VConnect V1.0

VConnect Customization

VConnect for Windows Azure Pack customization guide

Version: 1.7.2.0
Published: 8/10/2015

Customization and Extensibility options for end-to-end process automation

Introduction

VConnect comes with many customization and extensibility options at varying levels.

Basic customization options Include:

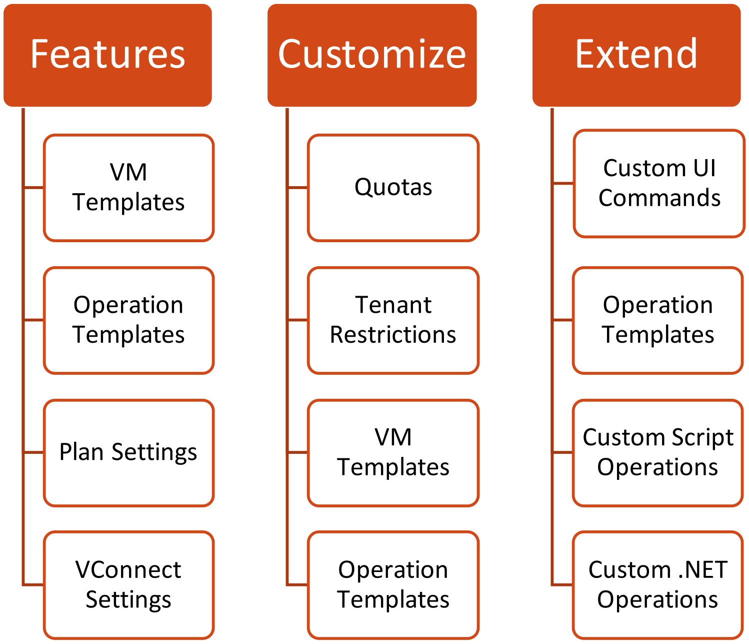
1) VConnect Plan Configuration Settings
2) VM Template Configuration and Customization
3) Leverage Static IP Pool Feature
4) Process Automation with VConnect UI Command extensions

Advanced customization options:

1) Customize and Extend Operation Template
2) Write your own Custom Operation in PowerShell
3) Write your own Custom Operation in .Net
4) Customize and Extend VConnect Scriptlets
5) Customize columns shown in Virtual Machine list grid

Following sections discuss each of these options in detail.

© Copyright 2015-2016 CLOUD ASSERT LLC



Basic Customizations

VConnect Plan Configuration

VConnect Plan settings offers variety of options to customize the experience of all users subscribing to a specific plan.

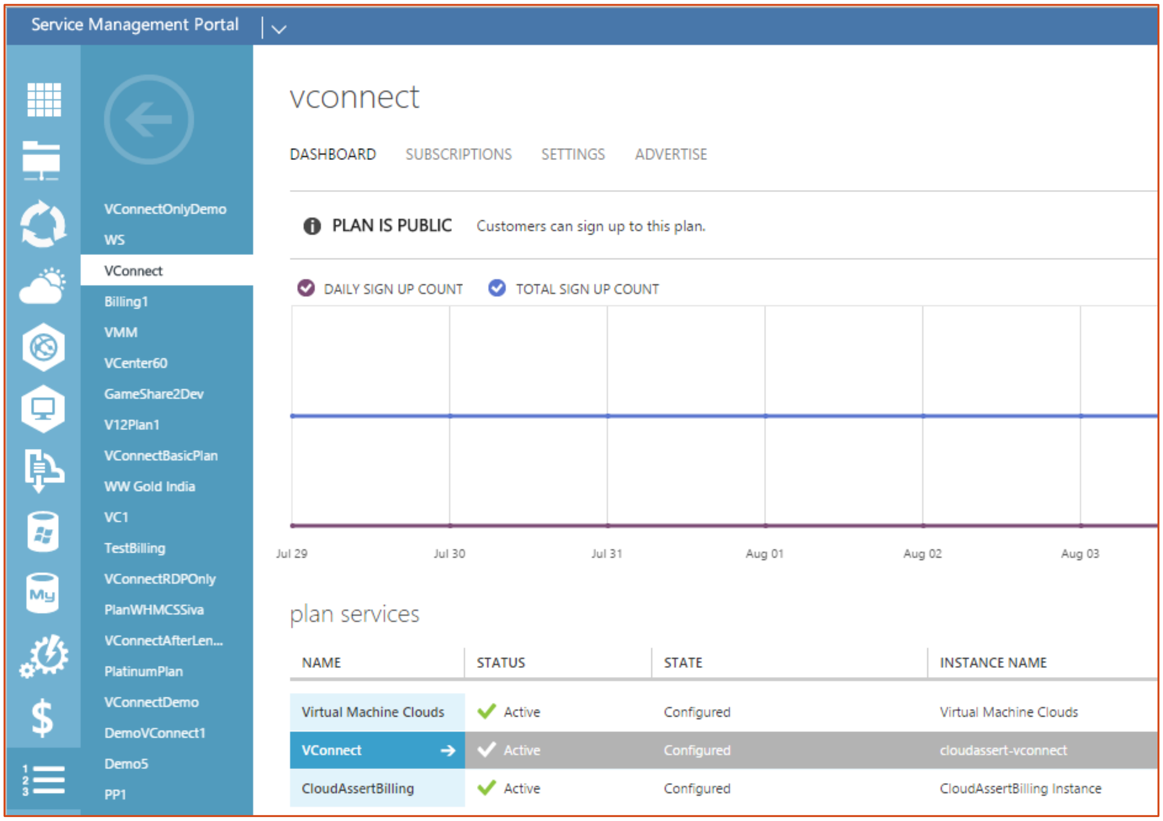
Each plan can have different settings and thus enabling administrators to provide unique experience for different set of customers.

Example Scenarios enabled by these settings:

- Target a specific set of VMWare resources to a group of users.
    - For example: one plan can target ClusterA with high performance Storage. And another plan can target a TestCluster
- Restrict VM management operations that a set of users belonging to a plan can do.
    - For example: MarkettingProdApplicationsPlan can restrict end users from ‘Deleting’ the VM.
- Set different quotas per Plan
    - For example: GenomeResearchTeamPlan could have higher a higher number of cores and memory quota.

How-To: VConnect Plan Configuration

Admin Portal → Plans→ Select a Plan → Click on ‘VConnect’ service



Service Management Portal

VConnectOnlyDemo
WS
VConnect
Billing1
VMM
VCenter60
GameShare2Dev
V12Plan1
VConnectBasicPlan
WW Gold India
VC1
TestBilling
VConnectRDPOnly
PlanWHMCSSiva
VConnectAfterLen...
PlatinumPlan
VConnectDemo
DemoVConnect1
Demo5
PP1

vconnect

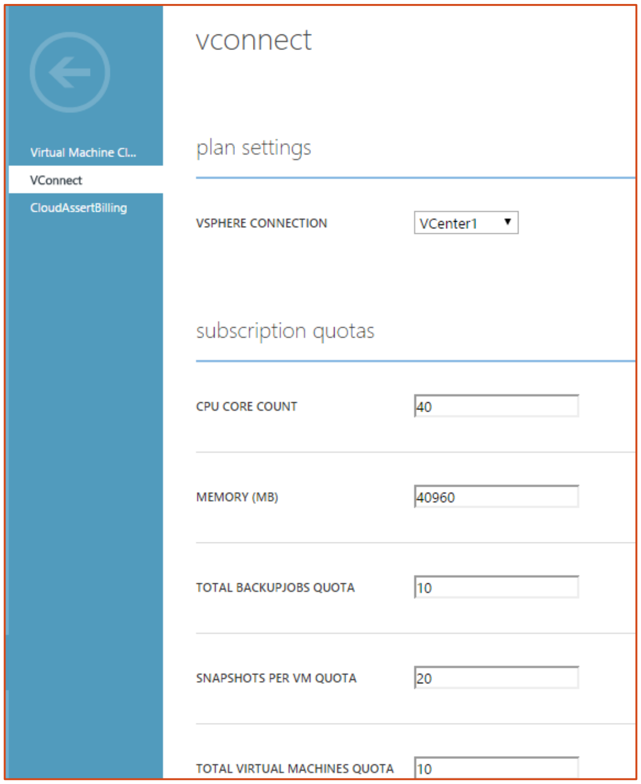
DASHBOARD SUBSCRIPTIONS SETTINGS ADVERTISE

PLAN IS PUBLIC Customers can sign up to this plan.

plan services

NAME	STATUS	STATE	INSTANCE NAME
Virtual Machine Clouds	✓ Active	Configured	Virtual Machine Clouds
VConnect →	✓ Active	Configured	cloudassert-vconnect
CloudAssertBilling	✓ Active	Configured	CloudAssertBilling Instance

VConnect Quota Settings*:*



Virtual Machine Cl...
VConnect
CloudAssertBilling

vconnect

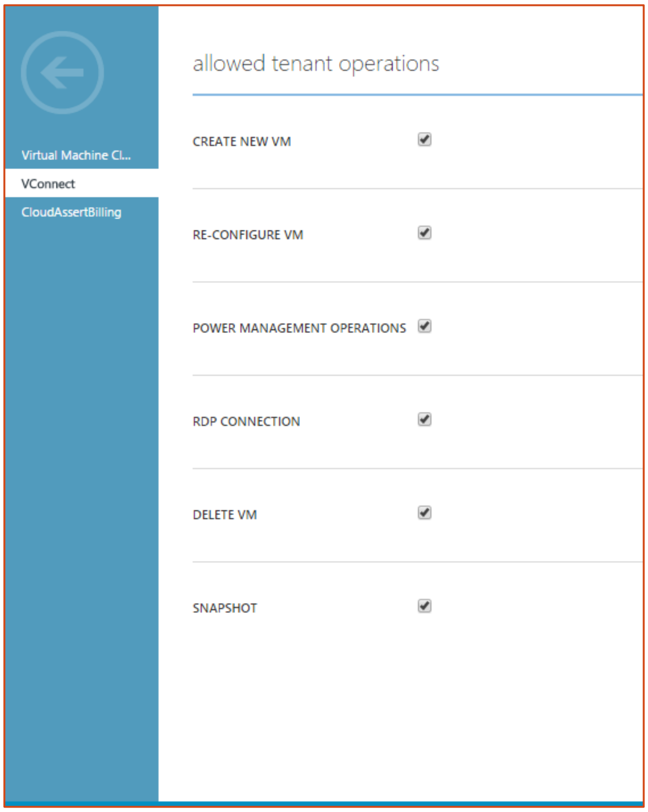
plan settings

VSPHERE CONNECTION: VCenter1 ▾

subscription quotas

CPU CORE COUNT	40
MEMORY (MB)	40960
TOTAL BACKUPJOBS QUOTA	10
SNAPSHOTS PER VM QUOTA	20
TOTAL VIRTUAL MACHINES QUOTA	10

VConnect Tenant Restrictions:



Virtual Machine Cl...
VConnect
CloudAssertBilling

allowed tenant operations

CREATE NEW VM ☑

RE-CONFIGURE VM ☑

POWER MANAGEMENT OPERATIONS ☑

RDP CONNECTION ☑

DELETE VM ☑

SNAPSHOT ☑

VM Template Configuration

Configure VM Templates

Example scenarios enabled by this method:

- Domain Join a VM
- Specify the type of Data Store to place the VM in
- Add installation scripts on start of the VM
- Specify a network name to join the VM to
- Decide which parameters your end users can override

Following table shows list of available parameters out of the box for a Windows VM Template. Administrator can override default values for these parameters that suits the needs. And also set ‘Is Configurable by Tenant’ user and ‘Is required’ options per parameter.

Table: Parameters available in VM Templates.

Label	Description	Default	Type	Options
NetworkName	"Network switch name to which the created VM will be connected to"		String	

IpMode	"Specifies the IP configuration mode. The valid values are UseDhcp and UseStaticIP. Default is UseDhcp."		Option	"UseDhcp=UseDhcp;UseStaticIP=UseStaticIP"
IP Address Pool Name	"Applicable only when IpMode == UseStaticIP. Refers to the IP Address Pool configured in VConnect."		String	
DataStoreHint	"Leave it empty to choose the first available Data Store with capacity"		String	
AdminPassword	"Specifies a new OS administrator's password."		SecureString	
AutoLogonCount	"Specifies the number of times the virtual machine automatically logs in as administrator without prompting for user credentials. The valid values are in the range between 0 and Int32.MaxValue. Specifying 0 disables auto log-on."	2	Number	
ChangeSid	"Indicates that the customization should modify the system security identifier (SID)."	true	Boolean	
Domain	"Specifies a domain name."		String	
DomainUsername	"Specifies the user name you want to use for domain authentication."		String	
DomainPassword	"Specifies the user name you want to use for domain authentication."		SecureString	
FullName	"Specifies the administrator's full name."		String	
RunOnce Commands	"Specifies a list of commands. These commands run when a user logs in for the first time after the customization completes."		StringArray	
OrgName	"Organization Name"		String	

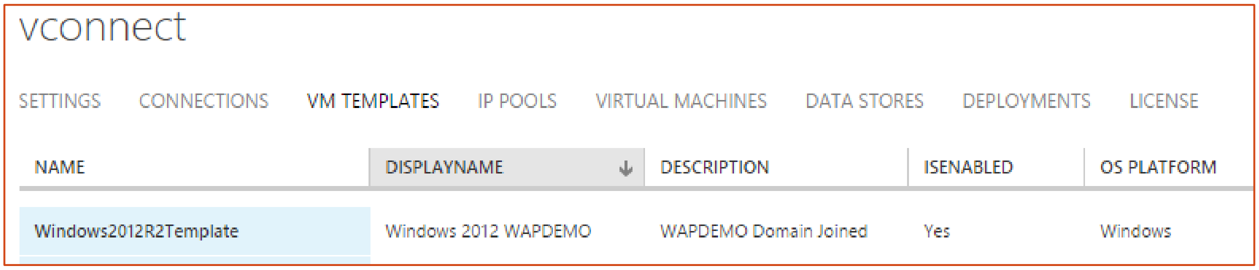
© Copyright 2015-2016 CLOUD ASSERT LLC

OSTimeZone	"Operating System TimeZone"	GMT (Greenwich Mean Time)	Option	
LicenseMaxConnections	"Specifies the maximum connections for server license mode. Use this parameter only if the LicenseMode parameter is set to Perserver."		String	""
LicenseMode	"Specifies the license mode of the Windows 2000/2003 guest operating system. The valid values are Perseat; Perserver and Notspecified. If Perserver is set use the LicenseMaxConnection parameter to define the maximum number of connections."		Option	"Notspecified=Notspecified;Perseat=Perseat;Perserver=Perserver"
ProductKey	"Specifies the MS product key. If the guest OS version is earlier than Vista;this parameter is required in order to make the customization unattended. For Vista or later the OS customization is unattended no matter if the ProductKey parameter is set."		SecureString	
WorkgroupName	"Workgroup of the VM"	Workgroup	String	

NOTE: Linux VM Template options are not the same as Windows VM Template options.

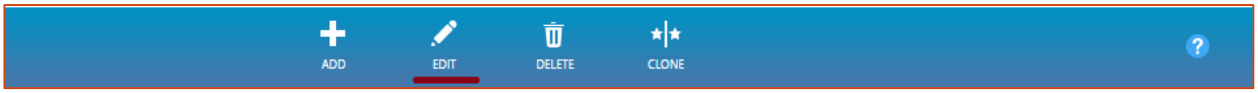
How-To:

Admin Portal → VM Templates → Select a Template → Edit

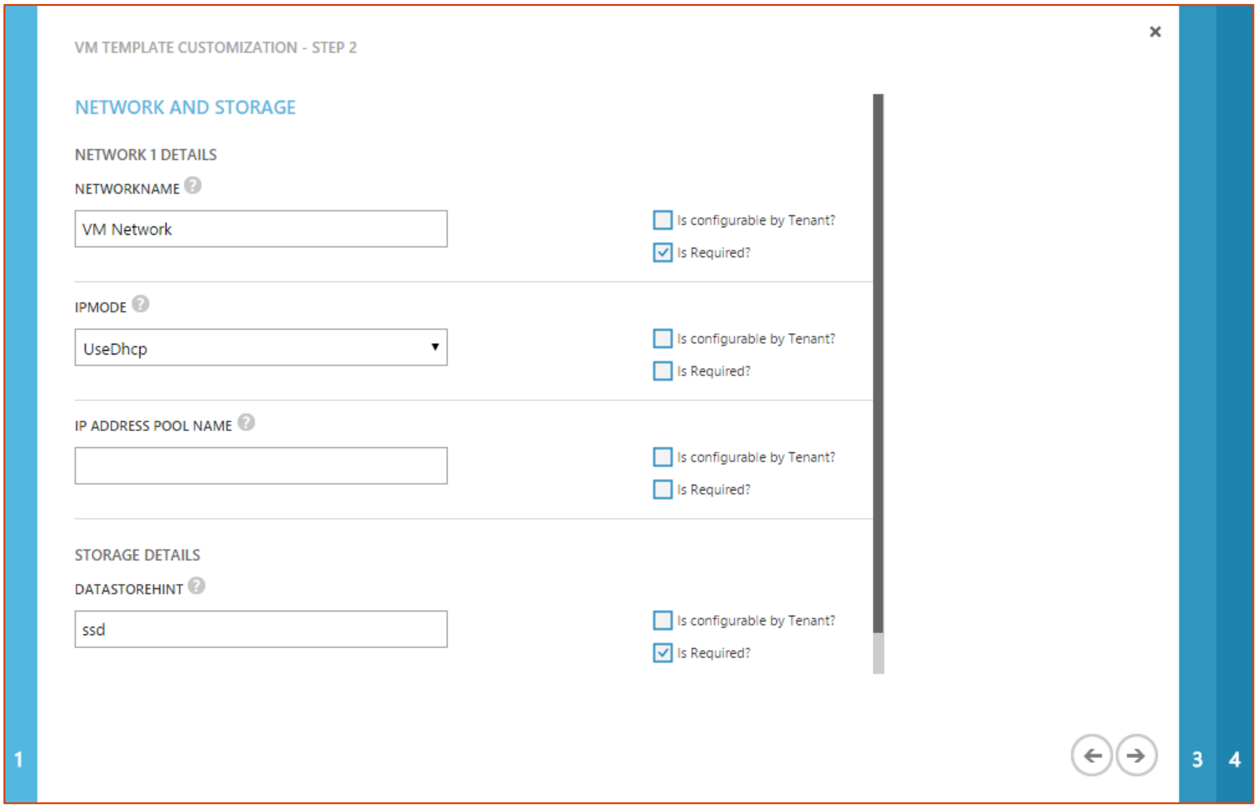


vconnect

NAME	DISPLAYNAME	DESCRIPTION	ISENABLED	OS PLATFORM
Windows2012R2Template	Windows 2012 WAPDEMO	WAPDEMO Domain Joined	Yes	Windows



VM Template Edit Dialog:



VM TEMPLATE CUSTOMIZATION - STEP 2

NETWORK AND STORAGE

NETWORK 1 DETAILS

NETWORKNAME: VM Network
□ Is configurable by Tenant?
☑ Is Required?

IPMODE: UseDhcp
□ Is configurable by Tenant?
□ Is Required?

IP ADDRESS POOL NAME:
□ Is configurable by Tenant?
□ Is Required?

STORAGE DETAILS

DATASTOREHINT: ssd
□ Is configurable by Tenant?
☑ Is Required?

Leverage Static IP Pool Feature

VConnect features IP Address Pool management. You can specific these values in the VM Template in ‘IP Mode’ and ‘IP Address Pool Name’ parameters.

IP Address Pools can be configured via the ‘IP Pools’ Tab.

Example scenarios enabled by this:

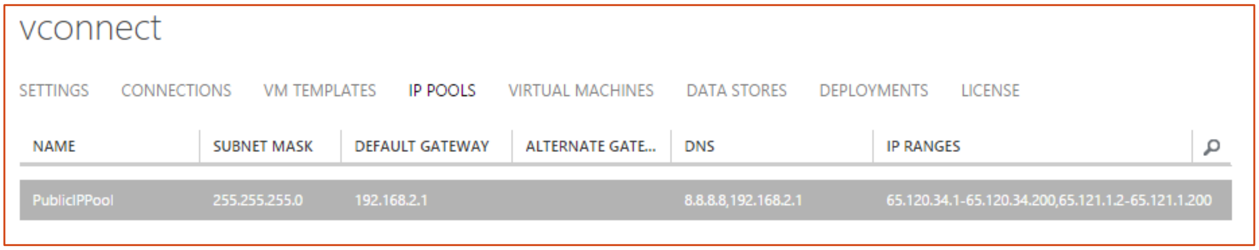
- Support non-DHCP environments with static list of known IP address ranges
- Support setting up public IP addresses for specific VM Templates

© Copyright 2015-2016 CLOUD ASSERT LLC

- Since IP Address Pool can be selected per VM Template, it is possible to have different type of VMs use different range of IP Addresses

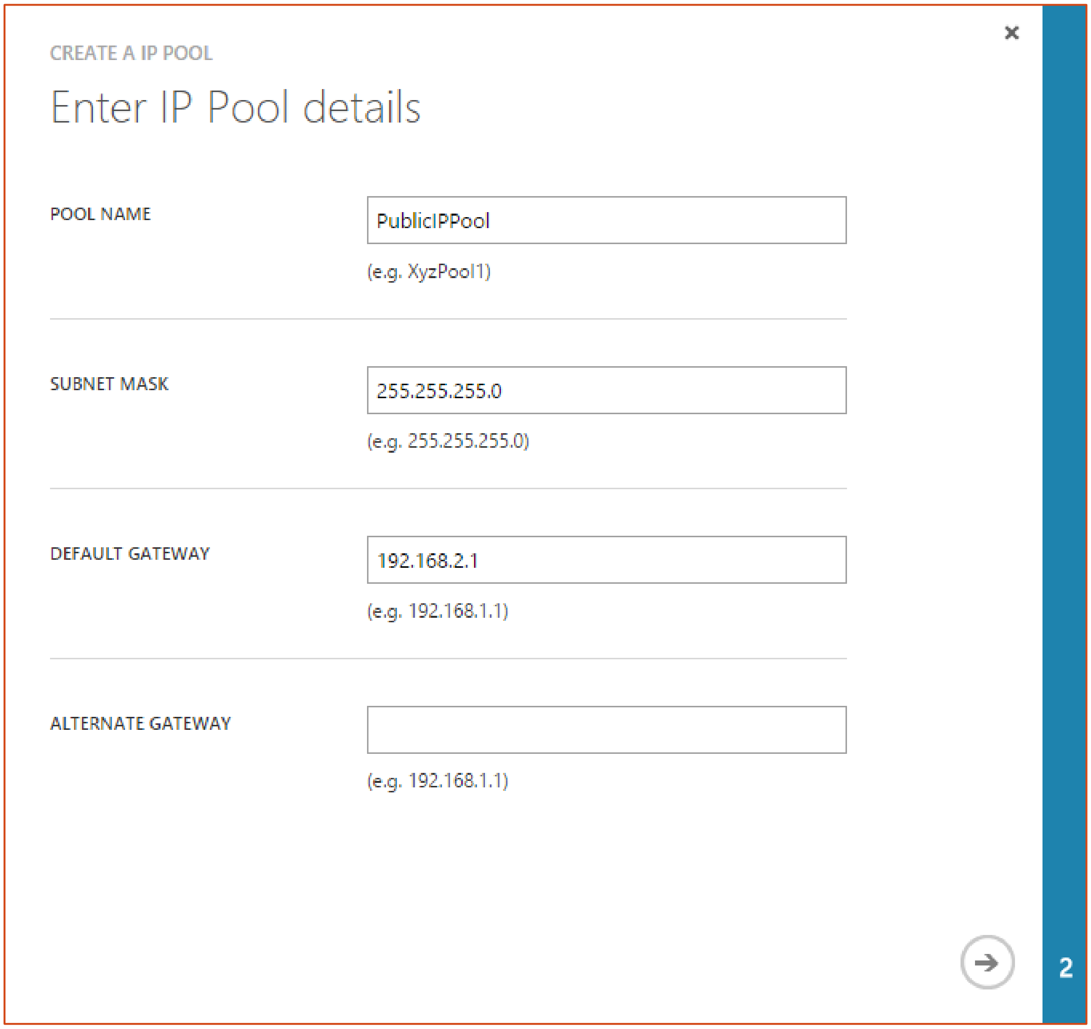
- How-To:

Admin Portal → Ip Pools → Add



NAME	SUBNET MASK	DEFAULT GATEWAY	ALTERNATE GATE...	DNS	IP RANGES
PublicIPPool	255.255.255.0	192.168.2.1		8.8.8.8,192.168.2.1	65.120.34.1-65.120.34.200,65.121.1.2-65.121.1.200

VConnect Add IP Pool Details and Ranges:



Enter IP Pool details

POOL NAME: PublicIPPool
(e.g. XyzPool1)

SUBNET MASK: 255.255.255.0
(e.g. 255.255.255.0)

DEFAULT GATEWAY: 192.168.2.1
(e.g. 192.168.1.1)

ALTERNATE GATEWAY:
(e.g. 192.168.1.1)

Process Automation with UI command extensions

© Copyright 2015-2016 CLOUD ASSERT LLC

VConnect features a much loved custom commands feature that enables administrators to augment Azure Pack portal User Interface with custom functionality without writing any UI Code.

Custom Commands in VConnect provides features for bringing in additional functionality with very few clicks inside VConnect management experience for end users.

Example scenarios made possible by this feature:

- Add a button to request for a VM configuration change
- Ask user to provide inputs for a Support Ticket request and submit to your own existing ticketing system
- Add any VMWare management functionality that is not exposed by VConnect currently, but you would need. For example, you could:
  - Add a button to ‘Add Disk’ and show users with options to select from
  - Add a button to ‘Replicate VM’ and enable VM replication feature
  - Support ‘FailOverSRM’ functionality

In addition to defining a command, Administrator can also define parameters to retrieve inputs from users though the Azure Pack User Interface when the user clicks the command button.

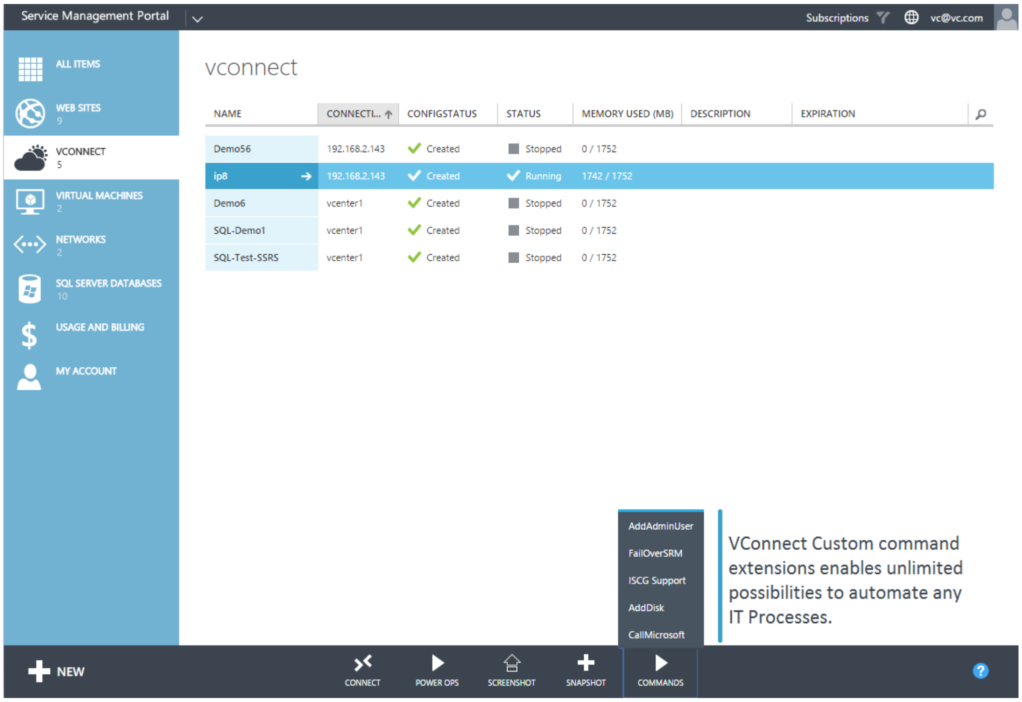
Following custom command parameters are supported.

Table: Supported Custom Command Parameter Types.

Parameter Type	Description
Text	Text input (string)
Password	Password input (Secured String)
Text (CSV)	Comma Separated Values input (String Array)
Number	Numeric input
Check Box	True or False input (Boolean)
Select Box	Options Input (Key Value Pair)

Custom commands provides limitless possibilities to extend, you can creatively bring in all typical VM management processes inside Azure Pack portal with this VConnect Custom Commands feature.

Custom Command Sample Screenshot:



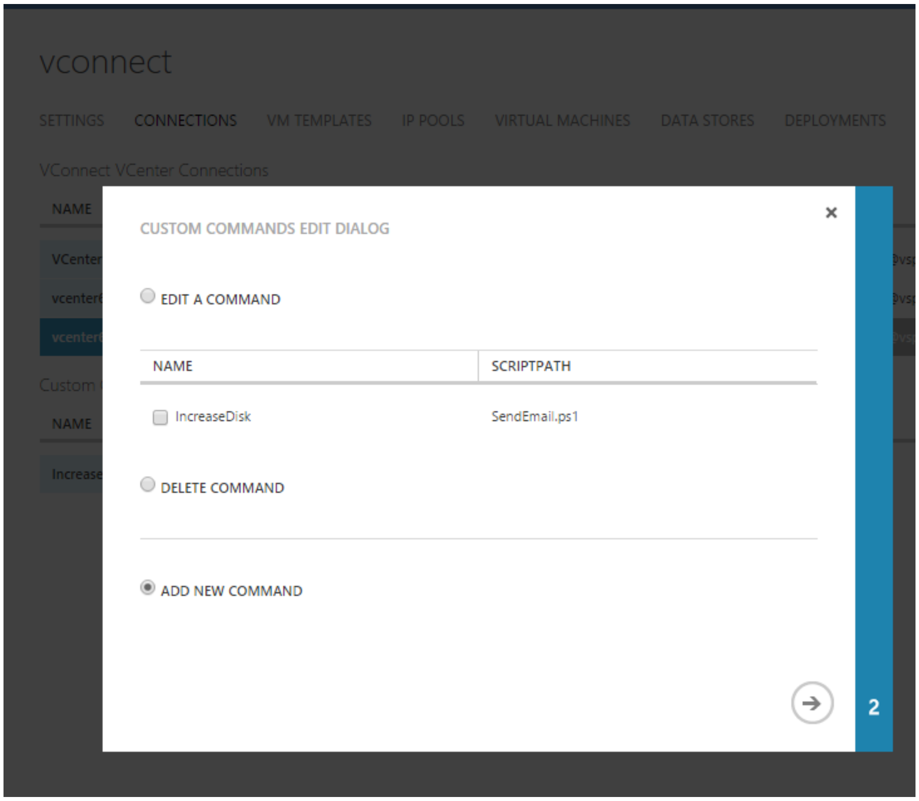
vconnect

NAME	CONNECTI... ↑	CONFIGSTATUS	STATUS	MEMORY USED (MB)	DESCRIPTION	EXPIRATION
Demo56	192.168.2.143	✓ Created	■ Stopped	0 / 1752		
ip8	192.168.2.143	✓ Created	✓ Running	1742 / 1752		
Demo6	vcenter1	✓ Created	■ Stopped	0 / 1752		
SQL-Demo1	vcenter1	✓ Created	■ Stopped	0 / 1752		
SQL-Test-SSRS	vcenter1	✓ Created	■ Stopped	0 / 1752		

VConnect Custom command extensions enables unlimited possibilities to automate any IT Processes.

How-To: Add Custom UI Command

Admin Portal → VConnect → Connections → Select a Connection → Custom Commands (button at the bottom) → Add New Command



CUSTOM COMMANDS EDIT DIALOG

○ EDIT A COMMAND

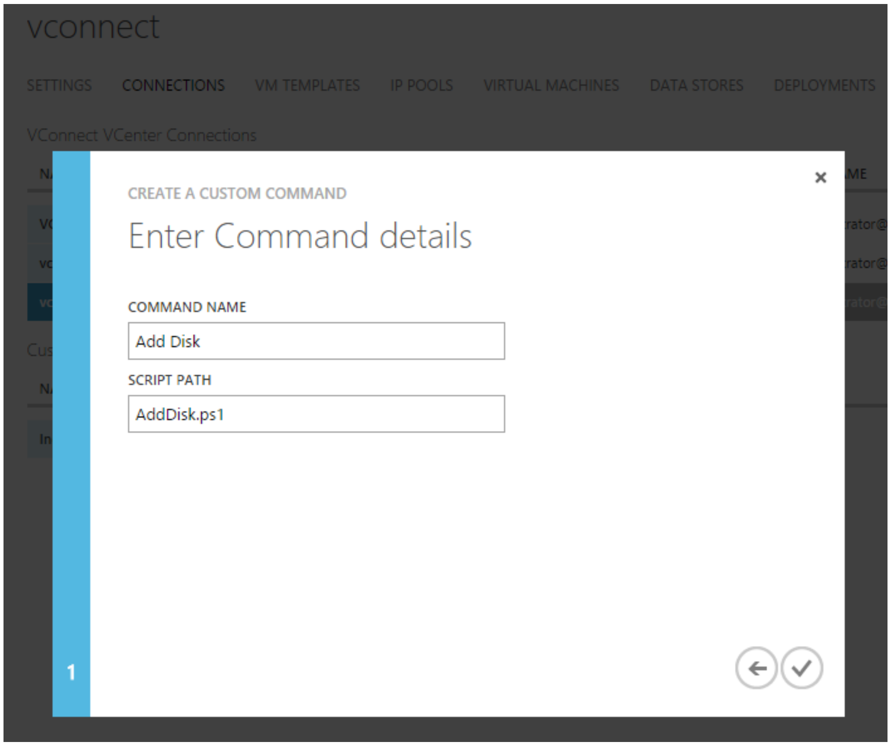
NAME	SCRIPTPATH
□ IncreaseDisk	SendEmail.ps1

○ DELETE COMMAND

● ADD NEW COMMAND

© Copyright 2015-2016 CLOUD ASSERT LLC

Add Custom Command Dialog:



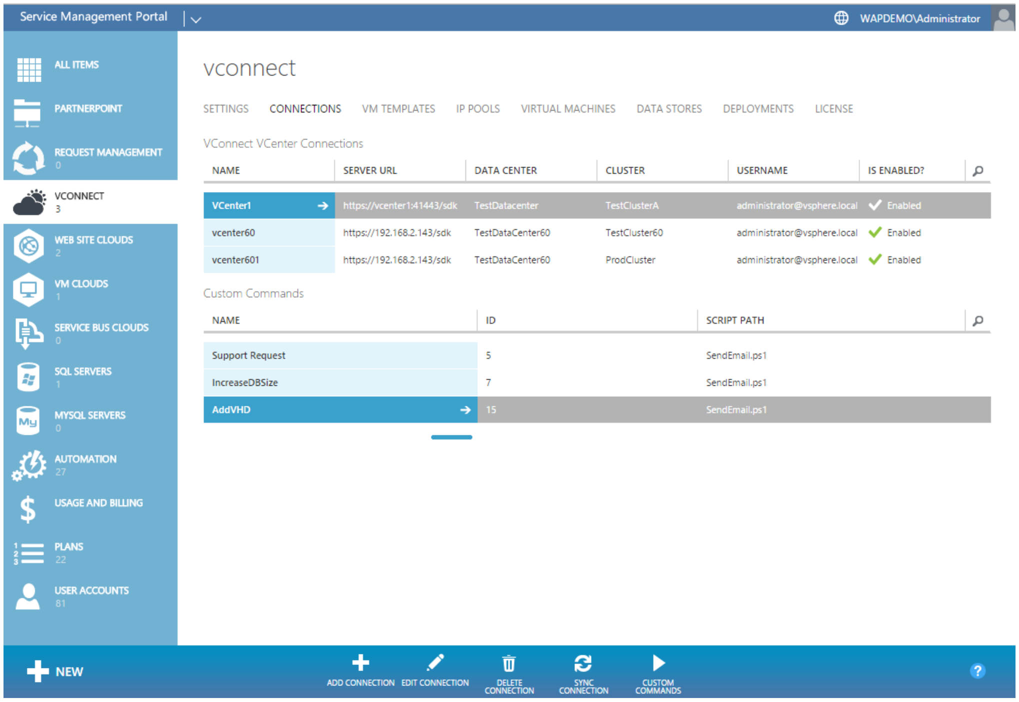
CREATE A CUSTOM COMMAND

Enter Command details

COMMAND NAME: Add Disk

SCRIPT PATH: AddDisk.ps1

Add custom parameters to receive user inputs for custom commands:



VConnect VCenter Connections

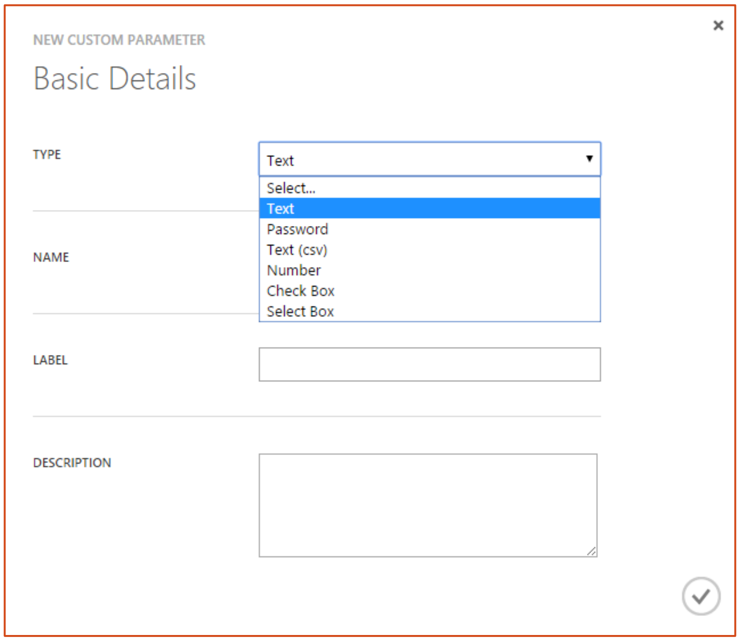
NAME	SERVER URL	DATA CENTER	CLUSTER	USERNAME	IS ENABLED?
VCenter1	https://vcenter1:41443/sdk	TestDatacenter	TestClusterA	administrator@vsphere.local	✓ Enabled
vcenter60	https://192.168.2.143/sdk	TestDataCenter60	TestCluster60	administrator@vsphere.local	✓ Enabled
vcenter601	https://192.168.2.143/sdk	TestDataCenter60	ProdCluster	administrator@vsphere.local	✓ Enabled

Custom Commands

NAME	ID	SCRIPT PATH
Support Request	5	SendEmail.ps1
IncreaseDBSize	7	SendEmail.ps1
AddVHD	15	SendEmail.ps1

© Copyright 2015-2016 CLOUD ASSERT LLC

Define Custom Parameter Type and values:



NEW CUSTOM PARAMETER

Basic Details

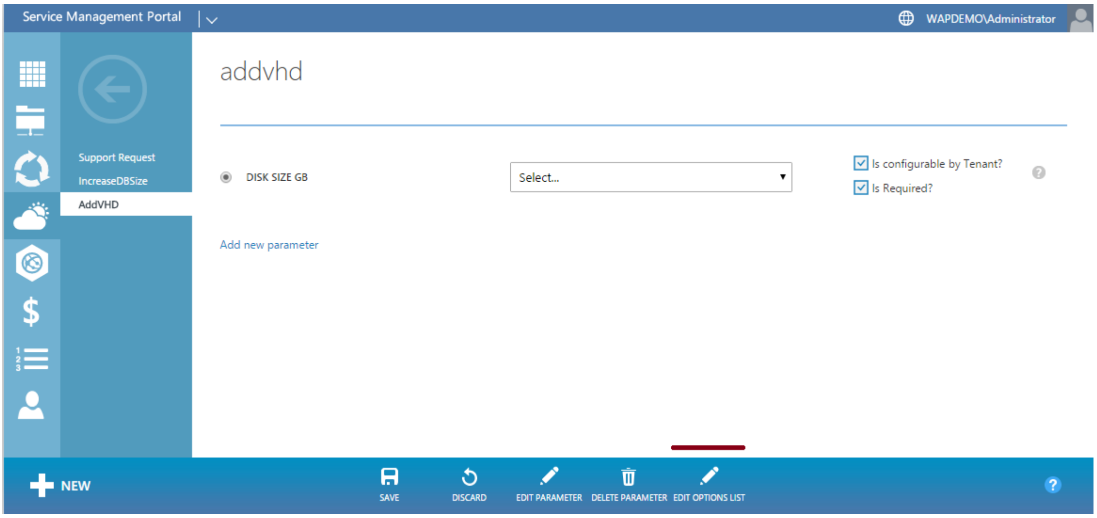
TYPE: Text

NAME

LABEL

DESCRIPTION

'Edit Options List' to set the options for the custom parameter:



Service Management Portal

WAPDEMO\Administrator

addvhd

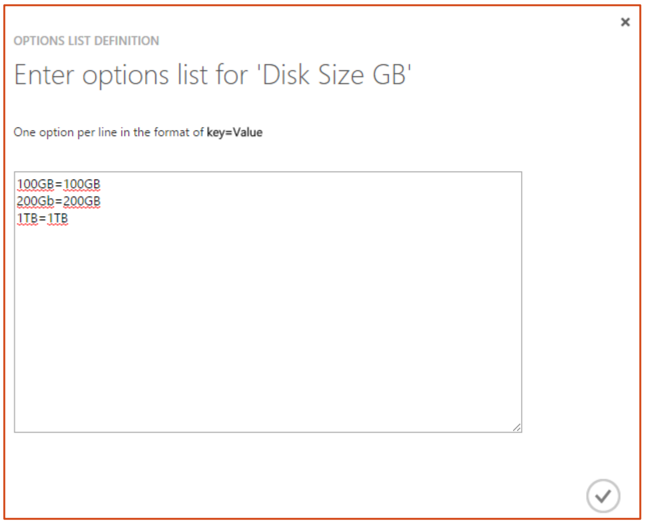
● DISK SIZE GB: Select...

☑ Is configurable by Tenant?

☑ Is Required?

© Copyright 2015-2016 CLOUD ASSERT LLC

Edit Options List Dialog:



×

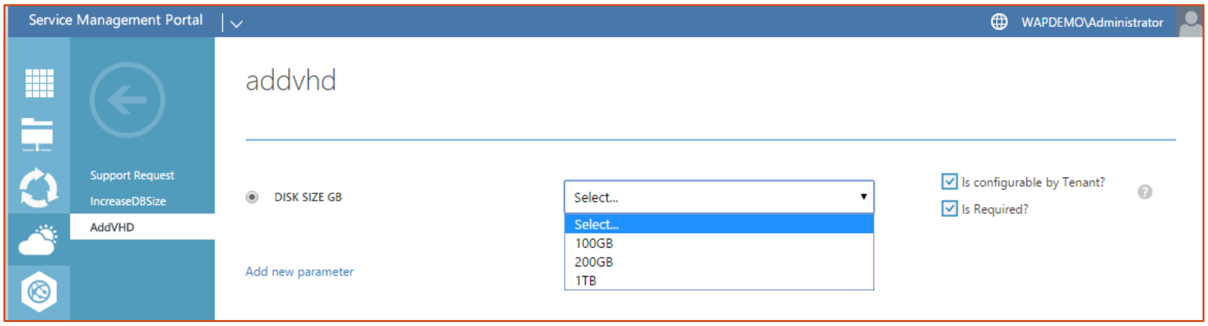
OPTIONS LIST DEFINITION

Enter options list for 'Disk Size GB'

One option per line in the format of **key=Value**

```
100GB=100GB
200Gb=200GB
1TB=1TB
```

Parameter Definition after Options set:



addvhd

● DISK SIZE GB

Select... ▾

Select...
100GB
200GB
1TB

☑ Is configurable by Tenant?

☑ Is Required?

Add new parameter

Advanced Customizations

VConnect has following customization and extension options for advanced scenarios.

1) Customize Operation Template
2) Write your own Custom Operation in PowerShell
3) Write your own Custom Operation in .Net
4) Customize VConnect Scriptlets

Customize Operation Template

VConnect employs a custom Operation Template model to drive the VM deployments. Basically all the steps in a VM Deployment are defined in a JSON file.

VConnect had this mechanism for a long time with the upcoming release, this template will be made available and administrators could potentially add additional steps or replace existing steps.

© Copyright 2015-2016 CLOUD ASSERT LLC

Currently there is a single template called ‘VCenterStandAloneVMCreateTemplate.json’ available at the VConnect install location. For example ‘C:\inetpub\MgmtSvc-CloudAssert-VConnect\bin\Templates’

Code: Operation Template Sample.

```
{
  "Version": "1.7.1",
  "Name": "DeployVMTemplate",
  "Operations": [
    {
      "Name": "CreateResourcePoolOp",
      "Label": "Create Resource Pool",
      "Provider": "VCenterOperationsProvider",
      "MaxRetryAttempts": 3
    },
    {
      "Name": "DeployVMFromTemplateOp",
      "Label": "Clone and Create VM",
      "Provider": "VCenterOperationsProvider",
      "MaxRetryAttempts": 3
    },
    {
      "Name": "GetNetworkSettingsOp",
      "Label": "Get Network Settings",
      "Provider": "VCenterOperationsProvider",
      "MaxRetryAttempts": 3
    },
    {
      "Name": "CustomizeVMOp",
      "Label": "Customize VM",
      "Provider": "VCenterOperationsProvider",
      "MaxRetryAttempts": 3,
      "Params": {
        "NicSettings": ""
      }
    },
    {
      "Name": "VMScriptOp",
      "Label": "Execute Script After Customization",
      "Provider": "VCenterOperationsProvider",
      "MaxRetryAttempts": 3,
      "Params": {
        "scriptfilename": "AfterVMCustomizationScript.ps1"
      }
    },
    {
      "Name": "FinishCreateVMOp",
      "Label": "Finalize and Mark VM Ready",
      "Provider": "VCenterOperationsProvider",
      "MaxRetryAttempts": 3
    }
  ]
}
```

An important aspect of these steps is that any operation can return a result details as a JSON object and those results are available for the next operation.

For example ‘GetNetworkSettingsOp’ returns ‘NicSettings’ JSON result details, which is consumed by the default ‘CustomizeVMOp’ to setup the network settings for the VM.

Custom Operations - Custom PowerShell Script Operation - VMScriptOp

VConnect introduced an operation called ‘VMScriptOp’ that enables adding any PowerShell Scripts as part of this deployment flow.

Example scenarios supported by this customization:

- Update a CMDB database during VM Provisioning
- Add an entry to a legacy inventory system
- Update VM configuration additionally
- Add the VM to a monitoring system
- Add the VM to an external backup system

An example of a custom VMScriptOp is shown below in GetNetworkSettingsOp.

How to implement custom IP Address and Network setting?

‘GetNetworkSettingsOp’ is a notable step in the default template, after which you can add a custom VMScriptOp PowerShell script to customize network settings. Any custom step that likes to set network settings is expected to return a standard format result for NicSettings as shown in the script examples below.

Code: Operation Template Custom Scrip Operation Sample

```
    {
         "Name": "GetNetworkSettingsOp",
         "Label": "Add Default Network Settings",
         "Provider": "VCenterOperationsProvider",
         "MaxRetryAttempts": 3
     },
     {
         "Name": "VMScriptOp",
         "Label": "Get Custom Network Settings",
         "Provider": "VCenterOperationsProvider",
         "MaxRetryAttempts": 3,
         "Params": {
         "scriptfilename": "GetNetworkSettingsScript.ps1"
        }
     },
```

The expected format for the result back should be in the format shown in an example script below:

Code: Custom Script Operation Script Sample

```
# This $nicsettings will be provided as input parameter by the system.
shown here for sample:
# $nicsettings = '[{"NetworkName":"VM
Network","IpMode":"UseDhcp","IpPoolName":null,"IpAddress":null,"SubnetMask"
:null,"DefaultGateway":null,"AlternateGateway":null,"Dns":null}]'

#-------- Start Script ----------------------------------------#
$nicSettingsInput = ConvertFrom-Json -InputObject $nicsettings
$nicSettingsOutput = @();

# Replace following with your own logic to query an extenal system to get
and allocate IP Address
$i = 190;
foreach($nicInput in $nicSettingsInput)
```

© Copyright 2015-2016 CLOUD ASSERT LLC

```
{
    $n = @{
            NetworkName = 'VM Network'
            IpMode = 'UseStaticIP'
            IpPoolName = 'custom'
            IpAddress = '192.168.2.' + $i
            SubnetMask = '255.255.255.0'
            DefaultGateway = '192.168.2.1'
            AlternateGateway = '192.168.2.1'
            Dns = @('8.8.8.8','192.168.2.1')
    }
    $nicSettingsOutput += $n;
    $i++;
}
# End of Network Settings Custom Logic

# Following Result format must be strictly followed to replace the default
$nicSettings with your value
$nJson =  ConvertTo-Json -InputObject $nicSettingsOutput -Compress;
$resultDataDict = @{ "NicSettings" = $nJson };
$resultData = ConvertTo-Json -InputObject $resultDataDict -Compress

$result = @{
                    IsSuccess = $true
                    Message = "Custom IP Network settings determined
successfully."
                    ErrorCode = 0
                    Details = $resultData
           }

New-Object PSObject -Property $result
```

IMPORTANT:

- You must use -Compress option in ConvertTo-Json
- You must Convert the NicSettings array object and the final resultDataDict to JSON before adding to $result.Details
- The result NicSettings param returned is an array of 1 or more network settings. This NicSettings result count MUST match the actual counts received as input param $nicSettings.

Custom Operations - Write your own Custom Operation in .Net

VConnect Operation Templates can be extended with custom operations written in .Net. Similar to PowerShell based VMScriptOp, but this is a full-fledged support adding Operations of any kind, not just involving VMWare VMs.

Example scenarios that this customization enables:

- Write complex operations in powerful language
- Attract developers to incorporate VConnect operations as part of their development deliverable
- Interact with wide range of enterprise and custom applications by leveraging existing client code

As of the latest release you would need to following assemblies:

CloudAssert.Operations.Common.dll

CloudAssert.WAP.VConnect.Public.Contracts.dll

These assemblies can be found in the VConnect install folder or contact support to obtain them.

Implementing a custom Operation is as simple as extending from IOperation or one of the derived classes like VConnectOperation and implementing couple of methods as shown in below sample.

Code: .Net Custom Operation Sample

```
    /// <summary>
    /// Sample IOperation implementation that can be used in OperationTemplate
    /// Implement: CloudAssert.WAP.VConnect.Public.Contracts.VConnectOperation
    /// </summary>
    public class UpdateCMDBOperation : 
CloudAssert.WAP.VConnect.Public.Contracts.VConnectOperation
    {
        public UpdateCMDBOperation(IVConnectOpsProviderRepositoryFactory factory, 
JsonParameters inputParams)
            : base(factory, inputParams)
        {

        }

        public override OperationResult Execute()
        {
            // Do your processing here

            return OperationResult.GetSuccess("UpdateCMDBOperation Succeeded");
        }

        public override OperationResult GetAsyncResult(string asyncState)
        {
            // Unless your operation returns 
OperationResultStatus.WaitingForAsyncCompletion,
            // you dont need to implement this.
            throw new NotImplementedException();
        }
    }
```

In order to take advantage of this new custom operation, you can include it in the Operation Template. For example in ‘VCenterStandAloneVMCreateTemplate.json’ add the following snippet:

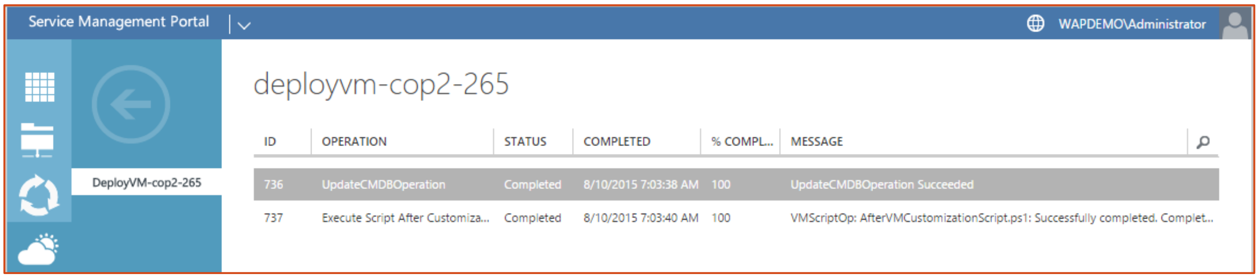
Code: Custom .Net Operation in Operation Template JSON

```
        {
          "Name": "EnterpriseITWebVConnectExtensions.UpdateCMDBOperation, 
EnterpriseITWebVConnectExtensions",
          "Label": "UpdateCMDBOperation",
          "Provider": "VCenterOperationsProvider",
          "MaxRetryAttempts": 1
        },
```

Where ‘Name’ is the fully qualified type name of format “Namespace.TypeName, AssemblyName”, as mentioned in this MSDN document for Type.AssemblyQualifiedName.

Leave the Provider as “VCenterOperationsProvider” or empty. You can potentially write your own provider for even more advanced scenarios, for now the default is sufficient.

Screenshot custom Operation as part of a sample deployment:



deployvm-cop2-265

ID	OPERATION	STATUS	COMPLETED	% COMPL...	MESSAGE
736	UpdateCMDBOperation	Completed	8/10/2015 7:03:38 AM	100	UpdateCMDBOperation Succeeded
737	Execute Script After Customiza...	Completed	8/10/2015 7:03:40 AM	100	VMScriptOp: AfterVMCustomizationScript.ps1: Successfully completed. Complet...

Customize VConnect Scriptlets:

VConnect enables extending and customizing default behavior by providing ability to override the default VM provisioning mechanisms.

VConnect by default uses Scriptlets, which are PowerShell scripts to provision Virtual Machines and to perform some of the standard operations. These Scriptlets are not available by default, but upon request we can provide these Scriptlets and place it in the VConnect installation folder.

These scriptlets can be modified to customize the default operations. Generally this type of customizations are not encouraged unless you work with the product team and it is mutually understood that the scenario you are trying to satisfy is not addressable with other customization mechanisms.

Example scenarios that this customization enables:

- Change how VConnect provisions at the lowest level of provisioning scripts

Customize the columns shown in Virtual Machine list grid

This is an advanced scenario that helps display custom properties of a Virtual Machine in the VConnect WAP Portal UI.

Example scenarios that this customization enables:

- Tag Virtual Machines with custom properties to track
- Properties are stored in VMWare VM annotations so not dependent on VConnect or WAP
- Can be read and updated by other tools and systems in an Enterprise environment

How-to update custom properties to display:

In VConnect Database table ‘AppSettings’, entry with Key = ‘CustomPropertiesToDisplay’ controls the custom properties displayed. Its value takes list of comma separated field names. By default this setting has `‘Description, Expiration’` values out of the box. You can however update to add your own custom properties to be pulled from the VMWare Virtual Machine Annotations. These names must correspond to a VMWare Virtual Machine Annotation property name, if not you may run into errors.

IMPORTANT: Because this setting is not exposed in Admin Portal UI yet, following workaround to update the database directly is provided. You must take a backup of the database before doing any updates to the database such as this.

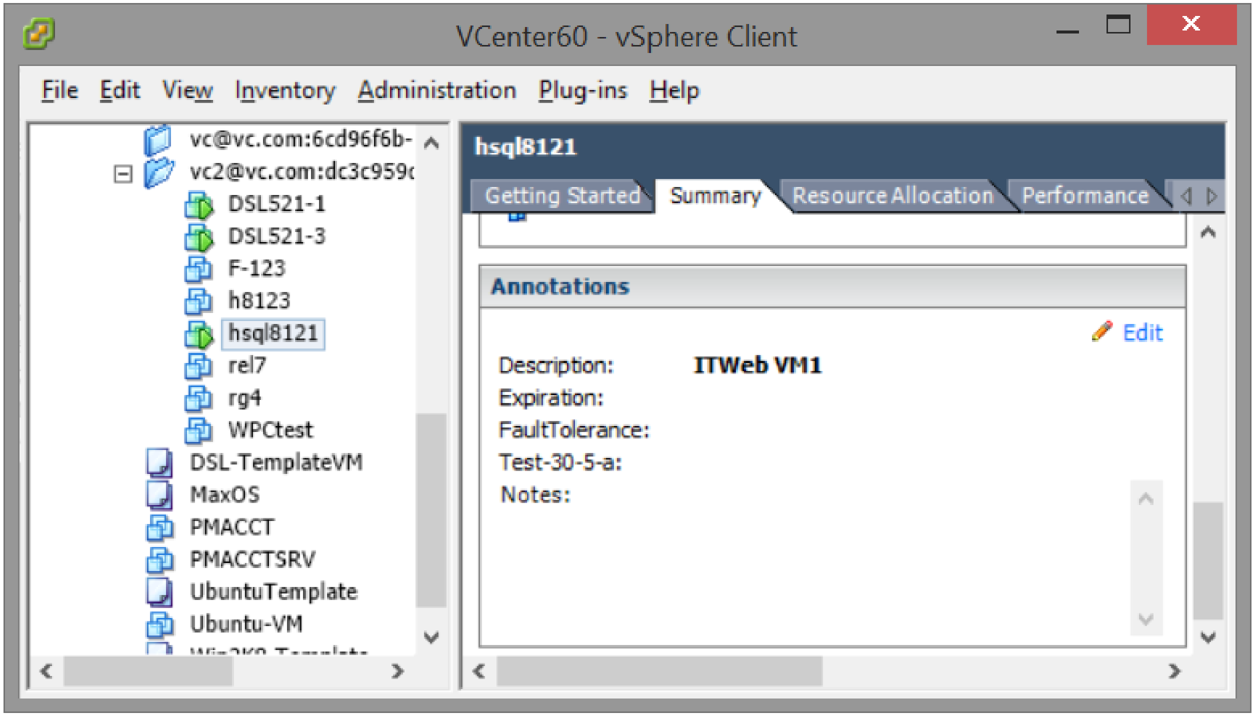
Table: SQL Query to update the custom properties to display

```
USE [VConnect12New]
GO
```

```
-- IMPORTANT: Take a backup of what was there already
SELECT *FROM AppSettings WHERE [Key] = 'CustomPropertiesToDisplay'
GO

-- Update with the custom properties to show.
-- Must match with VMWare VM Annotation Properties
UPDATE [dbo].[AppSettings]
   SET [Value] = 'Description,Expiration'
      ,[UpdatedByUser] = 'administrator'
      ,[LastUpdatedTime] = GETDATE()
WHERE [Key] = 'CustomPropertiesToDisplay'
GO
```

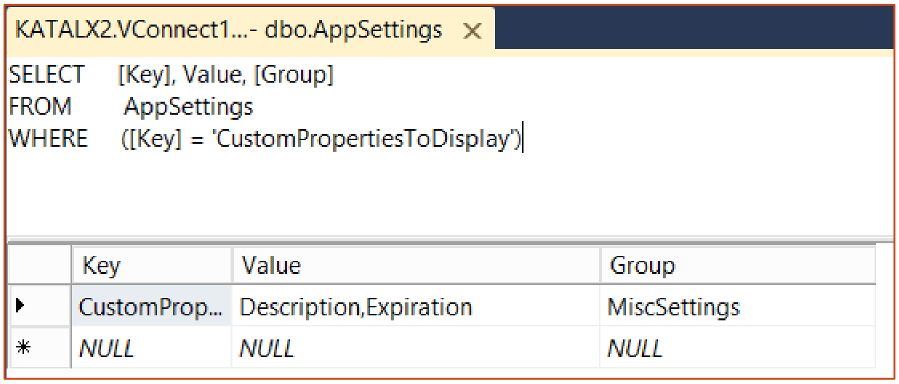
Screenshot: vSphere Virtual Machine summary showing the Annotations with custom properties used.



Description: ITWeb VM1
Expiration:
FaultTolerance:
Test-30-5-a:
Notes:

© Copyright 2015-2016 CLOUD ASSERT LLC

Screenshot: VConnect Database AppSettings table with the CustomPropertiesToDisplay settings

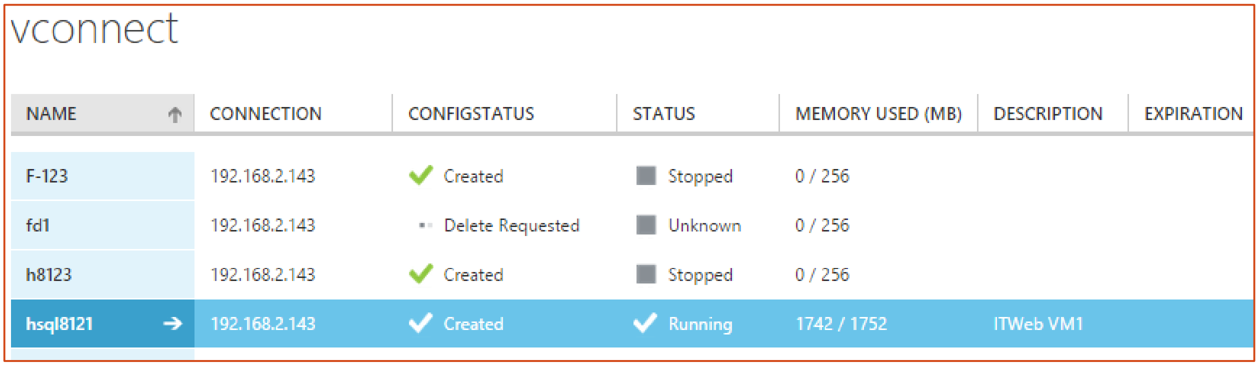


KATALX2.VConnect1...- dbo.AppSettings

```
SELECT    [Key], Value, [Group]
FROM       AppSettings
WHERE     ([Key] = 'CustomPropertiesToDisplay')
```

	Key	Value	Group
▶	CustomProp...	Description,Expiration	MiscSettings
*	*NULL*	*NULL*	*NULL*

Screenshot: VConnect Tenant Portal UI with custom properties displayed



vconnect

NAME ↑	CONNECTION	CONFIGSTATUS	STATUS	MEMORY USED (MB)	DESCRIPTION	EXPIRATION
F-123	192.168.2.143	✓ Created	■ Stopped	0 / 256		
fd1	192.168.2.143	Delete Requested	■ Unknown	0 / 256		
h8123	192.168.2.143	✓ Created	■ Stopped	0 / 256		
hsql8121 →	192.168.2.143	✓ Created	✓ Running	1742 / 1752	ITWeb VM1	

Enable Auto Sync Feature:

You might be aware that there is already an existing feature that helps administrators to onboard existing VMWare Virtual Machines to VConnect + WAP via 'Sync' command in the VConnect WAP Admin portal.

This advanced option allows to configure automatic periodic sync with VMWare vCenter clusters. Remember that use this option only if required as this might add additional workload on vCenter and VConnect.

Example scenarios that this customization enables:

- Create a VM through Request Management or other means, and see it appear in VConnect + WAP automatically
- Admins and users who like to still use vSphere or other client tools to create or delete VMs outside of the WAP + VConnect infrastructure can now use their existing tools along with WAP + VConnect
- VMs deleted outside due to non-compliance or expiry can be auto removed from WAP + VConnect
- VMs created for advanced and custom scenarios that is not yet supported by VConnect will be on boarded automatically into WAP + VConnect

Table: SQL Query to Enable Auto Sync

```
USE [VConnect12New]
GO

-- IMPORTANT: Take a backup of what was there already
SELECT *FROM AppSettings WHERE [Key] = 'IsAutoSyncSubscriptionAssets'
```

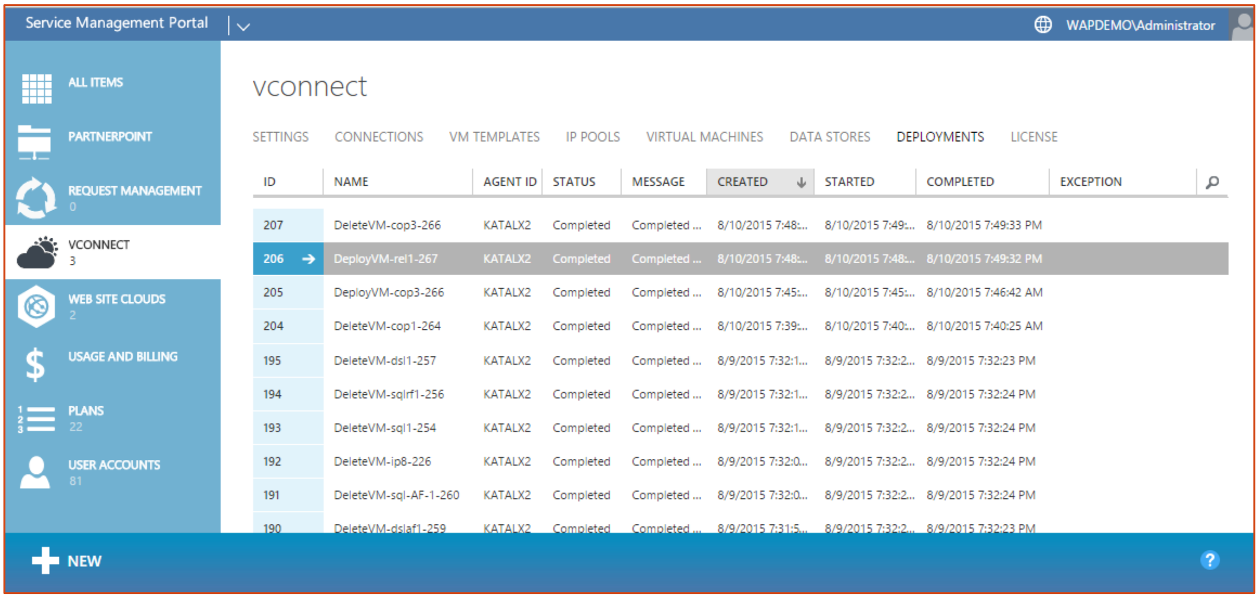
```
GO

UPDATE [dbo].[AppSettings]
   SET [Value] = 'true'
      ,[UpdatedByUser] = 'administrator'
      ,[LastUpdatedTime] = GETDATE()
WHERE [Key] = 'IsAutoSyncSubscriptionAssets'
GO
```

Appendix - FAQ

How to view ongoing operation template deployments?

Admin Portal → VConnect → Deployments

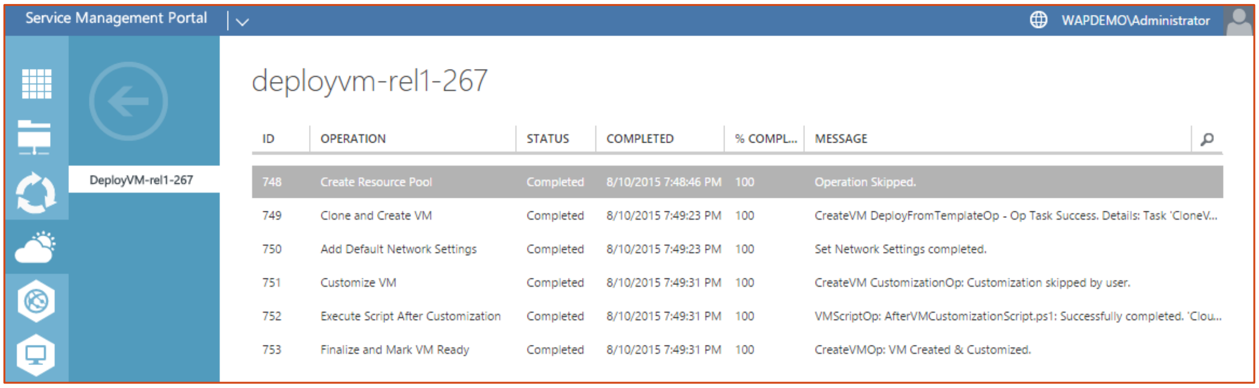


vconnect

SETTINGS | CONNECTIONS | VM TEMPLATES | IP POOLS | VIRTUAL MACHINES | DATA STORES | DEPLOYMENTS | LICENSE

ID	NAME	AGENT ID	STATUS	MESSAGE	CREATED	STARTED	COMPLETED	EXCEPTION
207	DeleteVM-cop3-266	KATALX2	Completed	Completed ...	8/10/2015 7:48:...	8/10/2015 7:49:...	8/10/2015 7:49:33 PM	
206 →	DeployVM-rel1-267	KATALX2	Completed	Completed ...	8/10/2015 7:48:...	8/10/2015 7:48:...	8/10/2015 7:49:32 PM	
205	DeployVM-cop3-266	KATALX2	Completed	Completed ...	8/10/2015 7:45:...	8/10/2015 7:45:...	8/10/2015 7:46:42 AM	
204	DeleteVM-cop1-264	KATALX2	Completed	Completed ...	8/10/2015 7:39:...	8/10/2015 7:40:...	8/10/2015 7:40:25 AM	
195	DeleteVM-dsl1-257	KATALX2	Completed	Completed ...	8/9/2015 7:32:1...	8/9/2015 7:32:2...	8/9/2015 7:32:23 PM	
194	DeleteVM-sqlrf1-256	KATALX2	Completed	Completed ...	8/9/2015 7:32:1...	8/9/2015 7:32:2...	8/9/2015 7:32:24 PM	
193	DeleteVM-sql1-254	KATALX2	Completed	Completed ...	8/9/2015 7:32:1...	8/9/2015 7:32:2...	8/9/2015 7:32:24 PM	
192	DeleteVM-ip8-226	KATALX2	Completed	Completed ...	8/9/2015 7:32:0...	8/9/2015 7:32:2...	8/9/2015 7:32:24 PM	
191	DeleteVM-sql-AF-1-260	KATALX2	Completed	Completed ...	8/9/2015 7:32:0...	8/9/2015 7:32:2...	8/9/2015 7:32:24 PM	
190	DeleteVM-dslaf1-259	KATALX2	Completed	Completed ...	8/9/2015 7:31:5...	8/9/2015 7:32:2...	8/9/2015 7:32:23 PM	

Deployment Operations Drilldown View:



deployvm-rel1-267

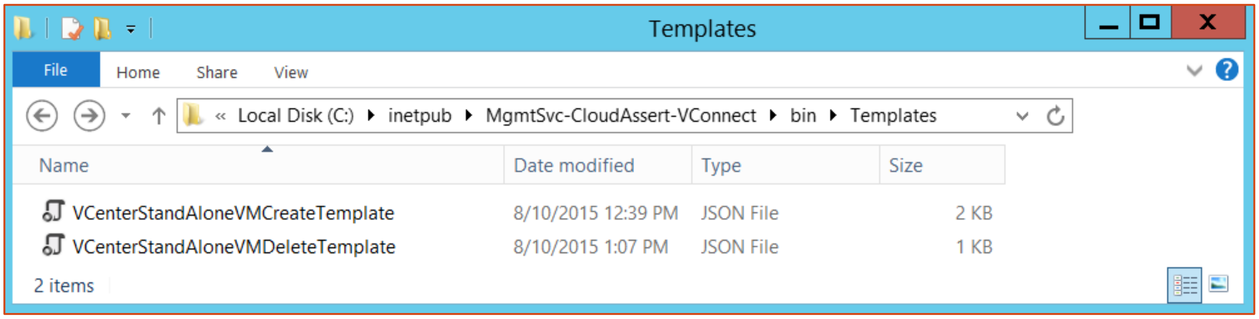
ID	OPERATION	STATUS	COMPLETED	% COMPL...	MESSAGE
748	Create Resource Pool	Completed	8/10/2015 7:48:46 PM	100	Operation Skipped.
749	Clone and Create VM	Completed	8/10/2015 7:49:23 PM	100	CreateVM DeployFromTemplateOp - Op Task Success. Details: Task 'CloneV...
750	Add Default Network Settings	Completed	8/10/2015 7:49:23 PM	100	Set Network Settings completed.
751	Customize VM	Completed	8/10/2015 7:49:31 PM	100	CreateVM CustomizationOp: Customization skipped by user.
752	Execute Script After Customization	Completed	8/10/2015 7:49:31 PM	100	VMScriptOp: AfterVMCustomizationScript.ps1: Successfully completed. 'Clou...
753	Finalize and Mark VM Ready	Completed	8/10/2015 7:49:31 PM	100	CreateVMOp: VM Created & Customized.

© Copyright 2015-2016 CLOUD ASSERT LLC

VConnect V1.0

Where are the Operation Templates found in VConnect install folder?

Currently default template for VM Creation named ‘VCenterStandAloneVMCreateTemplate.json’ and default template for VM Deletion named ‘VCenterStandAloneVMDeleteTemplate.json’ are available under the install location similar to this ‘C:\inetpub\MgmtSvc-CloudAssert-VConnect\bin\Templates’.


Templates
File
Home
Share
View
« Local Disk (C:) ▸ inetpub ▸ MgmtSvc-CloudAssert-VConnect ▸ bin ▸ Templates

Name	Date modified	Type	Size
VCenterStandAloneVMCreateTemplate	8/10/2015 12:39 PM	JSON File	2 KB
VCenterStandAloneVMDeleteTemplate	8/10/2015 1:07 PM	JSON File	1 KB

2 items

Where to place the Custom PowerShell scripts used in the operations?

Drop the PowerShell scripts in each of the VConnect installed machine under ‘C:\inetpub\MgmtSvc-CloudAssert-VConnect\’

Where to place the .Net Custom Operation DLLs used in the operations?

Drop the custom .NET assemblies and all dependent files in each of the VConnect installed machine under ‘C:\inetpub\MgmtSvc-CloudAssert-VConnect\bin’

What are the default parameters provided by VConnect and available inside custom scripts?

Default Parameters that are usually passed to custom scripts from VConnect:

```
$VMTemplateName = $null
$IsPowerOn = $false
$CPUCores = 0
$MemoryMB = 0
$OSType = $null
$scriptfilename = 'GetNetworkSettingsScript.ps1'
$vmname = 'nic1'
$vmtemplateid = '6'
$vsphereconnectionid = '2'
$vmconfigdataid = '227'
$useremailid = 'vc@vc.com'
```

© Copyright 2015-2016 CLOUD ASSERT LLC

```
$subscriptionid = '1dcef7cb-da40-4c99-bcbc-bf54ff0de635'
$iscreateresourcepoolpertenant = 'false'
$name =
'CloudAssert.WAP.VConnect.VMWare.Operations.CodedOperationTemplates.DeployV
MTemplate'
$isskip_createresourcepoolop = 'true'
$nicsettings = '[{"NetworkName":"VM
Network","IpMode":"UseDhcp","IpPoolName":null,"IpAddress":null,"SubnetMask"
:null,"DefaultGateway":null,"AlternateGateway":null,"Dns":null}]'
$NetworkName1 = 'VM Network'
$IpMode1 = $null
$IpPoolName1 = $null
$DataStoreHint = 'Samsung'
$IsSkipCustomization = $true
$UserName = 'admin@vsphere.local'
$Password = 'xxxxxxxxxxxxxxx'
$Datacenter = 'TestDataCenter60'
$Cluster = 'TestCluster60'
$HostName = $null
$HostServerName = '192.168.290.443'
$HostServerPort = 443
$ResourcePoolName = $null
$FolderName = 'vc@vc.com:1dcef7cb-da40-4c99-bcbc-bf54ff0de635'
$IsSkipIfFileNotExist = $true
```