Types of failover operations in Hyper-V Replica – Part I – Test Failover

At a high level, Hyper-V Replica supports three types of Failover:

- Test Failover
- Planned Failover
- Unplanned Failover

Each of these is built to meet specific and different needs. As you might know by now, using Hyper-V Replica you can replicate between different primary and replica site deployments – your primary server could be a standalone server or part of a cluster and similarly your replica server could be a standalone server or part of a cluster. Independent of your deployment, the workflow and the set of features offered are the same and the 3 types of failovers work in all deployments.

For this article, let's assume the name of the virtual machine is ***VirtualMachine_Workload***.

Test Failover (TFO)

1. What is Test Failover?

Test Failover is an operation initiated on your replica virtual machine which allows you to test the sanity of the virtualized workload without interrupting your production workload or ongoing replication.

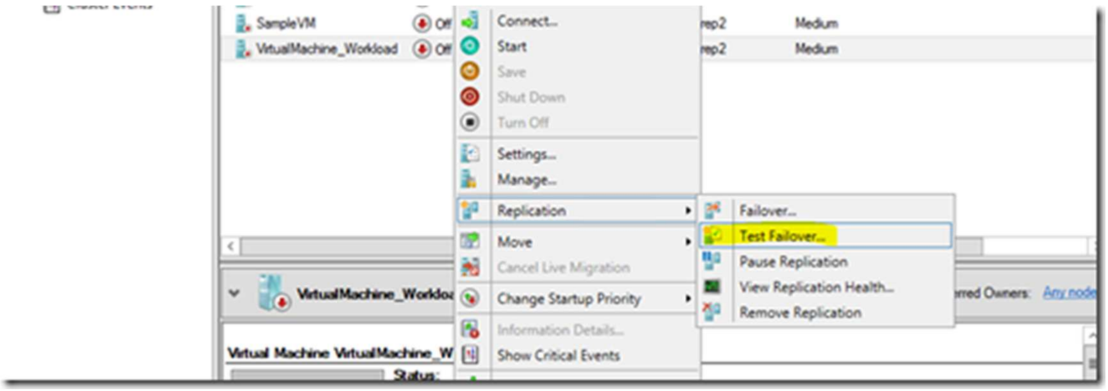
2. When should I use Test Failover?

Think of Test Failover as an ability to non-disruptively simulate your recovery procedure in an isolated network. You should initiate this operation if you wish to:

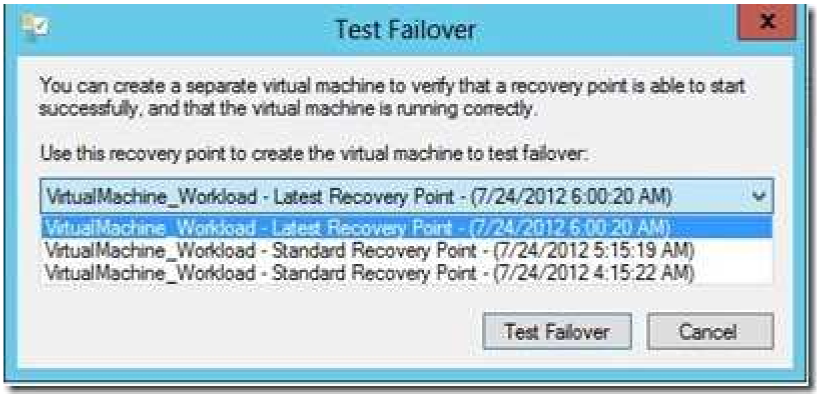
- Run minimal tests to validate if your replication is on track
- Train your personnel on what is to be done in case of a disaster.
- Test the recovery plan that you have built to test your preparation when disaster does strike.

3. How should I use this feature?

TFO is performed on the replica virtual machine by right-clicking on the VM and choosing the Test Failover operation (either from the Hyper-V Manager or from the Failover Clustering Manager)



You are given a choice to pick one of the available recovery points.



You can create a separate virtual machine to verify that a recovery point is able to start successfully, and that the virtual machine is running correctly.

Use this recovery point to create the virtual machine to test failover:

VirtualMachine_Workload - Latest Recovery Point - (7/24/2012 6:00:20 AM)

VirtualMachine_Workload - Latest Recovery Point - (7/24/2012 6:00:20 AM)
VirtualMachine_Workload - Standard Recovery Point - (7/24/2012 5:15:19 AM)
VirtualMachine_Workload - Standard Recovery Point - (7/24/2012 4:15:22 AM)

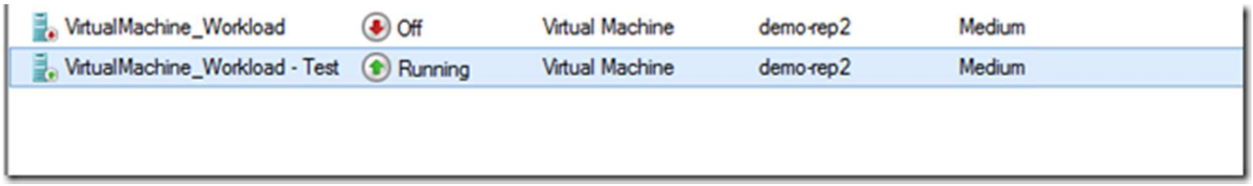
There are following types of recovery points available for your selection:

- Latest Recovery Point
- Standard Replicas (multiple)
- Application-Consistent Recovery Point (multiple)

"Latest Recovery Point" is the backup of Primary VHD of the Replica Virtual Machine which will be used to test the failover.

"Application-Consistent Recovery Point" and "Standard Replicas" are created by the Volume Shadow Copy Service (VSS) based on the interval you specified when configuring this Virtual Machine for replication.

After this, a NEW virtual machine is spun up on the replica site. The name of the new virtual machine is the name of the replica virtual machine with **" – Test"** appended. In our example, it would be *VirtualMachine_Workload – Test*



VirtualMachine_Workload	Off	Virtual Machine	demo-rep2	Medium
VirtualMachine_Workload - Test	Running	Virtual Machine	demo-rep2	Medium

In fact, "Test Failover" operation uses Hyper-V's "Export and Import" feature internally to create a new Virtual Machine and then rename it. The "Test Failover" operations include:

1. Export the Replica Virtual Machine to a temporary location (XML file, VHD files, etc.)
2. Modify the XML file of exported Virtual Machine to use a unique GUID
3. Register the newly created Virtual Machine with Hyper-V (VMMS.exe)
4. Rename the Virtual Machine name
5. Import the Virtual Machine onto the same Hyper-V server

NOTE: The Test Virtual Machine remains OFF after it has been imported successfully, by default, an imported Virtual Machine is not started automatically.

After turning Test Virtual Machine (*VirtualMachine_Workload – Test*) ON and once you are satisfied that it functions properly, you can always delete the test Virtual Machine from the Replica Server. Before you delete the test Virtual Machine, use "**Stop Test Failover**" action from the Right Click Context menu. This action will clean up the duplicate files (VHD and configuration files) for the test Virtual Machine.

NOTE: You must have enough resources available on the Replica Server in order to start the test Replica Virtual Machine and enough storage for "Test Failover" operation to create/store VHD files.

Tip: Only one "Test Failover" operation is allowed at one time for a Replica Virtual Machine. However, you can perform multiple "Test Failover" operations for different Replica Virtual Machines at the same time.

Tip: Since the "Test Failover" operation is performed at the Replica Site, there is no impact on the Primary Virtual Machine which continues to serve the client requests.

The TFO virtual machine should then be started in an isolated network and client tests can be run against the same to validate replication. You can pre-assign a network and an IP address using the guest IP address injection feature. Once satisfied that replication is kosher, you should do **"Stop Test Failover"** on the Replica virtual machine, which will clean up the duplicate virtual machine.

Since Test Failover does NOT impact your production workload and does NOT impact your ongoing replication, it is recommended that you perform TFO regularly. There are a couple of mechanisms which help you track the frequency of this event – BPA rules and replication health.

Network connectivity: The virtual NIC will be connected to the pre-designated test virtual switch.

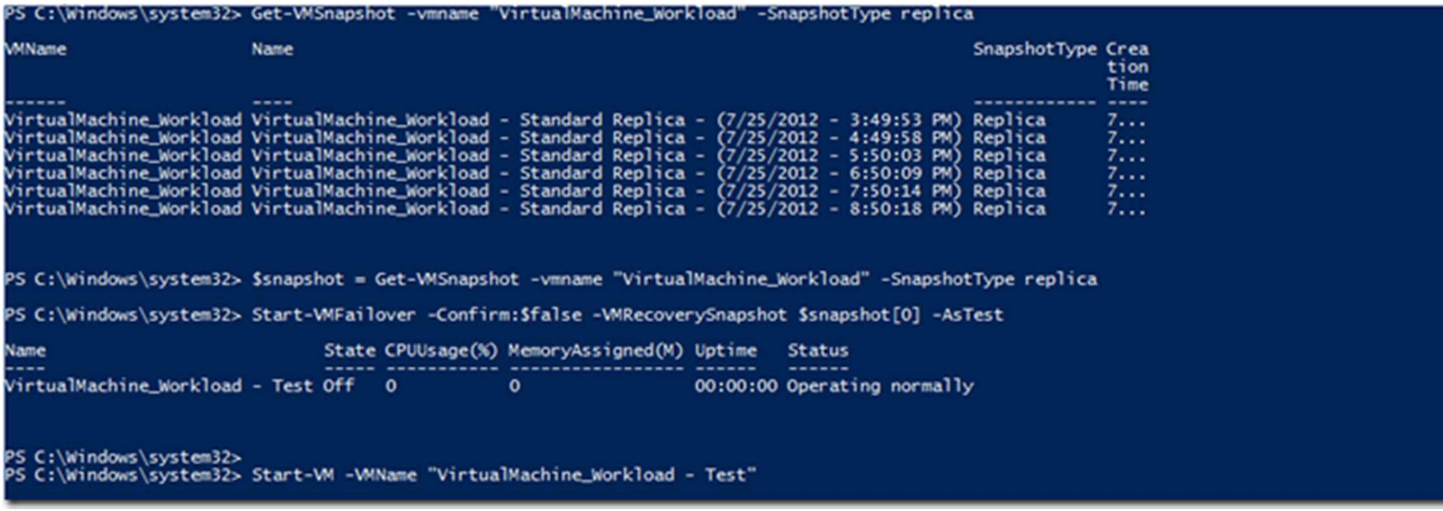
Virtual hard disks: Instead of slowly copying the virtual hard disks, differential disks are created and linked to the virtual hard disks of the replica VMs, working similarly to a checkpoint (or Hyper-V snapshot). This allows the test VM to be created quickly and use the disk contents of the replica VM as a starting point. The parent virtual hard disks are read from and newer data is written to and read from the differential virtual hard disks

The first step is to prepare the networking for a test failover.

- **Create one or more isolated virtual switches**: These networks will be used to bring your test failover VMs online without causing address, name, or service conflicts with production systems. Create these networks on the hosts in the secondary site where you will perform the test failover.
- **Assign VMs to isolated virtual switches**: Configure the connection of the virtual network in the replica VM for test purposes by expanding the properties of the VM's virtual NIC, browsing to test failover and selecting the test failover virtual switch under Virtual Switch.

Not only is this approach quick to start up, but it also means that replication will continue uninterrupted – we wouldn't want Mr. Murphy to cause a disaster during a BCP test and leave us hanging!

The above procedure can be achieved using Powershell using the following cmdlets.



```
PS C:\Windows\system32> Get-VMSnapshot -vmname "VirtualMachine_Workload" -SnapshotType replica

VMName                  Name                                                                  SnapshotType Crea
                                                                                                           tion
                                                                                                           Time
------                  ----                                                                  ------------ ----
VirtualMachine_Workload VirtualMachine_Workload - Standard Replica - (7/25/2012 - 3:49:53 PM) Replica      7...
VirtualMachine_Workload VirtualMachine_Workload - Standard Replica - (7/25/2012 - 4:49:58 PM) Replica      7...
VirtualMachine_Workload VirtualMachine_Workload - Standard Replica - (7/25/2012 - 5:50:03 PM) Replica      7...
VirtualMachine_Workload VirtualMachine_Workload - Standard Replica - (7/25/2012 - 6:50:09 PM) Replica      7...
VirtualMachine_Workload VirtualMachine_Workload - Standard Replica - (7/25/2012 - 7:50:14 PM) Replica      7...
VirtualMachine_Workload VirtualMachine_Workload - Standard Replica - (7/25/2012 - 8:50:18 PM) Replica      7...


PS C:\Windows\system32> $snapshot = Get-VMSnapshot -vmname "VirtualMachine_Workload" -SnapshotType replica

PS C:\Windows\system32> Start-VMFailover -Confirm:$false -VMRecoverySnapshot $snapshot[0] -AsTest

Name                           State CPUUsage(%) MemoryAssigned(M) Uptime   Status
----                           ----- ----------- ----------------- ------   ------
VirtualMachine_Workload - Test Off   0           0                 00:00:00 Operating normally


PS C:\Windows\system32>
PS C:\Windows\system32> Start-VM -VMName "VirtualMachine_Workload - Test"
```

Ref: https://www.petri.com/perform-hyper-v-replica-test-failover

1. What is Planned Failover?

PFO is an operation initiated on the primary VM which allows you to do an e2e validation of your recovery plan. PFO requires the VM to be shut down to ensure consistency.

PFO is *NOT* a substitute for High Availability which is achieved through clustering. PFO allows you to keep your business running with minimal downtime even during planned downtimes and guarantees zero data loss.

2. When should I use Planned Failover?

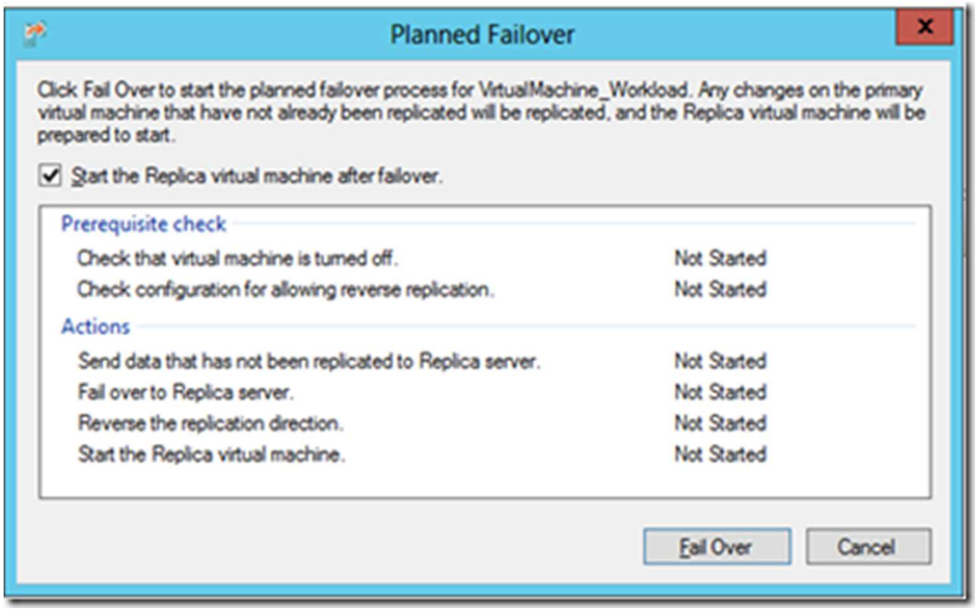
Planned Failover is used in the following cases

- I want to perform host maintenance on my primary and would like to run from the replica site.

- My primary site is expecting some power outage – I want to move over to the replica site.
- There's an impending typhoon – I want to proactively take action to ensure business continuity.
- My compliance requirements mandate that every quarter, I run my workloads from the replica site for a week.

3. How should I use Planned Failover?

After turning off the primary virtual machine, PFO is performed on the primary virtual machine by right-clicking on the VM and choosing the **Failover** operation (either from the Hyper-V Manager or from the Failover Clustering Manager).



Click Fail Over to start the planned failover process for VirtualMachine_Workload. Any changes on the primary virtual machine that have not already been replicated will be replicated, and the Replica virtual machine will be prepared to start.

☑ Start the Replica virtual machine after failover.

Prerequisite check	
Check that virtual machine is turned off.	Not Started
Check configuration for allowing reverse replication.	Not Started
Actions	
Send data that has not been replicated to Replica server.	Not Started
Fail over to Replica server.	Not Started
Reverse the replication direction.	Not Started
Start the Replica virtual machine.	Not Started

The following requirements must be met before the "Planned Failover" operation can start:

- Primary Server is configured to accept replication
- The Primary Virtual Machine is turned off

Usually, in a Hyper-V Replica environment, it is the Replica Server which accepts replication packets from a Primary Server running in production site but not the vice-versa - in some cases it is required by the Hyper-V Replica to reverse replicate the virtualized workload. The "Reverse Replicate" action requires this type of configuration, this will be touched on **again later in this** article.

A "Planned Failover" operation initiated for a Primary Virtual Machine is initiated manually by an administrator. Since it is a "planned event", Hyper-V Primary Server knows what all actions need to be taken for a Primary Virtual Machine. This is what happens as part of the operation:

1. An administrator initiates the "Planned Failover" action from a Right Click context menu of Primary Virtual Machine

2. Hyper-V server process (VMMS.exe) is notified of the action
3. VMMS.exe talks to Hyper-V VSS Writer to create a snapshot of the Primary Virtual Machine
4. Hyper-V VSS Writer creates a "Standard Replica" backup copy.
5. The Replica Server is notified of the event.
6. "Standard Replica" backup copy is sent to the Replica Server
7. The Replica Server applies the backup copy it has received and starts the Replica Virtual Machine

In case of a Planned Failover, there is no data loss. Since the Primary Server is informed of the action, it takes the necessary actions on the Primary Virtual Machine ensuring no data loss occurs because they are sent to Replica Server.

Please note – for the PFO to work from UI, remote WMI has to be enabled on the replica site and the user running the PFO should have the necessary privileges. More on remote WMI in a later blog

4. How does Planned Failover work?

PFO does the following

- Performs pre-requisite checks to ensure the operation can succeed. These pre-requisite checks are:
    - As PFO is a “planned” activity with zero data, the primary virtual machine should be shut down before initiating the operation.
    - Once the VM is failed over in the replica site, Hyper-V Replica starts replicating the changes back to the primary server. For this to work, the primary server should be configured to receive replication from the Hyper-V Replica Broker on the replica side if the replica is a cluster or the replica server if the replica is a standalone.
- Sends the last set of tracked changes which ensures zero data loss.
- Reverses the direction of replication – so if you were replicating from ***Cluster-P*** (on the primary site) to ***Cluster-R*** (on the replica site), then the replication will happen from ***Cluster-R*** to ***Cluster-P***. The primary virtual machine will become the replica virtual machine and vice-versa and all the recovery points are merged. This step is done at the end of PFO.
- If you have selected to start the virtual machine, the virtual machine is started on the ***Cluster-R*** site at the end of the operation. This boots off the latest point. You might decide not to start the virtual machine if the virtual machine is part of a multi-tiered application.

The above procedure can be achieved using Powershell using the following cmdlets.

Run these cmdlets on the primary side.

```
Stop-VM VirtualMachine_Workload

Start-VMFailover -VMName VirtualMachine_Workload –prepare
```

Run these cmdlets on the replica side.

```
Start-VMFailover -VMName VirtualMachine_Workload

Set-VMReplication -reverse -VMName VirtualMachine_Workload

Start-VM VirtualMachine_Workload
```

On a cluster, these cmdlets should be run against the node which is currently owning the virtual machine.

1. What is Unplanned Failover?

Unplanned Failover is an operation initiated on the replica VM when the primary VM/site is hit by a disaster. During Unplanned Failover, a check is done using Remote WMI to see if the primary VM is running. This is to protect against accidental administrator actions on the replica VM. This check prevents a 'split-brain' scenario where both the production and the replica VMs are running.

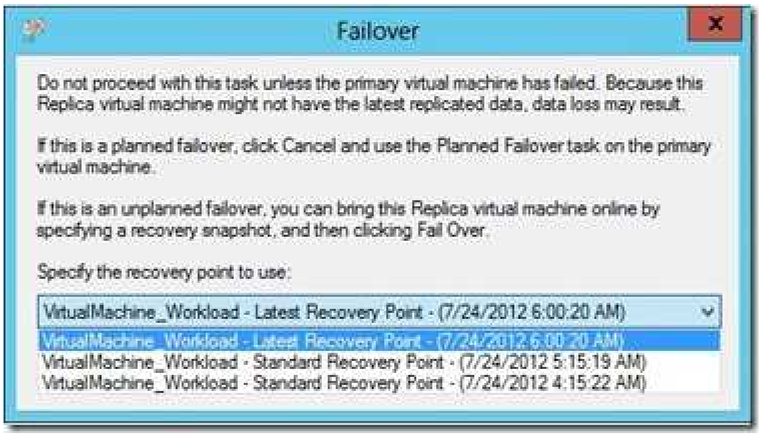
2. When should I use Unplanned Failover?

Unplanned Failover is used in the following cases

- My primary site is experiencing unexpected power outage or a natural disaster
- My primary site/VM has had a virus attack and I want to restore my business quickly with minimal data loss by restoring my replica VM

3. How should I use Unplanned Failover?

Unplanned failover is performed on the replica virtual machine by right-clicking on the VM and choosing the **Failover** operation (either from the Hyper-V Manager or from the Failover Clustering Manager).



Failover

Do not proceed with this task unless the primary virtual machine has failed. Because this Replica virtual machine might not have the latest replicated data, data loss may result.

If this is a planned failover, click Cancel and use the Planned Failover task on the primary virtual machine.

If this is an unplanned failover, you can bring this Replica virtual machine online by specifying a recovery snapshot, and then clicking Fail Over.

Specify the recovery point to use:

VirtualMachine_Workload - Latest Recovery Point - (7/24/2012 6:00:20 AM)

VirtualMachine_Workload - Latest Recovery Point - (7/24/2012 6:00:20 AM)
VirtualMachine_Workload - Standard Recovery Point - (7/24/2012 5:15:19 AM)
VirtualMachine_Workload - Standard Recovery Point - (7/24/2012 4:15:22 AM)

If you have turned on recovery history, Unplanned Failover can be performed against a previous point-in-time. This is usually done in case the most recent point is either corrupt or not application consistent. Once you failover, you should run some tests to check that the point-in-time is good. If the point-in-time has issues, you can cancel the failover using **“Cancel Failover”** on the replica VM. Then you can choose a different point-in-time and do a Failover.

After you have validated that the failed over VM is kosher, you should do a **‘Complete’** of the failover by performing an action on the replica virtual machine – this will ensure that the recovery points are merged.

The above procedure can be achieved using Powershell using the following cmdlets. Use Complete-VMFailover only after ensuring that the failed over VM serves the purpose.

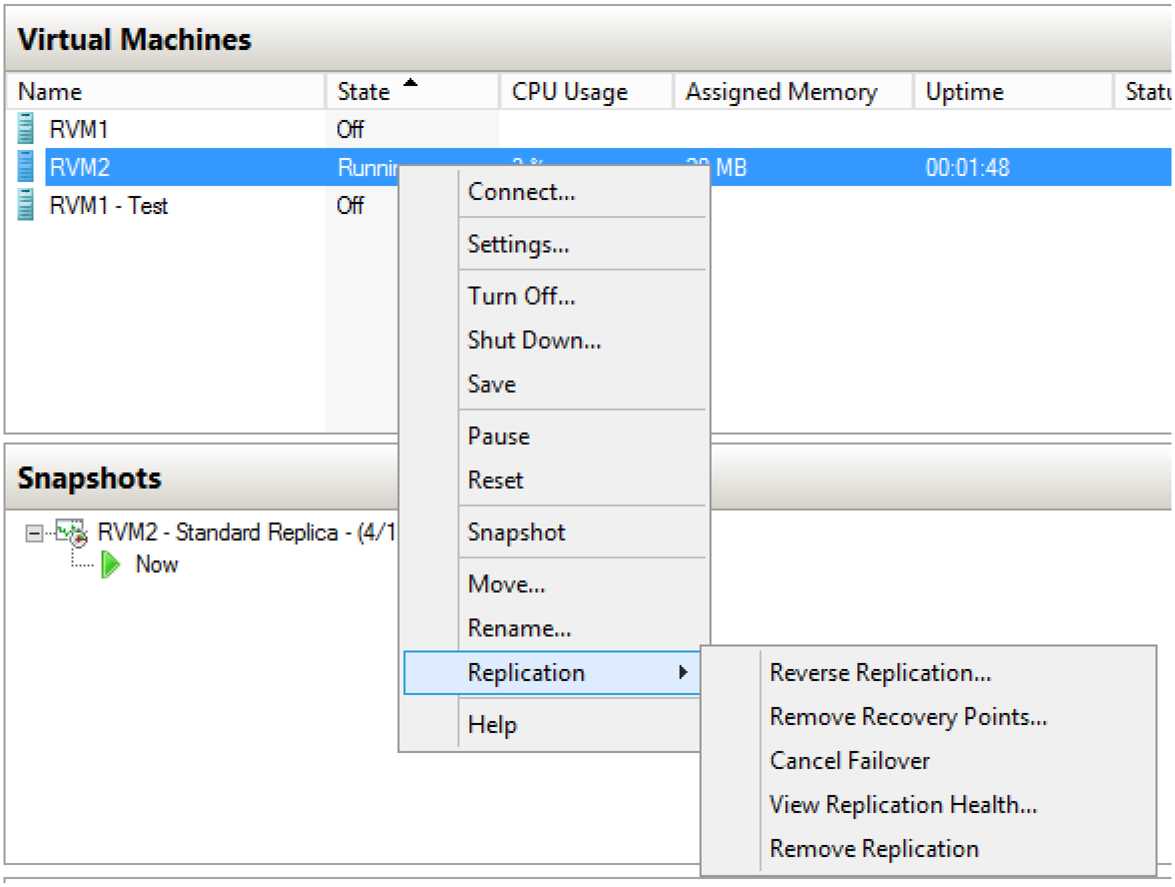
```
$snapshots = Get-VMSnapshot -VMName VirtualMachine_Workload -SnapshotType Replica

Start-VMFailover -Confirm:$false -VMRecoverySnapshot $snapshots[0]

Complete-VMFailover -VMName VirtualMachine_Workload -Confirm:$false
```

Cancel Failover and Reverse Replication

Cancel Failover" and ‘**Reverse Replication**" actions are available on the Replica Virtual Machine on which you initiated the "Failover" action. These two actions are shown in the below Figure 1.28.



Virtual Machines

Name	State	CPU Usage	Assigned Memory	Uptime	Statu
RVM1	Off				
RVM2	Runnir		MB	00:01:48	
RVM1 - Test	Off				

Snapshots

RVM2 - Standard Replica - (4/1

Now

"**Cancel Failover**" action allows you to cancel the failover and turn off the Replica Virtual Machine. The "**Cancel Failover**" action is created in a situation where a Replica Virtual Machine restored from a recovery point is not working normally. You can always initiate the "Cancel Failover" action to revert back the changes and then start over with a different recovery backup copy this time.

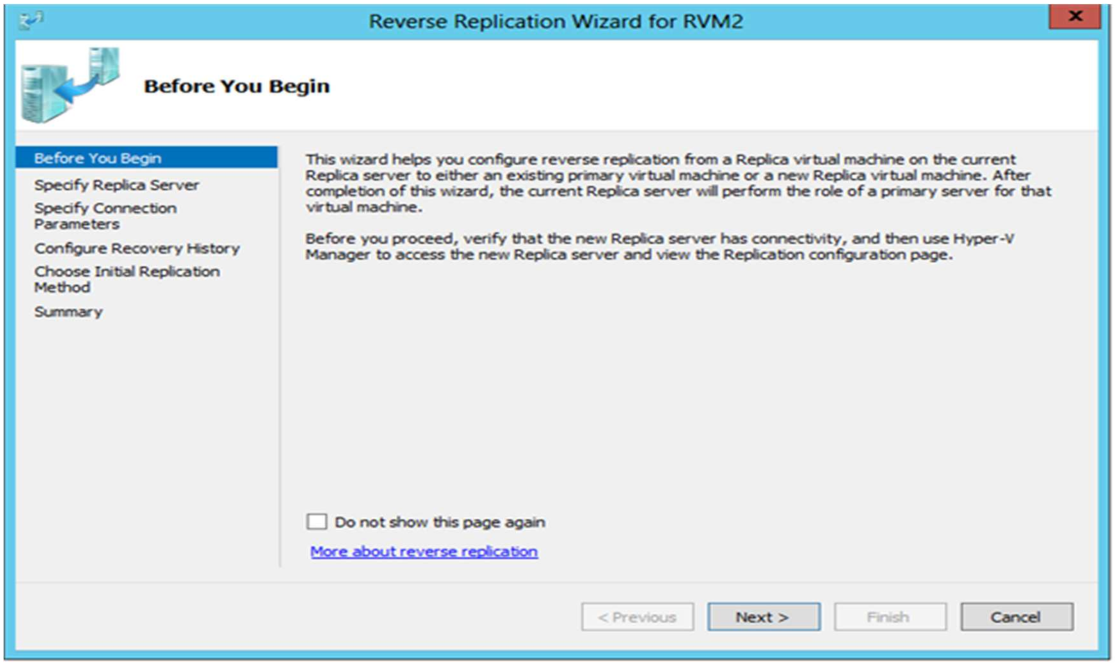
"**Reverse Replication**" is a manual failback option provided with Replica Virtual Machine! You may wonder; *Why not have automatic failback*? Well, Hyper-V Replica is a disaster recovery solution and not a high availability solution. Microsoft may develop further the Hyper-V Replica components in the next few years to provide the failback functionality but it is not implemented in the current version of Hyper-V Replica.

In case of any failure with the Primary Server or if the planned failover is initiated, the virtualized workload is brought online at the Replica Server to continue the service. At this point, Hyper-V Primary Server does not monitor the changes on the Primary Virtual Machine. In other words, there is no replication taking place from Primary Server to Replica Server. The virtual machine continues to run on Replica Server. If you need to change the replication order (e.g. from Replica Server to Primary Server), you must use "Reverse Replicate" action on Right Click context menu of the Replica virtual Machine.

Note: Primary Server must be configured to accept the incoming replication and Hyper-V Replica firewall rules must also be enabled.

When starting reverse replication, choose the Do not copy the initial replication option on the Initial Replication page. The VM on the original primary VM can be used, and a block-by-block comparison will be performed to synchronize between the replica VM and the original primary VM. Only the delta content needs to be sent over the network.

"**Reverse Replication**" action just allows you to failback the load from Replica Server to Primary Server. Clicking on the "**Reverse Replication**" brings the following screen as shown in the figure 1.29 below:



Reverse Replication Wizard for RVM2

Before You Begin

This wizard helps you configure reverse replication from a Replica virtual machine on the current Replica server to either an existing primary virtual machine or a new Replica virtual machine. After completion of this wizard, the current Replica server will perform the role of a primary server for that virtual machine.

Before you proceed, verify that the new Replica server has connectivity, and then use Hyper-V Manager to access the new Replica server and view the Replication configuration page.

□ Do not show this page again

Before You Begin
Specify Replica Server
Specify Connection Parameters
Configure Recovery History
Choose Initial Replication Method
Summary
More about reverse replication
< Previous
Next >
Finish
Cancel

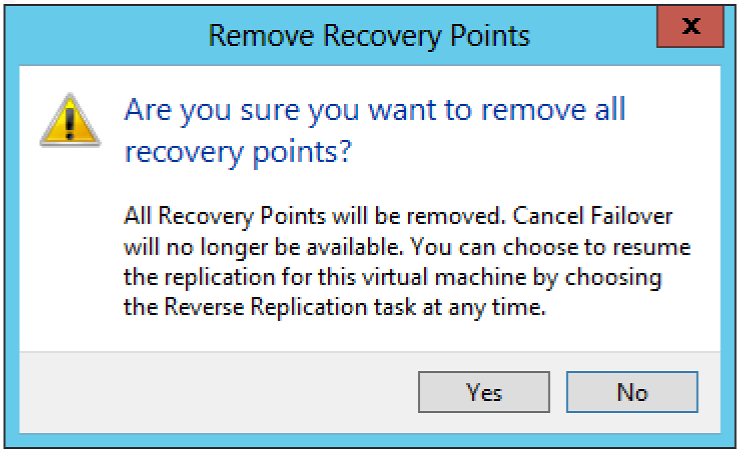
As stated in figure 1.29, this is the same wizard which you used to enable a Virtual Machine for replication earlier in this article. The configuration pages provided in this wizard are similar to those when you enable a Virtual Machine to participate in the Hyper-V Replica.

The wizard checks to see if the destination server is able to accept the incoming replication traffic generated from this server and this is where it is required for Primary Server to become a Replica Server.

Note: "Planned Failover" and "Reverse Replication" actions can be performed only if Primary Server is allowed to accept incoming replication requests.

You might want to remove all the recovery points associated with a Replica Virtual Machine. You would want to do that if recovery points are corrupted or you just don't need them. To remove all the recovery points, select "**Remove Recovery Points**" action, as shown on the Right Click menu in Figure 1.29 above.

When you click this action, a warning message will be displayed which requires your confirmation before recovery points associated with this Replica Virtual Machine can be removed from the local disk.



Are you sure you want to remove all recovery points?

All Recovery Points will be removed. Cancel Failover will no longer be available. You can choose to resume the replication for this virtual machine by choosing the Reverse Replication task at any time.

Figure 1.30 – "Remove recovery points" confirmation message

Recovery Points for a Replica Virtual Machine can be removed only if the Virtual Machine has been failed over using “Failover” action. The “Remove Recovery Points” action does not provide you a list of available recovery points rather it deletes all the recovery points associated with this Virtual Machine at a time.

As suggested in the Figure above, "Cancel Failover" option will no longer be available on the Right Click Context Menu if all recovery points are deleted.

Tip: Remove recovery points for a Virtual Machine if you want to start over or if you think there is no use of all the recovery points of a Virtual Machine.

To summarize, the following actions are available on the Right Click Context Menu of Primary and Replica Virtual Machines:

Characteristics

The table below calls out the characteristics of the 3 failovers

Characteristics

The table below calls out the characteristics of the 3 failovers

	Test Failover	**Planned Failover**	**Unplanned Failover**
Operation initiated on	Replica VM	Initiated on the primary VM and completed on the replica VM	Replica VM

Is a duplicate VM created during the operation?	Yes	No	No
How long is the operation run?	Short	Depends on maintenance window or regulation requirement	Depends on when the primary is brought back up
Recommended frequency	Once a month	Once in 6 months	Never (ok, fine – whenever you have a disaster😊)
What happens to the replication of the primary VM during the duration of this operation	Continues	Continues. In this operation, a role-reversal happens, the primary VM becomes the replica VM and replication continues back to the primary site (that initiated the operation).	Stopped
Is there data loss?	None	None	There can be data loss
Is there down time?	None	Planned downtime	Unplanned downtime
When to use	• Run minimal tests to validate if your replication is on track • Train your personnel on what is to be done in case of a disaster. • Test the recovery plan that you have built to test your preparation when disaster does strike.	• Perform host maintenance on your primary and would like to run from the replica site. • Your primary site is expecting some power outage – you want to move over to the replica site. • There's an impending typhoon – you want to proactively take action to ensure business continuity. • Your compliance requirements mandate that every quarter, you run your workloads from the replica site for a week.	• Your primary site is experiencing unexpected power outage or a natural disaster • Your primary site/VM has had a virus attack and you want to restore your business quickly with minimal data loss by restoring your replica VM

Summary

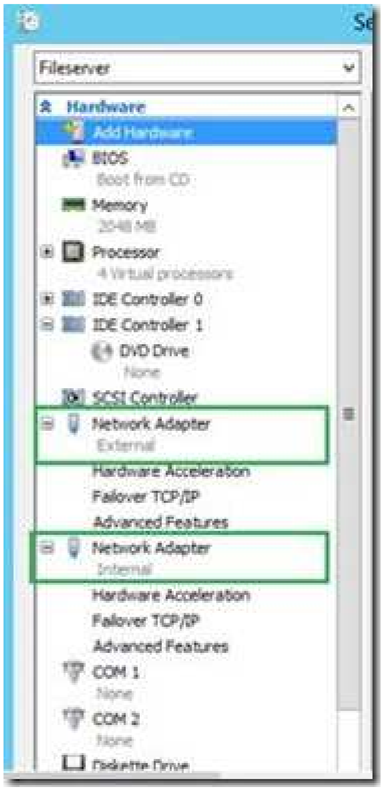
In closing, use **Test Failover** frequently to check the fidelity of your replication and test your recovery plans. Use **Planned Failover** occasionally for either planned maintenance or disaster simulation or compliance reasons. Use **Unplanned Failover** when your primary site is hit by a disaster.

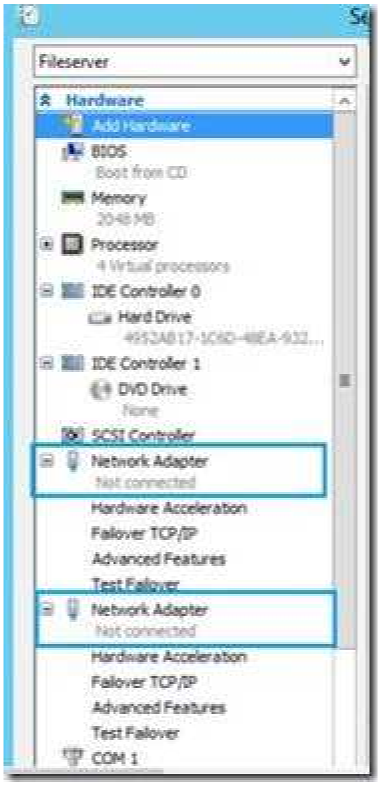
Inject IP address into the VM during failover

Hyper-V Replica reduces the Recovery Time Objective (RTO) providing the ability to configure the static IP address of the Replica VM before it is failed over. This IP address setting is injected into the failed over VM

Network Adapter status on Replica VM

When replication is enabled for a VM, the replica VM's network adapters are disconnected by default.





Primary Replica

Inject IP address from UI

Administrators need to connect the replica VM to the appropriate switch in the Replica server. The IP address which needs to be used in the guest VM during failover can be configured now.

Something might be bothering you about the failover to the disaster-recovery site in a different location: The VM has a TCP/IP configuration that is unlikely to work in a separate location, which will almost certainly be on a different subnet. As part of the Hyper-V Replica functionality, an additional Failover TCP/IP configuration is available on the VM when replication has been enabled. This configuration (found under the Network Adapter configuration of the VM) allows an alternative IPv4 or IPv6 configuration to be specified on the replica VM. The network configuration is injected into the VM during a failover, as shown in Figure

Strategy (Example)

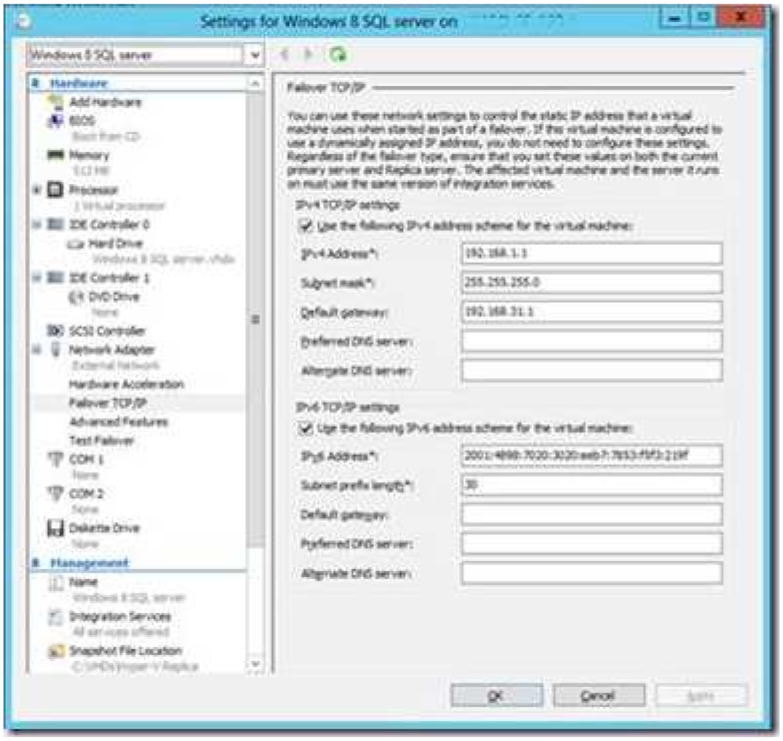
Your replication site is another site, so the network addressing is different from your primary one, and you will need to set new IP addresses for all the replicated VM when failing over.

Hyper-V replica gives you the possibility of injecting a new IP address when failing over. Go to the replica VM, Settings, NIC and you will find it.

So the day you will ask Hyper-V to failover to the replica VM, it will-inject this IP in Windows. When the VM will start, it will update it DNS record in your DNS server, and the A record will be automatically changed.

If you can find a way of keeping the same addressing in your DR site, it will be easiest foir you. But i have never seen this situation because in this case you will be always forced to failover all the datatcenter, you cannot fail-over just one VM, becasue you need to change all the routing into this DR site

Open the Hyper-V manager and open the settings of the replica VM. Click on **Network Adapter** and click on the **Failover TCP/IP** below the setting.



Failover TCP/IP

You can use these network settings to control the static IP address that a virtual machine uses when started as part of a failover. If this virtual machine is configured to use a dynamically assigned IP address, you do not need to configure these settings. Regardless of the failover type, ensure that you set these values on both the current primary server and Replica server. The affected virtual machine and the server it runs on must use the same version of integration services.

IPv4 TCP/IP settings

☑ Use the following IPv4 address scheme for the virtual machine:

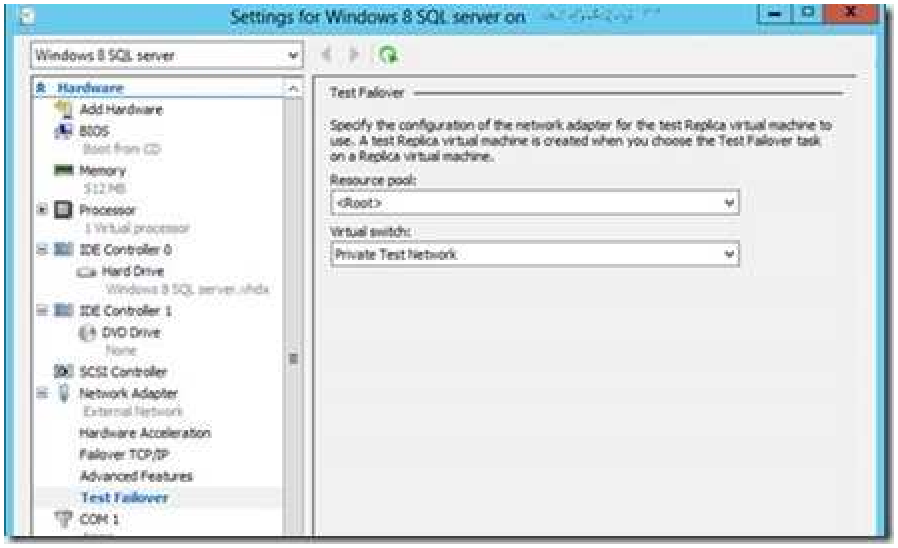
IPv4 Address*: 192.168.1.1
Subnet mask*: 255.255.255.0
Default gateway: 192.168.31.1
Preferred DNS server:
Alternate DNS server:

IPv6 TCP/IP settings

☑ Use the following IPv6 address scheme for the virtual machine:

IPv6 Address*: 2001:4898:7020:3020:aeb7:7853:f5f3:219f
Subnet prefix length*: 30
Default gateway:
Preferred DNS server:
Alternate DNS server:

Enter the IP (v4/v6) details including the address, subnet mask and DNS server(s) information. To verify the settings, invoke the "Test Failover" operation. It is recommended that this operation is run in an isolated network which can be achieved by using the **Test Failover** setting under the Replica VM's network adapter setting. In the picture below, the replica VM is connected to one such private network.



Test Failover

Specify the configuration of the network adapter for the test Replica virtual machine to use. A test Replica virtual machine is created when you choose the Test Failover task on a Replica virtual machine.

Resource pool: <Root>

Virtual switch: Private Test Network

By default, the network settings under **Test Failover** is not-connected to any switch. Once the above step is performed, when a test failover is invoked, the newly created VM will be connected to "Private Test Network" switch and the IP address provided under "**Failover TCP/IP**" will be injected into the test VM.

How does Guest IP injection work?

The Replica VM is blocked from starting unless the Failover workflow is initiated. The **Failover TCP/IP** settings which are provided are stored in the VM configuration file till then. When the replica VM is failed over, the KVP (Key Value Pair) Exchange integration component running within guest operating system picks up the staged settings and applies it inside the VM. Any failure to apply the settings is logged on root partition event viewer.

Few points to note:

- This feature is available only when the latest integration services installed for the VM.
- At the time of writing this article, this feature is available for Windows Guest OS'es only.
- This feature is supported only for synthetic network adapters.
- This feature cannot be used to inject IP addresses into non-replicated VMs

Inject IP address using PowerShell

The above functionality can be achieved using PowerShell as well. To set a IPv4 **Failover TCP/IP** settings on the replica VM, issue the following cmdlet:

```
Set-VMNetworkAdapterFailoverConfiguration 'Windows 8 SQL Server'
-IPv4Address 192.168.1.1
-IPv4SubnetMask 255.255.255.0
-IPv4DefaultGateway 192.168.31.1
```

To connect the VM to a different switch which would be used for Test Failover, use the following cmdlet:

```
Set-VMNetworkAdapter 'Windows 8 SQL Server'
-TestReplicaSwitchName 'Private Test Network'
```

where 'Private Test Network' is the name of a virtual switch which provides an isolated network environment.

Inject IP address using WMI

A frequent question which we get is around providing the ability to inject **multiple** IP addresses on the same network adapter.

Though this cannot be achieved using UI or PowerShell, the same can be achieved in WMI. This address set is represented by WMI class Msvm_FailoverNetworkAdapterSettingData. A WMI snippet is given below which allows you to achieve the above functionality:

```
#Get vm object
```

```
$vm = Get-WmiObject -Namespace 'root\virtualization\v2' -Class 'Msvm_ComputerSystem' | Where-Object { $_.ElementName -eq 'Windows 8 SQL Server' }


# Get active settings
$vmSettings = $vm.GetRelated('Msvm_VirtualSystemSettingData') | Where-Object { $_.VirtualSystemType -eq 'Microsoft:Hyper-V:System:Realized' }


# Get all network adapters
$nwAdapters = $vmSettings.GetRelated('Msvm_SyntheticEthernetPortSettingData')


# Find associated failover network settings
$failoverNetworkSettings = @()

foreach($nwAdapter in $nwAdapters)
{
    $failoverNetworkSettings = $failoverNetworkSettings + $nwAdapters.GetRelated("Msvm_FailoverNetworkAdapterSettingData")
}


#Set two IPv4 addresses for first network adapter
$settingForFirstAdapter = $failoverNetworkSettings[0]


#Each field is an array so multiple inputs can be given
$settingForFirstAdapter.IPAddresses = {'192.168.1.1', '192.168.1.2'}
$settingForFirstAdapter.Subnets = {'255.255.255.0', '255.255.255.0'}
$settingForFirstAdapter.DefaultGateways = {'192.168.31.1'}


# Address family values for settings IPv4 , IPv6 Or Boths
# For IPv4:   ProtocolIFType = 4096;
# For IPv6:   ProtocolIFType = 4097;
# For IPv4/V6:ProtocolIFType = 4098;

$settingForFirstAdapter.ProtocolIFType = 4096


#Set the failover IP address using replication service object
$replicationService = $vm.GetRelated('Msvm_ReplicationService')

$replicationService.SetFailoverNetworkAdapterSettings($vm.Path, {$settingForFirstAdapter.GetText(1)})

```

Ref:

https://blogs.technet.microsoft.com/virtualization/2012/07/31/types-of-failover-operations-in-hyper-v-replicapart-ii-planned-failover/

https://blogs.technet.microsoft.com/virtualization/2012/05/28/inject-ip-address-into-the-vm-during-failover/

https://www.simple-talk.com/content/article.aspx?article=1870

http://www.aidanfinn.com/?p=14537 – IP Assignment Strategies in Hyper-v Replica

https://www.simple-talk.com/sysadmin/virtualization/a-practical-guide-to-microsoft-hyper-v-replica-part-i/

https://www.simple-talk.com/sysadmin/virtualization/a-practical-guide-to-microsoft-hyper-v-replica-part-ii/