**Connect-AzAccount**

**Disk Encryption (Windows VM)**

1. Virtual disks on Windows VMs are encrypted at rest by using BitLocker.

2. There's no charge for encrypting virtual disks in Azure.

3. Disks are encrypted by using cryptographic keys that are secured in an Azure Key Vault.

4. You control these cryptographic keys and can audit their use.

5. Cryptographic keys are used to encrypt and decrypt virtual disks attached to your VM.

**The process for encrypting a VM is as follows:**

1. Create a cryptographic key in an Azure Key Vault.

2. Configure the cryptographic key to be usable for encrypting disks.

3. Enable disk encryption for your virtual disks.

4. The required cryptographic keys are requested from Azure Key Vault.

5. The virtual disks are encrypted using the provided cryptographic key.

**Requirements and limitations**

1. Supported scenarios and requirements for disk encryption:

2. Enabling encryption on new Windows VMs from Azure Marketplace images or custom VHD images.

3. Enabling encryption on existing Windows VMs in Azure.

4. Enabling encryption on Windows VMs that are configured by using Storage Spaces.

5. Disabling encryption on OS and data drives for Windows VMs.

6. Standard tier VMs, such as A, D, DS, G, and GS series VMs.

**VVIMP: All resources (including the Key Vault, Storage account, and VM) must be in the same Azure region and subscription. Every key vault must have a name that is unique across Azure.**

Disk encryption isn't currently supported in the following scenarios:

1. Basic tier VMs.

2. VMs created by using the Classic deployment model.

3. Updating the cryptographic keys on an already encrypted VM.

4. Integration with on-premises Key Management Service.

**Steps to Disk Encryption: (PowerShell)**

1. The first step is to create an Azure Key Vault to store your cryptographic keys.

Azure Key Vaults can store keys, secrets, or passwords that allow you to securely implement them in your applications and services.

For virtual disk encryption, you'll create a Key Vault to store a cryptographic key that is used to encrypt or decrypt your virtual disks.

*$rgName = "myResourceGroup"*

*$location = "East US"*

*New-AzResourceGroup -Location $location -Name $rgName*

Register-AzResourceProvider -ProviderNamespace "Microsoft.KeyVault" # Enable Azure keyvault within subscription

*$keyVaultName = "myKeyVault$(Get-Random)"*

#This line generates the unique name for keyvault by adding randomly generated string at the end.

*New-AzKeyVault -Location $location -ResourceGroupName $rgName -VaultName $keyVaultName -EnabledForDiskEncryption -Sku "Premium"*

#This command will create a new keyvault. The -sku parameter is required only if you wanted to add key to HSM module which requires additional cost.

For both protection models (Software-protected keys and HSM), the Azure platform needs to be granted access to request the cryptographic keys when the VM boots to decrypt the virtual disks.

*Set-AzKeyVaultAccessPolicy -VaultName $keyVaultName -ObjectId* 6897acd1-543d-4ce0-8536-cde48dc1fbd6 *-PermissionsToKeys create,import,delete,list,get -PermissionsToSecrets set,delete -PassThru*

In above objectid should replace with the actual objectid value of the currently logged in global admin account. In here -PermissionsToKeys define the permissions allocated for keys and -PermissionsToSecrets defines the permissions allocated for secrets.

2. Create a cryptographic key in your Key Vault with Add-AzureKeyVaultKey. The following example creates a key named myKey:

*Add-AzKeyVaultKey -VaultName $keyVaultName -Name "myKey" -Destination "Software"*

3. Create a virtual machine.

*$cred = Get-Credential*

*New-AzVm `*

*-ResourceGroupName $rgName `*

*-Name "myVM" `*

*-Location $location `*

*-VirtualNetworkName "myVnet" `*

*-SubnetName "mySubnet" `*

*-SecurityGroupName "myNetworkSecurityGroup" `*

*-PublicIpAddressName "myPublicIpAddress" `*

*-Credential $cred*

4. Encrypt your VM with Set-AzVMDiskEncryptionExtension using the Azure Key Vault key. The following example retrieves all the key information then encrypts the VM named myVM:

*$keyVault = Get-AzKeyVault -VaultName $keyVaultName -ResourceGroupName $rgName;*

*$diskEncryptionKeyVaultUrl = $keyVault.VaultUri;*

*$keyVaultResourceId = $keyVault.ResourceId;*

*$keyEncryptionKeyUrl = (Get-AzKeyVaultKey -VaultName $keyVaultName -Name myKey).Key.kid;*

*Set-AzVMDiskEncryptionExtension -ResourceGroupName $rgName `*

*-VMName "myVM" `*

*-DiskEncryptionKeyVaultUrl $diskEncryptionKeyVaultUrl `*

*-DiskEncryptionKeyVaultId $keyVaultResourceId `*

*-KeyEncryptionKeyUrl $keyEncryptionKeyUrl `*

*-KeyEncryptionKeyVaultId $keyVaultResourceId*

5. Once the encryption process completes and the VM has rebooted, review the encryption status with

*Get-AzVmDiskEncryptionStatus -ResourceGroupName $rgName -VMName "myVM"*

6. The output is similar to the following example:

OsVolumeEncrypted : Encrypted

DataVolumesEncrypted : Encrypted

OsVolumeEncryptionSettings : Microsoft.Azure.Management.Compute.Models.DiskEncryptionSettings

ProgressMessage : OsVolume: Encrypted, DataVolumes: Encrypted

**Disk Encryption (Linux VM)**

**Requirements and limitations**

* Azure Disk Encryption uses the DM-Crypt feature of Linux to provide volume encryption for the OS and data disks.
* Azure Disk Encryption is not available on Basic, A-series VMs, or on virtual machines that do not meet these minimum memory requirements:
* Linux VMs when only encrypting data volumes, 2 GB required.
* Linux VMs when encrypting both data and OS volumes, and where the root (/) file system usage is 4GB or less, 8 GB is required.
* Linux VMs when encrypting both data and OS volumes, and where the root (/) file system usage is greater than 4GB, The root file system usage \* 2. For instance, a 16 GB of root file system usage requires at least 32GB of RAM is required.
* Once the OS disk encryption process is complete on Linux virtual machines, the VM can be configured to run with less memory.
* Azure Linux VM must have dm-crypt & vfat modules running.
* Data disks of Linux VM (which required encryption) must be listed under /etc/fstab correctly.
* Before starting encryption, be sure to stop all services and processes that could be writing to mounted data disks and disable them, so that they do not restart automatically after a reboot. These could keep files open on these partitions, preventing the encryption procedure to remount them, causing failure of the encryption.
* To get a token to connect to your key vault, the Linux VM must be able to connect to an Azure Active Directory endpoint, [login.microsoftonline.com].
* To write the encryption keys to your key vault, the Linux VM must be able to connect to the key vault endpoint.
* The Linux VM must be able to connect to an Azure storage endpoint that hosts the Azure extension repository and an Azure storage account that hosts the VHD files.

1. Create Resource Group

*$rgName = "myResourceGroup"*

*$location = "EastUS" #For Linux this works not EAST US*

*New-AzResourceGroup -Name $rgName -Location $location*

2. Create KeyVault

*Register-AzResourceProvider -ProviderNamespace "Microsoft.KeyVault"*

# Enable Azure keyvault within subscription

*$keyVaultName = "myKeyVault$(Get-Random)"*

#This line generates the unique name for keyvault by adding randomly generated string at the end.

*New-AzKeyVault -Location $location -ResourceGroupName $rgName -VaultName $keyVaultName -EnabledForDiskEncryption -Sku "Premium"*

#This command will create a new keyvault. The -sku parameter is required only if you wanted to add key to HSM module which requires additional cost.

For both protection models (Software-protected keys and HSM), the Azure platform needs to be granted access to request the cryptographic keys when the VM boots to decrypt the virtual disks.

*Set-AzKeyVaultAccessPolicy -VaultName $keyVaultName -ObjectId xxxxxxxxxxxxxxxx -PermissionsToKeys create,import,delete,list,get -PermissionsToSecrets set,delete -PassThru*

In above objectid should replace with the actual objectid value of the currently logged in global admin account. In here -PermissionsToKeys define the permissions allocated for keys and -PermissionsToSecrets defines the permissions allocated for secrets.

3. Create a cryptographic key in your Key Vault with Add-AzureKeyVaultKey. The following example creates a key named myKey:

*Add-AzKeyVaultKey -VaultName $keyVaultName -Name "myKey" -Destination "Software"*

4. Create a new Linux VM

*$cred = Get-Credential*

*New-AzVm `*

*-ResourceGroupName $rgName `*

*-Name "myVM" `*

*-Location $location `*

*-VirtualNetworkName "myVnet" `*

*-SubnetName "mySubnet" `*

*-SecurityGroupName "myNetworkSecurityGroup" `*

*-PublicIpAddressName "myPublicIpAddress" `*

*-OpenPorts 22 `*

*-Image Canonical:UbuntuServer:16.04-LTS:latest `*

*-Size Standard\_D2s\_v3 `*

*-Credential $cred*

5. Encrypt the Linux VM

*$keyVault = Get-AzKeyVault -VaultName $keyVaultName -ResourceGroupName $rgName;*

*$diskEncryptionKeyVaultUrl = $keyVault.VaultUri;*

*$keyVaultResourceId = $keyVault.ResourceId;*

*$keyEncryptionKeyUrl = (Get-AzKeyVaultKey -VaultName $keyVaultName -Name myKey).Key.kid;*

*Set-AzVMDiskEncryptionExtension -ResourceGroupName $rgName `*

*-VMName "myVM" `*

*-DiskEncryptionKeyVaultUrl $diskEncryptionKeyVaultUrl `*

*-DiskEncryptionKeyVaultId $keyVaultResourceId `*

*-KeyEncryptionKeyUrl $keyEncryptionKeyUrl `*

*-KeyEncryptionKeyVaultId $keyVaultResourceId `*

*-VolumeType All `*

*-SkipVmBackup*

#VolumeType defines which type of disks need to be encrypted. In this case both OS & Data disks.

# VMBackup extension is a recovery safeguard for encrypted disks. However, this is not compatible with the managed disks. Therefore, the encryption will fail if you not using –SkipVmBackup

6. Once the encryption process completes and the VM has rebooted, review the encryption status with

*Get-AzVmDiskEncryptionStatus -ResourceGroupName $rgName -VMName "myVM"*