Windows PowerShell Get-Help on Cmdlet 'New-AzVmss'

PS:\>Get-HELP New-AzVmss -Full

NAME

New-AzVmss

SYNOPSIS

Creates a virtual machine scale set.

SYNTAX

New-AzVmss [[-ResourceGroupName] <System.String>] [-VMScaleSetName] <System.String> [-AllocationMethod {Static | Dynamic}] [-AsJob] [-BackendPoolName <System.String>]
[-BackendPort <System.Int32[]>] [-CapacityReservationGroupId <System.String>] -Credential <System.Management.Automation.PSCredential> [-DataDiskSizeInGb
<System.Int32[]>] [-DefaultProfile <Microsoft.Azure.Commands.Common.Authentication.Abstractions.Core.IAzureContextContainer>] [-DiskControllerType <System.String>]
[-DomainNameLabel <System.String>] [-EdgeZone <System.String>] [-EnableAutomaticOSUpgrade] [-EnableSecureBoot <System.Nullable`1[System.Boolean]>] [-EnableUltraSSD]
[-EnableVtpm <System.Nullable`1[System.Boolean]>] [-EncryptionAtHost] [-EvictionPolicy <System.String>] [-FrontendPoolName <System.String>] [-HostGroupId
<System.String>] [-IfMatch <System.String>] [-IfNoneMatch <System.String>] [-ImageName <System.String>]

[-ImageReferenceId <System.String>] [-InstanceCount
<System.Int32>] [-LoadBalancerName <System.String>] [-Location <System.String>] [-MaxPrice <System.Double>] [-NatBackendPort <System.Int32[]>] [-OrchestrationMode
<System.String>] [-PlatformFaultDomainCount <System.Int32>] [-Priority <System.String>] [-ProximityPlacementGroupId <System.String>] [-PublicIpAddressName
<System.String>] [-ScaleInPolicy <System.String[]>] [-SecurityGroupName <System.String>] [-SecurityType {TrustedLaunch | ConfidentialVM | Standard}]
[-SharedGalleryImageId <System.String>] [-SinglePlacementGroup] [-SkipExtensionsOnOverprovisionedVMs] [-SubnetAddressPrefix <System.String>] [-SubnetName
<System.String>] [-SystemAssignedIdentity] [-UpgradePolicyMode {Automatic | Manual | Rolling}] [-UserAssignedIdentity <System.String>] [-UserData <System.String>]
[-VirtualNetworkName <System.String>] [-VmSize <System.String>] [-VnetAddressPrefix <System.String>] [-Zone <System.Collections.Generic.List`1[System.String]>]
[-Confirm] [-WhatIf] [<CommonParameters>]

New-AzVmss [-ResourceGroupName] <System.String> [-VMScaleSetName] <System.String> [-VirtualMachineScaleSet]
<Microsoft.Azure.Commands.Compute.Automation.Models.PSVirtualMachineScaleSet> [-AsJob] [-DefaultProfile
<Microsoft.Azure.Commands.Common.Authentication.Abstractions.Core.IAzureContextContainer>] [-EdgeZone <System.String>] [-IfMatch <System.String>] [-IfNoneMatch
<System.String>] [-Confirm] [-WhatIf] [<CommonParameters>]

DESCRIPTION

The New-AzVmss cmdlet creates a Virtual Machine Scale Set (VMSS) in Azure. Use the simple parameter set (`SimpleParameterSet`) to quickly create a pre-set VMSS and
associated resources.

Use the default parameter set (`DefaultParameter`) for more advanced scenarios when you need to precisely configure each component of the VMSS and each associated
resource before creation. For default parameter set, first use the [New-AzVmssConfig](https://learn.microsoft.com/en-us/powershell/module/az.compute/new-azvmss)
cmdlet to create a virtual machine scale set object.

Then use the following cmdlets to set different properties of the virtual machine scale set object:
 -

[Add-AzVmssNetworkInterfaceConfiguration](https://learn.microsoft.com/en-us/powershell/module/az.compute/add-azvmssnetworkinterfaceconfiguration) to set the network profile.
 - [Set-AzVmssOsProfile](https://learn.microsoft.com/en-us/powershell/module/az.compute/set-azvmssosprofile) to set the OS profile.
 - [Set-AzVmssStorageProfile](https://learn.microsoft.com/en-us/powershell/module/az.compute/set-azvmssstorageprofile) to set the storage profile.
 -

[Get-AzComputeResourceSku](https://learn.microsoft.com/en-us/powershell/module/az.compute/get-azcomputeresourcesku) can also be used to find out available virtual machine sizes for your subscription and region.

See other cmdlets for virtual machine scale set here (https://learn.microsoft.com/en-us/powershell/module/az.compute/#vm-scale-sets).

 See Quickstart: Create a virtual machine scale set with Azure PowerShell (https://learn.microsoft.com/en-us/azure/virtual-machine-scale-sets/quick-create-powershell)for tutorial.

PARAMETERS

-AllocationMethod <System.String>

Allocation method for the Public IP Address of the Scale Set (Static or Dynamic). If no value is supplied, allocation will be static.

Required? false
Position? named
Default value Static
Accept pipeline input? False
Accept wildcard characters? false

-AsJob <System.Management.Automation.SwitchParameter>

Run cmdlet in the background and return a Job to track progress.

Required? false

Position? named

Default value False

Accept pipeline input? False

Accept wildcard characters? false

-BackendPoolName <System.String>

The name of the backend address pool to use in the load balancer for this Scale Set. If no value is provided, a new backend pool will be created, with the same

name as the Scale Set.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-BackendPort <System.Int32[]>

Backend port numbers used by the Scale Set load balancer to communicate with VMs in the Scale Set. If no values are specified, ports 3389 and 5985 will be used

for Windows VMS, and port 22 will be used for Linux VMs.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-CapacityReservationGroupId <System.String>

Id of the capacity reservation Group that is used to allocate.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-Credential <System.Management.Automation.PSCredential>

The administrator credentials (username and password) for VMs in this Scale Set.

 Username
 Restriction:
 Windows: Cannot contain special

characters \/""[]:|<>+=;,?*@& or end in \".\"
 Linux: Username must only contain letters, numbers, hyphens, and underscores and may not start with a hyphen or

number.
 Disallowed values: \"administrator\", \"admin\", \"user\", \"user1\", \"test\", \"user2\", \"test1\", \"user3\", \"admin1\", \"1\", \"123\", \"a\",

\"actuser\", \"adm\", \"admin2\", \"aspnet\", \"backup\", \"console\", \"david\", \"guest\", \"john\", \"owner\", \"root\", \"server\", \"sql\", \"support\",

\"support_388945a0\", \"sys\", \"test2\", \"test3\", \"user4\", \"user5\".
 Minimum-length: 1 character
 Max-length: 20 characters for Windows, 64

characters for Linux
 Password
 Must have 3 of the following: 1 lower case character, 1 upper case character, 1 number, and 1 special character.
 The

value must be between 12 and 123 characters long.

Required? true

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-DataDiskSizeInGb <System.Int32[]>

Specifies the sizes of data disks in GB.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-DefaultProfile <Microsoft.Azure.Commands.Common.Authentication.Abstractions.Core.IAzureContextContainer>

The credentials, account, tenant, and subscription used for communication with azure.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-DiskControllerType <System.String>

Specifies the disk controller type configured for the VM and VirtualMachineScaleSet. This property is only supported for virtual machines whose operating system

disk and VM sku supports Generation 2 (https://learn.microsoft.com/en-us/azure/virtual-machines/generation-2), please check the HyperVGenerations capability

returned as part of VM sku capabilities in the response of Microsoft.Compute SKUs api for the region contains V2

(https://learn.microsoft.com/rest/api/compute/resourceskus/list) .
 For more information about Disk Controller Types supported please refer to

https://aka.ms/azure-diskcontrollertypes.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-DomainNameLabel <System.String>

The domain name label for the public Fully-Qualified domain name (FQDN) for this Scale Set. This is the first component of the domain name that is automatically

assigned to the Scale Set. Automatically assigned Domain names use the form (`<DomainNameLabel>.<Location>.cloudapp.azure.com`). If no value is supplied, the

default domain name label will be the concatenation of `<ScaleSetName>` and `<ResourceGroupName>`.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-EdgeZone <System.String>

Sets the edge zone name. If set, the query will be routed to the specified edgezone instead of the main region.

Required? false

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-EnableAutomaticOSUpgrade <System.Management.Automation.SwitchParameter>

Whether OS upgrades should automatically be applied to scale set instances in a rolling fashion when a newer version of the image becomes available.

Required? false

Position? named

Default value False

Accept pipeline input? False

Accept wildcard characters? false

-EnableSecureBoot <System.Nullable`1[System.Boolean]>

Specifies whether secure boot should be enabled on the virtual machine.

Required? false

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-EnableUltraSSD <System.Management.Automation.SwitchParameter>

Use UltraSSD disks for the VMs in the scale set.

Required? false

Position? named

Default value False

Accept pipeline input? False

Accept wildcard characters? false

-EnableVtpm <System.Nullable`1[System.Boolean]>

Specifies whether vTPM should be enabled on the virtual machine.

Required? false

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-EncryptionAtHost <System.Management.Automation.SwitchParameter>

This parameter will enable the encryption for all the disks including Resource/Temp disk at host itself. Default: The Encryption at host will be disabled unless

this property is set to true for the resource.

Required? false

Position? named

Default value False

Accept pipeline input? False

Accept wildcard characters? false

-EvictionPolicy <System.String>

The eviction policy for the low priority virtual machine scale set. Only supported values are 'Deallocate' and 'Delete'.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-FrontendPoolName <System.String>

The name of the frontend address pool to use in the Scale Set load balancer. If no value is supplied, a new Frontend Address Pool will be created, with the same

name as the scale set.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-HostGroupId <System.String>

Specifies the dedicated host group the virtual machine scale set will reside in.

Required? false

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-IfMatch <System.String>

used to make a request conditional for the PUT and other non-safe methods. The server will only return the requested resources if the resource matches one of the

listed ETag values. Omit this value to always overwrite the current resource. Specify the last-seen ETag value to prevent accidentally overwriting concurrent

changes.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-IfNoneMatch <System.String>

Used to make a request conditional for the GET and HEAD methods. The server will only return the requested resources if none of the listed ETag values match the

current entity. Used to make a request conditional for the GET and HEAD methods. The server will only return the requested resources if none of the listed ETag

values match the current entity. Set to '*' to allow a new record set to be created, but to prevent updating an existing record set. Other values will result in

error from server as they are not supported.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-ImageName <System.String>

The alias of the image for VMs in this Scale Set. If no value is provided, the "Windows Server 2016 DataCenter" image will be used. The available aliases are:

Win2022AzureEdition, Win2022AzureEditionCore, Win2019Datacenter, Win2016Datacenter, Win2012R2Datacenter, Win2012Datacenter, UbuntuLTS, Ubuntu2204, CentOS85Gen2,

Debian11, OpenSuseLeap154Gen2, RHELRaw8LVMGen2, SuseSles15SP3, FlatcarLinuxFreeGen2.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-ImageReferenceId <System.String>

Specified the shared gallery image unique id for vmss deployment. This can be fetched from shared gallery image GET call.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-InstanceCount <System.Int32>

The number of VM images in the Scale Set. If no value is provided, 2 instances will be created.

Required? false

Position? named

Default value 2

Accept pipeline input? False

Accept wildcard characters? false

-LoadBalancerName <System.String>

The name of the load balancer to use with this Scale Set. A new load balancer using the same name as the Scale Set will be created if no value is specified.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-Location <System.String>

The Azure location where this Scale Set will be created. If no value is specified, the location will be inferred from the

location of other resources referenced

in the parameters.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-MaxPrice <System.Double>

The max price of the billing of a low priority virtual machine scale set.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-NatBackendPort <System.Int32[]>

Backend port for inbound network address translation.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-OrchestrationMode <System.String>

Specifies the orchestration mode for the virtual machine scale set. Possible values: Uniform, Flexible

Required? false

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-PlatformFaultDomainCount <System.Int32>

Fault Domain count for each placement group.

Required? false

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-Priority <System.String>

The priority for the virtual machine in the scale set. Only supported values are 'Regular', 'Spot' and 'Low'. 'Regular' is for regular virtual machine. 'Spot' is

for spot virtual machine. 'Low' is also for spot virtual machine but is replaced by 'Spot'. Please use 'Spot' instead of 'Low'.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-ProximityPlacementGroupId <System.String>

The resource id of the Proximity Placement Group to use with this scale set.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-PublicIpAddressName <System.String>

The name of the public IP Address to use with this scale set. A new Public IPAddress with the same name as the Scale Set will be created if no value is provided.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-ResourceGroupName <System.String>

Specifies the name of the resource group of the VMSS. If no value is specified, a new ResourceGroup will be created using the same name as the Scale Set.

Required? true

Position? 0

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-ScaleInPolicy <System.String[]>

The rules to be followed when scaling-in a virtual machine scale set. Possible values are: 'Default', 'OldestVM' and 'NewestVM'. 'Default' when a virtual

machine scale set is scaled in, the scale set will first be balanced across zones if it is a zonal scale set. Then, it will be balanced across Fault Domains as

far as possible. Within each Fault Domain, the virtual machines chosen for removal will be the newest ones that are not protected from scale-in. 'OldestVM' when

a virtual machine scale set is being scaled-in, the oldest virtual machines that are not protected from scale-in will be chosen for removal. For zonal virtual

machine scale sets, the scale set will first be balanced across zones. Within each zone, the oldest virtual machines that are not protected will be chosen for

removal. 'NewestVM' when a virtual machine scale set is being scaled-in, the newest virtual machines that are not protected from scale-in will be chosen for

removal. For zonal virtual machine scale sets, the scale set will first be balanced across zones. Within each zone, the newest virtual machines that are not

protected will be chosen for removal.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-SecurityGroupName <System.String>

The name of the network security group to apply to this Scale Set. If no value is provided, a default network security group with the same name as the Scale Set

will be created and applied to the Scale Set.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-SecurityType <System.String>

Specifies the SecurityType of the virtual machine. It has to be set to any specified value to enable UefiSettings. UefiSettings will not be enabled unless this

property is set.

Required? false

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-SharedGalleryImageId <System.String>

Specified the shared gallery image unique id for vm deployment. This can be fetched from shared gallery image GET call.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-SinglePlacementGroup <System.Management.Automation.SwitchParameter>

Use this to create the Scale set in a single placement group, default is multiple groups

Required? false

Position? named

Default value False

Accept pipeline input? False

Accept wildcard characters? false

-SkipExtensionsOnOverprovisionedVMs <System.Management.Automation.SwitchParameter>

Specifies that the extensions do not run on the extra overprovisioned VMs.

Required? false

Position? named

Default value False

Accept pipeline input? False

Accept wildcard characters? false

-SubnetAddressPrefix <System.String>

The address prefix of the Subnet this ScaleSet will use. Default Subnet settings (192.168.1.0/24) will be applied if no value is provided.

Required? false

Position? named

Default value 192.168.1.0/24

Accept pipeline input? False

Accept wildcard characters? false

-SubnetName <System.String>

The name of the subnet to use with this Scale Set. A new Subnet will be created with the same name as the Scale Set if no value is provided.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-SystemAssignedIdentity <System.Management.Automation.SwitchParameter>

If the parameter is present then the VM(s) in the scale set is(are) assigned a managed system identity that is auto generated.

Required? false

Position? named

Default value False

Accept pipeline input? False

Accept wildcard characters? false

-UpgradePolicyMode <Microsoft.Azure.Management.Compute.Models.UpgradeMode>

The upgrade policy mode for VM instances in this Scale Set. Upgrade policy could specify Automatic, Manual, or Rolling upgrades.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-UserAssignedIdentity <System.String>

The name of a managed service identity that should be assigned to the VM(s) in the scale set.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-UserData <System.String>

UserData for the Vmss, which will be base-64 encoded. Customer should not pass any secrets in here.

Required? false

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-VirtualMachineScaleSet <Microsoft.Azure.Commands.Compute.Automation.Models.PSVirtualMachineScaleSet>

Specifies the VirtualMachineScaleSet object that contains the properties of the VMSS that this cmdlet creates.

Required? true

Position? 2

Default value None

Accept pipeline input? True (ByValue)

Accept wildcard characters? false

-VirtualNetworkName <System.String>

The name fo the Virtual Network to use with this scale set. If no value is supplied, a new virtual network with the same name as the Scale Set will be created.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-VMScaleSetName <System.String>

Specifies the name of the VMSS that this cmdlet creates.

Required? true

Position? 1

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-VmSize <System.String>

The size of the VM instances in this scale set. Get-AzComputeResourceSku (https://learn.microsoft.com/en-us/powershell/module/az.compute/get-azcomputeresourcesku)

can be used to find out available sizes for your subscription and region. A default size (Standard_DS1_v2)will be used if no Size is specified.

Required? false

Position? named

Default value Standard_DS1_v2

Accept pipeline input? False

Accept wildcard characters? false

-VnetAddressPrefix <System.String>

The address prefix for the virtual network used with this Scale Set. Default virtual network address prefix settings (192.168.0.0/16) will be used if no value is

supplied.

Required? false

Position? named

Default value 192.168.0.0/16

Accept pipeline input? False

Accept wildcard characters? false

-Zone <System.Collections.Generic.List`1[System.String]>

A list of availability zones denoting the IP allocated for the resource needs to come from.

Required? false

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-Confirm <System.Management.Automation.SwitchParameter>

Prompts you for confirmation before running the cmdlet.

Required? false

Position? named

Default value False

Accept pipeline input? False

Accept wildcard characters? false

-WhatIf <System.Management.Automation.SwitchParameter>

Shows what would happen if the cmdlet runs. The cmdlet is not run.

Required? false

Position? named

Default value False

Accept pipeline input? False

Accept wildcard characters? false

<CommonParameters>

This cmdlet supports the common parameters: Verbose, Debug,

ErrorAction, ErrorVariable, WarningAction, WarningVariable,

OutBuffer, PipelineVariable, and OutVariable. For more information, see

about_CommonParameters (https:/go.microsoft.com/fwlink/?LinkID=113216).

INPUTS

System.String

Microsoft.Azure.Commands.Compute.Automation.Models.PSVirtualMachineScaleSet

System.Collections.Generic.List`1[[System.String, System.Private.CoreLib, Version=4.0.0.0, Culture=neutral, PublicKeyToken=7cec85d7bea7798e]]

OUTPUTS

Microsoft.Azure.Commands.Compute.Automation.Models.PSVirtualMachineScaleSet

NOTES

---- Example 1: Create a VMSS using the SimpleParameterSet ----

```
$vmssName = 'VMSSNAME'
# Create credentials, I am using one way to create credentials, there are others as well.
# Pick one that makes the most sense according to your use case.
$vmPassword = ConvertTo-SecureString "PASSWORD" -AsPlainText -Force
$vmCred = New-Object System.Management.Automation.PSCredential('USERNAME', $vmPassword)
```

$securityTypeStnd = "Standard"

#Create a VMSS using the default settings

New-AzVmss -Credential $vmCred -VMScaleSetName $vmssName -SecurityType $securityTypeStnd

The command above creates the following with the name `$vmssName` : * A Resource Group

* A virtual network

* A load balancer

* A public IP

* the VMSS with 2 instances

The default image chosen for the VMs in the VMSS is `2016-Datacenter Windows Server` and the SKU is `Standard_DS1_v2`

---- Example 2: Create a VMSS using the DefaultParameterSet ----

Common

$LOC = "WestUs";

$RGName = "rgkyvms";

New-AzResourceGroup -Name $RGName -Location $LOC -Force;

SRP

$STOName = "sto" + $RGName;

$STOType = "Standard_GRS";

New-AzStorageAccount -ResourceGroupName $RGName -Name $STOName -Location $LOC -Type $STOType;

```
    $STOAccount = Get-AzStorageAccount -ResourceGroupName $RGName -Name $STOName;

    # NRP
    $SubNet = New-AzVirtualNetworkSubnetConfig -Name ("subnet" + $RGName) -AddressPrefix "10.0.0.0/24";
     $VNet = New-AzVirtualNetwork -Force -Name ("vnet" + $RGName) -ResourceGroupName $RGName -Location $LOC
-AddressPrefix "10.0.0.0/16" -DnsServer "10.1.1.1" -Subnet
    $SubNet;
    $VNet = Get-AzVirtualNetwork -Name ('vnet' + $RGName) -ResourceGroupName $RGName;
    $SubNetId = $VNet.Subnets[0].Id;

      $PubIP = New-AzPublicIpAddress -Force -Name ("pubip" + $RGName) -ResourceGroupName $RGName -Location
$LOC -AllocationMethod Dynamic -DomainNameLabel ("pubip" +
    $RGName);
    $PubIP = Get-AzPublicIpAddress -Name ("pubip"  + $RGName) -ResourceGroupName $RGName;

    # Create LoadBalancer
    $FrontendName = "fe" + $RGName
    $BackendAddressPoolName = "bepool" + $RGName
    $ProbeName = "vmssprobe" + $RGName
    $InboundNatPoolName  = "innatpool" + $RGName
    $LBRuleName = "lbrule" + $RGName
    $LBName = "vmsslb" + $RGName

    $Frontend = New-AzLoadBalancerFrontendIpConfig -Name $FrontendName -PublicIpAddress $PubIP
    $BackendAddressPool = New-AzLoadBalancerBackendAddressPoolConfig -Name $BackendAddressPoolName
    $Probe = New-AzLoadBalancerProbeConfig -Name $ProbeName -RequestPath healthcheck.aspx -Protocol http -Port 80
-IntervalInSeconds 15 -ProbeCount 2
                $InboundNatPool     =     New-AzLoadBalancerInboundNatPoolConfig     -Name     $InboundNatPoolName
-FrontendIPConfigurationId `
      $Frontend.Id -Protocol Tcp -FrontendPortRangeStart 3360 -FrontendPortRangeEnd 3367 -BackendPort 3370;
    $LBRule = New-AzLoadBalancerRuleConfig -Name $LBRuleName `
      -FrontendIPConfiguration $Frontend -BackendAddressPool $BackendAddressPool `
      -Probe $Probe -Protocol Tcp -FrontendPort 80 -BackendPort 80 `
```

```
    -IdleTimeoutInMinutes 15 -EnableFloatingIP -LoadDistribution SourceIP;
$ActualLb = New-AzLoadBalancer -Name $LBName -ResourceGroupName $RGName -Location $LOC `
    -FrontendIpConfiguration $Frontend -BackendAddressPool $BackendAddressPool `
    -Probe $Probe -LoadBalancingRule $LBRule -InboundNatPool $InboundNatPool;
$ExpectedLb = Get-AzLoadBalancer -Name $LBName -ResourceGroupName $RGName

# New VMSS Parameters
$VMSSName = "vmss" + $RGName;

$AdminUsername = "Admin01";
$AdminPassword = "p4ssw0rd@123" + $RGName;

$PublisherName = "MicrosoftWindowsServer"
$Offer         = "WindowsServer"
$Sku           = "2012-R2-Datacenter"
$Version       = "latest"

$VHDContainer = "https://" + $STOName + ".blob.core.windows.net/" + $VMSSName;

$ExtName = "CSETest";
$Publisher = "Microsoft.Compute";
$ExtType = "BGInfo";
$ExtVer = "2.1";

#IP Config for the NIC
$IPCfg = New-AzVmssIpConfig -Name "Test" `
    -LoadBalancerInboundNatPoolsId $ExpectedLb.InboundNatPools[0].Id `
    -LoadBalancerBackendAddressPoolsId $ExpectedLb.BackendAddressPools[0].Id `
    -SubnetId $SubNetId;

#VMSS Config
$securityTypeStnd = "Standard";
 $VMSS = New-AzVmssConfig -Location $LOC -SkuCapacity 2 -SkuName "Standard_E4-2ds_v4" -UpgradePolicyMode
```

"Automatic" -SecurityType $securityTypeStnd `

| Add-AzVmssNetworkInterfaceConfiguration -Name "Test" -Primary $True -IPConfiguration $IPCfg `

| Add-AzVmssNetworkInterfaceConfiguration -Name "Test2" -IPConfiguration $IPCfg `

| Set-AzVmssOsProfile -ComputerNamePrefix "Test" -AdminUsername $AdminUsername -AdminPassword $AdminPassword `

| Set-AzVmssStorageProfile -Name "Test" -OsDiskCreateOption 'FromImage' -OsDiskCaching "None" `

-ImageReferenceOffer $Offer -ImageReferenceSku $Sku -ImageReferenceVersion $Version `

-ImageReferencePublisher $PublisherName -VhdContainer $VHDContainer `

| Add-AzVmssExtension -Name $ExtName -Publisher $Publisher -Type $ExtType -TypeHandlerVersion $ExtVer -AutoUpgradeMinorVersion $True

#Create the VMSS

New-AzVmss -ResourceGroupName $RGName -Name $VMSSName -VirtualMachineScaleSet $VMSS;

The complex example above creates a VMSS, following is an explanation of what is happening: * The first command creates a resource group with the specified name and

location. The second command uses the New-AzStorageAccount * cmdlet to create a storage account. The third command then uses the Get-AzStorageAccount * cmdlet to get

the storage account created in the second command and stores the result in the $STOAccount variable. The fifth command uses the New-AzVirtualNetworkSubnetConfig *

cmdlet to create a subnet and stores the result in the variable named $SubNet. The sixth command uses the New-AzVirtualNetwork * cmdlet to create a virtual network

and stores the result in the variable named $VNet. The seventh command uses the Get-AzVirtualNetwork * to get information about the virtual network created in the

sixth command and stores the information in the variable named $VNet. The eighth and ninth command uses the New-AzPublicIpAddress and Get- AzureRmPublicIpAddress * to

create and get information from that public IP address.

* The commands store the information in the variable named $PubIP. The tenth command uses the New-AzureRmLoadBalancerFrontendIpConfig * cmdlet to create a frontend

load balancer and stores the result in the variable named $Frontend. The eleventh command uses the New-AzLoadBalancerBackendAddressPoolConfig * to create a backend

address pool configuration and stores the result in the variable named $BackendAddressPool. The twelfth command uses

the New-AzLoadBalancerProbeConfig * to create a

probe and stores the probe information in the variable named $Probe. The thirteenth command uses the New-AzLoadBalancerInboundNatPoolConfig * cmdlet to create a load

balancer inbound network address translation (NAT) pool configuration. The fourteenth command uses the New-AzLoadBalancerRuleConfig * to create a load balancer rule

configuration and stores the result in the variable named $LBRule. The fifteenth command uses the New-AzLoadBalancer * cmdlet to create a load balancer and stores the

result in the variable named $ActualLb. The sixteenth command uses the Get-AzLoadBalancer * to get information about the load balancer that was created in the

fifteenth command and stores the information in the variable named $ExpectedLb. The seventeenth command uses the New-AzVmssIpConfig * cmdlet to create a VMSS IP

configuration and stores the information in the variable named $IPCfg. The eighteenth command uses the New-AzVmssConfig * cmdlet to create a VMSS configuration object

and stores the result in the variable named $VMSS. The nineteenth command uses the New-AzVmss * cmdlet to create the VMSS.

-------- Example 3: Create a VMSS with a UserData value --------

```
$ResourceGroupName = 'RESOURCE GROUP NAME';
$vmssName = 'VMSSNAME';
$domainNameLabel = "dnl" + $ResourceGroupName;
# Create credentials, I am using one way to create credentials, there are others as well.
# Pick one that makes the most sense according to your use case.
$vmPassword = ConvertTo-SecureString 'PASSWORD' -AsPlainText -Force;
$vmCred = New-Object System.Management.Automation.PSCredential('USERNAME', $vmPassword);

$text = "UserData value to encode";
$bytes = [System.Text.Encoding]::Unicode.GetBytes($text);
$userData = [Convert]::ToBase64String($bytes);
$securityTypeStnd = "Standard";

#Create a VMSS
```

New-AzVmss -ResourceGroupName $ResourceGroupName -Name $vmssName -Credential $vmCred -DomainNameLabel $domainNameLabel -Userdata $userData -SecurityType $securityTypeStnd;

$vmss = Get-AzVmss -ResourceGroupName $ResourceGroupName -VMScaleSetName $vmssName -InstanceView:$false -Userdata;

Create a VMSS with a UserData value

Example 4: Create a Vmss with the security type TrustedLaunch

```
$rgname = "rgname";
$loc = "eastus";

# VMSS Profile & Hardware requirements for the TrustedLaunch default behavior.
$vmssSize = 'Standard_D4s_v3';
$vmssName1 = 'vmss1' + $rgname;
$imageName = "Win2022AzureEdition";
$adminUsername = "<Username>";
$adminPassword = "<Password>" | ConvertTo-SecureString -AsPlainText -Force;
$vmCred = New-Object System.Management.Automation.PSCredential ($adminUsername, $adminPassword);

# VMSS Creation
$result = New-AzVmss -Credential $vmCred -VMScaleSetName $vmssName1 -ImageName $imageName -SecurityType "TrustedLaunch";
# Validate that for -SecurityType "TrustedLaunch", "-Vtpm" and -"SecureBoot" are "Enabled/true"
# $result.VirtualMachineProfile.SecurityProfile.UefiSettings.VTpmEnabled;
# $result.VirtualMachineProfile.SecurityProfile.UefiSettings.SecureBootEnabled;
```

This example Creates a new VMSS with the new Security Type 'TrustedLaunch' and the necessary UEFISettings values, VTpmEnabled and SecureBootEnalbed are true. Please

check the Trusted Launch feature page (https://aka.ms/trustedlaunch)for more information.

Example 5: Create a Vmss in Orchestration Mode: Flexible by default

```
# Create configration object
$vmssConfig = New-AzVmssConfig -Location EastUs2 -UpgradePolicyMode Manual -SinglePlacementGroup $true

# VMSS Creation
New-AzVmss -ResourceGroupName TestRg -VMScaleSetName myVMSS -VirtualMachineScaleSet $vmssConfig
```

This example Creates a new VMSS and it will default to OrchestrationMode Flexible.

Example 6: Create a new VMSS with TrustedLaunch turned on by default.

```
$rgname = "<Resource Group>";
$loc = "<Azure Region>";
New-AzResourceGroup -Name $rgname -Location $loc -Force;

$vmssName = 'vmss' + $rgname;
$vmssSize = 'Standard_D4s_v3';
$imageName = "Win2022AzureEdition";
$publisherName = "MicrosoftWindowsServer";
$offer = "WindowsServer";
$sku = "2022-Datacenter-Azure-Edition";
$adminUsername = "<Username>";
$password = "<Password>";

# NRP
$vnetworkName = 'vnet' + $rgname;
$subnetName = 'subnet' + $rgname;
$subnet = New-AzVirtualNetworkSubnetConfig -Name $subnetName -AddressPrefix "10.0.0.0/24";
  $vnet = New-AzVirtualNetwork -Name $vnetworkName -ResourceGroupName $rgname -Location $loc -AddressPrefix
"10.0.0.0/16" -Subnet $subnet;
```

```
$vnet = Get-AzVirtualNetwork -Name $vnetworkName -ResourceGroupName $rgname;
$subnetId = $vnet.Subnets[0].Id;

$ipCfg = New-AzVmssIpConfig -Name 'test' -SubnetId $subnetId;

$vmss = New-AzVmssConfig -Location $loc -SkuCapacity 2 -SkuName $vmssSize -UpgradePolicyMode 'Manual' `
  | Add-AzVmssNetworkInterfaceConfiguration -Name 'test' -Primary $true -IPConfiguration $ipCfg `
  | Set-AzVmssOsProfile -ComputerNamePrefix 'test' -AdminUsername $adminUsername -AdminPassword $password;

# Create TL Vmss
  $result = New-AzVmss -ResourceGroupName $rgname -VMScaleSetName $vmssName -VirtualMachineScaleSet $vmss;
$vmssGet = Get-AzVmss -ResourceGroupName $rgname -VMScaleSetName $vmssName;

# Verify $vmssGet.VirtualMachineProfile.SecurityProfile.SecurityType is TrustedLaunch.
# Verify $vmssGet.VirtualMachineProfile.SecurityProfile.UefiSettings.VTpmEnabled is True.
# Verify $vmssGet.VirtualMachineProfile.SecurityProfile.UefiSettings.SecureBootEnabled is True.
# Verify $vmssGet.VirtualMachineProfile.StorageProfile.ImageReference.Sku is 2022-Datacenter-Azure-Edition.
```

The virtual machine scale set above has Trusted Launch enabled by default. Please check the Trusted Launch feature page (https://aka.ms/trustedlaunch)for more information.

RELATED LINKS

Online Version: https://learn.microsoft.com/powershell/module/az.compute/new-azvmss

Get-AzVmss

Remove-AzVmss

Restart-AzVmss

Set-AzVmss

Start-AzVmss

Stop-AzVmss