# Puppet PowerShell Cmdlets for Azure

# Puppet PowerShell Cmdlets for Azure

Microsoft now provides a version of the Azure SDK that contains PowerShell cmdlets for provisioning, enabling, and disabling the Puppet extension handler on Windows virtual machines (VMs). Essentially, this provides a command-line interface that allows users to deploy Puppet Enterprise agents to Windows VMs in Azure.

There are three Puppet-specific cmdlets in the Azure SDK:

- Set-AzureVMPuppetExtension
- Get-AzureVMPuppetExtension
- Remove-AzureVMPuppetExtension

## Before You Begin

- Get a Microsoft Azure account.
- Install the ***Azure SDK***

## Import Azure Credentials Into Your PowerShell Session

To work with the Puppet cmdlets, you need to import Azure credentials into your PowerShell session.

1. Download subscription credentials for accessing Azure.
This can be done by executing a cmdlet. It will launch your browser and download the credentials file:

```
PS C:\> Get-AzurePublishSettingsFile
```

2. Import the credentials file into your current PowerShell session.

```
PS C:\> Import-AzurePublishSettingsFile -PublishSettingsFile
"${env:USERPROFILE}\Downloads\<subscription>-<date>-credentials.publishsettings"
```

Output:

```
Verbose: Setting: <subscription> as the default and current subscription. To view other
subscriptions, use `Get-AzureSubscription`.
```

3. Ensure the appropriate subscription and storage accounts are selected. See the Azure console for information about the `subscriptionName` and `storageAccountName`:

```
PS C:\> Select-AzureSubscription -SubscriptionName <subscriptionName>

PS C:\> Set-AzureSubscription -CurrentStorageAccountName <storageAccountName>
-SubscriptionName <subscriptionName>
```

## Set-AzureVMPuppetExtension

The `Set-AzureVMPuppetExtension` cmdlet is used to add the Puppet Enterprise agent and extension handler onto a Windows VM. This can be accomplished while provisioning a new Windows VM, or can be added to an existing VM. The following steps show how to add the PE agent and extension handler to an existing VM.

1. Define a VM configuration, specifying the hostname, VM size, and the image that you want to use:

```
PS C:\> $vmName = "agent-demo"

PS C:\> $vm = New-AzureVMConfig -Name $vmName -instanceSize Medium -ImageName
"a699494373c04fc0bc8f2bb1389d6106__Windows-Server-2012-R2-201402.01-en.us-127GB.vhd"
```

To obtain a list of images available to your subscription, execute the following:

```
PS C:\> Get-AzureVMImage
...
ImageName            : a699494373c04fc0bc8f2bb1389d6106__Windows-Server-2012-R2-201402.01-en.
us-127GB.vhd
...
```

2. Configure the username and password for the VM. You will need to enter these credentials below when you RDP to the host.

```
PS C:\> $vm = Add-AzureProvisioningConfig -VM $vm -AdminUsername "azureuser" -Password "<your
password>" -Windows
```

3. Add the Puppet Enterprise extension handler, specifying the FQDN of the puppet master. This FQDN should match the public FQDN of the puppet master, and will be of the form: `puppetmaster.cloudapp.net`

```
PS C:\> $vm = Set-AzureVMPuppetExtension -VM $vm -PuppetMasterServer <fqdn>
```

4. Provision the VM in the specified region:

```
PS C:\> New-AzureVM -Location 'West US' -ServiceName $vmName -VM $vm
```

Output:

```
OperationDescription                    OperationId
OperationStatus
--------------------                    -------------------------
New-AzureVM
0e3926ee-88eb-890a-97f6-3018c3971598    Succeeded
New-AzureVM
59d97f0e-f5e5-8dd8-9f54-39b7d85c1b2a    Succeeded
```

5. Wait for VM to provision.

```
PS C:\> Get-AzureVM -Name $vmName -ServiceName $vmName
```

Output:

```
DeploymentName        : agent-demo
Name                  : agent-demo
Label                 :
VM                    : Microsoft.WindowsAzure.Commands.ServiceManagement.Model.PersistentVM
InstanceStatus        : Provisioning
IpAddress             : XX.XX.XX.XX
InstanceStateDetails  : Windows is preparing your computer for first use...
PowerState            : Started
InstanceErrorCode     :
InstanceFaultDomain   : 0
InstanceName          : agent-demo
InstanceUpgradeDomain : 0
InstanceSize          : Medium
AvailabilitySetName   :
DNSName               : http://agent-demo.cloudapp.net/
ServiceName           : agent-demo
OperationDescription  : Get-AzureVM
OperationId           : 108447e3-88b3-810c-9646-35fec0535314
OperationStatus       : Succeeded
```

6. Eventually, it will report it is ready:

```
InstanceStatus        : ReadyRole
```

7. Optionally, RDP to the newly provisioned VM:

```
PS C:\> Get-AzureRemoteDesktopFile -Name $vmName -ServiceName $vmName -LocalPath
"${env:TEMP}\${vmName}.rdp" -launch
```

In the Remote Desktop Connection message box, click **Connect.**
Then enter the user name and password specified in the call to `Add-AzureProvisioningConfig`.

## Get-AzureVMPuppetExtension

The `Get-AzureVMPuppetExtension` cmdlet will get the current status of the Puppet extension handler.

```
PS C:\> Get-AzureVMPuppetExtension -VM $vm
```

Output:

```
ExtensionName        : PuppetEnterpriseAgent
Publisher            : PuppetLabs
Version              : 3.2.1
PrivateConfiguration :
PublicConfiguration  :
ReferenceName        : PuppetAgent
State                : Enable
```

Notice that the state is `Enable`, meaning the Puppet Enterprise agent is running.

The agent can be disabled using the `Set-AzureVMPuppetExtension` and `Update-AzureVM` cmdlets. Due to a bug, the `-Disable` flag is currently required for all Puppet cmdlets.

```
PS C:\> Set-AzureVMPuppetExtension -VM $vm -Disable -PuppetMasterServer <fqdn>
```

Output:

```
AvailabilitySetName               :
ConfigurationSets                 : {agent-demo, Microsoft.WindowsAzure.Commands.
                                     ServiceManagement.Model.PersistentVMMo
                                    del.NetworkConfigurationSet}
DataVirtualHardDisks              : {}
Label                             : agent-demo
OSVirtualHardDisk                 : Microsoft.WindowsAzure.Commands.ServiceManagement.Model.
PersistentVMModel.OSVirtual

                                    HardDisk
RoleName                          : agent-demo
RoleSize                          : Medium
RoleType                          : PersistentVMRole
WinRMCertificate                  :
X509Certificates                  : {}
NoExportPrivateKey                : False
NoRDPEndpoint                     : False
NoSSHEndpoint                     : False
DefaultWinRmCertificateThumbprint :
ProvisionGuestAgent               : True
ResourceExtensionReferences       : {PuppetEnterpriseAgent}
DataVirtualHardDisksToBeDeleted   :
```

Then run:

```
PS C:\> Update-AzureVM -ServiceName $vmName -Name $vmName -VM $vm
```

Output:

```
OperationDescription                    OperationId
OperationStatus
--------------------                    ------------------------
Update-AzureVM
eff7834b-baef-89d4-b4d4-7c72deb083f0    Succeeded
```

Note that `Set-AzureVMPuppetExtension` modifies the local `$vm` object, and that `Update-AzureVM` is required to update the VM object in Azure.

Continue to page 6.

## Remove-AzureVMPuppetExtension

The following code removes the Puppet Enterprise extension. This uninstalls the Puppet Enterprise agent from the VM. As previously, this is a two-step process:

```
PS C:\> Remove-AzureVMPuppetExtension -VM $vm
```

Output:

```
AvailabilitySetName               :
ConfigurationSets                 : {agent-demo, Microsoft.WindowsAzure.Commands.
                                     ServiceManagement.Model.PersistentVMMo
                                    del.NetworkConfigurationSet}
DataVirtualHardDisks              : {}
Label                             : agent-demo
OSVirtualHardDisk                 : Microsoft.WindowsAzure.Commands.ServiceManagement.Model.
PersistentVMModel.OSVirtual
                                    HardDisk
RoleName                          : agent-demo
RoleSize                          : Medium
RoleType                          : PersistentVMRole
WinRMCertificate                  :


X509Certificates                  : {}
NoExportPrivateKey                : False
NoRDPEndpoint                     : False
NoSSHEndpoint                     : False
DefaultWinRmCertificateThumbprint :
ProvisionGuestAgent               : True
ResourceExtensionReferences       : {}
DataVirtualHardDisksToBeDeleted   :
```

Then run:

```
PS C:\> Update-AzureVM -ServiceName $vmName -Name $vmName -VM $vm
```

Output:

```
OperationDescription                  OperationId
OperationStatus
--------------------                  -------------------------
Update-AzureVM
3e01076a-91ac-8821-a755-31171801b61f    Succeeded
```