Hyper-V Replica in depth

Mike Resseler
Veeam Product Strategy Specialist, MVP, Microsoft Certified IT Professional, MCSA, MCTS, MCP
Contents

Introduction ................................................................. 4
Why replication? ............................................................ 4
Hyper-V Replica ............................................................ 4
Overview ................................................................. 5
  Asynchronous replication ............................................. 5
  Requirements ............................................................ 6
Installation and configuration ........................................ 6
  Plan before you begin! ................................................. 7
  Firewall settings ....................................................... 9
  Certificate ............................................................... 10
  Enabling a host as replica server ................................. 11
  Enabling the Hyper-V Replica Broker role ...................... 14
  Replicating your first virtual machine ........................... 17
  Extended replication ................................................. 23
Failover types ............................................................. 25
  Planned failover ........................................................ 25
  Test failover ............................................................ 28
  Failover ................................................................. 31
Network configuration on the replica ................................ 33
Initial replication method .............................................. 35
PowerShell cmdlets .................................................... 38
Resource usage ........................................................ 39
Monitoring ............................................................... 46
  Best practice analyzer .............................................. 46
  Monitoring through the GUI ....................................... 49
Backup interoperability ................................................ 52
Some additional supported features. ................................. 53
  Online resizing of disks ........................................... 53
  Upgrading to R2 .................................................. 53
  Replicating VDI VMs and deduplication ......................... 53
  Saving disk space (general). .................................... 54

Best practices ............................................................ 54
  Separate disk for the paging file ............................... 54
  Think about your performance .................................. 54
  Use network throttling ............................................ 55

Azure Site Recovery .................................................... 55

Conclusion ................................................................. 56

Appendix A: Resources used ........................................ 56

About the author ......................................................... 57

About Veeam Software ............................................... 57
Introduction

When Windows Server 2012 hit the market in 2012 a new feature called Hyper-V Replica hit the shelf. In 2013, Windows Server 2012 R2 released and improved that feature. This white paper looks in depth at Hyper-V Replica—what it is, how it works and what capabilities it offers—and it includes specific use cases. By the end of this white paper, you should be able to conclude whether this feature suits your environment and know all the steps to successfully implement it.

Why replication?

Let’s get one thing straight from the start: Replication isn’t a backup replacement! I have had this discussion over and over again—you simply can’t replace your backup strategy with replication. The two complement each other in your entire DR strategy and can give you very good recovery time objectives (RTO) but replication alone won’t protect you against data corruption, upgrades gone awry, patch issues, user mistakes and much more.

The whole idea behind replication is that you have a standby copy of your production server ready to start when your production server is down and you need to be online very fast. With an updated standby replica, you can resume services very fast with a minimum of data loss. Replication brings additional possibilities to your DR plan whenever something happens to your virtual machines (VMs). Again, replication is certainly not a replacement for backup or archiving, but it will enhance your DR plan by providing additional opportunities for failover and recovery.

Hyper-V Replica

Hyper-V Replica is a feature that is built-in with Windows Server 2012 and 2012 R2 and can be used to replicate VMs from one host to another. Since there are some nice improvements in Windows Server 2012 R2, the focus of this white paper will be on Windows Server 2012 R2 and not on Windows Server 2012. There are many other third-party solutions that also offer replication, but this white paper is not a comparison of replication tools. Hyper-V Replica is a feature of Windows Server 2012 R2 and therefore comes at no additional cost. Replication on the SAN level can be quite expensive and brings additional challenges with underlying hardware. Software replication also exists and can result in additional costs. (Veeam® Backup & Replication™ is a 2-in-1: backup and replication™ solution and offers additional features and support for more than Hyper-V.) Again, this paper is not a comparison of replication tools but is intended to guide you through the Windows Server 2012 R2 solution starting from what it is, how it works, preparations, installation, configuration, and some tips and tricks.
Last but not least, it is important to understand when to use Hyper-V Replica. High availability (HA) is achieved when using features such as clustering or HA features that are built in to the service or application itself. Replica is used for failover when the application goes down (or worse, when the entire site goes down). When something goes wrong, the administrator has the ability to do a failover of the production environment to the replica environment and thereby minimize downtime. There will still be downtime and probably some data loss, but with replication it can be minimized.

**Overview**

Hyper-V Replica provides asynchronous replication of Hyper-V VMs between two or more Hyper-V hosts. It is fully storage agnostic. This means that it doesn’t need any shared storage or any specific storage hardware. It is also application agnostic. Every workload that can be virtualized in Hyper-V can be replicated.

The replication engine works over a standard IP-based network and can be encrypted during transmission.

Hyper-V Replica is very simple to configure and does not require either shared storage or any particular storage hardware. Any server workload that can be virtualized can be replicated as well.

Hyper-V Replica can work with standalone servers, failover clusters or a mixture of both. These servers can be in a local area network (LAN) or in a wide area network (WAN). Replica servers don’t need to be in the same domain; in fact, they don’t need to be domain joined at all.

**Asynchronous replication**

Hyper-V Replica is an asynchronous replication. That means that whatever you do on the production server won’t be applied to the replica immediately. Depending on the interval of synchronization (discussed later) there will always be a difference between the production VM and the replica.

Asynchronous replication does have its advantages, including less load on the production machine and network.

When deciding between replication solutions, you must consider what your requirements are. If you can’t afford to lose a single second of data, then asynchronous replication won’t work for you, and you are looking into very expensive synchronous mechanisms. If you can afford to lose a bit of data, then this solution might be exactly what you need.
Requirements

There are no real hardware or software requirements to use Hyper-V Replica in your environment. From a hardware point of view, you will need two physical machines running Windows Server 2012 or Windows Server 2012 R2 that have the Hyper-V role enabled. However, when working with Hyper-V Replica you need to think about some soft requirements:

- Hyper-V-capable hardware
- Storage! Make sure you have enough storage both on the primary as replica server
- Network connectivity
- Firewall rules in place
- A certificate, if desired (see the certificate section)
- Hyper-V role enabled on the replica and primary host(s)

Installation and configuration

This section gives an overview of installation and configuration and shows how easy it is to do. If the Hyper-V role is not already installed, add it through PowerShell *(Install-WindowsFeature -Name Hyper-V – IncludeManagementTools)* or through the GUI (Add Roles and Features Wizard (ARFW)) as shown in Figure 1:

![Figure 1: Add Roles and Features Wizard](image)
Plan before you begin!

Since it is easy to get started with Hyper-V Replica, many administrators forget to plan first. This may be fine for a proof of concept (POC) but for a production environment, it is always best to plan first.

The documentation found on http://technet.microsoft.com/en-us/library/jj134153.aspx has some good questions / steps that you can use to prepare yourself for implementation. Without reinventing the wheel, I'll list them here and give some additional information.

**Question 1:** Will both the primary and replica servers be behind the same firewall?

In most use cases, the replica server and primary server will be separated from each other by means of a firewall. In order to make sure that the traffic works you need to make sure that this traffic is allowed. By default you will be using port 80 (Kerberos over HTTP) and 443 (certificate-based over HTTPS). If you are planning to change these ports when enabling replication, you need to act accordingly. You can find more information in the section on firewall settings.

**Question 2:** Will either the primary or replica server be part of a failover cluster?

I have seen many proofs of concept that test replication from host to host without the use of a cluster. Of course this is mostly done for financial reasons and because there is not enough hardware available to do the testing. But in real life you will probably run a Hyper-V cluster. To be able to do effective replication, that means you will need to configure a replica broker as a role. And if you also have a cluster running at the replica side, you will also need to work with a broker. You can find out about enabling the broker in the section on enabling the Hyper-V Replica Broker role.

**Question 3:** Does the replication data sent between the primary and replica servers need to be encrypted?

When you are replicating within the same site, you probably are not going to encrypt the transferred data. But when you are replicating over the WAN, you will want to do this. Hyper-V replication can be encrypted by the means of certificate-based encryption, but in order to do this, you will need an appropriate security certificate. The details of such a certificate can be found in the section about certificates.

**Question 4:** Which VHDs need to be replicated?

When starting replication, you can choose which VHDs from a specific VM need to be replicated. Some VHDs that are not needed can be excluded to save traffic and disk space. A best practice about page files can be found in the best practices section.
Question 5: How many recovery points do you need?

This is a tricky question. When using Hyper-V Replica, you can store the latest received replication data, but you can also store one or more additional recovery points. These recovery points can be created every hour so that you can recover to an earlier point in time. In Windows Server 2012 you are limited to 15 recovery points and in Windows Server 2012 R2 you can go to a maximum of 24. Note that this will require additional resources in your environment. This is covered in the section on resource usage.

Question 6: Do you need applications to remain consistent?

Replication automatically ensures consistency in the state of the operating system and in many cases in the state of the application as well. But not every application will remain consistent. If those applications need to be consistent, make sure you use VSS on a set schedule or use other methods to provide application consistency whenever it is a non-VSS-aware application.

Question 7: Can you do the initial replication method at any time or do you need to plan it, or even ship it first?

If you start replicating a VM, the entire VM needs to be transferred to the other location first. If this is within a local area network and the infrastructure has enough resources, this can be done immediately. However, in many cases, this will not be on the local area network or the Hyper-V hosts won't have sufficient resources available to do the initial replication. In that case you risk bandwidth issues or performance issues. It might be necessary to schedule the initial replication or even ship the VM when it is a large one. You can find out more about this in the section on initial replication method.

Question 8: Will you use extended replication?

Last but not least, starting from Windows Server 2012 R2 you can not only replicate from server 1 to server 2, but you also can extend that replication to server 3. This additional replication can take your DR strategy a step further but requires additional planning and resources. Find more in the section about extended replication.
Firewall settings

First the good news: If you are only working with Windows Firewall, then it is rather easy to implement the right settings. The Windows Firewall with Advanced Security has predefined rules that you can enable easily.

Perform the following procedure on the Hyper-V Replica (host) server:

1. Open Windows Firewall with Advanced Security and click Inbound Rules
2. Right-click Hyper-V Replica HTTPSListener (TCP-In) and click Enable Rule (when using certificate-based authentication)
3. Right-click Hyper-V Replica HTTPListener (TCP-In) and click Enable Rule (when using Kerberos authentication)

It becomes a bit more difficult when you want to do this when working with clusters. Of course you can jump on every Hyper-V host and perform this manually one by one, but you always have PowerShell to make life easier:

```powershell
get-clusternode | ForEach-Object {Invoke-command -computername $_.name -scriptblock {Enable-Netfirewallrule -displayname "Hyper-V Replica HTTPListener (TCP-In)"}}
```

Or when you work with certificate-based authentication:

```powershell
get-clusternode | ForEach-Object {Invoke-command -computername $_.name -scriptblock {Enable-Netfirewallrule -displayname "Hyper-V Replica HTTPSListener (TCP-In)"}}
```

Of course, when you are using standalone hosts, these commands (without the get-clusternode part) can also be used to enable the firewall rules.

If you have another firewall or have changed the ports, then you will need to create these rules yourself on your specific firewall; or when you have changed the ports and use Windows firewall, you can adapt them to match your ports.
Certificate

If you already have a certificate in your environment or you already have a certificate server, then you can easily use an existing certificate, but it must comply with the following rules:

- Not expired or revoked
- Includes both client and server authentication extensions for enhanced key usage (EKU) and an associated private key
- Terminates at a valid root certificate in the Trusted Root Certification Authorities store on the replica server
- If the VM is hosted by a standalone server, the subject common name (CN) must be equal to (or the subject alternative name—DNS name—should contain) the FQDN of the host. If the VM is hosted by a failover cluster, the subject CN must be equal to (or the subject alternative name—DNS name—should contain) the FQDN of the Hyper-V Replica Broker.

If you are uncertain whether or not your certificate will actually work you can use the `certutil` utility to validate it. Run `certutil -store my` on both the primary and replica servers. The output from that command should indicate that the certificate passed the encryption test.

Self-signed certificate

In case you want to work with a self-signed certificate, you can use the following procedure to create one:

1. Make sure that the `Makecert.exe` utility is copied locally to the primary server AND replica server (`Makecert.exe` can be retrieved when you install the Windows 8 Software Development Kit (SDK))

2. Run the following command: `makecert -pe -n "CN=PrimaryTestRootCA" -ss root -sr LocalMachine -sky signature -r "PrimaryTestRootCA.cer"` to create a self-signed root authority certificate (you need to run this from an elevated command prompt)

3. Create a new certificate that is signed by the root certificate that you just created. Run (elevated prompt!) `makecert -pe -n "CN=<FQDN>" -ss my -sr LocalMachine -sky exchange -eku 1.3.6.1.5.5.7.3.1,1.3.6.1.5.5.7.3.2 -in "PrimaryTestRootCA" -is root -ir LocalMachine -sp "Microsoft RSA SChannel Cryptographic Provider" -sy 12 PrimaryTestCert.cer` (where FQDN is the name of the replica server or the replica broker FQDN)
4. On the replica server, create a self-signed root authority certificate:
   `makecert -pe -n "CN=ReplicaTestRootCA" -ss root -sr LocalMachine -sky signature -r "ReplicaTestRootCA.cer"`

5. Again create a new certificate that is signed by the root certificate: `makecert -pe -n "CN=<FQDN>" -ss my -sr LocalMachine -sky exchange -eku 1.3.6.1.5.5.7.3.1,1.3.6.1.5.5.7.3.2 -in "ReplicaTestRootCA" -is root -ir LocalMachine -sp "Microsoft RSA SChannel Cryptographic Provider" -sy 12 ReplicaTest.cer`

6. Copy `ReplicaTestRootCA.cer` from the replica server to the primary server and import it with the following command: `certutil -addstore -f Root "ReplicaTestRootCA.cer"`

7. Copy `PrimaryTestRootCA.cer` from the primary server to the replica server and import it with the following command: `certutil -addstore -f Root "PrimaryTestRootCA.cer"`

8. Make sure that the certificate revocation check is disabled. By default, this is required but self-signed certificates don’t support this. To disable the certificate revocation check, edit the registry on the primary and replica servers by using the command: `reg add "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Virtualization\Replication" /v DisableCertRevocationCheck /d 1 /t REG_DWORD /f`

**Enabling a host as replica server**

You need to configure a Hyper-V host to be able to receive replication. You can do this easily through Hyper-V Manager or through Failover Cluster Manager when the servers are part of a failover cluster.

Right-click on a host and select **Hyper-V Settings**... or select **Hyper-V Settings**... from the **Actions** pane (see Figure 2):
Then you can go to **Replication Configuration** and start configuring your replica server.
Figure 3 shows three blocks that can be explained as follows:

**Block 1: Enable**

It’s rather simple, but you need to select this checkbox to enable this server as a replica server.

**Block 2: Authentication and ports**

Here you will decide whether you are going to use Kerberos authentication or certificate-based authentication. Kerberos is the easiest to use, but it requires that both servers are in the same domain (or mutually trusted domains), and it doesn’t provide encryption. Certificate-based authentication requires a bit more work (i.e., you need certificates), but it will encrypt your traffic and works between servers that do not have a mutual trust.

**Block 3: Authorization and storage**

You can choose to allow authentication from any server or select specific servers. The first option is easy to use, but I advise you to use that only in a POC or test / dev environment. For the second option, you can specify servers based on FQDN (wildcards supported) or a trust group. A trust group is a security “tag” that can be used to make sure that replication between all servers with that specific tag is allowed.

![Add Authorization Entry](image)

*Figure 4: Authorization entries*

Don’t forget to change the default location where the replicated data will be stored (see Figure 4).
That’s it. This host is ready to accept replication. Note that I advise that you do the same thing on the primary server as well. When you are going to use failover, and the replica direction needs to be reversed, your primary server becomes replica and vice versa. But if this isn’t configured initially, that won’t work.

**Enabling the Hyper-V Replica Broker role**

When you are running a Hyper-V cluster, you need to enable the Hyper-V Replica Broker role. You can’t work with the Hyper-V Manager any more. The procedure for this is very easy. Open the Failover Cluster Manager, select the cluster and click **Configure Role** in the actions pane. The **High Availability Wizard** appears (see Figure 5):

![Figure 5: High Availability Wizard](image)

Press **Next** on the **Before You Begin** page.

![Figure 6: Choose the Hyper-V Replica Broker role](image)
On the **Select Role** page (shown in Figure 6), select the **Hyper-V Replica Broker** role and press **Next**.

![Figure 7: Client Access Point configuration](image)

As shown in Figure 7, give your role a NETBIOS name and IP address (referred to as the “client access point” in the Technet documentation) and press **Next**.

![Figure 8: Confirmation window](image)

Confirm your settings (shown in Figure 8) and press **Next**.
Figure 9: Configuration done

Figure 9 shows that high availability was successfully configured for the role. You are ready.

As always, you can do the same through PowerShell:

```powershell
$BrokerName = "HVR-Broker"
Add-ClusterServerRole -Name $BrokerName –StaticAddress 192.168.1.251
Add-ClusterResource -Name "Virtual Machine Replication Broker" -Type "Virtual Machine Replication Broker" -Group $BrokerName
Add-ClusterResourceDependency "Virtual Machine Replication Broker" $BrokerName
Start-ClusterGroup $BrokerName
```

You can configure the replication the same way that you did through Hyper-V, but the difference is that now it is configured on the cluster level, which means you don’t have to do this separately on each server. To do this, right-click on the Hyper-V Replica Broker role in the Cluster Manager and choose **Replication Settings**.
A few notes on the use of a broker. After installing the broker role, you still can use the Hyper-V console to replicate the VMs, but if you are making your VMs highly available (HA) in the cluster, you want to enable the replication through the Failover Cluster Manager.

And here is a last, very important note: Everything else that you are going to read about from here on will need to be done through the Failover Cluster Manager!

Also, configuration, failovers and other operations will be done from the Failover Cluster Manager at that moment! So this applies to everything you see in the next sections.

**Replicating your first virtual machine**

Now that everything is prepared, you can finally start replicating your first virtual machine. Remember that you asked yourselves a couple of questions first, and now you need to make sure that you have the answers to those questions to do the configuration correctly.

In Hyper-V Manager, on the primary Hyper-V host, right-click on the chosen VM and select **enable replication** as shown in Figure 11.
A new wizard will appear. On the **Specify Replica Server** page, type in the name of the replica server (or browse) and press **Next** (see Figure 12):

**Figure 11: Enable Replication on a VM**

**Figure 12: Type the name of the replica server (or browse)**
On the **Specify Connection Parameters** page you will be able to choose the authentication type you want to use for the replication (see Figure 13). However, if the primary server and the replica server are able to contact and *talk* to each other (meaning that they are in the same domain with network connection), they will figure out the type of authentication based on their configuration. However, when this is not possible, you will need to choose manually what type of authentication you are using and that must match on both sides. Note also the checkbox for **compress the data that is transmitted over the network** (shown in Figure 13). Selecting this checkbox will decrease the load on the network, but it will increase the resources needed on the Hyper-V hosts. Depending on your infrastructure, you will need to make the right decision here.

![Figure 13: Connection parameters](image)

Now you need to choose the VHD(x) disks that you want to replicate (see Figure 14). Later I will describe a best practice for this, but for now, just remember that you can select the disks you need for replication. Note that if you deselect a disk, the data won’t be available at the other side, and it won’t even be created on the replica VM. That means that you will need to create a “dummy” if you need that disk to boot the replica VM!
On the next page you need to choose the replication frequency. In Windows Server 2012 there was no option, and it replicated every five minutes (if possible). In Windows Server 2012 R2, you have the option of 30 seconds, 5 minutes or 15 minutes. While it would be nice to choose 30 seconds for every replication, thus having a lower possibility of losing data, you need to take into consideration your resources and bandwidth. For example, when your hypervisors are heavily loaded, then you will probably not want to replicate every 30 seconds. In this example, I will choose five minutes (see Figure 15).
Now you will get the option to choose whether you want to create additional recovery points (Figure 16).

As already explained, this is where you configure your additional recovery points. Note that you can do this WITHOUT VSS snapshots, but the consequence is that the data inside won’t be quiesced.
In this example I want an additional recovery point every hour AND I will do a VSS snapshot every four hours (see Figure 17).

![Figure 18: Initial Replication Method](image)

On the **Choose Initial Replication Method** page you will choose how and when to perform the initial replication (see Figure 18). More information on this step can be found in the section on **initial replication method**.

![Figure 19: Summary screen](image)
Finally you are at the summary screen (shown in Figure 19) and you can finish.

![Enable Replication for Core](image)

Figure 20: Replication is successful

You will get a notification that replication is enabled successfully (Figure 20) and that you can change some additional settings in the network, but we will come back to that later.

Congratulations! You have finished your first replication. Note that the initial replica can take a while, and from then on, it will replicate the changes at the chosen interval if everything is OK.

**Extended replication**

Starting from Windows Server 2012 R2, you will be able to do extended replication. This means that you can replicate from the primary server to the replica server, and then from the replica server once more to a third server. There are quite a few reasons why you would want to do this. For example, inside your own company you might want to replicate from a host (or cluster) to another host (or cluster) in the same data center or same building. That would allow you to replicate every 30 seconds, and if something goes wrong you can act very rapidly. However, you also want to have a copy outside of the same location or data center and then replicate that data to a third server. This is a great way of providing that additional protection in your data center.

Extending the replication is very easy to do, but there is one thing you need to know. You cannot replicate every 30 seconds; the only options now are five minutes and 15 minutes. Also, if your first replication is every 15 minutes, your extended replication cannot be five.
The rules for extended replication are:

- The replication range for extended replication must be longer than the first replication
- If you are using application-consistent snapshots, those are forwarded as well, but you cannot change the replication frequency for them
- Finally, whatever VHD(x) you are replicating will be extended further and no changes are possible

There are, however, things you can change:

- The replication frequency (see above)
- The authentication method
- The initial replication method

Enabling extended replication is very easy. Start from the replica VM, right-click and choose **Extend Replication** as shown in Figure 21:

![Figure 21: Extend Replication](image)

Now you can follow the wizard, which is more or less the same as in the section about **replicating your first virtual machine**, except you will have fewer choices to make.

And of course, this can also be done through PowerShell:

```powershell
Enable-VMReplication -VMName <vmname> -ReplicaServerName <extended_server_name> -ReplicaServerPort <Auth_port> -AuthenticationType <Certificate/Kerberos> -ReplicationFrequencySec <300/900>
```
Failover types

When your setup is in place and working, there are a few types of failover that you can perform. Each has different characteristics, as shown in Table 1.

Table 1: Failover Types

<table>
<thead>
<tr>
<th></th>
<th>Planned Failover</th>
<th>Test Failover</th>
<th>Failover</th>
</tr>
</thead>
<tbody>
<tr>
<td>From where started?</td>
<td>From the primary VM, ended on replica VM</td>
<td>From the replica VM</td>
<td>From the replica VM</td>
</tr>
<tr>
<td>Duplicate VM?</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Time frame?</td>
<td>Depends on the use-case</td>
<td>Short</td>
<td>Depends on how long it takes to fix the issue</td>
</tr>
<tr>
<td>Recommended frequency</td>
<td>Every six months</td>
<td>Once a month</td>
<td>Never (only in emergencies)</td>
</tr>
<tr>
<td>Replication</td>
<td>Continues, in reversed mode</td>
<td>Continues</td>
<td>Stopped</td>
</tr>
<tr>
<td>Data Loss</td>
<td>No</td>
<td>No</td>
<td>Possible</td>
</tr>
<tr>
<td>Downtime</td>
<td>Yes (planned)</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Planned failover

A planned failover is something that you perform when you know in advance you are going to need a failover. This could be a hurricane coming your way and you want to be proactive, or maybe you just need to do some maintenance on the primary server. The nice thing about this failover type is that you won’t have noticeable downtime or data loss, so this procedure can also be used as a test of your disaster recovery procedures (note that this can be done with Test Failover as well).

Follow this procedure:

Shut down the production VM. Planned failover will give you an error notification when it is still running. You do need to do this manually since the wizard won’t do it for you.

Right-click on the production VM that runs on the primary server and choose Planned Failover…as shown in Figure 22.
Figure 22: Starting your planned failover

By default, the checkbox for **Reverse the replication direction after failover** is not selected. It is advised to select this checkbox as shown in Figure 23, so that the production VM will receive the changes back with no data loss when the planned failover procedure is finished. One note of caution here: At that moment, the primary server also becomes a replica server, so the host must be configured to allow replication (see the section on enabling a host as replica server).

Figure 23: Failover

You will see the planned failover happening (Figure 24) and if all goes well, you will finally get a message that the failover has succeeded. At that time, the replica VM is running rather than the production VM.
Figure 24: Planned Failover

A few words of caution are needed here. If you have configured alternative network settings, then they are working now. Imagine that your replica VMs are in another data center and you are not using stretched VLANs—at that moment your network config probably is different. That also means that you need to redirect your users who connect to that server accordingly. DNS updates are not done automatically by the planned failover wizard, so it might be that additional manual work is needed from your side.

And of course, you can do this entirely by using PowerShell.

On the primary server (“core” is the name of my VM):

```
$PrimaryVM1 = "Core"
Stop-VM $PrimaryVM1
Start-VMFailover -VMName $PrimaryVM1 –prepare
```

Then jump to the replica server:

```
$ReplicaVM1 = "Core"
Start-VMFailover -VMName $ReplicaVM1
Start-VM $ReplicaVM1
```

And finally, when you also want to reverse the replication:

```
Set-VMReplication -reverse -VMName $ReplicaVM1
```

The replica VM up and running, there is no data loss and we have reversed the direction. Now let’s assume that whatever happened or needed to happen at the primary server is done, and we want to start the procedure again. To be able to do that, we are going to perform another planned failover, but this time it will start from the replica server. So we will shut down the VM again (downtime!) and follow the same procedure as above.

---

1. When this is done on a failover cluster, these PowerShell commands should be run against the server that is the current owner of that VM.
The failover will happen again, and you will reverse the replication back to your starting point (see Figure 25).

![Planned Failover](image)

**Figure 25: Going back to the original situation**

If you are using extended replication, note that you will have more options here. In this example, you can fail over from A to B and reverse the replication to A. But in an A,B,C scenario, you might want to fail over to B and use C as the new replica or fail over completely to C—all new scenarios that you can use.

### Test failover

As with backups, you should test from time to time to see whether your replicated VMs are actually working in case you need them in a disaster.

By using test failover, you will create a temporary VM on the replica server. You can play with this VM, do some tests and more without interrupting the replication between the production VM and the replica VM.

There are more possibilities (use cases) where you can use this failover type. Let’s say you have a certain application (e.g., an email service) that you would like to upgrade. And for the sake of simplicity, let’s say that the email service exists out of a domain controller (needed for authentication, service accounts, etc.) and the actual email server.

On the replication side, you can create the test failover VMs, put them in a quarantined network (in order not to interfere with the running production servers, as explained in the section on network configuration on the replica) and then use those two test failover VMs to perform some upgrade tests.

This situation is ideal for performing tests on a real-life environment (after all, the VMs are replicas of your production environment) and it’s perfect for building a change management plan.
In Figure 26 you can see how to build such a setup.

The procedure of doing a test failover is very simple, but it requires a final step that can't be forgotten after all the testing!

On the replica server, right-click on the replica VM and select Replication... - Test Failover...(see Figure 27):

If you are using multiple recovery points, select the recovery point that you want to use and press Test Failover (shown in Figure 28):
In your list of VMs, you now see a VM with the same name and “– Test” as addition (see Figure 29). Note that you will need to start the VM manually.

After all of your testing and other tasks with those test VMs, you need to clean up. Right-click on the replica VM and select Replication – Stop Test Failover to stop the test, and this will clean up everything (see Figure 30).
A last note: When you are using extended replication, you can do the test failover on the extended replica without hurting the chain. This will provide additional possibilities in testing your environment.

Failover

Failover is something that you don’t want to perform. You need to do this when there is a problem, and when it is necessary, you need to be prepared. An important note: Do not wait for a problem to occur before you familiarize yourself with this procedure. Try it out in a test / dev / POC environment and make sure the steps are well documented. I have seen too many situations where IT had everything in place, but the moment they needed to act (under pressure), things went wrong. Don’t get me wrong—if you need to do a failover, there will be pressure, but if you have practiced, you should be able to do it.

Let’s say you have a situation on your hands and your server is down at your production environment. It looks like it is a serious problem and you decide to go with the replica.

As shown in Figure 31, go to the replica server, right-click on the replica VM and choose replication > failover...

Figure 31: Start the failover from the replica server
You will get a clear warning that you cannot do this unless the primary VM is down and that there is a possibility that the replica does not have the latest data, so data loss might occur. When you are certain that you need to perform this failover, select the recovery point (if you have multiple) and press the **Fail Over** button (shown in Figure 32).

![Failover](image)

**Figure 32: Choose the recovery point**

You will see that the replica VM will start running (Figure 33).

![Virtual Machines](image)

**Figure 33: Failover has succeeded**

You need to use the PowerShell cmdlet **Complete-VMFailover** (shown in Figure 34) to make sure that the failover is completed.

![PowerShell](image)

**Figure 34: Complete the failover**
As you can see in the **Replication Health** window, the failover is now complete (Figure 35). The moment you get the primary server back online (or, in a worst case scenario, you have built a new one) you can reverse the replication to make sure that everything will be back to normal after a while.

**Network configuration on the replica**

If you are using replication, there is always the possibility that the network settings on the replica VM need to be changed. All the original settings are replicated as well, but imagine that your replicated VM is located in another data center with a different subnet; then you would need to adjust the network configuration before you start that replica.

Luckily this is anticipated in Hyper-V Replica.

Open the settings of your VM and go to **Network Adapter > Failover TCP/IP**
Now you can modify IPv4 and/or IPv6 settings of the VM when a failover occurs. Note that this is only necessary when you are not using a DHCP server (see Figure 36).

Whenever you perform a test failover, you want to make sure that the VM is not connected to the network when you start the VM. By default, the test VM will not be connected to the virtual switch. But you can easily change this by opening those settings again and selecting **network adapter > test failover** and connecting it to an appropriate network. Of course, this is a separated network, so you won’t have the issue that both servers are online and could reach each other (see Figure 37).
Initial replication method

When you start replication, the entire VM needs to be replicated to the replica server. Depending on the size of that VM and the bandwidth you have available, this can take a long time. But probably more important is the load that this can generate on your production environment. Your bandwidth can get consumed and additional compute resources will be used during the process of initial replication. This is something you may want to avoid during production hours, so it’s a good idea to define the time for initial replication.

Luckily the system anticipates this. As soon as you enable replication for a VM, the wizard asks you when you want to do the replication.
As you can see in Figure 38, you can choose from three methods:

- Send initial copy over the network
  - Start replication immediately
  - Start replication on (select date and time)
- Send initial copy using external media
- Use an existing virtual machine on the replica server as the initial copy

The first method is used when you have the bandwidth available. You then have the option to send it immediately or choose a date and time to send the initial replication. This is certainly handy when you have the bandwidth but want to make sure the resources are only used after business hours.

The second method (send initial copy using external media) is very handy when your bandwidth is limited or your VM is large in size. We will go into detail soon as this might be a bit confusing the first time that you use it.

The last method uses an existing virtual machine on the replica server. This VM can be from a recovered backup, for example, or from a previous replication that you had disabled and want to re-enable (and the bits are still on the replica server).

Now let’s say that we want to use the second method because we need to replicate over a WAN and our bandwidth is limited. In this case the primary server will create a (dummy) VM on the replica server and save the VM to a specified folder (Figure 39). When you look into that folder you will see a folder called `<nameVM>_GUID`. 
You can then move or ship that external media to the location of the replica server and attach it to that server. Go to Hyper-V Manager, right-click the (dummy) VM and choose **Replication > Import Initial Replica…** (see Figure 40):

Now you need to provide the location of the files and press the **complete initial replication** button (see Figure 41). The rest will be done automatically for you.

Please note that replication will start after that. Depending on the interval between the import and the export (based on how long it took to ship that data), it can take some time before the replica VM is up to date.
PowerShell cmdlets

Although I am a big fan of PowerShell, it is impossible to memorize all of the cmdlets unless they are used on a daily basis. Starting with Windows Server 2012 there are more than 2,300 cmdlets and in R2 that number increased, so it’s necessary to check get-help regularly. But that works great if you know all of the cmdlets and can figure out the syntax. And because all Hyper-V Replica cmdlets are in the same module as all of the Hyper-V cmdlets, you can’t use that to discover the cmdlets.

Luckily there are only two commands you need to memorize to get all of the cmdlets that are available for Hyper-V Replica:

```powershell
Get-command –Module Hyper-V | where {$_ .Name –like "*replication*"}
Get-command –Module Hyper-V | where {$_ .Name – like "*failover*"}
```

The first command shows you all the general cmdlets for Hyper-V Replica (Figure 42):

![Figure 42: PowerShell cmdlets](image)

The second command shows you all the cmdlets for failover tasks (Figure 43):

![Figure 43: PowerShell cmdlets](image)
When testing Hyper-V Replica, I always advise people to start using the GUI. However, I also recommend that you use Hyper-V on a core installation in production, so it is certainly worth spending some time using the PowerShell cmdlets. Anything you can do in the GUI can also be done through PowerShell. I also recommend that you use these PowerShell cmdlets to get more health information and display it, for example, on a SharePoint portal or a simple webpage. Last but not least, whenever you need to do an actual failover, keep in mind that this is a stressful period. This means something serious is wrong, and you need to act upon it. You probably know exactly what to do, but under stress you might skip a step or use the wrong option. Building your failover plans by using PowerShell scripts (even if they are one-liners) can help you to avoid making mistakes and it can speed up the recovery.

**Resource usage**

It isn’t easy to calculate the amount of resources needed to run Hyper-V Replica. After all, it will require some additional CPU, memory, network bandwidth, storage and IOPS. Microsoft has released a tool you can use to calculate resource needs. Of course, this calculation and will never be “precise,” but it should at least provide a good view of what can happen in your environment when you enable Hyper-V Replica, and it shows whether your Hyper-V hosts, storage and network can handle that additional resource usage.


The documentation explains very well what the prerequisites are, so I will simply give you an overview of how it works.

When you run the tool, you will need to decide the duration for its calculations. In this example I have chosen the default of 30 minutes, but in a real-life environment you might want to increase that number (see Figure 44). It is also highly recommended that you run the tool during production hours to get more representative data.
Select your primary host(s) or cluster (Figure 45):
Choose the replica host or the replica cluster FQDN (Figure 46):

![Figure 46: Selecting replica host or cluster](image)

If you have or want extended replica, you can calculate this as well, as shown Figure 47. I am not using extended replica (or planning it), so I won’t do those calculations.

![Figure 47: Choose extended replica site details](image)

Now you choose which VMs you are planning to replicate and which drives (Figure 48).
Finally, choose your network information. You can also select a certificate if you want to use certificate-based authentication. And if you want to use extended replication, you can add the estimated bandwidth (and, optionally, the certificate) here as well (Figure 49).

Review the summary and return when you need to make adjustments, and then choose Next (Figure 50).
The system will now start running during the time frame that you selected and it will make the calculations (Figure 51).

When the calculations are finished, you can press view report to see the result (Figure 52).
Next I will look at a few different sections of the report.

As you can see in Figure 53, the report is divided into different parts. The general information is mostly your input and this is followed by information about the processor, memory, IOPS, storage and network (Figures 54-56).

**Processor**

Description: Details of the CPU impact on the Primary, Replica, and Extended Replica servers/cluster. For details on how to interpret the values shown, refer to subsection 4.1 Processor, under the section "Understanding the report" of the documentation.

<table>
<thead>
<tr>
<th>Host name</th>
<th>Host type</th>
<th>CPU Utilization before enabling Hyper-V Replica</th>
<th>CPU Utilization after enabling Hyper-V Replica</th>
</tr>
</thead>
<tbody>
<tr>
<td>HYPERV01</td>
<td>PRIMARY</td>
<td>0.94%</td>
<td>3.98%</td>
</tr>
<tr>
<td>HYPERV02</td>
<td>REPLICA</td>
<td>-</td>
<td>3% Additional overhead</td>
</tr>
</tbody>
</table>

**Figure 52: Capacity calculated**

**Figure 53: HTML report**

**Figure 54: Processor details**
You can see what the CPU utilization was before enabling Hyper-V Replica and the amount used after enabling the replica. Don’t pay too much attention to the specific figures shown here as I ran this report on a very small demo environment. This will of course be different in a real production environment. However, you can already see that there will be overhead.

### Memory

Detailing: Details of the Memory impact (in GB) on the Primary, Replica, and Extended Replica servers/cluster. For details on how to interpret the values shown, refer to subsection 2.3 Memory, under the section ‘Understanding the report’ of the documentation.

<table>
<thead>
<tr>
<th>Host name</th>
<th>Role type</th>
<th>Total RAM</th>
<th>Average RAM usage before enabling replication (GB)</th>
<th>Estimated RAM usage after enabling replication (GB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>HYPERV01</td>
<td>PRIMARY</td>
<td>16</td>
<td>7.85</td>
<td>8.33</td>
</tr>
<tr>
<td>HYPERV02</td>
<td>REPLICA</td>
<td>-</td>
<td>-</td>
<td>3% Additional overhead</td>
</tr>
</tbody>
</table>

**Figure 55: Memory details**

The same principle can be applied to the memory. Looking at the data before and after enabling the Hyper-V replica, you can see that there is overhead both on the primary and the replica server. Look at these numbers very carefully and plan accordingly. Many primary servers are already heavily used on the memory side and the additional load can cause performance issues.

### IOPS - Primary

Detailing: Details of the IOPS impact on the Primary storage subsystem. For details on how to interpret the values shown, refer to subsection 2.2 IOPS, under the section ‘Understanding the report’ of the documentation.

<table>
<thead>
<tr>
<th>Host name</th>
<th>Virtual Machines</th>
<th>Peak VM Write IOPS</th>
<th>Average VM Write IOPS after enabling replication</th>
<th>Average VM Write IOPS after enabling replication</th>
<th>IOPS during initial replication</th>
</tr>
</thead>
<tbody>
<tr>
<td>HYPERV01</td>
<td>BR01</td>
<td>8</td>
<td>0.85</td>
<td>1.23</td>
<td>4.75</td>
</tr>
<tr>
<td>HYPERV01</td>
<td>DC01</td>
<td>17.8</td>
<td>0.85</td>
<td>1.1</td>
<td>4.25</td>
</tr>
<tr>
<td>HYPERV01</td>
<td>Test01-restarted</td>
<td>4.9</td>
<td>0.24</td>
<td>0.31</td>
<td>1.19</td>
</tr>
<tr>
<td>TOTAL</td>
<td>-</td>
<td>2.03</td>
<td>1.64</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

**Figure 56: IOPS details**

As always, IOPS is very important in a virtual environment and can make the difference between a fluid, running environment and an environment with performance issues. Because you work with a log file, as explained, that also means you have more IOPS on the primary server. Logically, because you are sending that log file to the replica server and it needs to be applied to the replica VM, you will see that additional IOPS are necessary on that side as well.
Monitoring

There are different ways you can monitor Hyper-V Replica in your environment, and the best ways are monitoring frameworks such as Microsoft System Center Operations Manager. By using the Hyper-V Management Pack and possibly some extensions, you can monitor the solution at all times and be notified proactively or when there is an immediate issue.

I won’t go into detail about how these solutions work as not everybody has those solutions and some may want to use different monitoring solutions.

Instead we will have a look at what is in the box.

Best practice analyzer

The best practice analyzer (BPA) can be run on your Hyper-V hosts through the Service Manager component of System Center. It’s important to note that the rules for Hyper-V Replica are in the same best practice analyzer as the rules for Hyper-V in general. This can make it a bit more confusing if you are only looking for Hyper-V Replica rules, but because each rule has a specific number, you can filter them out easily.

Although I like to run a BPA scan from time to time, I don’t see this as a golden rule. These best practices are based on a huge amount of data that Microsoft has gathered from various companies in different verticals and sizes. I like to run through the rules where I am non-compliant to see if this is something to consider for my environment as well. This is particularly helpful because the rules include additional information and links to technical documentation, which can help you in improving your setup.

Below is the table of the rules for Hyper-V Replica:
Table 2: Hyper-V Replica BPA rules

<table>
<thead>
<tr>
<th>Rule #</th>
<th>Rule explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>37</td>
<td>A replica server must be configured to accept replication requests</td>
</tr>
<tr>
<td>38</td>
<td>Replica servers should be configured to identify specific primary servers authorized to send replication traffic</td>
</tr>
<tr>
<td>39</td>
<td>Compression is recommended for replication traffic</td>
</tr>
<tr>
<td>40</td>
<td>Configure guest operating systems for VSS-based backups to enable application-consistent snapshots for Hyper-V Replica</td>
</tr>
<tr>
<td>41</td>
<td>Integration services must be installed before primary or replica virtual machines can use an alternate IP address after a failover</td>
</tr>
<tr>
<td>42</td>
<td>Authorization entries should have distinct tags for primary servers with virtual machines that are not part of the same security group.</td>
</tr>
<tr>
<td>43</td>
<td>To participate in replication, servers in failover clusters must have a Hyper-V Replica Broker configured</td>
</tr>
<tr>
<td>44</td>
<td>Certificate-based authentication is recommended for replication.</td>
</tr>
<tr>
<td>45</td>
<td>Virtual hard disks with paging files should be excluded from replication</td>
</tr>
<tr>
<td>46</td>
<td>Configure a policy to throttle the replication traffic on the network</td>
</tr>
<tr>
<td>47</td>
<td>Configure the failover TCP/IP settings that you want the replica virtual machine to use in the event of a failover</td>
</tr>
<tr>
<td>48</td>
<td>Resynchronization of replication should be scheduled for off-peak hours</td>
</tr>
<tr>
<td>49</td>
<td>Certificate-based authentication is configured, but the specified certificate is not installed on the replica server or failover cluster nodes</td>
</tr>
<tr>
<td>50</td>
<td>Replication is paused for one or more virtual machines on this server</td>
</tr>
<tr>
<td>51</td>
<td>Test failover should be attempted after initial replication is complete</td>
</tr>
<tr>
<td>52</td>
<td>Test failovers should be carried out at least monthly to verify that failover will succeed and that virtual machine workloads will operate as expected after failover</td>
</tr>
<tr>
<td>53</td>
<td>VHDX-format virtual hard disks are recommended for virtual machines that have recovery history enabled in replication settings</td>
</tr>
<tr>
<td>54</td>
<td>Recovery snapshots should be removed after failover</td>
</tr>
</tbody>
</table>

When you added your different hosts to Server Manager, you can browse to the Hyper-V tab and run the BPA scan on all of your servers at once (Figure 57):
Hyper-V Replica in depth

Figure 57: Run BPA for multiple servers

The result will be shown in Server Manager and you can work with many filters to sort the results (Figure 58):

Figure 58: Results of a BPA scan

But there are other ways to do this as well. As always, PowerShell is going to be your friend:

- Invoke-BpaModel -ModelId Microsoft/Windows/Hyper-V
- Get-BpaResult -ModelId Microsoft/Windows/Hyper-V
- Get-BpaResult -ModelId Microsoft/Windows/Hyper-V -Filter Noncompliant

By using these commands, you can run a BPA scan and get the results. And because it is PowerShell, you can save the results, format them and more.
Monitoring through the GUI

When everything is in place, you can monitor health through Hyper-V Manager. The first thing you can do is add a column to Hyper-V Manager that will show you the replication status.

Go to the Hyper-V Manager, right-click on the columns bar in the Virtual Machines window and select Add/Remove Columns... (Figure 59):

![Add/Remove Columns](image)

*Figure 59: Adding a column*

In the **Add/Remove Columns** window, choose **Replication Health** and press **Add** and then **OK** (Figure 60):

![Add/Remove Columns](image)

*Figure 60: Add/Remove Columns window*
Figure 60: Add the Replication Health column

Now you will be able to see the replication health at a glance in your virtual machines window (Figure 61).

<table>
<thead>
<tr>
<th>Virtual Machines</th>
<th>Name</th>
<th>State</th>
<th>CPU Usage</th>
<th>Assigned Memory</th>
<th>Uptime</th>
<th>Status</th>
<th>Replication Health</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tom's Test</td>
<td>Off</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Not Applicable</td>
</tr>
<tr>
<td>DriveCC</td>
<td>Off</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Critical</td>
</tr>
<tr>
<td>Core</td>
<td>Off</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Critical</td>
</tr>
<tr>
<td>KiNDIC</td>
<td>Off</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Critical</td>
</tr>
<tr>
<td>Appliance</td>
<td>Off</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Critical</td>
</tr>
</tbody>
</table>

Figure 61: Looking at the Replication Health

As you can see, I have a few VMs with no replication health status (Not Applicable) and two VMs that are Critical and one with a Warning. Because I noticed that something is wrong, I am going to look a bit deeper into the health state.

Right-click on the VM, choose **Replication > View Replication Health**…(Figure 62):

In the window that launches, you can see various statistics about your replication, such as the average latency, average size, maximum size and more. In this case, there is a warning: The replication was stopped during a period of time because one of the hosts was down (Figure 63).
In Figure 64, you see a replication health window with a critical state. This also gives you more information on what can be wrong with the replica.
Another way to review replication health is to select the Replication tab located at the bottom of Hyper-V Manager (shown in Figure 65).

![Figure 65: Replication tab](image)

And last, but certainly not least, you can get more information by using PowerShell and, in this case, the `Measure-VMReplication` cmdlets (Figure 66).

![Figure 66: Health through PowerShell](image)

**Backup interoperability**

In an ideal world, you would now redirect your backups to the replica VM and take advantage of the fact that you are not disturbing your production workloads with backups. But the ideal world does not exist. Before you start planning your backup strategy, consult with your vendor to see how it supports Hyper-V Replica.

In addition to backup, you also need to think also about restore. Imagine that you restore the primary VM—what will happen? Will the replica VM require a full resync? And what happens when you restore the replica VM (if supported)—will you need to resync fully or can it use the recovered replica VM as a base for its initial replication?

Make sure you know the answers to these questions BEFORE you are in a problematic situation.
Some additional supported features

Online resizing of disks

Starting with Windows Server 2012 R2, online resizing of your virtual disks is supported. The nice thing is that this feature is also supported when that specific VM is being replicated. However, keep in mind a few points:

- There is no need for resynchronization
- There is certainly no need to delete and to re-enable the replication
- BUT! The resizing needs to be performed manually on both sides (PowerShell can also be used)
- And when you shrink your disks, older recovery points (if you use them) will not be usable anymore!

Upgrading to R2

If you have already set up an entire replication strategy in Windows Server 2012 and want to use the additional functionality of R2, the upgrade process doesn’t have to be complicated (in contrast to what many people would suspect).

On a very high level, you need to follow these steps:

1. Upgrade your replica servers first (replication from Windows Server 2012 to 2012 R2 works but not vice versa)
2. Consider doing a planned failover to minimize downtime and if you want to have no downtime at all, you should consider cross-live migration when the primary server is in a clustered configuration
3. Upgrade your primary server

Replicating VDI VMs and deduplication

If you are replicating VDI VMs, then you are allowed to use data deduplication. To save on disk space, you can enable data deduplication on the replica VM as well.
**Saving disk space (general)**

The above is supported when using VDI, but if you are replicating server VMs, it is a no-go. However, there is a way to save on disk space on the replica side by making use of dynamic disks.

At a high-level, the procedure is as follows:

1. Enable replication using online IR or out-of-band IR
2. Wait for the initial replication to complete
3. Pause the replication on the replica server—this part is important!
4. Use the Edit Disk and Convert option to convert the disk from fixed to dynamic
5. Replace the fixed disk with the dynamic disk
6. Resume the replication

**Best practices**

**Separate disk for the paging file**

I consider this not only a best practice for Hyper-V Replica, but for every VM you are running. Some backup solutions that protect the VM as an entity won’t exclude the paging file in the backup, but most of the time they do have the ability to exclude a disk. Veeam Backup & Replication will automatically exclude paging files for you if you were wondering.

More important, the paging file is a file that changes a lot; thus, when you do incremental replication, these blocks are the biggest candidates to be transferred. And you actually don’t need them. In any case, when you start your replica, you “boot” the VM, so the page file is empty at that moment.

**Think about your performance**

Performance is one of the main issues I have seen AFTER companies have enabled this technology. They get into problems with their resources because they didn’t plan well. Go back to the planning part of this document and make sure you have followed all the steps.
**Use network throttling**

If you are not using a dedicated network, or you are replicating over the WAN that is shared with many other solutions, then network throttling is a must.

With Windows Server 2012 QoS that is rather easy to do and there are a few options to choose from:

- **Throttling based on the destination subnet:**
  
  ```powershell
  New-NetQosPolicy “Replication Traffic to 10.0.0.0/8” –DestinationAddress 10.0.0.0/8 –MinBandwidthWeightAction 40
  ```

- **Throttling based on the destination port:**
  
  ```powershell
  New-NetQosPolicy “Replication Traffic to 8080” –DestinationPort 8080 –ThrottleRateActionBytesPerSecond 100000
  ```

- **Throttling based on Application Name**
  
  ```powershell
  New-NetQosPolicy “Replication Traffic from vmms.exe”
  -IPPortMatchCondition 80 -AppPathNameMatchCondition *vmms.exe -IPProtocolMatchCondition TCP –ThrottleRateActionBitsPerSecond 100000
  ```

**Azure Site Recovery**

Last but not least, Microsoft offers a service called Azure Site Recovery. This service allows you to create a disaster recovery plan that is orchestrated when you need to execute it. Many of you will probably say that it isn’t necessary, but my experience shows that this can be very important. During the creation and testing of a disaster recovery plan, most IT administrators are relaxed and working under normal circumstances. However, having been a DR consultant, I have seen that when a true disaster happens, the nerves, time pressure and “looking-over-the-shoulder syndrome” all come into play and can cause administrators to fail at the simplest tasks. And there is no blame in that—humans are not made to be flawless under every circumstance.

With Azure Site Recovery, you can automate all of the DR steps, so in the event of a true disaster, you only need to push one button and let Azure Site Recovery do the work.
In a nutshell, here is what this service delivers:

- Automation
- Remote health monitoring
- Recovery plans (that you can customize)
- Recovery plan testing (with no impact!)
- Orchestrated recovery
- The potential to replicate to (and recover in) Azure

Of course, because it is a service in Azure, it comes with a price tag and you will need additional tools such as System Center Virtual Machine Manager (and clouds deployed in SCVMM) to be able to use it.

For more information, visit http://azure.microsoft.com/en-us/services/site-recovery/

**Conclusion**

Hyper-V Replica is a very good technology that you can use in your infrastructure. As with many technologies, planning is key; and to make this a successful implementation, it is necessary to calculate the requirements and set up the correct infrastructure.

If you don’t yet have a replication tool that is integrated with your backup solution and you are running Hyper-V, then this is certainly something you should explore.

**Appendix A: Resources used**


About the author

Mike Resseler is a Product Strategy Specialist for Veeam. Mike is focused on technologies around Hyper-V and System Center. With years of experience in the field he presents on many occasions on large events such as MMS, TechEd and TechDays. Mike has been an awarded the MVP for System Center Cloud and Datacenter Management since 2010 and received the Hyper-V MVP since 2014. His major hobby is discussing and developing solid Disaster Recovery scenarios. Additionally, he has enterprise-class experience in Private Cloud architecture, deployment with marked focus on protection from the bottom to the top. He holds certifications in many Microsoft Technologies such as MCITP.

Follow Mike on @MikeResseler or @Veeam and on Google+.

About Veeam Software

Veeam® enables the Always-On Business™ by providing solutions that deliver Availability for the Modern Data Center™, which provides recovery time and point objectives (RTPO™) of less than 15 minutes for all applications and data. Veeam recognizes the challenges in keeping a business up and running at all times and addresses them with solutions that provide high-speed recovery, data loss avoidance, verified protection, leveraged data and complete visibility. Veeam Backup & Replication™ leverages technologies that enable the modern data center, including VMware vSphere, Microsoft Hyper-V, NetApp storage, and HP 3PAR StoreServ and StoreVirtual Storage, to help organizations meet RTPO, save time, mitigate risks, and dramatically reduce capital and operational costs. Veeam Availability Suite™ provides all of the benefits and features of Veeam Backup & Replication along with advanced monitoring, reporting and capacity planning for the backup infrastructure. Veeam Management Pack™ for System Center is the most comprehensive, intuitive and intelligent extension for app-to-metal management of Hyper-V and vSphere infrastructures, and includes monitoring and reporting for Veeam Backup & Replication. The Veeam Cloud Provider (VCP) program offers flexible monthly and perpetual licensing to meet the needs of hosting, managed service and cloud service providers. The VCP program currently includes more than 6,000 service provider partners worldwide.

Founded in 2006, Veeam currently has 27,000 ProPartners and more than 121,500 customers worldwide. Veeam’s global headquarters are located in Baar, Switzerland, and the company has offices throughout the world. To learn more, visit http://www.veeam.com.
Veeam Availability Suite v8

AVAILABILITY™
for the Modern Data Center

RTPO <15 min for ALL applications and data

To learn more, visit www.veeam.com