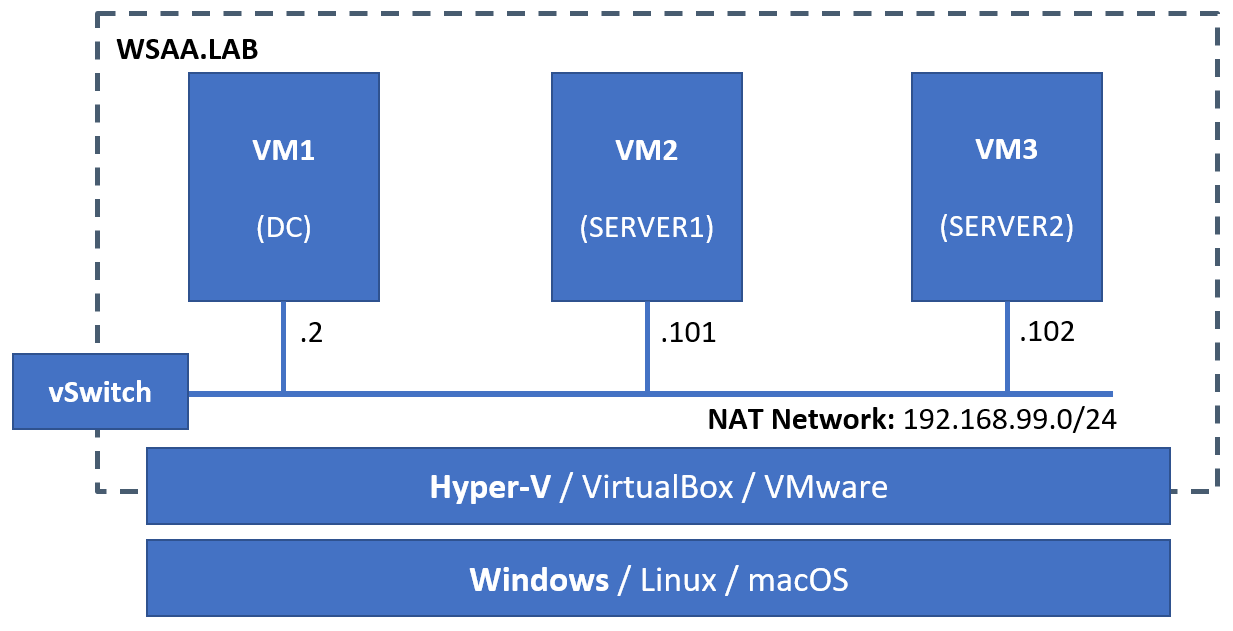
# Practice M2: Virtualization

For the purpose of this lab and the course, we will consider that we are working in a pure Windows environment either on-premise or in the cloud and using **Hyper-V** as a virtualization solution. All tasks can be achieved under different configuration (another host OS or virtualization solution) with the appropriate adjustments

The infrastructure we will be working with is like:



## Part 1: Virtualization 101

Our goal for this part is to try to establish a small infrastructure with three VMs in a manual manner

One of the machines will act as a domain controller and the rest as member servers

All machines could be with **Desktop Experience** and with **Standard** edition of **Windows Server 2019**

*Please note, that while the following steps can be executed in virtual machines, we will do them on a physical machine, because otherwise we will need a technique called* ***nested virtualization*** *(we will talk about it a little bit later)*

### Turn on the Hyper-V *(on the host)*

To enable **Hyper-V** with all its tools on a **Windows Server** machine we must execute:

**Install-WindowsFeature -Name Hyper-V -IncludeManagementTools -Restart**

Or use the **Server Manager** and from there go to **Manage > Add Roles and Features** and add the **Hyper-V** role

The above procedure is the same for **Windows Server** version **2012**, **2012 R2**, **2016**, and **2019**. And it is applicable to both **Standard** and **Datacenter** edition

If we are working on **Windows 10** (**Enterprise**, **Pro**, or **Education**), then the above procedure should be adjusted. The command will become:

**Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Hyper-V -All**

*If we are using* ***Windows 10 Home****, then even though not officially supported, there is still a way to install and activate the* ***Hyper-V*** *role. You can search with you favorite search engine for how-to enable hyper-v on windows 10 home*

### Configure Hyper-V *(on the host)*

Start the **Hyper-V Manager** and select the **Hyper-V** server

In the **Actions** menu click on **Hyper-V Settings**

Explore the available configuration options

Let’s put both the virtual disks and virtual machines in a same custom folder

Got to **Virtual Hard Disks** and specify a folder, for example **D:\VM**

Go to **Virtual Machines** and specify the same folder

Confirm with **OK**

Now on, all new objects will go by default to the folder specified

### Create a VM with Hyper-V Manager *(on the host)*

With the **Hyper-V Manager** open, select the **Hyper-V** server

In the **Actions** menu click on **New** and then on **Virtual Machine**

On the **Before You Begin** screen click on **Next**

Next will appear the **Specify Name and Location** screen

Enter for example **SERVER1** and click on **Next**

On the **Specify Generation** screen note that by default **Generation 1** is selected. Change it to **Generation 2** and click **Next**

Next, the **Assign Memory** screen will appear

Leave the default values and click **Next**

On the **Configure Networking** screen accept the default values and click **Next**

Next, the **Connect Virtual Hard Disk** screen will appear. Please note that by default the size of the disk is 127 GB, but because it is a dynamic disk it won’t need that much space available. Click **Next**

On the **Installation Options** screen accept the default values (**Install an operating system later**) and click **Next**

Review the information on the **Summary** screen and click **Next**

After a while, we have our first VM ready

### Create a VM with PowerShell *(on the host)*

Now, we will repeat what we did graphically but this time with **PowerShell**. It is a matter of execution a single command

Let’s borrow the command from the slides

Open a **PowerShell** session

Execute the following command

**New-VM -Name SERVER2 -Generation 2 -MemoryStartupBytes 1gb**

It won’t work. The reason is that we launched a regular session

Now, close the **PowerShell** windows and open a new session, but this time with **Run As Administrator** and repeat the command. It must succeed

Go to **Hyper-V Manager** and examine the virtual machine just created. It appears that there is no hard disk. Leave it as it is, we will add one later. Of course, you can delete if you like

Return to the **PowerShell** session and execute a modified command

**New-VM -Name SERVER3 -Generation 2 -MemoryStartupBytes 1gb -NewVHDSizeBytes 127gb -NewVHDPath SERVER3.vhdx**

Now it will execute successfully

Go to the **Hyper-V Manager** and examine the result

### Adjust VM settings *(on the host)*

Select a VM (for example **SERVER1**) in the **Virtual Machines** section of the **Hyper-V Manager**

Click on **Settings** in its context menu

Navigate to the **SCSI Controller** in the left section

In the right section select **DVD Drive** and click **Add**

Then change the **Media** selection to **Image file** and navigate to the installation media

Click on **Apply** to confirm that we want to add the DVD drive

Navigate to the **Firmware** option in the left section

Adjust the order to become **DVD Drive**, **Hard Drive**, and then **Network Adapter** in the right section

Confirm with **Apply**

*Don’t forget to navigate to* ***Checkpoints*** *in the left section and turn off the* ***Use automatic checkpoints*** *if currently on*

Save all changes by closing the window with the **OK** button

### Guest OS installation *(on the host)*

Return to the **Hyper-V** **Manager** and select the **SERVER1** VM

Click on **Connect** in the VM’s menu and then start the VM

Follow the steps required to install the chosen OS

*We will skip forward by importing a ready machine and on a latter stage will go in detail about the import process*

### Prepare a VM for export *(on the host)*

This is the time to add or remove components, or adjust some settings

After we are satisfied by the state of our virtual machine, we can move on

It is good practice to **generalize** or **sysprep** a **Windows** VM especially if are going to use in an AD environment

With the VM running (for example **SERVER1**) select it and click on **Connect** in the VM’s menu

A session window will open. Log on to the machine

Open a command line session with **Run As Administrator**

Navigate to **C:\Windows\System32\Sysprep** folder

Execute the **Sysprep.exe** tool

In **System Cleanup Action** select **Enter System Out-of-Box Experience (OOBE)**

Select the **Generalize** option

In the **Shutdown Options** drop down select **Shutdown** and click on **OK**

After a while, once the generalization is done, the virtual machine will turn off

### Export a VM *(on the host)*

While still in the **Hyper-V Manager** select the virtual machine (for example **SERVER1**) and click **Export** in the VM’s menu

Turn off the VM

An **Export Virtual Machine** dialog box will appear

Select a destination folder and initiate the process by clicking on **Export**

The dialog box will close immediately but depending on the size of the VM the process will run for a while

We can monitor the progress in the **Virtual Machines** section of the **Hyper-V Manager**

As usual, the same can be achieved with the help of **PowerShell**. The export command is this

**Export-VM -Name SERVER1 -Path D:\Backup**

Now, using the exported VM, we can create multiple machines out of it

### Explore VMs *(on the host)*

We can always go to the **Hyper-V Manager** and have a quick look at the **Virtual Machines** section in order to see what VMs we have, which are in what state, and so on

Of course, we can do the same with **PowerShell**, just execute this simple command

**Get-VM**

Here we can do even more. We can ask for more details by executing this

**Get-VM | Select \***

### Delete VMs *(on the host)*

Because we have a copy of our machine it is safe to delete the original

Navigate to the **Virtual Machines** section in the **Hyper-V Manager**

Select the VM in question (for example **SERVER1**) and from its menu select the **Delete** option

Confirm the operation by answering **Delete** on the question dialog that appear

After a while, our machine will be gone

As usual, we will look at how it is done in **PowerShell**

Again, it is a matter of executing a single command

**Remove-VM -Name SERVER1**

We will be asked if we are sure, we must confirm by entering **Y** and pressing the **Enter** key

Please note, that the act of deleting a VM doesn’t delete the associated virtual hard disks, so we must take care for this all by ourselves

Navigate to the folder where the virtual disks are stored and delete the ones you don’t need anymore

### Import a VM *(on the host)*

Let’s practice the import process a bit

Return to the **Hyper-V Manager**

In the **Actions** menu select the **Import Virtual Machine** option

New **Import Virtual Machine** wizard will open

On the **Before You Begin** window click **Next**

Then, the **Locate Folder** screen will appear. Navigate to the folder where the export took place

Click on **Next**

Then, the **Select Virtual Machine** window will appear. Check if this is the correct machine and click on **Next**

In the **Choose Import Type** screen select the **Copy the virtual machine (create a new unique ID)** option and click **Next**

Then, on the **Choose Folders for Virtual Machine Files** screen leave everything as it is and click **Next**

On the **Choose Folders to Store Virtual Hard Disks** screen accept the defaults and click **Next**

If everything with our choices so far is fine and we have a green light to continue, the last screen that will appear is the **Summary** or **Completing Import Wizard**. Otherwise, an error message will be show and we must take the required corrective measures

Assuming that everything is fine, click on **Finish**

### Prepare the environment *(on the host)*

Using what we learned so far, let’s create the desired setup – one domain controller and three member servers

Try to import the same once more

It won’t be a successful attempt

You should have noticed that the **Import Wizard** won’t ask us for a name neither for the VM nor for the virtual disk

Perhaps the right approach is a mixture of what we know so far

A valid approach is to create the virtual machines without virtual disks

Then copy the virtual disk from the export (usually where the VMs are located) as many times as needed and each time adjust the name

Alternatively, we can prepare first the disks and then during the machine creation to link each machine to an existing disk

Open each VM settings and add the corresponding virtual disk

Now, we have the infrastructure at least sort of

As our machines are not connected, we cannot join the member servers to the domain

We will continue with this in the second part

## Part 2: Virtualization 102

In this part we will experiment with disks and network related stuff and complete our infrastructure eventually

### Explore network infrastructure

Navigate to the **Hyper-V Manager**

Click on the **Virtual Switch Manager** in the **Actions** section

Normally, we should see just the **Default Switch** that was created during the installation

We can examine its type and settings

### Isolated network

Let’s create a new **Private network** switch

While still in the **Virtual Switch Manager**, click on **New virtual network switch** option in the left section

Select **Private** for switch type in the right section

Click on **Create Virtual Switch** button

Enter a name, for example **Private Switch**

Confirm with **OK**

### NAT network

Let’s create a new **NAT switch**

Open a **PowerShell** session with elevated privileges and execute the following

**New-VMSwitch -SwitchName "NAT vSwitch" -SwitchType Internal**

**New-NetIPAddress -IPAddress 192.168.99.1 -PrefixLength 24 -InterfaceAlias "vEthernet (NAT vSwitch)"**

**New-NetNAT -Name "NAT Network" -InternalIPInterfaceAddressPrefix 192.168.99.0/24**

You may need to adjust the above lines to match your requirements

Close the session window

### External network

Open the **Virtual Switch Manager** again

Click on **New virtual network switch** option in the left section

Select **External** for switch type in the right section

Click on **Create Virtual Switch** button

Enter a name, for example **External Switch**

Select a network adapter to be used

Confirm with **OK**

A dialog window will appear informing us that there could be some connectivity disruption

If we are sure, we must confirm with **Yes**

### VM network configuration

Let’s start building the network infrastructure

Open the settings for the first (DC) VM and navigate to the **Network Adapter**

In the **Virtual switch** drop-down on the right select the **NAT Switch** created earlier

Confirm with **OK**

Go to the second VM (member server 1) and open its settings

Navigate to the **Network Adapter** option in the left section

In the **Virtual switch** drop-down on the right select the **NAT Switch** created earlier

Confirm with **Apply**

Close the window with **OK**

Repeat the procedure for the third VM as well

### Finalizing the infrastructure

Now, we can finish the domain setup and joining phase if we want

Before we do it, we must set the appropriate names and static IP addresses to the NICs of the VMs connected to the **Private Switch** or to the **NAT Switch** depending on the option you chose

If we have all machines joined to a domain, we can manage everything from a single machine

For the sake of the rest of the tasks in this section, even a stand-alone machine with internet connectivity will do the job

### Nested virtualization

Before we turn our member servers to **Hyper-V** hosts, we must adjust a few settings

First, stop them

Open a **PowerShell** session with **Run as Administrator** on the host

Turn on nested virtualization for the first member machine (for example **SERVER2**)

**Set-VMProcessor -VMName SERVER2 -ExposeVirtualizationExtensions $true**

Then disable **Dynamic Memory** for the VM

**Set-VMMemory -VMName SERVER2 -DynamicMemoryEnabled $false**

Turn on the MAC address spoofing

**Set-VMNetworkAdapter -VMName SERVER2 -MACAddressSpoofing on**

Repeat the same procedure for the other member server (for example **SERVER3**)

Don’t forget to set the memory to **4 GB** at least

Then power on member servers and check the assigned memory

Open a connection to one of the member servers and install the **Hyper-V** role

### Create fixed virtual disk (VHDX)

Navigate to the **Hyper-V Manager** and click on the **New** and then **Hard Disk**

On the **Before You Begin** screen click on **Next**

Then, on the **Choose Disk Format** screen, ensure that the **VHDX** option is selected and click **Next**

Then, on the **Choose Disk Type** screen, select the **Fixed size** option and click **Next**

On the **Specify Name and Location** screen, for name enter **Spare1.vhdx** and click **Next**

On the next screen (**Configure Disk**) ensure that the **Create a new blank virtual hard disk** is selected, adjust the size to **1GB** and click **Next**

After reviewing the information on the **Summary** screen, confirm the creation with **Finish**

As usual, we must look at how this can be done with **PowerShell**

Let’s create a second fixed disk, this time **500MB**

The command will be this one

**New-VHD -Path D:\VM\Spare2.vhdx -Fixed -SizeBytes 500mb**

### Attaching and detaching virtual disk

Go to the settings of the first VM (for example **SERVER1**)

Select the **SCSI Controller** in the left section

Then select the **Hard Drive** option in the right section and click **Add**

Click the **Browse** button and navigate to the **Spare1.vhdx** file created earlier

Confirm with **OK**

*Please note that this can be done while the VM is up and running*

Connect to the VM and open the **Disk Management** tool

You will see the newly added disk there ready to be used

Let’s bring it online and initialize it with a **GPT** partition

Then create a **Simple Volume** around **500MB** in size and format it with **NTFS**

Mount the disk and navigate to its root folder

Create a simple **Readme.txt** file with an arbitrary text

Return to the VM settings and detach the disk

Go to the settings of the second VM (for example **SERVER2**) and attach it there

Connect to the VM and check if the disk is usable

It appears that we must go through **Disk Management** tool again to bring the disk online

After this manipulation, the disk should be usable

Before we move on, attach the second virtual disk (**Spare2.vhdx**) to one of the VMs, initialize it and format its full capacity

### Resizing virtual disks (expand)

Let’s try to expand the first disk (**Spare1.vhdx**) while it is mounted

Navigate to the **Hyper-V Manager**

In the **Actions** section click on the **Edit Disk** option

Navigate to the **Spare1.vhdx** disk file and click **Next**

The only offered option is to expand it, so click **Next** again

Set **New size** to **2GB** and click **Next**

Confirm with **Finish**

The process will take a while. Once done, we can go to the VM and check if the disk size was expanded

We can do the same to the second virtual disk (**Spare2.vhdx**) with **PowerShell**

**Resize-VHD -Path D:\VM\Spare2.vhdx -SizeBytes 2gb**

### Resizing virtual disks (shrink)

Let’s try to shrink down the size of the disk (**Spare1.vhdx**) we just expanded by removing all unallocated space

This can be done with a single **PowerShell** command

**Resize-VHD -Path D:\VM\Spare1.vhdx -ToMinimumSize**

### Convert virtual disk from fixed to dynamic

Our second disk, even though is 2GB in size, has just a 500MB partition

So, let’s convert it to dynamic and free the allocated space

Navigate to the **Hyper-V Manager**

In the **Actions** section click on the **Edit Disk** option

Navigate to the **Spare2.vhdx** disk file and click **Next**

On the **Choose Action** screen we will be presented only with **Expand** (almost always available) and **Shrink** (there is some unallocated space) options

So, it appears that we cannot convert it right now. Let’s detach it from the VM. First, close the tool by clicking **Cancel** and then detach the disk

Once done, return to the **Edit Disk** tool

Now, we are presented with four options, including the **Convert** one. Select it and click **Next**

Keep the **VHDX** format and click **Next**

Change the **Disk Type** to **Dynamically expanding** and click **Next**

Select a location and name for the new disk, for example **D:\VM\Spare3.vhdx** and click **Next**

Confirm with **Finish**

The process will take a while. Once done, use the **Inspect Disk** tool to check the result

The disk is successfully converted to dynamic and shrunk down while the maximum capacity is left intact

### Checkpoints

Turn off one of the member machines (choose one with additional disk attached)

Select the VM (for example **SERVER2**)

Go to **Settings** and then select **Checkpoints** in the left section

Check if they are enabled and which type is selected

Make sure that the **Production checkpoints** is selected, and the **Use automatic checkpoints** option deselected

Confirm with **OK**

Now, from the context menu of the VM select the **Checkpoint** option to create one

Navigate to the folder where the VMs are stored and examine what additional files appeared

Beside the files for the main disk and the VM itself, there is a differencing disk created for the additionally attached virtual disk

We can confirm this with the help of the **Inspect Disk** tool and get some information about the differencing disk

## Part 3: Virtualization 103

### Preparation

Before we continue further, we must make sure that we have the required infrastructure

We will need the following:

* VM1 – domain controller (DC), 1 vCPU, 1536 MB RAM, OS Disk, IP: 192.168.99.2
* VM2 – Hyper-V machine 1 (SERVER1), 1 vCPU, 3072 MB RAM, OS Disk, IP: 192.168.99.101
* VM3 – Hyper-V machine 2 (SERVER2), 1 vCPU, 3072 MB RAM, OS Disk, IP: 192.168.99.102

Of course, the above can be adjusted both in terms of number (less likely) of machines and/or allocated resources

Now is the time to start from scratch. Why not prepare a script (or use the one provided with the materials)

Make sure that both Hyper-V servers are prepared for nested virtualization:

* Dynamic memory is disabled, and memory size is setup to **2 GB** or more
* Virtualization extensions are exposed to both VM
* MAC address spoofing is enabled for network adapters of both VM

Using the **Server Manager** or **PowerShell** install the **Hyper-V** role on both member servers – **SERVER1** and **SERVER2**

Remember to:

* set the default virtual machine stores for example to **C:\HV**. It can be done later as well
* create switch with same name and type on both virtual hypervisors
* make sure that the user you are using is part of the local Hyper-V Administrators group of the hosts

### VM creation / import

Log in to **SERVER1** and open **Hyper-V Manager**

Either create a small virtual machine or import an existing one

Should you decide to create virtual machine, name it **VM1** and create it with the following characteristics:

* Generation 2
* 512 MB RAM
* 10 GB HDD
* No need to be connected to a switch

Mount a Linux ISO (something minimal) and install the OS on the VM. Please bear in mind that resources are needed

Alternatively, you can import it from a template but again a minimal one

Or if you are limited on resources, just create an empty virtual machine

Power on the VM no matter if there is or is not an OS installed

### Shared-nothing live migration

#### Virtual host (hypervisor) preparation

Log on to **SERVER1** with an account that has administrative privileges and start the **Hyper-V Manager**

Click on **Hyper-V Settings** under **Actions** menu

Select the **Live Migrations** item

Turn on the **Enable incoming and outgoing live migrations** option

Select the **Use any available network for live migration**

Click on **Advanced Features** under **Live Migrations** if you want to change **Authentication protocol** and **Performance options**

Leave them as they are and click **OK**

Repeat the same steps on **SERVER2** as well

#### VM movement

Being on **SERVER1** and in **Hyper-V Manager**, connect to **SERVER2** as well

Then, select the VM that we are about to move, for example **VM1**

Invoke its context menu and select **Move**

On the welcome screen click **Next**

Select **Move the virtual machine** option and click **Next**

Enter **SERVER2** in the **Name** text field and click **Next**

Select the **Move the virtual machine’s data to a single location** option and click **Next**

Enter **C:\HV** in the **Folder** text box and click **Next**

Check the information on the **Summary** screen and click **Finish**

Monitor the process. After a while, the machine should appear on **SERVER2**

### Hyper-V replica

This exercise relies on the successful completion of the previous one. We must have a virtual machine running on **SERVER2**

#### Virtual host (hypervisor) preparation

Log on to **SERVER1** (it will be our target server) with an account that has administrative privileges and start the **Hyper-V Manager**

Click on **Hyper-V Settings** under **Actions** menu

Select the **Replication Configuration** option

Tick the **Enable this computer as a Replica server** option

Select the **Use Kerberos (HTTP)** option

For the purpose of this lab select the **Allow replication from any authentication server** option

Click **OK**

A dialog box may appear reminding you that you should take care after a firewall rule. Write down the name (***Hyper-V Replica HTTP Listener (TCP-In)***) of the rule, click **OK** and then check if it is enabled in the firewall and if not – enable it

Now, log on to **SERVER2** (it will be our source server) and open the **Hyper-V Manager**

Repeat the set of replication settings here as well

Select the running virtual machine **VM1** and select **Enable Replication** from its context menu

A wizard will appear. On the welcome screen click **Next**

Enter **SERVER1** in the **Replica server** text box and click **Next**

Accept the default values on the **Specify Connection Parameters** screen and click **Next**

On the next screen, select the virtual hard disk drives that we want to replicate. In our case the VM has just one hard disk and it is selected. Click **Next**

Change the replication frequency to **30 seconds** and click **Next**

Make sure that the **Maintain only the latest recovery point** option is selected and click **Next**

Accept the default values on the **Initial Replication Method** window and click **Next**

Check the information on the summary screen and click **Finish**

After a while, the replication will be set up. In our case it will happen almost immediately

Return to **SERVER1** and check if there is a copy of the **VM1** machine. There should be one but in power off state

Select the virtual machine and switch to **Replication** tab (in the bottom part of the screen) to examine what is going on

Alternatively, in the context menu you can select **Replication > View Replication Health** for more detailed information

#### Failover

Being still on **SERVER1**, again in the context menu, select **Replication > Test Failover** to run a test failover

A dialog box will appear asking us which recovery point to be used as a base. Accept the offered one and click **Test Failover**

You can start the test copy and explore it. Once done, select the original virtual machine

In the context menu choose **Replication > Stop Test Failover**

A dialog box will appear warning us that the test copy of the VM will be deleted. Click on **Stop Test Failover**

Now, return to **SERVER2** and select the virtual machine that is being replicated

Shut down the virtual machine

Invoke its context menu and select **Replication > Planned Failover**

Make sure that both options are selected

Click **Fail Over** button

At first, a set of prerequisite checks will be attempted. The process will continue only if all checks pass

Once the process is complete, click **Close**

Now, if you switch to **SERVER1**, you will see that its copy of the virtual machine is running and replicating back to **SERVER2**

#### Pause and stop replication

While still on **SERVER1** select the virtual machine

Invoke its context menu and click **Replication > Pause Replication** to temporary stop the replication

Or **Replication > Remove Replication** to terminate it

Go with the second option. A dialog box will appear. Confirm with **Remove Replication**