Add a custom VM image to Azure Stack Hub

sethmanheim

# Add a custom VM image to Azure Stack Hub

In Azure Stack Hub, as an operator you can add your custom virtual machine (VM) image to the marketplace and make it available to your users. You can add VM images to the Azure Stack Hub Marketplace through the administrator portal or Windows PowerShell. Use either an image from the global Azure Marketplace as a base for your custom image, or create your own using Hyper-V.

A user in the tenant portal in Azure Stack Hub can also add a custom virtual machine (VM) image by following the guidance in step 1. A user can create their custom image as a virtual hard disk (VHD) and upload the image to a storage account on Azure Stack Hub. They can then create a VM from the VHD.

Custom images come in two forms: **generalized** and **specialized**.

* **Generalized image**
* A generalized disk image is one that has been prepared with **Sysprep** to remove any unique information (such as user accounts), enabling it to be reused to create multiple VMs. This is a good option for marketplace items.
* **Specialized image**
* A specialized disk image is a copy of a virtual hard disk (VHD) from an existing VM that contains the user accounts, applications, and other state data from your original VM. This is typically the format in which VMs are migrated to Azure Stack Hub.

## Step 1: Create the custom VM image

### Windows - Create a custom generalized VHD

#### VHD is from outside Azure

Follow the steps in [Upload a generalized VHD and use it to create new VMs in Azure](/azure/virtual-machines/windows/upload-generalized-managed) to correctly **Sysprep** your VHD and make it generalized.

#### VHD is from Azure

Prior to generalizing the VM, make sure of the following:

Before the Azure Stack 1910 release:

* When you provision the VM on Azure, use PowerShell and provision it without the -ProvisionVMAgent flag.
* Remove all VM extensions using the **Remove-AzureRmVMExtension** cmdlet from the VM before generalizing the VM in Azure. You can find which VM extensions are installed by going to Windows (C:) > WindowsAzure > Logs > Plugins.

Remove-AzureRmVMExtension -ResourceGroupName winvmrg1 -VMName windowsvm -Name "CustomScriptExtension"

On or after the Azure Stack 1910 release:

* The preceding steps do not apply to VHDs brought from Azure to an Azure Stack Hub that is on or beyond the 1910 release.

Follow the instructions in [this article](/azure/virtual-machines/windows/download-vhd) to correctly generalize and download the VHD before porting it to Azure Stack Hub.

### Windows - Specialized

Follow the steps [here](/azure/virtual-machines/windows/create-vm-specialized#prepare-the-vm) to prepare the VHD correctly. To deploy VM extensions, make sure that the VM agent .msi available [in this article](/azure/virtual-machines/extensions/agent-windows#manual-installation) is installed in the VM before VM deployment. If the VM agent is not present in the VHD, extension deployment will fail. You do not need to set the OS profile while provisioning, or set $vm.OSProfile.AllowExtensionOperations = $true.

### Linux - Generalized

#### VHD from outside Azure

If the VHD is from outside Azure, follow the appropriate instructions to generalize the VHD:

* [CentOS-based Distributions](/azure/virtual-machines/linux/create-upload-centos?toc=%2fazure%2fvirtual-machines%2flinux%2ftoc.json)
* [Debian Linux](/azure/virtual-machines/linux/debian-create-upload-vhd?toc=%2fazure%2fvirtual-machines%2flinux%2ftoc.json)
* [Red Hat Enterprise Linux](azure-stack-redhat-create-upload-vhd.md)
* [SLES or openSUSE](/azure/virtual-machines/linux/suse-create-upload-vhd?toc=%2fazure%2fvirtual-machines%2flinux%2ftoc.json)
* [Ubuntu Server](/azure/virtual-machines/linux/create-upload-ubuntu?toc=%2fazure%2fvirtual-machines%2flinux%2ftoc.json)

#### VHD from Azure

If the VHD is from Azure, follow these instructions to generalize and download the VHD:

1. Stop the **waagent** service:

* sudo waagent -force -deprovision  
  export HISTSIZE=0  
  logout
* The Azure Linux Agent versions that work with Azure Stack Hub [are documented here](azure-stack-linux.md#azure-linux-agent). Make sure that the sysprepped image has an Azure Linux agent version that is compatible with Azure Stack Hub.

1. Stop deallocate the VM.
2. Download the VHD.
   1. To download the VHD file, generate a shared access signature (SAS) URL. When the URL is generated, an expiration time is assigned to the URL.
   2. On the menu of the blade for the VM, select **Disks**.
   3. Select the operating system disk for the VM, and then select **Disk Export**.
   4. Set the expiration time of the URL to 36000.
   5. Select **Generate URL**.
   6. Generate the URL.
   7. Under the URL that was generated, select **Download the VHD file**.
   8. You might need to select **Save** in the browser to start the download. The default name for the VHD file is **abcd**.
   9. You can now port this VHD to Azure Stack Hub.

[!IMPORTANT]  
You can find a script in the article [Sample script to upload a VHD to Azure and create a new VM](/azure/virtual-machines/scripts/virtual-machines-windows-powershell-upload-generalized-script) to upload the VHD to an Azure Stack Hub user storage account and create a VM. Make sure to provide $urlOfUploadedImageVhd as the Azure Stack Hub storage account+container URL. For a generalized VHD, make sure to use FromImage value when setting -CreateOption FromImage.

### Linux - Specialized

Specialized VHDs should not be used as the base VHD for a marketplace item. Use generalized VHDs for this. However, specialized VHDs are a good fit for when you need to migrate VMs from on-premises to Azure Stack Hub

#### VHD from outside Azure

Step 1: Follow the appropriate instructions to make the VHD suitable for Azure. Use this article until the step to install the Linux agent, and then proceed to step 2 before installing the agent:

* [CentOS-based Distributions](/azure/virtual-machines/linux/create-upload-centos?toc=%2fazure%2fvirtual-machines%2flinux%2ftoc.json)
* [Debian Linux](/azure/virtual-machines/linux/debian-create-upload-vhd?toc=%2fazure%2fvirtual-machines%2flinux%2ftoc.json)
* [Red Hat Enterprise Linux](azure-stack-redhat-create-upload-vhd.md)
* [SLES or openSUSE](/azure/virtual-machines/linux/suse-create-upload-vhd?toc=%2fazure%2fvirtual-machines%2flinux%2ftoc.json)
* [Ubuntu Server](/azure/virtual-machines/linux/create-upload-ubuntu?toc=%2fazure%2fvirtual-machines%2flinux%2ftoc.json)

[!IMPORTANT] Do not run the last step: (sudo waagent -force -deprovision) as this will generalize the VHD.

Step 2: If a Linux specialized VHD is brought from outside of Azure to Azure Stack Hub, to run VM extensions and disable provisioning do the following:

To identify what version of Linux Agent is installed in the source VM image, run the following commands. The version number that describes the provisioning code is WALinuxAgent-, not the Goal state agent:

waagent -version

For example:

waagent -version  
WALinuxAgent-2.2.45 running on centos 7.7.1908  
Python: 2.7.5  
Goal state agent: 2.2.46

To disable the Linux Agent provisioning with Linux Agent lower than 2.2.4, set the following parameters in **/etc/waagent.conf**: Provisioning.Enabled=n, and Provisioning.UseCloudInit=n.

In scenarios in which you want to run extensions:

1. Set the following parameter in **/etc/waagent.conf**:
   * Provisioning.Enabled=n
   * Provisioning.UseCloudInit=n
2. To ensure walinuxagent provisioning is disabled, run: mkdir -p /var/lib/waagent && touch /var/lib/waagent/provisioned
3. If you have cloud-init in your image, disable cloud init:

* touch /etc/cloud/cloud-init.disabled  
  sudo sed -i '/azure\_resource/d' /etc/fstab

1. Execute a logout.

To disable provisioning with Linux Agent 2.2.45 and later, make the following configuration option changes:

* Provisioning.Enabled and Provisioning.UseCloudInit are now ignored.

In this version, currently there is no Provisioning.Agent option to disable provisioning completely; however, you can add the provisioning marker file, and with the following settings, provisioning is ignored:

1. In **/etc/waagent.conf** add this configuration option: Provisioning.Agent=Auto.
2. To ensure walinuxagent provisioning is disabled, run: mkdir -p /var/lib/waagent && touch /var/lib/waagent/provisioned.
3. Disable cloud-init install by running the following:

* touch /etc/cloud/cloud-init.disabled  
  sudo sed -i '/azure\_resource/d' /etc/fstab

1. Log out.

#### VHD from Azure

You can use [this guidance](/azure/virtual-machines/linux/upload-vhd#requirements) to prepare the VHD.

[!IMPORTANT]  
You can use the following PowerShell example to upload the VHD to an Azure Stack Hub user storage account:

# Provide values for the variables  
$resourceGroup = 'myResourceGroup'  
$location = 'Orlando'  
$storageaccount = 'mystorageaccount'  
$storageType = 'Standard\_LRS'  
$storageaccounturl = 'https://resourcegrpabc.blob.orlando.azurestack.corp.microsoft.com/container'  
$localPath = 'C:\Users\Public\Documents\Hyper-V\VHDs\generalized.vhd'  
$vhdName = 'myUploadedVhd.vhd'  
  
New-AzResourceGroup -Name $resourceGroup -Location $location  
New-AzStorageAccount -ResourceGroupName $resourceGroup -Name $storageAccount -Location $location `  
 -SkuName $storageType -Kind "Storage"  
$urlOfUploadedImageVhd = ($storageaccounturl + '/' + $vhdName)  
Add-AzVhd -ResourceGroupName $resourceGroup -Destination $urlOfUploadedImageVhd `  
 -LocalFilePath $localPath

For a specialized VHD, make sure to use “attach” semantics using -CreateOption Attach, [as in this example](/azure/virtual-machines/scripts/virtual-machines-windows-powershell-sample-create-vm-from-managed-os-disks), to create a VM from this VHD.

### Considerations

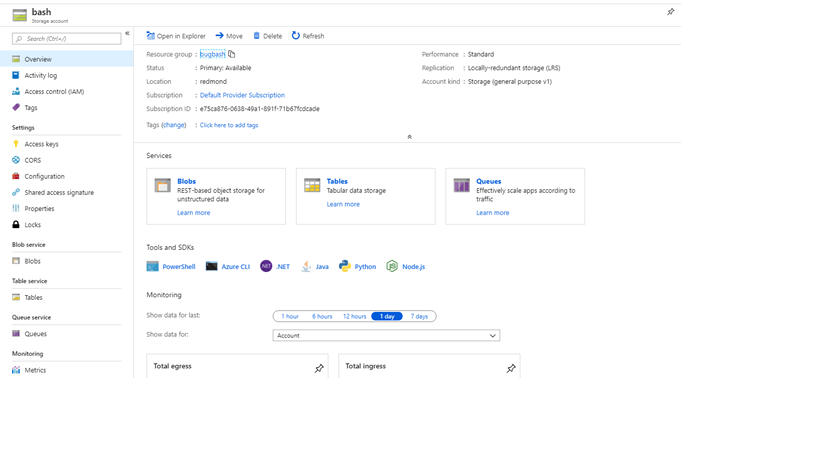
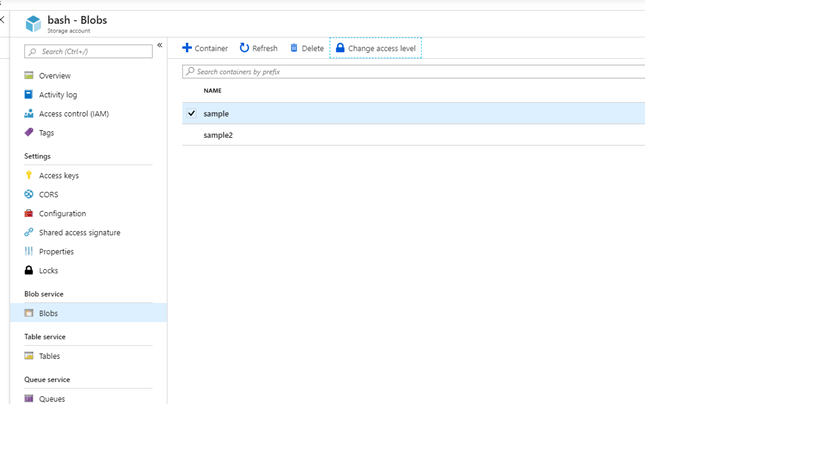
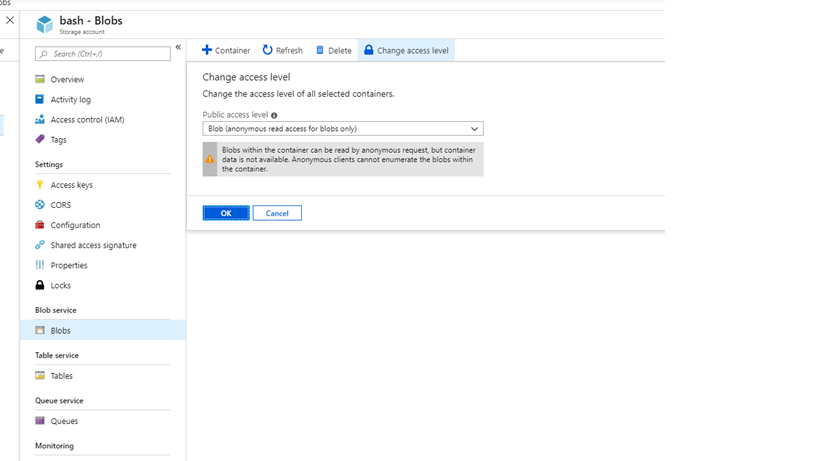
Before you upload the image, it’s important to consider the following:

* Azure Stack Hub only supports generation 1 VMs in the fixed disk VHD format. The fixed-format structures the logical disk linearly within the file, so that disk offset **X** is stored at blob offset **X**. A small footer at the end of the blob describes the properties of the VHD. To confirm if your disk is fixed, use the **Get-VHD** PowerShell cmdlet.
* Azure Stack Hub does not support dynamic disk VHDs.

## Step 2: Upload a storage account

Upload the VM Image to a storage account as an Azure Stack Hub operator.

1. [Install PowerShell for Azure Stack Hub](azure-stack-powershell-install.md).
2. Sign in to Azure Stack Hub as an operator. For instructions, see [Sign in to Azure Stack Hub as an operator](azure-stack-powershell-configure-admin.md).
3. Images must be referenced by a blob storage URI. Prepare a Windows or Linux operating system image in VHD format (not VHDX), and then upload the image to a storage account in Azure Stack Hub.
   * If the VHD is in Azure, you can use a tool such as [Azcopy](/azure/storage/common/storage-use-azcopy) to directly transfer the VHD between Azure and your Azure Stack Hub storage account if you are running on a connected Azure Stack Hub.
   * On a disconnected Azure Stack Hub, if your VHD is in Azure, you must download the VHD to a machine that has connectivity to both Azure and Azure Stack Hub. Then copy the VHD to this machine from Azure before you transfer the VHD to Azure Stack Hub using any of the common [storage data transfer tools](../user/azure-stack-storage-transfer.md) that can be used across Azure and Azure Stack Hub.
   * One such tool used in this example is the **Add-AzureRmVHD** command to upload a VHD to a storage account in the Azure Stack Hub Administrator portal:
   * Add-AzureRmVhd -Destination "https://bash.blob.redmond.azurestack.com/sample/vhdtestingmgd.vhd" -LocalFilePath "C:\vhd\vhdtestingmgd.vhd"
4. Make a note of the blob storage URI where you upload the image. The blob storage URI has the following format: \*\*<storageAccount>/<blobContainer>/<targetVHDName>\*.vhd\*\*.
5. To make the blob anonymously accessible, go to the storage account blob container where the VM image VHD was uploaded. Select **Blob**, and then select **Access policy**. Optionally, you can generate a shared access signature (SAS) for the container, and include it as part of the blob URI. This step ensures the blob is available to be used. If the blob isn’t anonymously accessible, the VM image will be created in a failed state.

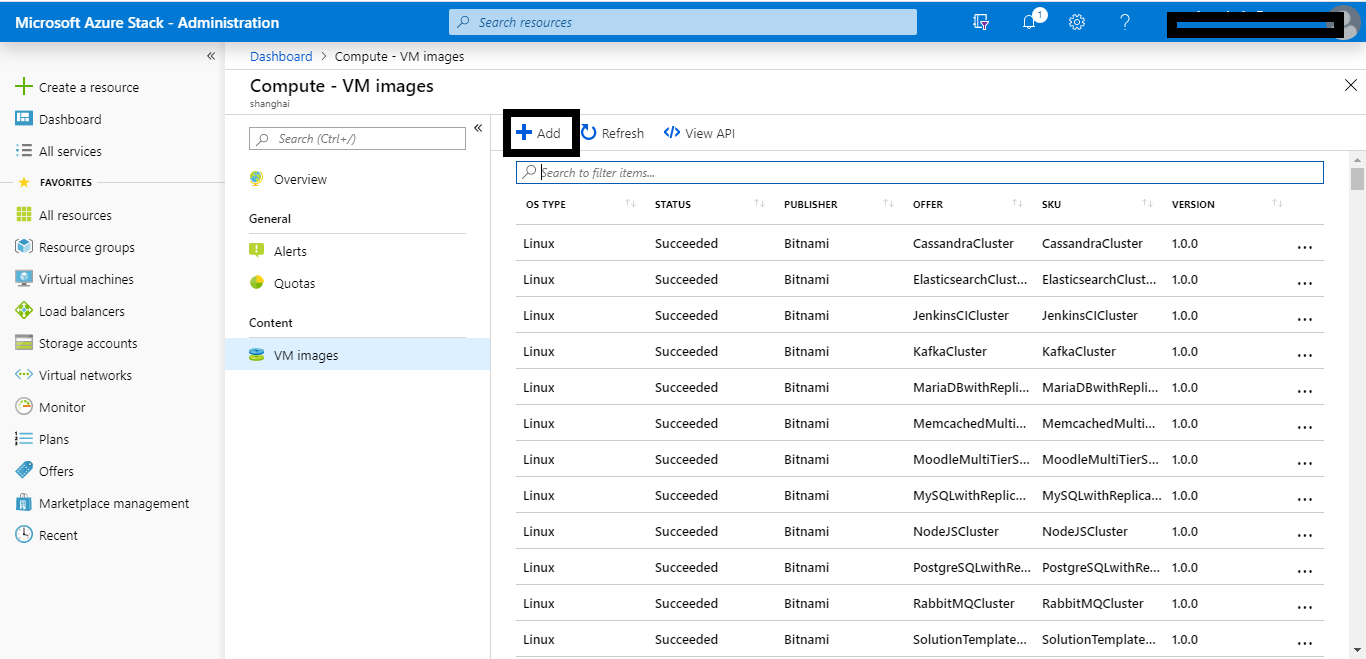
* 
* Go to storage account blobs
* 
* Set blob access to public
* 
* Set blob access to public

1. You can also use the preceding steps to upload the image to a storage account on the user portal (by signing in as the user) and create a VM directly from it. In this case, it will be a custom VHD that is not available through the marketplace. You will also not need to follow step 3.

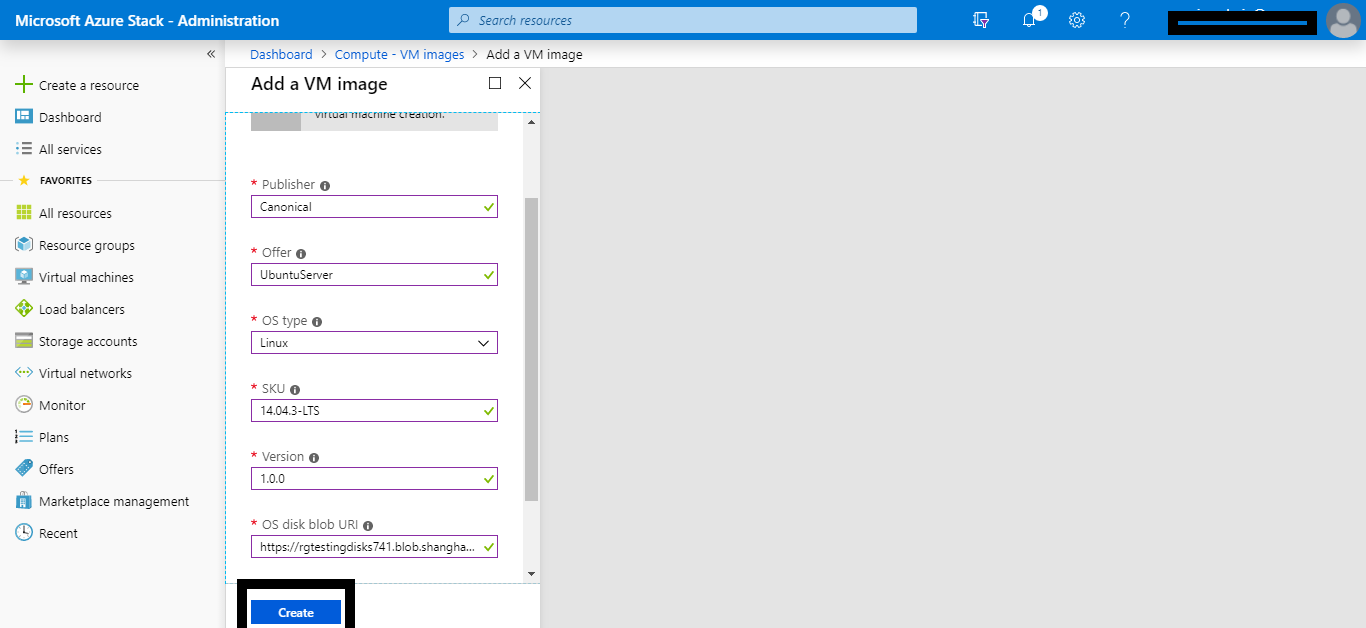
## Step 3, Option 1: Add using the portal

Add the VM image as an Azure Stack Hub operator using the portal.

1. Sign in to Azure Stack Hub as an operator. In the menu, select **All services** > **Compute** under **VM Images** > **Add**.

* 
* Custom image sideloading UI

1. Under **Create image**, enter the **Publisher**, **Offer**, **SKU**, **Version**, and OS disk blob URI. Then, select **Create** to begin creating the VM image.

* 
* Custom image sideloading UI
* When the image is successfully created, the VM image status changes to **Succeeded**.

1. When you add an image, it is only available for Azure Resource Manager-based templates and PowerShell deployments. To make an image available to your users as a marketplace item, publish the marketplace item using the steps in the article [Create and publish a Marketplace item](azure-stack-create-and-publish-marketplace-item.md). Make sure you note the **Publisher**, **Offer**, **SKU**, and **Version** values. You will need them when you edit the Resource Manager template and Manifest.json in your custom .azpkg.

## Step 3, Option 2: Add using PowerShell

Add a VM image as an Azure Stack Hub operator using PowerShell.

1. [Install PowerShell for Azure Stack Hub](azure-stack-powershell-install.md).
2. Sign in to Azure Stack Hub as an operator. For instructions, see [Sign in to Azure Stack Hub as an operator](azure-stack-powershell-configure-admin.md).
3. Open PowerShell with an elevated prompt, and run:

* Add-AzsPlatformimage -publisher "<publisher>" `  
   -offer "<Offer>" `  
   -sku "<SKU>" `  
   -version "<#.#.#>" `  
   -OSType "<OS type>" `  
   -OSUri "<OS URI>"
* The **Add-AzsPlatformimage** cmdlet specifies values used by the Azure Resource Manager templates to reference the VM image. The values include:
  + **publisher**  
    For example: Canonical  
    The **publisher** name segment of the VM image that users use when they deploy the image. Don’t include a space or other special characters in this field.
  + **offer**  
    For example: UbuntuServer  
    The **offer** name segment of the VM image that users use when they deploy the VM image. Don’t include a space or other special characters in this field.
  + **sku**  
    For example: 14.04.3-LTS  
    The **SKU** name segment of the VM image that users use when they deploy the VM image. Don’t include a space or other special characters in this field.
  + **version**  
    For example: 1.0.0  
    The version of the VM image that users use when they deploy the VM image. This version is in the format **#.#.#**. Don’t include a space or other special characters in this field.
  + **osType**  
    For example: Linux  
    The **osType** of the image must be either **Windows** or **Linux**.
  + **OSUri**  
    For example: https://storageaccount.blob.core.windows.net/vhds/Ubuntu1404.vhd  
    You can specify a blob storage URI for an osDisk.
  + For more information, see the PowerShell reference for the [Add-AzsPlatformimage](/powershell/module/azs.compute.admin/add-azsplatformimage) cmdlet.

1. When you add an image, it is only available for Azure Resource Manager-based templates and PowerShell deployments. To make an image available to your users as a marketplace item, publish the marketplace item using the steps in the article [Create and publish a Marketplace item](azure-stack-create-and-publish-marketplace-item.md). Make sure you note the **Publisher**, **Offer**, **SKU**, and **Version** values. You will need them when you edit the Resource Manager template and Manifest.json in your custom .azpkg.

## Remove using the portal

To remove the VM image as an Azure Stack Hub operator using the Azure Stack Hub portal, follow these steps:

1. Open the Azure Stack Hub [administrator portal](https://portal.azure.com/signin/index).
2. If the VM image has an associated Marketplace item, select **Marketplace management**, and then select the VM marketplace item you want to delete.
3. If the VM image does not have an associated Marketplace item, navigate to **All services > Compute > VM Images**, and then select the ellipsis (**…**) next to the VM image.
4. Select **Delete**.

## Remove using PowerShell

To remove the VM image as an Azure Stack Hub operator using PowerShell, follow these steps:

1. [Install PowerShell for Azure Stack Hub](azure-stack-powershell-install.md).
2. Sign in to Azure Stack Hub as an operator.
3. Open PowerShell with an elevated prompt, and run:

* Remove-AzsPlatformImage `  
   -publisher "<Publisher>" `  
   -offer "<Offer>" `  
   -sku "<SKU>" `  
   -version "<Version>" `
* The **Remove-AzsPlatformImage** cmdlet specifies values used by the Azure Resource Manager templates to reference the VM image. The values include:
  + **publisher**  
    For example: Canonical  
    The **publisher** name segment of the VM image that users use when they deploy the image. Don’t include a space or other special characters in this field.
  + **offer**  
    For example: UbuntuServer  
    The **offer** name segment of the VM image that users use when they deploy the VM image. Don’t include a space or other special characters in this field.
  + **sku**  
    For example: 14.04.3-LTS  
    The **SKU** name segment of the VM image that users use when they deploy the VM image. Don’t include a space or other special characters in this field.
  + **version**  
    For example: 1.0.0  
    The version of the VM image that users use when they deploy the VM image. This version is in the format **#.#.#**. Don’t include a space or other special characters in this field.
  + For more info about the **Remove-AzsPlatformImage** cmdlet, see the Microsoft PowerShell [Azure Stack Hub Operator module documentation](/powershell/azure/azure-stack/overview).

## Next steps

* [Create and publish a custom Azure Stack Hub Marketplace item](azure-stack-create-and-publish-marketplace-item.md)
* [Provision a virtual machine](../user/azure-stack-create-vm-template.md)