaling Plan

$scalingPlan = New-AzWvdScalingPlan `

    -ResourceGroupName $resourceGroupName `
```

```
    -Name $scalingPlanName `
    -Location $location `
    -TimeZone "Eastern Standard Time" `
    -ExclusionTag "DoNotScale" `
    -ScheduleList @(
        New-AzWvdScalingSchedule `
            -Name "WeekdaySchedule" `
            -DaysOfWeek Monday,Tuesday,Wednesday,Thursday,Friday `
            -RampUpStartTime "08:00" `
            -RampUpLoadBalancingAlgorithm "BreadthFirst" `
            -RampUpMinimumHostsPercentage 25 `
            -RampUpCapacityThresholdPercentage 60 `
            -PeakStartTime "09:00" `
            -RampDownStartTime "18:00" `
            -RampDownMinimumHostsPercentage 10 `
            -RampDownForceLogoffUsers False `
            -RampDownNotificationMessage "Sessions will log off soon
due to scale down." `
            -RampDownWaitTimeMinutes 15 `
            -OffPeakStartTime "20:00" `
            -OffPeakMinimumHostsPercentage 5
    )

# Link Scaling Plan to Host Pool
Add-AzWvdScalingPlanHostPoolReference `
    -ResourceGroupName $resourceGroupName `
    -ScalingPlanName $scalingPlanName `
    -HostPoolReference @(@{
```

```
        HostPoolArmPath =
"/subscriptions/$subscriptionId/resourceGroups/$resourceGroupName/pr
oviders/Microsoft.DesktopVirtualization/hostPools/$hostPoolName"
        ScalingPlanEnabled = $true
    })

Write-Host "Scaling Plan '$scalingPlanName' created and linked to
host pool '$hostPoolName'."
```

💡 Replace `<your-subscription-id>` and other values as per your environment.

🧱 2. ARM Template to Create a Scaling Plan

You can deploy this using Azure CLI, PowerShell, or directly in the Azure Portal.

📄 `scalingPlan.json`:

json

CopyEdit

```
{
  "$schema":
"https://schema.management.azure.com/schemas/2019-04-01/deploymentTe
mplate.json#",
  "contentVersion": "1.0.0.0",
  "parameters": {
    "scalingPlanName": {
      "type": "string"
    },
    "location": {
      "type": "string"
    },
    "hostPoolResourceId": {
      "type": "string"
```

```
    }
  },
  "resources": [
    {
      "type": "Microsoft.DesktopVirtualization/scalingPlans",
      "apiVersion": "2022-02-10-preview",
      "name": "[parameters('scalingPlanName')]",
      "location": "[parameters('location')]",
      "properties": {
        "timeZone": "Eastern Standard Time",
        "exclusionTag": "DoNotScale",
        "hostPoolReferences": [
          {
            "hostPoolArmPath": "[parameters('hostPoolResourceId')]",
            "scalingPlanEnabled": true
          }
        ],
        "schedules": [
          {
            "name": "WeekdaySchedule",
            "daysOfWeek": ["Monday", "Tuesday", "Wednesday",
"Thursday", "Friday"],
            "rampUpStartTime": "08:00",
            "rampUpLoadBalancingAlgorithm": "BreadthFirst",
            "rampUpMinimumHostsPercentage": 25,
            "rampUpCapacityThresholdPercentage": 60,
            "peakStartTime": "09:00",
            "rampDownStartTime": "18:00",
            "rampDownMinimumHostsPercentage": 10,
            "rampDownForceLogoffUsers": false,
```

```
            "rampDownNotificationMessage": "You will be logged off
soon for scaling down.",
            "rampDownWaitTimeMinutes": 15,
            "offPeakStartTime": "20:00",
            "offPeakMinimumHostsPercentage": 5
          }
        ]
      }
    }
  ]
}
```

🚀 How to Deploy the ARM Template

Using Azure CLI:
bash

CopyEdit

```
az deployment group create \
  --name avd-scaling-deploy \
  --resource-group AVD-RG \
  --template-file scalingPlan.json \
  --parameters scalingPlanName=MyScalingPlan \
               location=eastus \

hostPoolResourceId="/subscriptions/<sub-id>/resourceGroups/AVD-RG/pr
oviders/Microsoft.DesktopVirtualization/hostPools/MyPooledHostPool"
```

2.Publish Remote app

Publishing a **RemoteApp** in **Azure Virtual Desktop (AVD)** and assigning it to a **user** using a **free Azure subscription** is possible — as long as you stay within your usage limits. Here's a **step-by-step guide** to help you publish a RemoteApp (like Notepad) and make it available to a user.

🧰 Prerequisites

1. **Azure Free Subscription** (active)
2. A **pooled or personal host pool** with:
    - At least 1 running Windows VM (session host)
    - Proper user permissions
    - Azure AD Join or AD DS setup
3. The application (e.g., `notepad.exe`) must be installed on the session host.

✅ Step-by-Step: Publish RemoteApp and Assign to User

Step 1: Connect to Your Session Host

1. Go to **Azure Portal** > **Virtual Machines**
2. Connect via **RDP** to your session host
3. Verify the app (e.g., Notepad) exists at a known path (e.g., `C:\Windows\System32\notepad.exe`)

Step 2: Go to Azure Virtual Desktop

1. In Azure Portal, search for **Azure Virtual Desktop**
2. Navigate to **Application Groups**
3. Find the **RemoteApp application group** linked to your host pool
    - It usually ends in `-RemoteApp` (or create one if it doesn't exist)

Step 3: Add RemoteApp

1. Click on the application group (e.g., `MyHostPool-RemoteApp`)
2. Under **Applications**, click **+ Add**
3. Fill in details:
    - **App name**: Notepad
    - **File path**: `C:\Windows\System32\notepad.exe`
    - **Icon path**: (Optional)
    - **Command-line arguments**: (Leave empty unless needed)

✅ Click **Save**

Step 4: Assign User

1. In the same RemoteApp group, go to **Assignments**
2. Click **+ Add**
3. Select the user(s) you want to assign this RemoteApp to
    - The user must be in **Azure AD** and have permissions to log in

---

Step 5: Give AVD Access

Make sure the assigned user is:

- **Assigned to the desktop or RemoteApp group**
- **Given RBAC access** (optional, for visibility)
- Licensed for AVD (Free subscriptions support AVD with eligible licenses — for test, this is usually sufficient if you log in as the same account)

Step 6: User Access via Web or Client

1. Go to https://rdweb.wvd.microsoft.com/arm/webclient
2. Sign in with the assigned user's account
3. Launch the **RemoteApp (e.g., Notepad)** from the browser or **Remote Desktop Client**

🧪 Optional PowerShell for Publishing

powershell

CopyEdit

```
Install-Module -Name Az.DesktopVirtualization -Force

# Connect to Azure
Connect-AzAccount

# Add Notepad to RemoteApp group
New-AzWvdApplication `
  -ResourceGroupName "AVD-RG" `
  -ApplicationGroupName "MyHostPool-RemoteApp" `
  -Name "Notepad" `
  -FilePath "C:\Windows\System32\notepad.exe" `
  -FriendlyName "Notepad" `
  -ShowInPortal $true
```

📝 Notes on Free Subscription

Feature	Status
RemoteApp Support	✅ Available
Azure AD Join (no AD DS)	✅ Recommended for test use

AVD Licensing	✅ Covered under eligible M365 plans (trial OK)
Limits	❗ B-series VM may throttle; session limit is low
Cost	⚠ RemoteApp may still incur compute + storage

PowerShell automation script to help you publish **multiple RemoteApps** to an **Azure Virtual Desktop (AVD) RemoteApp Application Group**, ideal for setting up quickly in test or production — even with a **free Azure subscription**.

🎯 What This Script Does

- Connects to your Azure subscription
- Publishes multiple RemoteApps to a RemoteApp Application Group
- Assigns them to a user

📦 Requirements

- PowerShell 5.x+ with **Az.DesktopVirtualization** module installed:

powershell

CopyEdit

```
Install-Module -Name Az.DesktopVirtualization -Force
```

- A working **AVD host pool** with a linked **RemoteApp group**
- Session host VMs with the apps installed
- Azure AD users

🧠 Example Apps to Publish

We'll publish common Windows tools like:

- Notepad
- Paint
- Calculator
- WordPad

🚀 PowerShell Script: Publish Multiple RemoteApps

powershell

CopyEdit

```
# Login to Azure
Connect-AzAccount

# Set variables
$resourceGroup = "AVD-RG"
$applicationGroup = "MyHostPool-RemoteApp"  # Your RemoteApp app group name
$apps = @(
    @{ Name = "Notepad";     Path = "C:\Windows\System32\notepad.exe";     FriendlyName = "Notepad" },
    @{ Name = "Paint";       Path = "C:\Windows\System32\mspaint.exe";      FriendlyName = "Paint" },
    @{ Name = "Calculator";  Path = "C:\Windows\System32\calc.exe"; FriendlyName = "Calculator" },
    @{ Name = "WordPad";     Path = "C:\Program Files\Windows NT\Accessories\wordpad.exe"; FriendlyName = "WordPad" }
)

# Loop and create applications
foreach ($app in $apps) {
```

```
    try {
        Write-Host "Publishing RemoteApp: $($app.Name)"
        New-AzWvdApplication `
            -ResourceGroupName $resourceGroup `
            -ApplicationGroupName $applicationGroup `
            -Name $app.Name `
            -FilePath $app.Path `
            -FriendlyName $app.FriendlyName `
            -ShowInPortal $true

        Write-Host "✓ Published: $($app.FriendlyName)"
-ForegroundColor Green
    }
    catch {
        Write-Warning "⚠ Failed to publish $($app.Name): $_"
    }
}

# (Optional) Assign a user to the RemoteApp group
# Replace with your Azure AD user principal name (email)
$userPrincipal = "your-user@yourdomain.com"
Write-Host "Assigning user $userPrincipal to app group..."

Add-AzWvdApplicationGroupUser `
    -ResourceGroupName $resourceGroup `
    -ApplicationGroupName $applicationGroup `
    -UserPrincipalName $userPrincipal

Write-Host "✅ RemoteApps published and user assigned!"
-ForegroundColor Cyan
```

📝 Notes

- **File paths must be valid on the session hosts.**
    - Use RDP to verify if you're unsure.
- You can extend this by:
    - Reading from a CSV of apps
    - Adding command-line args
    - Auto-detecting installed apps (more advanced)
- **Permissions**: Ensure the user is allowed to log in to the host pool.

🧪 Sample Output

sql

CopyEdit

```
Publishing RemoteApp: Notepad
✓ Published: Notepad
Publishing RemoteApp: Paint
✓ Published: Paint
Assigning user your-user@yourdomain.com to app group...
✅ RemoteApps published and user assigned!
```

3.Fslogix profile storage

Refer- https://www.youtube.com/playlist?list=PLYQGZ_g5WcelFhhkKLGfxZnHCZBSbZoOX

Setting up **FSLogix profile storage** in Azure for **Azure Virtual Desktop (AVD)** using a **free Azure subscription** is possible — but you need to optimize carefully to **stay within your free quota**. FSLogix uses a central file share (usually an **Azure Files share**) to store user profiles, which are mounted during AVD login as VHD(X) files.

✅ Overview: FSLogix Profile Storage with Free Azure Subscription

Component	Free?	Notes
FSLogix	✅ Free with AVD-eligible licenses (e.g., M365 E3/E5)	
Azure Files	⚠️ Limited	Can be used within free storage quota (5 GB/month LRS in some tiers)
AVD VMs	✅	Use free credits or B-series for low cost
Windows License	✅	M365 E3/E5 required (not included in Azure credits)

🚀 Step-by-Step: FSLogix Profile Storage Setup

🔹 Step 1: Create a Storage Account

1. Go to **Azure Portal > Storage accounts > Create**
2. Use these settings:
    - **Name**: e.g., `fslogixstorage123`
    - **Region**: Same as your AVD VM
    - **Performance**: **Standard**
    - **Redundancy**: **Locally-redundant storage (LRS)**
    - **Tier**: **Hot**
3. Click **Review + Create**

💡 LRS is cheaper and fine for FSLogix in dev/test.

🔹 Step 2: Create a File Share for FSLogix

1. Go to your Storage Account > **File shares**
2. Click **+ File Share**
    - **Name**: `profiles`
    - **Quota**: Start small (e.g., 10 GB)
3. Click **Create**

🔹 Step 3: Set NTFS-Style Permissions

FSLogix requires **NTFS permissions**, not just Azure RBAC.

Option 1: Join storage account to Active Directory (Recommended)

- Go to **Storage Account > Configuration**
- Set **Active Directory (AD DS)** integration (requires domain-joined environment)
- Configure with your domain join settings

⚠️ This needs an on-prem DC or **Azure AD DS** (not free).

Option 2: Use Azure File Sync (for dev/testing only)

- If you don't have domain join capabilities, use **Azure File Sync** and a small VM as an intermediary for setting NTFS permissions.

🔹 Step 4: Configure AVD VMs for FSLogix

1. On your AVD session hosts, install **FSLogix**:
    - Download: https://www.microsoft.com/en-us/download/details.aspx?id=100607
2. Create or update the following registry settings:

reg

CopyEdit

```
[HKEY_LOCAL_MACHINE\SOFTWARE\FSLogix\Profiles]
```

```
"Enabled"=dword:00000001
"VHDLocations"="\\fslogixstorage123.file.core.windows.net\profiles"
"SizeInMBs"=dword:00020000 ; (optional, sets max size)
"VolumeType"="VHDX"
```

> You may also use a **Group Policy template** to set these.

🔹 Step 5: Map Network Drive in VM and Test

Use PowerShell or File Explorer on the AVD VM to test:

powershell

CopyEdit

```
net use Z: \\fslogixstorage123.file.core.windows.net\profiles /user:Azure\<StorageAccountKey>
```

You can find the **access key** in the Storage Account > **Access keys**.

🔹 Step 6: Log In with AVD User and Check Profile

When the user logs in, FSLogix will mount a profile container (e.g., `user_sid.vhdx`) from the file share.

🧠 Tips to Stay Within Free Tier

Resource	How to Save
Storage	Use **LRS**, and limit quota of file share
VMs	Use **B1ms/B2s** VMs within 750-hour free CPU time
Bandwidth	Avoid unnecessary data movement across regions
Licensing	Make sure you use M365 E3/E5 or eligible licenses to legally use FSLogix

---

✅ Summary

Yes, you can set up **FSLogix profile storage** using a free Azure subscription by:

- Creating a **Storage Account + File Share**
- Configuring **NTFS permissions**
- Installing FSLogix on session hosts
- Pointing FSLogix to the share via registry or GPO

set up a **Group Policy Object (GPO)** to configure **FSLogix profiles** for your Azure Virtual Desktop (AVD) environment, using the official **FSLogix ADMX templates**.

🧾 What You Need

- A **domain-joined environment** (either on-prem AD or Azure AD DS)
- FSLogix installed on session host VMs
- Group Policy Management Console (GPMC) available on your management machine

---

📁 Step 1: Download FSLogix ADMX Templates

Download from Microsoft:
🔗 FSLogix Download Page

Once downloaded:

- Extract the ZIP file
- Navigate to `fslogix.admx`
- Copy `fslogix.admx` to `C:\Windows\PolicyDefinitions`
- Copy corresponding language file (e.g., `fslogix.adml`) to `C:\Windows\PolicyDefinitions\en-US`

If you're using a central store (`\\domain\SYSVOL\domain\Policies\PolicyDefinitions`), copy the files there instead.

📘 Step 2: Create a GPO for FSLogix

1. Open **Group Policy Management Console (GPMC)**
2. Right-click your **OU or domain**, choose **Create a GPO** (e.g., `FSLogix_Profile_Config`)
3. Right-click the new GPO → **Edit**

⚙️ Step 3: Configure FSLogix Settings via GPO

Navigate to:

yaml

CopyEdit

```
Computer Configuration >
 Administrative Templates >
 FSLogix >
 Profile Containers
```

Configure the following:

Setting Name	Value
Enabled	Enabled
VHD Location	`\\<storageaccount>.file.core.windows.net\profiles`
Size in MBs (optional)	e.g., `30720` (for 30 GB max size)
Volume Type	`VHDX`

Flip Flop Profile Directory Name	Enabled (recommended)
Delete local profile when VHD should apply	Enabled

You can also configure **redirection**, **cloud cache**, and **profile versioning** if needed.

📥 Example Registry Equivalent (For Reference)

If you'd prefer to push this via script or registry, here's the equivalent:

r

CopyEdit

```
[HKEY_LOCAL_MACHINE\SOFTWARE\FSLogix\Profiles]
"Enabled"=dword:00000001
"VHDLocations"="\\fslogixstorage123.file.core.windows.net\profiles"
"SizeInMBs"=dword:00007800 ; 30 GB
"VolumeType"="VHDX"
"DeleteLocalProfileWhenVHDShouldApply"=dword:00000001
```

✅ Final Step: Link GPO and Reboot VM

- Link the GPO to the OU where your **AVD session host VMs** are located.
- Reboot the VMs or run `gpupdate /force`.

🧠 Tips

- Make sure the **AVD host pool** VMs can access the file share with appropriate NTFS permissions.

- You can monitor profile mounting via the event log:
  `Applications and Services Logs > FSLogix > Operational`

4.App masking

App Masking in Azure Virtual Desktop (AVD) is typically achieved using **FSLogix App Masking**, a feature of **FSLogix**—a Microsoft product included with many Windows licenses (like Microsoft 365 E3/E5, Windows 10/11 Enterprise multi-session, etc.). However, with a **free Azure subscription**, you must be mindful of the **limited resources** and **licensing constraints**.

🧩 What is App Masking?

App Masking hides applications from users based on **Active Directory (AD) group membership** or other criteria, without uninstalling them. This is useful in multi-user environments like AVD to manage who sees what apps.

🧾 Requirements for App Masking in AVD

1. **Azure Virtual Desktop Environment:**
   - Azure AD-joined or hybrid-joined session host(s)
   - Host pool(s), application group(s), etc.
2. **FSLogix Installed on Session Hosts:**
   - Comes with AVD entitlement if you're using eligible licenses.
3. **Group Policy or registry for FSLogix configuration**
4. **Active Directory Integration (Azure AD DS or AD DS synced with Azure AD):**
   - Needed to define user groups for App Masking rules.

💡 Using App Masking in a Free Azure Subscription

Here's how you can **implement App Masking in AVD with limited/free Azure credits**:

✅ Step 1: Set Up AVD in Free Tier

- Use the **Azure free account** to create:
    - One Windows 10/11 multi-session VM
    - A basic AVD host pool
    - A small managed disk (64 GB SSD recommended)
    - Use **B-series or D-series VMs** (within 750 hours/month free quota)

✅ Step 2: Set Up Active Directory

App Masking requires users and groups:

- Use **Azure AD DS (costs apply)** or
- Create a **Domain Controller VM (Windows Server with AD DS role)** using minimal resources

✅ Step 3: Install FSLogix

- Download FSLogix from Microsoft Download Center
- Install on the AVD session host

✅ Step 4: Create App Masking Rules

- Install FSLogix Rule Editor (part of FSLogix)
- Use the **FSLogix Rule Editor** to:
    - Create an **App Masking Rule Set (.fxr)**
    - Specify which apps to hide and from whom
    - Save rules in a shared location (e.g., `\\YourFS01\FSLogixRules`)

✅ Step 5: Apply Masking Rules

- Deploy rules to session hosts (via Group Policy or copy manually)
- Ensure FSLogix service is running and rules are applied at user login

---

💵 Cost Considerations

Component	Free?	Notes
Azure Virtual Desktop	✅	Free service; pay for VM, storage, etc.
Windows 10/11 Multisession	✅	Requires eligible license (M365 E3/E5, etc.)
FSLogix	✅	Free with eligible license
Azure AD DS	❌	Not free; consider domain controller VM instead
VMs (AVD hosts)	✅ (limited)	Use low-tier VMs within free limits

---

🔒 Licensing Reminder

FSLogix and AVD App Masking are only **licensed for use with certain Microsoft subscriptions**, such as:

- Microsoft 365 E3/E5
- Windows 10/11 Enterprise with Software Assurance

If you're using only a free Azure account with no licenses, you may not be fully compliant.

🧠 Summary

You **can implement basic App Masking** with a free Azure account by:

- Creating a small AVD deployment
- Installing FSLogix and setting up App Masking rules
- Using a simple AD setup for user/group control

However, be careful of:

- Resource and cost limits
- Licensing compliance for FSLogix

5.MSIX app attach and app deployment

Deploying applications using **MSIX App Attach in Azure Virtual Desktop (AVD)** is fully possible with a **free Azure subscription**, but it requires careful resource and cost management, as well as compliance with prerequisites. Here's a step-by-step guide tailored to a **free-tier** deployment.

✅ What is MSIX App Attach?

MSIX App Attach is a way to dynamically deliver applications to users in AVD without installing them on the system. Apps are packaged as **MSIX**, then attached (mounted) to the session host VM when the user logs in.

🧾 Prerequisites for MSIX App Attach

Requirement	Free Tier Feasibility	Notes
AVD Host Pool	✅	Create using Azure free credits
Windows 10/11 Enterprise multi-session VM	✅	Needed for AVD
MSIX Package	✅	You can create your own using MSIX Packaging Tool
Azure Files / File Share (SMB)	✅ (with limits)	Use for storing MSIX VHD/X files
FSLogix (optional)	✅	Recommended for user profile management
PowerShell / Azure CLI	✅	Used for attaching apps

🚀 Step-by-Step: Deploy MSIX App Attach in AVD on a Free Azure Subscription

1. Set up AVD (Host Pool & VM)

- Use Azure portal or PowerShell to create:
    - A **host pool** with at least one session host (Windows 10/11 multisession)
    - An **application group**
    - Assign users from Azure AD

💡 Use B2s VM to stay within free tier CPU/memory limits.

---

2. Create or Get an MSIX App Package

- Use Microsoft's **MSIX Packaging Tool** to convert a traditional installer (like `.exe` or `.msi`) into an `.msix` package.
- Convert `.msix` to `.vhd` or `.vhdx`:
    - You can use the MSIX ToolKit or manual PowerShell methods.

powershell

CopyEdit

```
Expand-MSIXPackage -PackagePath "C:\App\App.msix" -DestinationPath "C:\ExpandedApp"
```

```
Convert-MSIXToVHD -AppPath "C:\ExpandedApp" -DestinationPath "C:\App\App.vhdx"
```

---

3. Upload MSIX VHD/VHDX to Azure Files

- Create an **Azure Files share** in a **Storage Account**.
- Upload the `.vhdx` file to a new container (e.g., `msixapps`).

- Configure NTFS permissions for AVD session host(s) to access this share.

🔐 Use **private endpoints or secure access** if you're deploying in production.

---

4. Register and Add MSIX Package in AVD

Use **PowerShell** or **Azure CLI** to register the package:

powershell

CopyEdit

```
Add-AzWvdMsixPackage -ResourceGroupName "<your-rg>" `
  -HostPoolName "<your-hostpool>" `
  -ImagePath
"\\<storageaccount>.file.core.windows.net\<share>\<app>.vhdx" `
  -PackageAlias "MyApp" `
  -DisplayName "MyApp" `
  -IsActive $true `
  -IsRegularRegistration $true
```

---

5. Add MSIX App to Application Group

In the Azure Portal:

- Go to **AVD > Application Groups > [Your App Group] > Applications**
- Click **Add MSIX app**
- Select your **MSIX package**
- Assign it to users

---

6. Test the Deployment

- Connect to AVD via the Remote Desktop client or web client.

- Login with a user assigned to the application group.
- Verify that the app appears as expected.

⚠️ Tips for Free Tier Use

Area	Recommendation
VM Sizing	Use `B2s` or similar small instances
Storage	Keep file shares minimal; use `Standard` tier
Monitoring	Use manual checks; avoid enabling diagnostic logs unless needed
Licensing	Ensure you have a valid Microsoft 365 or Windows license (E3/E5/Business Premium)

✅ Summary

You **can** deploy apps using **MSIX App Attach in AVD** under a free Azure subscription by:

1. Creating a basic AVD environment
2. Packaging your app as MSIX ⟶ VHDX
3. Hosting it in an Azure Files share
4. Registering and assigning it via Azure

6.Full Patching process

detailed and structured process for fully patching Azure Virtual Desktop (AVD) session hosts using a **golden image** and **Azure Compute Gallery (ACG)**, incorporating:

- 🔒 **Snapshot of the old image** (for rollback)
- 🛠️ **Updating patches and installing software**
- 💾 **Creating a new image version**

- 📦 **Deploying the updated image to session hosts**

🛠️ Full AVD Patching Workflow with Snapshot + Azure Compute Gallery

---

📋 Step-by-Step Process

✅ 1. Identify the Master (Golden) VM

This is your base VM used to create session hosts.

- It should:
    - Use a supported OS (Windows 10/11 Enterprise multi-session or Server)
    - Not currently be used for production logins
    - Be in a **Stopped (deallocated)** state before taking a snapshot

---

📸 2. Take a Snapshot of the Existing OS Disk

Before making any changes, create a snapshot of the VM's disk to **safeguard against rollback**.

🔧 **PowerShell Example**
powershell

CopyEdit

```
$rgName = "AVD-ResourceGroup"
$vmName = "AVD-Master-VM"

# Get VM details
$vm = Get-AzVM -Name $vmName -ResourceGroupName $rgName

# Create snapshot config
$snapshotConfig = New-AzSnapshotConfig `
```

```
  -SourceUri $vm.StorageProfile.OsDisk.ManagedDisk.Id `
  -Location $vm.Location `
  -CreateOption Copy

# Create snapshot
New-AzSnapshot -Snapshot $snapshotConfig -SnapshotName
"$vmName-Snapshot-BeforeUpdate" -ResourceGroupName $rgName
```

✅ **Now you have a rollback point.**

🔄 3. Start the VM and Apply Patches/Updates

🔧 **Windows Updates:**
powershell

CopyEdit

```
Install-WindowsUpdate -AcceptAll -AutoReboot
```

Use `sconfig` or manual Windows Update if needed.

🔧 **App Updates:**

- Install or update user-requested software (e.g., Chrome, Office, custom apps)
- Remove obsolete software or temporary files

🔧 **Optional Tweaks:**

- Update registry
- Apply GPO changes
- Optimize the VM using Microsoft's AVD optimization script

🧹 4. Clean Up Before Image Capture

Delete temp files:

powershell
CopyEdit
```
Cleanmgr /sagerun:1
```

-
- Clear event logs, browser caches, etc.
- Optionally remove local user profiles

🧼 5. Generalize the Image (Sysprep)

Run Sysprep to prepare the image for capture:

powershell

CopyEdit

```
C:\Windows\System32\Sysprep\Sysprep.exe
```

- Select: **OOBE + Generalize**
- Shutdown: **Shutdown**

🔲 The VM will shut down — don't restart it!

📦 6. Capture the Image to Azure Compute Gallery

Option A: Via Azure Portal

- Go to the VM > "Capture"
- Choose:
    - Add to existing **Azure Compute Gallery**
    - Set version (e.g., `1.0.1`)
    - Deallocate VM if prompted

Option B: PowerShell

powershell

CopyEdit

```
$imageVersionName = "1.0.1"

New-AzGalleryImageVersion `
  -ResourceGroupName $rgName `
  -GalleryName "AVDGallery" `
  -GalleryImageDefinitionName "AVD-Golden-Image" `
  -GalleryImageVersionName $imageVersionName `
  -Location $vm.Location `
  -Source $vm.Id `
  -PublishingProfileEndOfLifeDate (Get-Date).AddYears(3)
```

✅ New version is published.

🧰 7. Create New VMs (Session Hosts) from Updated Image

Option A: Manually (Small-Scale Testing)

- Use Azure portal to deploy VM or scale set from new image version
- Join to AVD host pool and domain

Option B: Automated (Production Use)

Use ARM templates, PowerShell, or Azure DevOps to deploy session hosts:

powershell

CopyEdit

```
New-AzVM `
  -Name "AVD-SessionHost-01" `
  -ResourceGroupName $rgName `
```

```
  -Image (Get-AzGalleryImageVersion -ResourceGroupName $rgName
-GalleryName "AVDGallery" -GalleryImageDefinitionName
"AVD-Golden-Image" -GalleryImageVersionName $imageVersionName) `
  -Location $vm.Location
```

📌 Use **AVD deployment scripts** to auto-join the host pool, install FSLogix, etc.

🔁 8. Update or Replace Old Session Hosts

Best Practice:

- Put old hosts in **drain mode**
- Validate new hosts with test users
- Gradually replace old hosts (or scale in)
- Delete or deallocate old hosts **after** confirming stability

---

🧠 Optional: Automate via Azure Image Builder

If you need recurring or CI/CD-like image creation, consider:

- Azure Image Builder (AIB) to automate patching
- Integration with DevOps pipelines or GitHub Actions

🧾 Summary Table

Step	Action
1	Identify and stop master VM
2	Take snapshot of OS disk
3	Start VM and patch OS & apps

4	Clean up (temp files, logs, etc.)
5	Run Sysprep (generalize)
6	Capture image and publish to Azure Compute Gallery
7	Create new VMs/session hosts from new image
8	Drain old hosts, validate, decommission if needed

📘 Recommendations

- **Keep previous image versions** in ACG for rollback or testing
- Use **tags** and **naming standards** for version tracking
- Automate using pipelines if frequent updates are required
- Test new image versions in a **staging AVD host pool**

➔ Real time example steps-

1.Taking snapshot of old image-

Go to the VM and search for golden image (e.g. deveimg) and stop the VM

Then copy the naming convention from ACG (e.g. search deveing and coly the naming convention and change the current date)

After VM stopped, Go to the settings->Disks->click on disk name-> create snapshot

Put the naming convention and choose snapshot type full and go for review and create

2.Patching process-

Update the patches on the master virtual machine and install/upgrade any software if we have a request from the user.

Move the master VM out of domain

Power off the master VM

Capture the snapshot

Create disk from the snapshot which will be a tmp disk

Create a tmp VM from the tmp disk

Perform sysprep

Capture the image

Delete the temp VM

Power on the master VM

Join the master VM back to domain.

Creation session host with latest image captured.

Got to VM-search for deveimg(for developer) image

Take RDP and check for update the machine and also check machine is not in workgroup (WG)

Then shut down the VM and azure portal VM left panel select settings-disk

Then create snapshot (AVD_azweavddeveimg_beforesysprep_24JAN2025) select full

Then create disk from snapshot- (azweavddevetmp_Osdisk01)

3.Then create VM-

Take the RDP of VM and go to file explorer-C drive->windows->system32->sysprep

Then shift + delete on panther file then double click on sysprep (generalize and shutdown)

After sysprep go to VM then create image-naming convention

7.Automation account and alerting

set up an **Azure Automation Account** and configure **VM alerting** in **Azure Virtual Desktop (AVD)** using a **free Azure subscription**, but there are some **limitations and cost considerations** you should be aware of. Here's a full breakdown.

✅ Overview

Feature	Free Subscription Support	Notes
Azure Automation Account	✅ Free up to 500 minutes/month	For running runbooks, updates, etc.
VM Alerting (via Azure Monitor)	✅ Basic alerts are free	Some features may consume credits
Log Analytics Workspace	✅ Limited free ingestion	5 GB/month free data ingestion; 31-day retention

---

🚀 Step-by-Step: Set Up Azure Automation and Alerts in AVD

Step 1: Create an Azure Automation Account

1. Go to Azure Portal ⟶ Search for **Automation Accounts**
2. Click **Create**
    - Choose a **Resource Group**
    - Name: e.g., `AVDAutoAccount`
    - Region: Choose your region
3. Enable **System-assigned managed identity** (helps with VM access)
4. Click **Review + Create**

> 💡 The free tier includes **500 minutes/month** for job runtime. Stay within that to avoid charges.

Step 2: Link to a Log Analytics Workspace

Automation can use Log Analytics for monitoring.

1. Create or use an existing **Log Analytics Workspace**
2. Go to **Automation Account > Linked Workspace**
    - Link your Log Analytics workspace

Step 3: Create Runbooks for VM Monitoring or Auto-Start/Stop

You can create PowerShell or Python runbooks. Example: Start/Stop VM based on a schedule.

Sample Runbook: Stop VM
powershell

CopyEdit

```
param (
    [string]$vmName,
    [string]$resourceGroupName
)

Stop-AzVM -Name $vmName -ResourceGroupName $resourceGroupName -Force
```

💡 Schedule the runbook using the **Schedules** blade in the Automation Account.

Step 4: Enable Diagnostics and Configure Alerts

You can create alerts for metrics like CPU usage, VM down, etc.

A. Enable Monitoring:

1. Go to **AVD VM > Monitoring > Diagnostic settings**
2. Enable and send logs/metrics to:
    - **Log Analytics Workspace**
    - (Optional) Archive to Storage or send to Event Hub

B. Create Alerts:

1. Go to **Monitor > Alerts > + New Alert Rule**
2. Scope: Select the AVD VM
3. Condition: Choose metric like "CPU Percentage > 80%"
4. Action: Email, webhook, or Automation Runbook
5. Alert Details: Name and save

📊 **Free alerts**:

- 250 metric alert rules per subscription

- 5 GB/month free ingestion in Log Analytics

✅ What You Can Do for Free:

Task	Free Tier Feasibility
Create and run Automation Runbooks	✅ up to 500 minutes/month
Set metric-based alerts (CPU, memory)	✅ up to 250 alerts
Link with Log Analytics	✅ with 5 GB/month data limit
Use action groups for emails/webhooks	✅ free

⚠️ Considerations

- Avoid enabling **guest-level diagnostics** unless necessary—they increase Log Analytics data ingestion.
- Don’t exceed **500 Automation runtime minutes/month** without watching costs.
- Always check the **Cost Management + Billing** blade to monitor usage.

🧠 Summary

Yes, with a **free Azure subscription**, you can:

- Set up an **Automation Account**
- Create **runbooks** for starting/stopping VMs
- Use **Azure Monitor alerts** to trigger actions
- Use **Log Analytics** within the free data limits

PowerShell runbook in Azure Automation to **start or stop multiple VMs** based on a **tag value**, such as `AutoStart = true` or `AutoShutdown = true`. This is ideal for scaling automation

efficiently, especially when managing multiple Azure Virtual Desktop (AVD) session hosts or regular VMs in a **free Azure subscription**.

✅ Overview

- The script looks for VMs in a specific **Resource Group**.
- It checks for a **custom tag** (e.g., `AutoShutdown = true` or `AutoStart = true`).
- It **starts or stops all matching VMs** automatically.

📘 Runbook: Multi-VM Auto Start

➕ Name: `Start-MultipleVMs-ByTag`

powershell

CopyEdit

```
param (
    [Parameter(Mandatory=$true)]
    [string]$resourceGroupName,

    [Parameter(Mandatory=$false)]
    [string]$tagName = "AutoStart",

    [Parameter(Mandatory=$false)]
    [string]$tagValue = "true"
)

# Connect to Azure
Connect-AzAccount -Identity

# Get all VMs in the resource group with matching tag
```

```
$vms = Get-AzVM -ResourceGroupName $resourceGroupName | Where-Object
{
    $_.Tags.ContainsKey($tagName) -and $_.Tags[$tagName] -eq
$tagValue
}

# Start each VM
foreach ($vm in $vms) {
    Write-Output "Starting VM: $($vm.Name)"
    Start-AzVM -ResourceGroupName $resourceGroupName -Name $vm.Name
}
```

📕 Runbook: Multi-VM Auto Stop

➕ Name: `Stop-MultipleVMs-ByTag`

powershell

CopyEdit

```
param (
    [Parameter(Mandatory=$true)]
    [string]$resourceGroupName,

    [Parameter(Mandatory=$false)]
    [string]$tagName = "AutoShutdown",

    [Parameter(Mandatory=$false)]
    [string]$tagValue = "true"
)

# Connect to Azure
```

```
Connect-AzAccount -Identity


# Get all VMs in the resource group with matching tag
$vms = Get-AzVM -ResourceGroupName $resourceGroupName | Where-Object {
    $_.Tags.ContainsKey($tagName) -and $_.Tags[$tagName] -eq $tagValue
}


# Stop each VM
foreach ($vm in $vms) {
    Write-Output "Stopping VM: $($vm.Name)"
    Stop-AzVM -ResourceGroupName $resourceGroupName -Name $vm.Name -Force
}
```

🏷️ Step: Add Tags to Your VMs

For any VM you want to automate:

1. Go to the VM → **Tags**
2. Add a tag like:
    - Key: `AutoShutdown`, Value: `true`
    - OR
    - Key: `AutoStart`, Value: `true`

🔁 Bonus: Schedule Runbooks

- Create **two schedules** in the Automation Account:
    - One for `Start-MultipleVMs-ByTag` (e.g., 8:00 AM)
    - One for `Stop-MultipleVMs-ByTag` (e.g., 7:00 PM)
- Pass the `resourceGroupName` as a **parameter** during scheduling

✅ Summary

With these tag-based runbooks, you can:

- Dynamically **control multiple VMs** with a single script
- Avoid hardcoding VM names
- Add/remove VMs from automation just by tagging them

8.Automation script, powershell, ARM templates

PowerShell-based automation script to help you manage an **Azure Virtual Desktop (AVD)** environment using the **Az module** and **AVD PowerShell cmdlets**.

✅ What This Script Does

This script will:

1. Login to Azure and set the subscription
2. Create a **resource group**
3. Create a **host pool**
4. Create a **workspace**
5. Register the host pool with the workspace
6. Add a **session host (VM)** to the host pool
7. Assign **users** to the application group

⚙️ Prerequisites

1. Install Azure PowerShell:

powershell

CopyEdit

```
Install-Module Az -Scope CurrentUser -Repository PSGallery -Force
```

```
Install-Module -Name Az.DesktopVirtualization -Scope CurrentUser
-Force
```

2. Log in:

powershell

CopyEdit

```
Connect-AzAccount
```

📝 Automation Script

powershell

CopyEdit

```
# VARIABLES – Change as needed
```

```
$resourceGroupName = "AVD-RG"
```

```
$location = "EastUS"
```

```
$hostPoolName = "AVDHostPool"
```

```
$workspaceName = "AVDWorkspace"
```

```
$vmPrefix = "AVD-VM"
```

```
$vmCount = 1
```

```
$vmSize = "Standard_B2ms"
```

```
$imageOffer = "windows-10"
```

```
$imagePublisher = "MicrosoftWindowsDesktop"

$imageSku = "win10-21h2-avd"

$adminUsername = "avdadmin"

$adminPassword = ConvertTo-SecureString "Password123!" -AsPlainText
-Force

$domainJoinUser = "yourdomain\\domainadmin"

$domainJoinPassword = ConvertTo-SecureString "YourPassword"
-AsPlainText -Force

$subscriptionId = "your-subscription-id"

$userPrincipalName = "user@yourdomain.com"  # User to assign to the
desktop application group


# LOGIN & SET SUBSCRIPTION

Connect-AzAccount

Set-AzContext -SubscriptionId $subscriptionId


# RESOURCE GROUP

New-AzResourceGroup -Name $resourceGroupName -Location $location


# HOST POOL

New-AzWvdHostPool -ResourceGroupName $resourceGroupName -Name
$hostPoolName `

    -Location $location -FriendlyName "AVD Host Pool" -HostPoolType
"Pooled" `

    -LoadBalancerType "BreadthFirst" -MaxSessionLimit 3


# WORKSPACE

New-AzWvdWorkspace -ResourceGroupName $resourceGroupName -Name
$workspaceName `

    -Location $location -FriendlyName "AVD Workspace" -Description
"My AVD Workspace"
```

```
# APPLICATION GROUP (Desktop)

$appGroupName = "$hostPoolName-DAG"

New-AzWvdApplicationGroup -ResourceGroupName $resourceGroupName
-HostPoolName $hostPoolName `

    -Name $appGroupName -Location $location -FriendlyName "Desktop
App Group" -ApplicationGroupType "Desktop"


# ASSOCIATE APP GROUP WITH WORKSPACE

New-AzWvdWorkspaceAssociation -ResourceGroupName $resourceGroupName
-WorkspaceName $workspaceName `

    -ApplicationGroupName $appGroupName


# ADD USER TO APP GROUP

Add-AzWvdAppGroupUser -ResourceGroupName $resourceGroupName
-ApplicationGroupName $appGroupName `

    -UserPrincipalName $userPrincipalName


# DEPLOY VM (BASIC EXAMPLE)

for ($i = 1; $i -le $vmCount; $i++) {

    $vmName = "$vmPrefix-$i"


    # Create NIC

    $nicName = "$vmName-NIC"

    $vnet = Get-AzVirtualNetwork -Name "your-vnet-name"
-ResourceGroupName $resourceGroupName

    $subnet = Get-AzVirtualNetworkSubnetConfig -Name
"your-subnet-name" -VirtualNetwork $vnet

    $nic = New-AzNetworkInterface -Name $nicName -ResourceGroupName
$resourceGroupName `

        -Location $location -SubnetId $subnet.Id


    # VM Config
```

```
    $vmConfig = New-AzVMConfig -VMName $vmName -VMSize $vmSize | `
        Set-AzVMOperatingSystem -Windows -ComputerName $vmName
-Credential (New-Object PSCredential ($adminUsername,
$adminPassword)) -ProvisionVMAgent -EnableAutoUpdate | `
        Set-AzVMSourceImage -PublisherName $imagePublisher -Offer
$imageOffer -Skus $imageSku -Version "latest" | `
        Add-AzVMNetworkInterface -Id $nic.Id


    # Create the VM
    New-AzVM -ResourceGroupName $resourceGroupName -Location
$location -VM $vmConfig
}


Write-Host "AVD environment setup completed!"
```

⚠️ Notes

- This script assumes you already have a **virtual network and subnet** created.
- For production, passwords should be stored securely (e.g., in **Azure Key Vault**).
- You must replace `"your-subscription-id"`, `"your-vnet-name"`, and similar placeholders with actual values.
- To join session hosts to Azure AD DS or traditional AD, you'll need domain join automation (not shown here).

➕ Optional Enhancements

- **FSLogix Profile Container setup** for profile management
- Use **ARM templates** or **Bicep** for repeatable deployments
- Integrate **auto-scaling** with Azure Logic Apps or Azure Automation

automation scripts for managing an existing Azure Virtual Desktop (AVD) environment — like adding/removing session hosts, managing users, starting/stopping VMs, and monitoring.

Here are some practical **PowerShell scripts** using the Az and Az.DesktopVirtualization modules to help you manage an existing AVD deployment:

---

1. Connect to Azure and Set Subscription

powershell

CopyEdit

```
# Login and select subscription

Connect-AzAccount

Set-AzContext -SubscriptionId "your-subscription-id"
```

---

2. List Host Pools in a Resource Group

powershell

CopyEdit

```
$resourceGroupName = "YourResourceGroup"


Get-AzWvdHostPool -ResourceGroupName $resourceGroupName |
Format-Table Name, Location, HostPoolType, LoadBalancerType
```

---

3. List Session Hosts in a Host Pool

powershell

CopyEdit

```
$hostPoolName = "YourHostPoolName"


Get-AzWvdSessionHost -ResourceGroupName $resourceGroupName
-HostPoolName $hostPoolName |

    Format-Table Name, SessionHostType, Status, LastHeartBeat
```

---

4. Add a Session Host (VM) to a Host Pool

Assuming the VM is already deployed and domain joined, add it to the host pool:

powershell

CopyEdit

```
$sessionHostName = "your-vm-name.yourdomain.com"


New-AzWvdSessionHost -ResourceGroupName $resourceGroupName
-HostPoolName $hostPoolName -Name $sessionHostName
```

---

5. Remove a Session Host from Host Pool

powershell

CopyEdit

```
Remove-AzWvdSessionHost -ResourceGroupName $resourceGroupName
-HostPoolName $hostPoolName -Name $sessionHostName -Force
```

---

6. Add a User to an Application Group

powershell

CopyEdit

```
$appGroupName = "YourAppGroupName"  # e.g. HostPoolName-DAG for
Desktop App Group

$userUPN = "user@domain.com"


Add-AzWvdAppGroupUser -ResourceGroupName $resourceGroupName
-ApplicationGroupName $appGroupName -UserPrincipalName $userUPN
```

---

7. Remove a User from Application Group

powershell

CopyEdit

```
Remove-AzWvdAppGroupUser -ResourceGroupName $resourceGroupName
-ApplicationGroupName $appGroupName -UserPrincipalName $userUPN
-Force
```

---

8. Start / Stop Session Host VMs

You can start or stop session host VMs using Azure PowerShell:

powershell

CopyEdit

```
$vmName = "YourSessionHostVMName"

# Start VM
Start-AzVM -ResourceGroupName $resourceGroupName -Name $vmName

# Stop VM
Stop-AzVM -ResourceGroupName $resourceGroupName -Name $vmName -Force
```

---

9. Get User Sessions on a Host Pool

This requires **remote management** or AVD diagnostic tools. You can get session info with:

powershell

CopyEdit

```
Get-AzWvdUserSession -ResourceGroupName $resourceGroupName
-HostPoolName $hostPoolName
```

---

10. Restart Session Host

powershell

CopyEdit

```
Restart-AzVM -ResourceGroupName $resourceGroupName -Name $vmName
```

Bonus: Automate Session Host Scaling (Example)

To automate starting/stopping VMs based on time of day, combine the above VM start/stop commands with Azure Automation or Azure Functions triggered by a schedule.

Summary

These scripts will let you:

- Manage users and app groups
- Manage session hosts (add/remove/list)
- Control session host VMs (start/stop/restart)
- Get environment info