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

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

NAME

Set-AzVmssSecurityProfile

SYNOPSIS

This cmdlet allows users to set the SecurityType enum for Virtual Machines scale sets.

SYNTAX

Set-AzVmssSecurityProfile [-VirtualMachineScaleSet] <Microsoft.Azure.Commands.Compute.Automation.Models.PSVirtualMachineScaleSet> [[-SecurityType] <System.String>] [-DefaultProfile <Microsoft.Azure.Commands.Common.Authentication.Abstractions.Core.IAzureContextContainer>] [<CommonParameters>]

DESCRIPTION

Sets the Security Type of the VMSS

PARAMETERS

-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

-SecurityType <System.String>

Parameter to set the SecurityType on the VMs of the scale set.

Required? false

Position? 1

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

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

The virtual machine scale set profile.

Required? true

Position? 0

Default value None

Accept pipeline input? True (ByPropertyName, ByValue)

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

Microsoft.Azure.Management.Compute.Models.SecurityTypes

OUTPUTS

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

NOTES

---------- Example 1 ----------

```
$VMSS = Get-AzVmss -ResourceGroupName "ResourceGroup11" -VMScaleSetName "ContosoVM07"
$VMSS = Set-AzVmssSecurityProfile -VirtualMachineScaleSet $VMSS -SecurityType "TrustedLaunch"
```

The first command gets the virtual machine scale set named ContosoVM07 by using Get-AzVmss . The command stores it in the $VMSS variable. The second command sets the
SecurityType enum to "TrustedLaunch".

Example 2: Create a Confidential Vmss resource with encryption type VMGuestStateOnly.

```
# Common Variables
  $rgname = <Resource Group Name>
  $loc = "northeurope"
  New-AzResourceGroup -Name $rgname -Location $loc -Force
```

```
$vmssSize = "Standard_DC2as_v5"
$PublisherName = "MicrosoftWindowsServer"
$Offer = "WindowsServer"
$SKU = '2022-datacenter-smalldisk-g2'
$version = "latest"
$securityType = "ConfidentialVM"
$securityEncryptionType = "VMGuestStateOnly"
$secureboot = $true
$vtpm = $true

# 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" -Subnet $subnet
$vnet = Get-AzVirtualNetwork -Name ('vnet' + $rgname) -ResourceGroupName $rgname
$subnetId = $vnet.Subnets[0].Id

# New VMSS Parameters
$vmssName = 'vmss' + $rgname
$adminUsername = <User Name>
$adminPassword = <Password> | ConvertTo-SecureString -AsPlainText -Force

$imgRef = New-Object -TypeName 'Microsoft.Azure.Commands.Compute.Models.PSVirtualMachineImage'
$imgRef.PublisherName = $PublisherName
$imgRef.Offer = $Offer
$imgRef.Skus = $SKU
$imgRef.Version = $version

$vmssIPName = <IP Name>
$vmssNICName = <NIC Name>
$computerNamePrefix = <Name Prefix>
$ipCfg = New-AzVmssIpConfig -Name $vmssIPName -SubnetId $subnetId
```

```
$vmss = New-AzVmssConfig -Location $loc -SkuCapacity 2 -SkuName $vmssSize -UpgradePolicyMode 'Manual' `
    | Add-AzVmssNetworkInterfaceConfiguration -Name $vmssNICName -Primary $true -IPConfiguration $ipCfg `
        | Set-AzVmssOsProfile -ComputerNamePrefix $computerNamePrefix -AdminUsername $adminUsername -AdminPassword $adminPassword `
            | Set-AzVmssStorageProfile -OsDiskCreateOption 'FromImage' -OsDiskCaching 'ReadOnly' -SecurityEncryptionType $securityEncryptionType `
                -ImageReferenceOffer $imgRef.Offer -ImageReferenceSku $imgRef.Skus -ImageReferenceVersion $imgRef.Version `
                -ImageReferencePublisher $imgRef.PublisherName

# Confidential Vmss required parameters
$vmss = Set-AzVmssSecurityProfile -VirtualMachineScaleSet $vmss -SecurityType $securityType
$vmss = Set-AzVmssUefi -VirtualMachineScaleSet $VMSS -EnableVtpm $vtpm -EnableSecureBoot $secureboot

# Create Vmss
$result = New-AzVmss -ResourceGroupName $rgname -Name $vmssName -VirtualMachineScaleSet $vmss

# Validate
$vmssGet = Get-AzVmss -ResourceGroupName $rgname -Name $vmssName
# SecurityType value can be seen at $vmssGet.VirtualMachineProfile.SecurityProfile.SecurityType
```

Example 3: Create a Confidential Vmss resource with encryption type DiskWithVMGuestState and Image reference Disk Encryption set to EncryptedWithPmk.

```
# Common variables
$rgname = <Resource Group Name>
$loc = "northeurope"
New-AzResourceGroup -ResourceGroupName $rgName -Location $loc -Force
$secureBoot = $true
```

```
$vtpm = $true
$vmssName = "vmss" + $rgname

# VM variables
$vmName = <VM Name>
$vmSize = "Standard_DC2as_v5"
$vmssSize = "Standard_DC2as_v5"

$password = <Password>
$securePassword = $password | ConvertTo-SecureString -AsPlainText -Force
$username = <User Name>
$vmCred = New-Object System.Management.Automation.PSCredential ($username, $securePassword)



$imagePublisher = "MicrosoftWindowsServer"
$imageOffer = "windowsserver"
$imageSku = "2022-datacenter-smalldisk-g2"
$imageVersion = "latest"
$osDiskSecurityType = "DiskwithVMGuestState"
$vmSecurityType = "ConfidentialVM"

# Network variables
$NetworkName = [system.string]::concat($vmName, '-vnet')
$NICName = [system.string]::concat($vmName, '-nic')
$SubnetName = [system.string]::concat($vmName, '-subnet')
$SubnetAddressPrefix = "10.0.0.0/24"
$VnetAddressPrefix = "10.0.0.0/16"

# Setup Network
$SingleSubnet = New-AzVirtualNetworkSubnetConfig -Name $SubnetName -AddressPrefix $SubnetAddressPrefix
$Vnet = New-AzVirtualNetwork -Name $NetworkName -ResourceGroupName $rgName `
    -Location $loc -AddressPrefix $VnetAddressPrefix -Subnet $SingleSubnet
$NIC = New-AzNetworkInterface -Name $NICName -ResourceGroupName $rgName `
```

```
    -Location $loc -SubnetId $Vnet.Subnets[0].Id

# Setup CVM
$virtualMachine = New-AzVMConfig -VMName $vmName -VMSize $vmSize
$VirtualMachine = Set-AzVMOperatingSystem -VM $VirtualMachine -Windows -ComputerName $vmName `
        -Credential $vmCred -ProvisionVMAgent -EnableAutoUpdate
$VirtualMachine = Add-AzVMNetworkInterface -VM $VirtualMachine -Id $NIC.Id
$VirtualMachine = Set-AzVMSourceImage -VM $VirtualMachine -PublisherName $imagePublisher `
        -Offer $imageOffer -Skus $imageSku -Version $imageVersion
$VirtualMachine = Set-AzVMOSDisk -VM $VirtualMachine -StorageAccountType "StandardSSD_LRS" `
        -CreateOption "FromImage" -SecurityEncryptionType $osDiskSecurityType
$VirtualMachine = Set-AzVMSecurityProfile -VM $VirtualMachine -SecurityType $vmSecurityType
$VirtualMachine = Set-AzVMUefi -VM $VirtualMachine -EnableVtpm $true -EnableSecureBoot $true

New-AzVM -ResourceGroupName $rgName -Location $loc -VM $VirtualMachine

$cvm = Get-AzVM -VMName $vmName -ResourceGroupName $rgName

# Image Gallery variables
$galleryName = "rg" + $rgname
$definitionName = "def"+$rgname
$publisherName = "cvm01"
$versionName = "1.0.0"
# Platform Managed Key encryption
$cvmEncryptionType = "EncryptedWithPmk"
$replicaCount = 1
$storageAccountType = "Standard_LRS"
$osState = "Specialized"
$osType = "Windows"
$sourceImageId = $cvm.Id

# Setup Image Gallery
New-AzGallery -ResourceGroupName $rgName -Name $galleryName -location $loc
```

```
$imagePublisher = "MicrosoftWindowsServer"
$imageOffer = "windowsserver"
$imageSku = "2022-datacenter-smalldisk-g2"
$vmSecurityType = "ConfidentialVM"

# Setup Image Definition
$SecurityTypeTable = @{Name='SecurityType';Value='ConfidentialVM'}
$features = @($SecurityTypeTable)
New-AzGalleryImageDefinition -ResourceGroupName $rgName -GalleryName $galleryName -Name $definitionName `
   -Feature $features -Publisher $imagePublisher -Offer $imageOffer -Sku $imageSku -location $loc `
   -OsState $osState -OsType $osType -HyperVGeneration 'V2'

 $galDefinition = Get-AzGalleryImageDefinition -ResourceGroupName $rgname -GalleryName $galleryName -Name $definitionName

# Setup Image Version
$cvmOsDiskEncryption = @{CVMEncryptionType=$cvmEncryptionType}
$cvmEncryption = @{OSDiskImage = $cvmOsDiskEncryption}
 $region = @{Name = $loc; ReplicaCount = $replicaCount; StorageAccountType = $storageAccountType; Encryption = $cvmEncryption}
$targetRegions = @($region)

 # Pause the script to ensure the referenced VM is in the Succeeded state. The amount of time can vary and this just a precaution.
Start-Sleep -Seconds 360

                    New-AzGalleryImageVersion -ResourceGroupName $rgName -GalleryName $galleryName -GalleryImageDefinitionName $definitionName `
   -Name $versionName -Location $loc -SourceImageId $sourceImageId -ReplicaCount $replicaCount `
   -StorageAccountType $storageAccountType -TargetRegion $targetRegions

          $galVersion = Get-AzGalleryImageVersion -ResourceGroupName $rgname -GalleryName $galleryName
```

```
-GalleryImageDefinitionName $definitionName

    # NRP for vmss setup. This is not required if you want to reuse the previous NRP setup.
    $subnet = New-AzVirtualNetworkSubnetConfig -Name ('subnet' + $rgname) -AddressPrefix $SubnetAddressPrefix
        $vnet = New-AzVirtualNetwork -Force -Name ('vnet' + $rgname) -ResourceGroupName $rgname -Location $loc
-AddressPrefix $VnetAddressPrefix -Subnet $subnet
    $subnetId = $vnet.Subnets[0].Id
    $vmssIPName = <IP Name>
    $vmssNICName = <NIC Name>

    $ipCfg = New-AzVmssIpConfig -Name $vmssIPName -SubnetId $subnetId

    # Vmss setup
    $securityEncryptionType = ''DiskWithVMGuestState''
        $vmss = New-AzVmssConfig -Location $loc -SkuCapacity 2 -SkuName $vmssSize -UpgradePolicyMode 'Manual'
-ImageReferenceId $galDefinition.Id`
        | Add-AzVmssNetworkInterfaceConfiguration -Name $vmssNICName -Primary $true -IPConfiguration $ipCfg `
                    | Set-AzVmssStorageProfile -OsDiskCreateOption 'FromImage' -OsDiskCaching 'ReadOnly'
-SecurityEncryptionType $securityEncryptionType

    # Confidential Vmss required parameters
    $vmss = Set-AzVmssSecurityProfile -VirtualMachineScaleSet $vmss -SecurityType $vmSecurityType
    $vmss = Set-AzVmssUefi -VirtualMachineScaleSet $VMSS -EnableVtpm $vtpm -EnableSecureBoot $secureboot

    # Create Vmss
    $result = New-AzVmss -ResourceGroupName $rgname -Name $vmssName -VirtualMachineScaleSet $vmss

    # Validate
    $vmssGet = Get-AzVmss -ResourceGroupName $rgname -Name $vmssName
    # Verify the Vmss SecurityType at $vmssGet.VirtualMAchineProfile.SecurityProfile.SecurityType

    $vmssvms = Get-AzVmssVM -ResourceGroupName $rgname -VMScaleSetName $vmssName
        $vmssvm = Get-AzVmssVM -ResourceGroupName $rgname -VMScaleSetName $vmssName -InstanceId
```

$vmssvms[0].InstanceId

Verify the SecurityEncryptionType at $vmssvm.StorageProfile.OsDIsk.ManagedDisk.SecurityProfile.SecurityEncryptionType

Example 4: Create a Confidential Vmss resource with encryption type DiskWithVMGuestState and Image reference Disk Encryption set to EncryptedWithCmk.

```
# Common Variables
    $rgname = <Resource Group Name>;
    $loc = "northeurope";

    New-AzResourceGroup -ResourceGroupName $rgName -Location $loc -Force;

    $secureBoot = $true;
    $vtpm = $true;
    $vmssName = "vmss" + $rgname;

    # VM variables
    $vmName = "v" + $rgname;
    $vmSize = "Standard_DC2as_v5";
    $vmssSize = "Standard_DC2as_v5";
    $password = <Password>;
    $securePassword = $password | ConvertTo-SecureString -AsPlainText -Force;
    $username = <Username>;
    $vmCred = New-Object System.Management.Automation.PSCredential ($username, $securePassword);
    $imagePublisher = "MicrosoftWindowsServer";
    $imageOffer = "windowsserver";
    $imageSku = "2022-datacenter-smalldisk-g2";
    $imageVersion = "latest";
    $osDiskSecurityType = "DiskwithVMGuestState";
```

```
$vmSecurityType = "ConfidentialVM";
$deployCMK = $true;
$storageType = "StandardSSD_LRS";

# Network variables
$NetworkName = $vmname + "-vnet";
$NICName = $vmName + "-nic";
$SubnetName = $vmName + "-subnet";
$SubnetAddressPrefix = "10.0.0.0/24";
$VnetAddressPrefix = "10.0.0.0/16";

# Key Vault setup
$keyVaultName = "kv" + $rgname;
$keyName = "k" + $rgname;
$desName = "des" + $rgname;
$cvmAgent = Get-AzADServicePrincipal -ApplicationId "bf7b6499-ff71-4aa2-97a4-f372087be7f0";
    $kv = New-AzKeyVault -Name $keyVaultName -ResourceGroupName $rgName -Location $loc -Sku "Premium" -EnablePurgeProtection -SoftDeleteRetentionInDays 7;
    Set-AzKeyVaultAccessPolicy -ObjectId $cvmAgent.Id -VaultName $keyVaultName -ResourceGroupName $rgName -PermissionsToKeys "get","release";
                                                                    Start-BitsTransfer -Source https://cvmprivatepreviewsa.blob.core.windows.net/cvmpublicpreviewcontainer/skr-policy.json -Destination ".\skr-policy.json";
    $desKey = Add-AzKeyVaultKey -Name $keyName -VaultName $keyVaultName -KeyOps "wrapKey","unwrapKey" -KeyType "RSA-HSM" -Size 3072 `
        -Exportable -ReleasePolicyPath ".\skr-policy.json" -Destination "HSM";

    $desConfig = New-AzDiskEncryptionSetConfig -Location $loc -KeyUrl $desKey.Id -SourceVaultId $kv.ResourceId -IdentityType "SystemAssigned" `
    -EncryptionType "ConfidentialVmEncryptedWithCustomerKey";
        $des = New-AzDiskEncryptionSet -DiskEncryptionSet $desConfig -DiskEncryptionSetName $desName -ResourceGroupName $rgName;
$desIdentity = Get-AzADServicePrincipal -ObjectId $des.Identity.PrincipalId -ErrorAction 'SilentlyContinue';
```

```
Set-AzKeyVaultAccessPolicy -ObjectId $des.Identity.PrincipalId -ResourceGroupName $rgName -VaultName $keyVaultName -PermissionsToKeys "wrapKey","unwrapKey","get";

$des = Get-AzDiskEncryptionSet -ResourceGroupName $rgname -Name $desName;

# Setup Network
$SingleSubnet = New-AzVirtualNetworkSubnetConfig -Name $SubnetName -AddressPrefix $SubnetAddressPrefix;
$Vnet = New-AzVirtualNetwork -Name $NetworkName -ResourceGroupName $rgName `
   -Location $loc -AddressPrefix $VnetAddressPrefix -Subnet $SingleSubnet
$NIC = New-AzNetworkInterface -Name $NICName -ResourceGroupName $rgName `
   -Location $loc -SubnetId $Vnet.Subnets[0].Id;

# Setup Confidential VM
$virtualMachine = New-AzVMConfig -VMName $vmName -VMSize $vmSize;
$VirtualMachine = Set-AzVMOperatingSystem -VM $VirtualMachine -Windows -ComputerName $vmName `
        -Credential $vmCred -ProvisionVMAgent -EnableAutoUpdate;
$VirtualMachine = Add-AzVMNetworkInterface -VM $VirtualMachine -Id $NIC.Id;
$VirtualMachine = Set-AzVMSourceImage -VM $VirtualMachine -PublisherName $imagePublisher `
       -Offer $imageOffer -Skus $imageSku -Version $imageVersion;

$paramSetAzVmOsDisk = @{
   VM = $virtualMachine
   StorageAccountType = $storageType
   CreateOption = "FromImage"
   SecurityEncryptionType = $osDiskSecurityType
   ErrorAction = 'Stop'
   SecureVMDiskEncryptionSet = $des.Id
};
$VirtualMachine = Set-AzVMOSDisk @paramSetAzVmOsDisk;

$VirtualMachine = Set-AzVMSecurityProfile -VM $VirtualMachine -SecurityType $vmSecurityType;
$VirtualMachine = Set-AzVMUefi -VM $VirtualMachine -EnableVtpm $true -EnableSecureBoot $true;
```

```
# Create CVM to be used as Image reference
New-AzVM -ResourceGroupName $rgName -Location $loc -VM $VirtualMachine;
$cvm = Get-AzVM -VMName $vmName -ResourceGroupName $rgName;

# Image Gallery variables
$galleryName = "gal" + $rgname;
$definitionName = "def"+$rgname;
$publisherName = <Publisher Name>;
$versionName = "1.0.0";
# Customer Managed Key encryption
$cvmEncryptionType = "EncryptedWithCmk"
$replicaCount = 1;
$storageAccountType = "Standard_LRS";
$osState = "Specialized";
$osType = "Windows";
$sourceImageId = $cvm.Id;

# Setup Image Gallery
New-AzGallery -ResourceGroupName $rgName -Name $galleryName -location $loc;

# Setup Image Definition
$SecurityTypeTable = @{Name='SecurityType';Value='ConfidentialVM'};
$features = @($SecurityTypeTable);
New-AzGalleryImageDefinition -ResourceGroupName $rgName -GalleryName $galleryName -Name $definitionName `
   -Feature $features -Publisher $imagePublisher -Offer $imageOffer -Sku $imageSku -location $loc `
   -OsState $osState -OsType $osType -HyperVGeneration 'V2';

  $galDefinition = Get-AzGalleryImageDefinition -ResourceGroupName $rgname -GalleryName $galleryName -Name
$definitionName;

# Setup Image Version
$cvmOsDiskEncryption = @{CVMEncryptionType=$cvmEncryptionType; };
$cvmOsDiskEncryption.Add('CVMDiskEncryptionSetID', $des.Id);
```

```
$cvmEncryption = @{OSDiskImage = $cvmOsDiskEncryption};
$region = @{Name = $loc; ReplicaCount = $replicaCount; StorageAccountType = $storageAccountType; Encryption = $cvmEncryption};
$targetRegions = @($region);

# Pause the script to ensure the referenced VM is in the Succeeded state. The amount of time can vary and this just a precaution.
Start-Sleep -Seconds 360;

New-AzGalleryImageVersion -ResourceGroupName $rgName -GalleryName $galleryName -GalleryImageDefinitionName $definitionName `
-Name $versionName -Location $loc -SourceImageId $sourceImageId -ReplicaCount $replicaCount `
-StorageAccountType $storageAccountType -TargetRegion $targetRegions;

$galVersion = Get-AzGalleryImageVersion -ResourceGroupName $rgname -GalleryName $galleryName -GalleryImageDefinitionName $definitionName;

$securityEncryptionType = ''DiskWithVMGuestState'';

# NRP Vmss setup
$subnet = New-AzVirtualNetworkSubnetConfig -Name ('subnet2' + $rgname) -AddressPrefix $SubnetAddressPrefix;
$vnet = New-AzVirtualNetwork -Force -Name ('vnet2' + $rgname) -ResourceGroupName $rgname -Location $loc -AddressPrefix $VnetAddressPrefix -Subnet $subnet;
$vnet = Get-AzVirtualNetwork -Name ('vnet2' + $rgname) -ResourceGroupName $rgname;
$subnetId = $vnet.Subnets[0].Id;

$vmssIPName = <IP Name>
$vmssNICName = <NIC Name>
$ipCfg = New-AzVmssIpConfig -Name $vmssIPName -SubnetId $subnetId;

# Vmss setup
$vmss = New-AzVmssConfig -Location $loc -SkuCapacity 2 -SkuName $vmssSize -UpgradePolicyMode 'Manual' -ImageReferenceId $galDefinition.Id`
```

```
        | Add-AzVmssNetworkInterfaceConfiguration -Name $vmssNICName -Primary $true -IPConfiguration $ipCfg `
                        | Set-AzVmssStorageProfile -OsDiskCreateOption 'FromImage' -OsDiskCaching 'ReadOnly'
-SecurityEncryptionType $securityEncryptionType
    -SecureVMDiskEncryptionSet $des.Id;

        # Confidential Vmss required parameters
        $vmss = Set-AzVmssSecurityProfile -VirtualMachineScaleSet $vmss -SecurityType $vmSecurityType;
        $vmss = Set-AzVmssUefi -VirtualMachineScaleSet $VMSS -EnableVtpm $vtpm -EnableSecureBoot $secureboot;

        # Create Vmss
        $result = New-AzVmss -ResourceGroupName $rgname -Name $vmssName -VirtualMachineScaleSet $vmss;

        # Validate
        $vmssGet = Get-AzVmss -ResourceGroupName $rgname -Name $vmssName;
        # Verify Vmss SecurityType at $vmssGet.VirtualMachineProfile.SecurityProfile.SecurityType;

        $vmssvms = Get-AzVmssVM -ResourceGroupName $rgname -VMScaleSetName $vmssName;
                $vmssvm = Get-AzVmssVM -ResourceGroupName $rgname -VMScaleSetName $vmssName -InstanceId
$vmssvms[0].InstanceId;
                                                        # Verify the SEcurityEncryptionType at
$vmssvm.StorageProfile.OsDIsk.ManagedDisk.SecurityProfile.SecurityEncryptionType;

                                                    # Verify the Gallery Version encyrption at
$galVersion.PublishingProfile.TargetRegions.Encryption.OSDiskImage.SecurityProfile.ConfidentialVMEncryptionType
    $cvmEncryptionType;
```

RELATED LINKS

Online Version: https://learn.microsoft.com/powershell/module/az.compute/set-azvmsssecurityprofile