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

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

NAME

New-AzGalleryImageVersion

SYNOPSIS

Create a gallery image version.

SYNTAX

New-AzGalleryImageVersion [-ResourceGroupName] <System.String> [-GalleryName] <System.String> [-GalleryImageDefinitionName] <System.String> [-Name] <System.String>

[-AsJob] [-DataDiskImage <Microsoft.Azure.Management.Compute.Models.GalleryDataDiskImage[]>] [-DefaultProfile

<Microsoft.Azure.Commands.Common.Authentication.Abstractions.Core.IAzureContextContainer>] -Location <System.String> [-OSDiskImage

<Microsoft.Azure.Management.Compute.Models.GalleryOSDiskImage>] [-PublishingProfileEndOfLifeDate <System.DateTime>] [-PublishingProfileExcludeFromLatest]

[-ReplicaCount <System.Int32>] [-SourceImageId <System.String>] [-SourceImageVMId <System.String>] [-StorageAccountType <System.String>] [-Tag

<System.Collections.Hashtable>] [-TargetExtendedLocation <System.Collections.Hashtable[]>] [-TargetRegion <System.Collections.Hashtable[]>] [-Confirm] [-WhatIf]

[<CommonParameters>]

DESCRIPTION

Create a gallery image version.

PARAMETERS

-AsJob <System.Management.Automation.SwitchParameter>

Run cmdlet in the background

Required? false

Position? named

Default value False

Accept pipeline input? False

Accept wildcard characters? false

-DataDiskImage <Microsoft.Azure.Management.Compute.Models.GalleryDataDiskImage[]>

Data disk images. e.g. @{Source = @{Id = <source_id>}; Lun = 1; SizeInGB = 100; HostCaching = "ReadOnly" }

Required? false

Position? named

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

-GalleryImageDefinitionName <System.String>

The name of the gallery.

Required? true

Position? 2

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-GalleryName <System.String>

The name of the gallery.

Required? true

Position? 1

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-Location <System.String>

Resource location

Required? true

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-Name <System.String>

The name of the gallery image version.

Required? true

Position? 3

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-OSDiskImage <Microsoft.Azure.Management.Compute.Models.GalleryOSDiskImage>

OS disk image e.g. @{Source = @{Id = <source_id>}; SizeInGB = 100; HostCaching = "ReadOnly" }

Required? false

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-PublishingProfileEndOfLifeDate <System.DateTime>

The end of life date of the gallery Image Version.

Required? false

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-PublishingProfileExcludeFromLatest <System.Management.Automation.SwitchParameter>

If it is set, Virtual Machines deployed from the latest version of the Image Definition won't use this Image Version.

Required? false

Position? named

Default value False

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-ReplicaCount <System.Int32>

The number of replicas of the Image Version to be created per region.

Required? false

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-ResourceGroupName <System.String>

The name of the resource group.

Required? true

Position? 0

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-SourceImageId <System.String>

The ID of the source image from which the Image Version is going to be created.

Required? false

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-SourceImageVMId <System.String>

The resource Id of the source virtual machine. Only required when capturing a virtual machine to source this Gallery Image Version.

Required? false

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-StorageAccountType <System.String>

Specifies the storage account type to be used to store the image. This property is not updatable. Available values are Standard_LRS, Standard_ZRS and Premium_LRS.

Required? false

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-Tag <System.Collections.Hashtable>

Resource tags

Required? false

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-TargetExtendedLocation <System.Collections.Hashtable[]>

The target extended locations where the Image Version is going to be replicated to. This property is updatable.

Required? false

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-TargetRegion <System.Collections.Hashtable[]>

The target regions where the Image Version is going to be replicated to.

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

System.Collections.Hashtable

System.Int32

System.Management.Automation.SwitchParameter

System.DateTime

System.Collections.Hashtable[]

OUTPUTS

Microsoft.Azure.Commands.Compute.Automation.Models.PSGalleryImageVersion

NOTES

-- Example 1: Create an image version from a virtual machine --

$rgName = "myResourceGroup"

$galleryName = "myGallery"

$galleryImageDefinitionName = "myImage"

$galleryImageVersionName = "1.0.0"

$location = "eastus"

$sourceImageId = "/subscriptions/00000000-0000-0000-0000-000000000000/resourceGroups/myVMRG/providers/Microsoft.Compute/virtualMachines/myVM"

New-AzGalleryImageVersion -ResourceGroupName $rgName -GalleryName $galleryName -GalleryImageDefinitionName $galleryImageDefinitionName -Name $galleryImageVersionName

-Location $location -SourceImageId $sourceImageId

Add a new image version from a virtual machine into the image definition.

--- Example 2: Create an image version from a managed image ---

$rgName = "myResourceGroup"

$galleryName = "myGallery"

$galleryImageDefinitionName = "myImage"

$galleryImageVersionName = "1.0.0"

$location = "eastus"

$sourceImageId = "/subscriptions/00000000-0000-0000-0000-000000000000/resourceGroups/myImageRG/providers/Microsoft.Compute/images/myImage"

$storageAccountType = "StandardSSD_LRS"

New-AzGalleryImageVersion -ResourceGroupName $rgName -GalleryName $galleryName -GalleryImageDefinitionName $galleryImageDefinitionName -Name $galleryImageVersionName

-Location $location -StorageAccountType $storageAccountType -SourceImageId $sourceImageId

Add a new image version from a managed image into the image definition.

Example 3: Create an image version from an another image version

$rgName = "myResourceGroup"

$galleryName = "myGallery"

$galleryImageDefinitionName = "myImage"

$galleryImageVersionName = "1.0.0"

$location = "eastus"

$sourceImageId = "/subscriptions/00000000-0000-0000-0000-000000000000/resourceGroups/myImageRG/providers/Microsoft.Compute/galleries/myOtherGallery/images/myImageDefin

ition/versions/1.0.0"

New-AzGalleryImageVersion -ResourceGroupName $rgName -GalleryName $galleryName -GalleryImageDefinitionName $galleryImageDefinitionName -Name $galleryImageVersionName

-Location $location -SourceImageId $sourceImageId

Copy an image version into another image version

---- Example 4: Add a new image version from a managed disk ----

$rgName = "myResourceGroup"

$galleryName = "myGallery"

$galleryImageDefinitionName = "myImage"

$galleryImageVersionName = "1.0.0"

$location = "eastus"

$osDisk = @{Source = @{Id = "/subscriptions/00000000-0000-0000-0000-000000000000/resourceGroups/myDiskRG/providers/Microsoft.Compute/disks/myOSDisk" }}

New-AzGalleryImageVersion -ResourceGroupName $rgName -GalleryName $galleryName -GalleryImageDefinitionName $galleryImageDefinitionName -Name $galleryImageVersionName

-Location $location -OSDiskImage $osDisk

Create an image version from a managed disk

Example 5: Add a new image version from a managed disk and add additional data disks

```
$rgName = "myResourceGroup"
$galleryName = "myGallery"
$galleryImageDefinitionName = "myImage"
$galleryImageVersionName = "1.0.0"
$location = "eastus"
$osDisk = @{Source = @{Id = "/subscriptions/00000000-0000-0000-0000-000000000000/resourceGroups/myDiskRG/providers/Microsoft.Compute/disks/myOSDisk" }}
$dataDisk0 = @{Source = @{Id = "/subscriptions/00000000-0000-0000-0000-000000000000/resourceGroups/myDiskRG/providers/Microsoft.Compute/disks/myDataDisk" }; Lun = 0; }
$dataDisks = @($dataDisk0)
New-AzGalleryImageVersion -ResourceGroupName $rgName -GalleryName $galleryName -GalleryImageDefinitionName $galleryImageDefinitionName -Name $galleryImageVersionName
-Location $location -OSDiskImage $osDisk  -DataDiskImage $dataDisks
```

Create an image version by specifying OS and data disks

Example 6: Add a new image version from a snapshot of an OS disk

```
$rgName = "myResourceGroup"
$galleryName = "myGallery"
$galleryImageDefinitionName = "myImage"
$galleryImageVersionName = "1.0.0"
$location = "eastus"
$osSnapshot = @{Source = @{Id = "/subscriptions/00000000-0000-0000-0000-000000000000/resourceGroups/mySnapshotRG/providers/Microsoft.Compute/snapshots/myOSSnapshot" }}
New-AzGalleryImageVersion -ResourceGroupName $rgName -GalleryName $galleryName -GalleryImageDefinitionName $galleryImageDefinitionName -Name $galleryImageVersionName
-Location $location -OSDiskImage $osSnapshot
```

Create an image version from a disk snapshot

Example 7: Add a new image version from a snapshot of an OS disk and add additional data disks

```
$rgName = "myResourceGroup"
$galleryName = "myGallery"
$galleryImageDefinitionName = "myImage"
$galleryImageVersionName = "1.0.0"
$location = "eastus"
$osSnapshot = @{Source = @{Id = "/subscriptions/00000000-0000-0000-0000-000000000000/resourceGroups/mySnapshotRG/providers/Microsoft.Compute/snapshots/myOSSnapshot" }}
$dataSnapshot0 = @{Source = @{Id = "/subscriptions/00000000-0000-0000-0000-000000000000/resourceGroups/mySnapshotRG/providers/Microsoft.Compute/snapshots/myDataSnapshot" }; Lun = 0; }
$dataDisks = @($dataSnapshot0)
New-AzGalleryImageVersion -ResourceGroupName $rgName -GalleryName $galleryName -GalleryImageDefinitionName $galleryImageDefinitionName -Name $galleryImageVersionName
-Location $location -OSDiskImage $osSnapshot  -DataDiskImage $dataDisks
```

Create an image version by specifying snapshots for OS and data disks.

Example 8: Add a new image version from a combination of disks and snapshots

```
$rgName = "myResourceGroup"
$galleryName = "myGallery"
$galleryImageDefinitionName = "myImage"
$galleryImageVersionName = "1.0.0"
$location = "eastus"
```

$osSnapshot = @{Source = @{Id = "/subscriptions/00000000-0000-0000-0000-000000000000/resourceGroups/mySnapshotRG/providers/Microsoft.Compute/snapshots/myOSSnapshot" }}

$dataDisk0 = @{Source = @{Id = "/subscriptions/00000000-0000-0000-0000-000000000000/resourceGroups/myDiskRG/providers/Microsoft.Compute/disks/myDataDisk" }; Lun = 0; }

$dataDisks = @($dataDisk0)

New-AzGalleryImageVersion -ResourceGroupName $rgName -GalleryName $galleryName -GalleryImageDefinitionName $galleryImageDefinitionName -Name $galleryImageVersionName

-Location $location -OSDiskImage $osSnapshot -DataDiskImage $dataDisks

Create an image version by specifying a snapshot as an OS disk and a managed disk as a data disk.

Example 9: Add a new image version and copy it to additional regions.

$rgName = "myResourceGroup"

$galleryName = "myGallery"

$galleryImageDefinitionName = "myImage"

$galleryImageVersionName = "1.0.0"

$location = "eastus"

$sourceImageId = "/subscriptions/00000000-0000-0000-0000-000000000000/resourceGroups/myVMRG/providers/Microsoft.Compute/virtualMachines/myVM"

$replicaCount = 1

$storageAccountType = "Standard_ZRS"

$region_eastus = @{Name = 'East US';ReplicaCount = 3;StorageAccountType = 'Standard_LRS'}

$region_westus = @{Name = 'West US'}

$region_ukwest = @{Name = 'UK West';ReplicaCount = 2}

$region_southcentralus = @{Name = 'South Central US';StorageAccountType = 'Standard_LRS'}

```
$targetRegions = @($region_eastus, $region_westus, $region_ukwest, $region_southcentralus)

New-AzGalleryImageVersion -ResourceGroupName $rgName -GalleryName $galleryName -GalleryImageDefinitionName $galleryImageDefinitionName -Name $galleryImageVersionName
    -Location $location -SourceImageId $sourceImageId -ReplicaCount 1 -StorageAccountType $storageAccountType -TargetRegion $targetRegions
```

Create an image version in four regions. In this example, the global replica count is 1 and the global storage account type is Standard_ZRS. East US will have 3
replicas, each stored on Standard_LRS account storage. West US will inherit from global settings and have 1 replica stored on Standard_ZRS. UK West will have a
replica count of 2 stored on Standard_ZRS. South Central US will have one replica stored on Standard_LRS.

Example 10: Add a new image version with encryption in multiple regions

```
$rgName = "myResourceGroup"
$galleryName = "myGallery"
$galleryImageDefinitionName = "myImage"
$galleryImageVersionName = "1.0.0"
$location = "eastus"
$sourceImageId = "/subscriptions/00000000-0000-0000-0000-000000000000/resourceGroups/myVMRG/providers/Microsoft.Compute/virtualMachines/myVM"
$replicaCount = 1
$storageAccountType = "Standard_ZRS"

# East US regional settings
$eastUSdes = "/subscriptions/00000000-0000-0000-0000-000000000000/resourceGroups/myDESrg/providers/Microsoft.Compute/diskEncryptionSets/myEastUSDES"
$encryption_eastus_os = @{DiskEncryptionSetId = $eastUSdes }
$encryption_eastus_dd0 = @{DiskEncryptionSetId = $eastUSdes; Lun = 0 }
```

```
$encryption_eastus_dd = @($encryption_eastus_dd0)
$eastus_encryption = @{OSDiskImage = $eastus_encryption_os; DataDiskImages = $eastus_encryption_dd }
    $region_eastus = @{Name = 'East US';ReplicaCount = 3;StorageAccountType = 'Standard_LRS'; Encryption = $encryption_eastus}

# West US regional settings
                                                                $westUS2des =
"/subscriptions/00000000-0000-0000-0000-000000000000/resourceGroups/myDESrg/providers/Microsoft.Compute/diskEncryptionSets/myWestUSDES"
$encryption_westus_os = @{DiskEncryptionSetId = $westUSdes }
$encryption_westus_dd0 = @{DiskEncryptionSetId = $westUSdes; Lun = 0 }
$encryption_westus_dd = @($encryption_westus_dd0)
$westus_encryption = @{OSDiskImage = $encryption_westus_os; DataDiskImages = $encryption_westus_dd }
$region_westus = @{Name = 'West US'; Encryption = $westus_encryption}

# Create images
$targetRegions = @($region_eastus, $region_westus)
New-AzGalleryImageVersion -ResourceGroupName $rgName -GalleryName $galleryName -GalleryImageDefinitionName $galleryImageDefinitionName -Name $galleryImageVersionName
-Location $location -SourceImageId $sourceImageId -TargetRegion $targetRegions
```

Create an image version with encryption in two regions. Disk encryption sets are regional resources and a different disk encryption set must be used in each region.

Example 11: Create an image version and have it excluded from latest

```
$rgName = "myResourceGroup"
$galleryName = "myGallery"
$galleryImageDefinitionName = "myImage"
$galleryImageVersionName = "1.0.0"
$location = "eastus"
New-AzGalleryImageVersion -ResourceGroupName $rgName -GalleryName $galleryName -GalleryImageDefinitionName
```

$galleryImageDefinitionName -Name $galleryImageVersionName

-Location $location -SourceImageId $sourceImageId -PublishingProfileExcludeFromLatest

Add a new image version into an image definition but exclude it from being considered for latest version within its image definition.

Example 12: Create an image version and set its end-of-life date

$rgName = "myResourceGroup"

$galleryName = "myGallery"

$galleryImageDefinitionName = "myImage"

$galleryImageVersionName = "1.0.0"

$location = "eastus"

$endOfLifeDate = "2024-08-02T00:00:00+00:00"

New-AzGalleryImageVersion -ResourceGroupName $rgName -GalleryName $galleryName -GalleryImageDefinitionName $galleryImageDefinitionName -Name $galleryImageVersionName

-Location $location -SourceImageId $sourceImageId -PublishingProfileEndOfLifeDate $endOfLifeDate

This example has the end-of-life date for image version set to August 2, 2024 at mignight UTC. End-of-life dates can be specified for both the image definitions and

image versions. Image versions can still be used after the end-of-life dates.

--- Example 13: Create an image version for Confidential VM ---

$rgName = "myResourceGroup"

$galleryName = "myGallery"

$galleryImageDefinitionName = "cvmImage"

$galleryImageVersionName = "1.0.0"

$location = "North Europe"

$sourceImageId

"/subscriptions/00000000-0000-0000-0000-000000000000/resourceGroups/myVMRG/providers/Microsoft.Compute/virtualMachines/myVM"

$cvmDiskEncryptionSetId =

"/subscriptions/00000000-0000-0000-0000-000000000000/resourceGroups/myVMRG/providers/Microsoft.Compute/diskEncryptionSets/cvmDiskEncryptionSet"

$dataDiskEncryptionSetId =

"/subscriptions/00000000-0000-0000-0000-000000000000/resourceGroups/myVMRG/providers/Microsoft.Compute/diskEncryptionSets/dataDiskEncryptionSet"

$cvmOsDiskEncryption = @{CVMEncryptionType='EncryptedWithCmk'; CVMDiskEncryptionSetID=$cvmDiskEncryptionSetId}

$cvmDataDiskEncryption_lun0 = @{DiskEncryptionSetId = $dataDiskEncryptionSetId ; Lun = 0}

$cvmDataDiskEncryption = @($cvmDataDiskEncryption_lun0)

$cvmEncryption = @{OSDiskImage = $cvmOsDiskEncryption; DataDiskImages = $cvmDataDiskEncryption}

$region_northEurope = @{Name = 'NorthEurope';ReplicaCount = 3;StorageAccountType = 'Standard_LRS'; Encryption = $cvmEncryption}

$targetRegions = @($region_northEurope)

New-AzGalleryImageVersion -ResourceGroupName $rgName -GalleryName $galleryName -GalleryImageDefinitionName $galleryImageDefinitionName -Name $galleryImageVersionName

-Location $location -SourceImageId $sourceImageId -ReplicaCount 1 -StorageAccountType "Standard_LRS" -TargetRegion $targetRegions

In this example, Confidential VM (CVM) Image with Customer Managed Key (CMK) is created under Image definition which supports ConfidentialVM security type.

Example 14: Create an image version with a target extended location

```
$rgName = "myResourceGroup"
$galleryName = "myGallery"
$galleryImageDefinitionName = "cvmImage"
$galleryImageVersionName = "1.0.0"
$location = "EastUs"
$sourceImageId = "/subscriptions/00000000-0000-0000-0000-000000000000/resourceGroups/myVMRG/providers/Microsoft.Compute/virtualMachines/myVM"

$replicaCount = 1
$extendedLocation = @{Name = 'microsoftlosangeles1';Type='EdgeZone'}
$edgezone_losangeles = @{Location = "westus";ExtendedLocation=$extendedLocation;ReplicaCount = 3;StorageAccountType = 'StandardSSD_LRS'}
$targetExtendedLocations = @($edgezone_losangeles)

New-AzGalleryImageVersion -ResourceGroupName $rgName -GalleryName $galleryName -GalleryImageDefinitionName $galleryImageDefinitionName -Name $galleryImageVersionName
-Location $location -SourceImageId $sourceImageId -ReplicaCount 1 -StorageAccountType "Standard_LRS" -TargetExtendedLocation $targetExtendedLocations
```

This example creates a hashtable for target extended location properties and passes in with -TargetExtendedLocation parameter.

RELATED LINKS

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