



Windows PowerShell Get-Help on Cmdlet 'Set-AzVmssOsProfile'

PS:\>Get-HELP Set-AzVmssOsProfile -Full

NAME

Set-AzVmssOsProfile

SYNOPSIS

Sets the VMSS operating system profile properties.

SYNTAX

```

Set-AzVmssOsProfile [-VirtualMachineScaleSet]
<Microsoft.Azure.Commands.Compute.Automation.Models.PSVirtualMachineScaleSet> [[-ComputerNamePrefix]
<System.String>]
[[[-LinuxConfigurationDisablePasswordAuthentication] <System.Nullable`1[System.Boolean]>] [[-PublicKey]
<Microsoft.Azure.Management.Compute.Models.SshPublicKey[]>]
[[[-Secret] <Microsoft.Azure.Management.Compute.Models.VaultSecretGroup[]>] [[-AdminUsername] <System.String>]
[[[-AdminPassword] <System.String>] [[-CustomData]
<System.String>] [[-WindowsConfigurationProvisionVMAgent] <System.Nullable`1[System.Boolean]>]
[[[-WindowsConfigurationEnableAutomaticUpdate]
<System.Nullable`1[System.Boolean]>] [[-TimeZone] <System.String>] [[-AdditionalUnattendContent]
<Microsoft.Azure.Management.Compute.Models.AdditionalUnattendContent[]>] [[-Listener]
<Microsoft.Azure.Management.Compute.Models.WinRMListener[]>] [-DefaultProfile

```

<Microsoft.Azure.Commands.Common.Authentication.Abstractions.Core.IAzureContextContainer>] [-EnableHotpatching]
 [-LinuxConfigurationPatchMode <System.String>]
 [-LinuxConfigurationProvisionVMAgent <System.Nullable`1[System.Boolean]>] [-WindowsConfigurationPatchMode
 <System.String>] [-Confirm] [-WhatIf] [<CommonParameters>]

DESCRIPTION

The Set-AzVmssOsProfile cmdlet sets the Virtual Machine Scale Set operating system profile properties.

PARAMETERS

-AdditionalUnattendContent <Microsoft.Azure.Management.Compute.Models.AdditionalUnattendContent[]>

Specifies an unattended content object. You can use the Add-AzVMAdditionalUnattendContent to create the object.

Required?	false
Position?	8
Default value	None
Accept pipeline input?	True (ByPropertyName)
Accept wildcard characters?	false

-AdminPassword <System.String>

Specifies the administrator password to use for all the virtual machine instances in the VMSS.

Required?	false
Position?	3
Default value	None
Accept pipeline input?	True (ByPropertyName)
Accept wildcard characters?	false

-AdminUsername <System.String>

Specifies the administrator account name to use for all the virtual machine instances in the VMSS.
 Restriction:

 Windows: Cannot contain special
 characters V""[]|<>+=,;?*@& 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
 For a list of built-in system users on Linux that should not be used in this field, see Selecting User Names for Linux on Azure

(<https://learn.microsoft.com/azure/devops/organizations/settings/naming-restrictions>).

Required?	false
Position?	2
Default value	None
Accept pipeline input?	True (ByPropertyName)
Accept wildcard characters?	false

-ComputerNamePrefix <System.String>

Specifies the computer name prefix for all the virtual machine instances in the VMSS. Computer names must be 1 to 15 characters long.

Required?	false
Position?	1
Default value	None
Accept pipeline input?	True (ByPropertyName)
Accept wildcard characters?	false

-CustomData <System.String>

Specifies a base-64 encoded string of custom data. This is decoded to a binary array that is saved as a file on the virtual machine. The maximum length of the

binary array is 65535 bytes.
 For using cloud-init for your VM, see Using cloud-init to customize a Linux VM during creation

([/azure/virtual-machines/linux/tutorial-automate-vm-deployment](https://learn.microsoft.com/azure/virtual-machines/linux/tutorial-automate-vm-deployment)).

Required? false
Position? 4
Default value None
Accept pipeline input? True (ByPropertyName)
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

-EnableHotpatching <System.Management.Automation.SwitchParameter>

Enables customers to patch their Azure Vmss without requiring a reboot. For enableHotpatching, the 'provisionVMAGENT' must be set to true and 'patchMode' must be set to 'AutomaticByPlatform'.

Required? false
Position? named
Default value False
Accept pipeline input? True (ByPropertyName)
Accept wildcard characters? false

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

Indicates that this cmdlet disables password authentication.

Required? false
Position? 10
Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-LinuxConfigurationPatchMode <System.String>

Specifies the mode of VM Guest Patching to IaaS virtual machine or virtual machines associated to virtual machine scale set with OrchestrationMode as Flexible.

</>
 Possible values are:

 ImageDefault - The virtual machine's default patching configuration is used.

 AutomaticByPlatform - The

virtual machine will be automatically updated by the platform. The property provisionVMAgent must be true

Required? false

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

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

Indicates whether virtual machine agent should be provisioned on the virtual machine.

 When this property is not specified in the request body, default

behavior is to set it to true. This will ensure that VM Agent is installed on the VM so that extensions can be added to the VM later

Required? false

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-Listener <Microsoft.Azure.Management.Compute.Models.WinRMListener[]>

Specifies the Windows Remote Management (WinRM) listeners. This enables remote Windows PowerShell. You can use the Add-AzVmssWinRMListener cmdlet to create the listener.

Required?	false
Position?	9
Default value	None
Accept pipeline input?	True (ByPropertyName)
Accept wildcard characters?	false

-PublicKey <Microsoft.Azure.Management.Compute.Models.SshPublicKey[]>

Specifies the Secure Shell (SSH) public key object. You can use the Add-AzVMSSshPublicKey cmdlet to create the object.

Required?	false
Position?	11
Default value	None
Accept pipeline input?	True (ByPropertyName)
Accept wildcard characters?	false

-Secret <Microsoft.Azure.Management.Compute.Models.VaultSecretGroup[]>

Specifies the secrets object which contains the certificate references to place on the virtual machine. You can use the Add-AzVmssSecret cmdlet to create the secrets object.

Required?	false
Position?	12
Default value	None
Accept pipeline input?	True (ByPropertyName)
Accept wildcard characters?	false

-TimeZone <System.String>

Specifies the time zone of the virtual machine. e.g. \"Pacific Standard Time\".
 Possible values can be TimeZoneInfo.Id

(https://learn.microsoft.com/dotnet/api/system.timezonelocalizer.id?#System_TimeZoneInfo_Id) value from time zones returned by

[TimeZoneInfo.GetSystemTimeZones](https://learn.microsoft.com/dotnet/api/system.timezonetimezones).

Required?	false
Position?	7
Default value	None
Accept pipeline input?	True (ByPropertyName)
Accept wildcard characters?	false

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

Specifies the VMSS object. You can use the New-AzVmssConfig cmdlet to create the object.

Required?	true
Position?	0
Default value	None
Accept pipeline input?	True (ByPropertyName, ByValue)
Accept wildcard characters?	false

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

Indicates whether the virtual machines in the VMSS are enabled for automatic updates.

Required?	false
Position?	6
Default value	None
Accept pipeline input?	True (ByPropertyName)
Accept wildcard characters?	false

-WindowsConfigurationPatchMode <System.String>

Specifies the mode of VM Guest Patching to IaaS virtual machine or virtual machines associated to virtual machine scale set with OrchestrationMode as Flexible.

 Possible values are:

 Manual - You control the application of patches to a virtual machine. You do this by applying patches manually inside

the VM. In this mode, automatic updates are disabled; the property WindowsConfiguration.enableAutomaticUpdates must be false

 AutomaticByOS - The

virtual machine will automatically be updated by the OS. The property `WindowsConfiguration.enableAutomaticUpdates` must be true.
AutomaticByPlatform -

the virtual machine will automatically updated by the platform. The properties `provisionVMAgent` and `WindowsConfiguration.enableAutomaticUpdates` must be true

Required?	false
Position?	named
Default value	None
Accept pipeline input?	True (ByPropertyName)
Accept wildcard characters?	false

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

Indicates whether virtual machine agent should be provisioned on the virtual machines in the VMSS.

Required?	false
Position?	5
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

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

System.String

System.Nullable`1[[System.Boolean, System.Private.CoreLib, Version=4.0.0.0, Culture=neutral, PublicKeyToken=7cec85d7bea7798e]]

Microsoft.Azure.Management.Compute.Models.AdditionalUnattendContent[]

Microsoft.Azure.Management.Compute.Models.WinRMListener[]

Microsoft.Azure.Management.Compute.Models.SshPublicKey[]

Microsoft.Azure.Management.Compute.Models.VaultSecretGroup[]

OUTPUTS

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

NOTES

Example 1: Set the operating system profile properties for a VMSS

```
$vmss = New-AzVmssConfig -Location $Loc -SkuCapacity 2 -SkuName "Standard_A0" -UpgradePolicyMode "Automatic"  
-NetworkInterfaceConfiguration $NetCfg  
Set-AzVmssOsProfile -VirtualMachineScaleSet $vmss -ComputerNamePrefix "Test" -AdminUsername $AdminUsername  
-AdminPassword $AdminPassword
```

This command sets operating system profile properties for the \$vmss object. The command sets the computer name prefix for all the virtual machine instances in the VMSS to Test and supplies the administrator username and password.

Example 2: Set the operating system profile properties for a Vmss in Flexible mode with Hotpatching enabled.

```
# Setup variables.  
$loc = "eastus";  
$rgname = "<Resource Group Name>";  
$vmssName = "myVmssSlb";  
$vmNamePrefix = "vmSlb";  
$vmssInstanceCount = 5;  
$vmssSku = "Standard_DS1_v2";
```

```

$vnename = "myVnet";
$vnetAddress = "10.0.0.0/16";
$subnetname = "default-slb";
$subnetAddress = "10.0.2.0/24";
$securePassword = "<Password>" | ConvertTo-SecureString -AsPlainText -Force;
$cred = New-Object System.Management.Automation.PSCredential ("<Username>", $securePassword);

# VMSS Flex requires explicit outbound access.

# Create a virtual network.

$frontendSubnet = New-AzVirtualNetworkSubnetConfig -Name $subnetname -AddressPrefix $subnetAddress;
    $virtualNetwork = New-AzVirtualNetwork -Name $vnename -ResourceGroupName $rgname -Location $loc
-AddressPrefix $vnetAddress -Subnet $frontendSubnet;

# Create a public IP address.

$publicIP = New-AzPublicIpAddress `
    -ResourceGroupName $rgname `
    -Location $loc `
    -AllocationMethod Static `
    -Sku "Standard" `
    -IpAddressVersion "IPv4" `
    -Name "myLBPublicIP";

# Create a frontend and backend IP pool.

$frontendIP = New-AzLoadBalancerFrontendIpConfig `
    -Name "myFrontEndPool" `
    -PublicIpAddress $publicIP;

$backendPool = New-AzLoadBalancerBackendAddressPoolConfig -Name "myBackEndPool" ;

# Create the load balancer.

$lb = New-AzLoadBalancer `
    -ResourceGroupName $rgname `
    -Name "myLoadBalancer" `

```

```
-Sku "Standard" `
-Tier "Regional" `
-Location $loc `
-FrontendIpConfiguration $frontendIP `
-BackendAddressPool $backendPool;
```

Create a load balancer health probe for TCP port 80.

```
Add-AzLoadBalancerProbeConfig -Name "myHealthProbe" `
```

```
-LoadBalancer $lb `
-Protocol TCP `
-Port 80 `
-IntervalInSeconds 15 `
-ProbeCount 2;
```

Create a load balancer rule to distribute traffic on port TCP 80.

The health probe from the previous step is used to make sure that traffic is

only directed to healthy VM instances.

```
Add-AzLoadBalancerRuleConfig `
```

```
-Name "myLoadBalancerRule" `
-LoadBalancer $lb `
-FrontendIpConfiguration $lb.FrontendIpConfigurations[0] `
-BackendAddressPool $lb.BackendAddressPools[0] `
-Protocol TCP `
-FrontendPort 80 `
-BackendPort 80 `
-DisableOutboundSNAT `
-Probe (Get-AzLoadBalancerProbeConfig -Name "myHealthProbe" -LoadBalancer $lb);
```

Add outbound connectivity rule.

```
Add-AzLoadBalancerOutboundRuleConfig `
```

```
-Name "outboundrule" `
-LoadBalancer $lb `
-AllocatedOutboundPort '10000' `
```

```

-Protocol 'All' `

-IdleTimeoutInMinutes '15' `

-FrontendIpConfiguration $lb.FrontendIpConfigurations[0] `

-BackendAddressPool $lb.BackendAddressPools[0];

# Update the load balancer configuration.

Set-AzLoadBalancer -LoadBalancer $lb;

# Create IP address configurations.

# Instances will require explicit outbound connectivity, for example
# - NAT Gateway on the subnet (recommended)
# - Instances in backend pool of Standard LB with outbound connectivity rules
# - Public IP address on each instance
# See aka.ms/defaultoutboundaccess for more info.

$ipConfig = New-AzVmssIpConfig `
    -Name "myIPConfig" `
    -SubnetId $virtualNetwork.Subnets[0].Id `
    -LoadBalancerBackendAddressPoolsId $lb.BackendAddressPools[0].Id `
    -Primary;

# Create a config object.

# The Vmss config object stores the core information for creating a scale set.

$vmssConfig = New-AzVmssConfig `
    -Location $loc `
    -SkuCapacity $vmssInstanceCount `
    -SkuName $vmssSku `
    -OrchestrationMode 'Flexible' `
    -PlatformFaultDomainCount 1;

# Reference a virtual machine image from the gallery.

Set-AzVmssStorageProfile $vmssConfig `
    -OsDiskCreateOption "FromImage" `
    -ImageReferencePublisher "MicrosoftWindowsServer" `

```

```
-ImageReferenceOffer "WindowsServer" `
-ImageReferenceSku "2022-datacenter-azure-edition-core-smalldisk" `
-ImageReferenceVersion "latest";
```

Set up information for authenticating with the virtual machine.

```
Set-AzVmssOsProfile $vmssConfig `
-AdminUsername $cred.UserName `
-AdminPassword $cred.Password `
-ComputerNamePrefix $vmNamePrefix `
-WindowsConfigurationProvisionVMAgent $true `
-WindowsConfigurationPatchMode "AutomaticByPlatform" `
-EnableHotpatching;
```

Attach the virtual network to the config object.

```
Add-AzVmssNetworkInterfaceConfiguration `
-VirtualMachineScaleSet $vmssConfig `
-Name "network-config" `
-Primary $true `
-IPConfiguration $ipConfig `
-NetworkApiVersion '2020-11-01';
```

Define the Application Health extension properties.

```
$publicConfig = @{ "protocol" = "http"; "port" = 80; "requestPath" = "/healthEndpoint" };
$extensionName = "myHealthExtension";
$extensionType = "ApplicationHealthWindows";
$publisher = "Microsoft.ManagedServices";
```

Add the Application Health extension to the scale set model.

```
Add-AzVmssExtension -VirtualMachineScaleSet $vmssConfig `
-Name $extensionName `
-Publisher $publisher `
-Setting $publicConfig `
-Type $extensionType `
-TypeHandlerVersion "1.0" `
```

```
-AutoUpgradeMinorVersion $True;
```

```
# Create the virtual machine scale set.
```

```
$vmss = New-AzVmss `
```

```
-ResourceGroupName $rgname `
```

```
-Name $vmssName `
```

```
-VirtualMachineScaleSet $vmssConfig;
```

Set the operating system profile properties for a Vmss in Flexible mode with Hotpatching enabled

RELATED LINKS

Online Version: <https://learn.microsoft.com/powershell/module/az.compute/set-azvmssosprofile>

Add-AzVMAdditionalUnattendContent

Add-AzVmssWinRMListener

Add-AzVMSshPublicKey

Add-AzVmssSecret

New-AzVmssConfig