Hyper-V – VM 'Shared Nothing' Live Migration (SNLM)

Server 2012 introduces a new type of live migration: a shared-nothing live migration, i.e., *No shared storage*, no shared cluster membership -- all you need is a Gigabit Ethernet connection between the Server 2012 Hyper-V hosts. With this network connection, you can move a VM between Hyper-V hosts, including moving the VM's virtual hard disks (VHDs), memory content, processor, and device state with no downtime to the VM. However, you cannot use shared nothing-live migration to migrate between clusters directly. Instead, you need remove HA of VM and move.

'Shared-Nothing Live Migration' Requirements

- Same hardware or chipset to be either Intel or AMD on both Hyper-V Hosts unless you are using virtual machine's Processor Compatibility feature.
- Identically named Virtual Switch on the source and destination Hyper-V Hosts. If you do not specify the same name for the Virtual Network Switch then 'Shared-Nothing Live Migration' process pauses.
- Two Hyper-V Hosts connected via a reliable Network Connection; preferably Gigabyte Network Connection and Hyper-V role enabled or Windows Server 2012 Hyper-V Server free edition.
- Live migration must be enabled on the Hyper-V Hosts
- Both Hyper-V Hosts must be part of the same Active Directory Domain.
- Pass-Through disks are not supported by the 'Shared-Nothing Live Migration' technology. You can only use VHD files.
- Firewall exception for TCP over port 6600. This firewall rule exception is enabled in the Firewall when you initially enable the 'Live Migration'on the Hyper-V Host.
- SMB 3.0 File System. 'Shared-Nothing Live Migration' feature leverages SMB 3.0 introduced in Windows Server 2012.
- The following services enabled on the Network Adaptors you use for 'Shared-Nothing Live Migration' feature:
    - Client for Microsoft Networks
    - File and Printer Sharing for Microsoft Networks

Although the default settings might work in simple environments or for a basic test, most environments will want to switch to Kerberos for authentication and will want to use a specific network for live migration traffic, which will include both a copy of the VM memory and its storage. Using Kerberos allows administrators to initiate live migrations remotely; using a specific network helps to manage network traffic and to ensure that the required network bandwidth is available for live migrations. Let's look at authentication first and why it's a challenge for live migration in a non-clustered environment.

Authentication for Live Migration

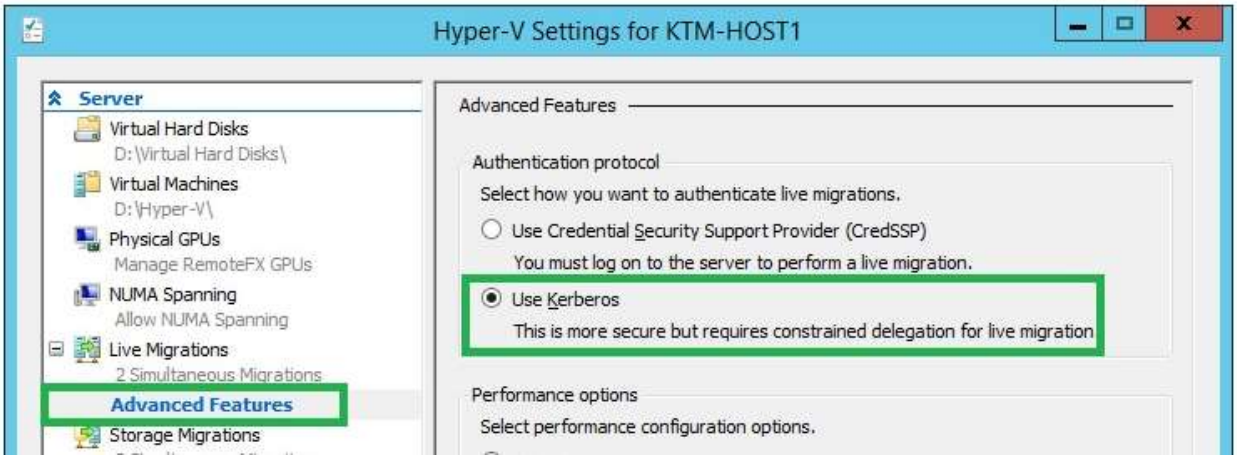
In a cluster environment in which all Hyper-V hosts are part of a failover cluster, all the Hyper-V hosts share a common cluster account. This account is used for communication between the hosts for authentication, simplifying (from an authentication perspective) operations such as migrations within a cluster. Outside of a cluster, each Hyper-V host has its own computer account, without a shared

credential; when operations are performed, the user account of the user who is performing the action is typically used for authentication.

With a live migration, actions are taken on the source and target Hyper-V servers (and on file servers, if the VM is stored on an SMB share), both of which require the actions to be authenticated. If the administrator who is performing the live migration is logged on to the source or target Hyper-V server and initiates a shared-nothing live migration from the local Hyper-V Manager, then that administrator's credentials can be used both locally and to run commands on the target Hyper-V server. In this scenario, **CredSSP** works fine, allowing the administrator's credentials to be used on the remote server from the client -- basically a single authentication hop from the local machine that is performing the action to a remote server.

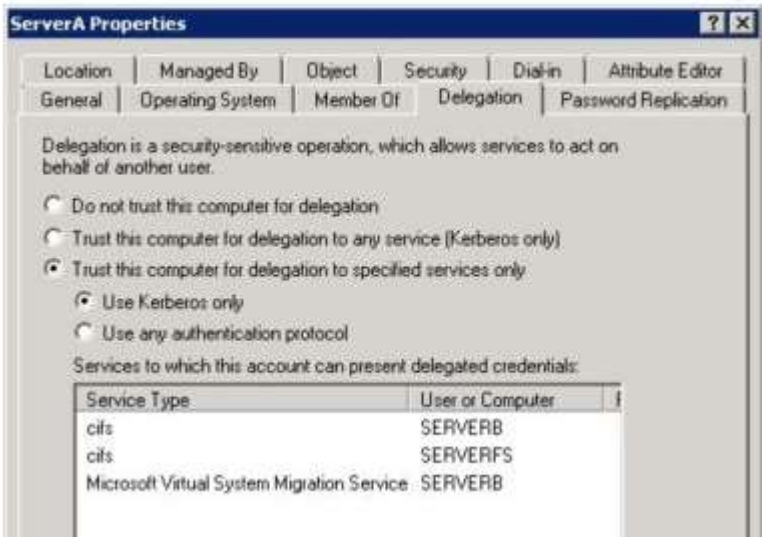
However, the whole goal for Server 2012 (and management in general) is remote management and automation. Having to actually log on to the source or target Hyper-V server each time you require a live migration outside of a cluster is a huge inconvenience for remote management. If a user was logged on to the local computer running Hyper-V Manager and tried to initiate a live migration between Hyper-V hosts A and B, that attempt would fail. The user's credentials would be used on Hyper-V host A (which is one hop from the client machine), but Hyper-V host A would be unable to use those credentials on Host B to complete the live migration. The problem is that **CredSSP** doesn't allow credentials to be passed to a system that is more than one hop away. This is where the option to use **Kerberos** enables full remote management: Kerberos supports constrained delegation of authentication. Therefore, when a user performs an action on a remote server, that remote server can use the user's credentials for authentication on a second remote server.

Note: CredSSP often fails in migration, it is best to choose Kerberos Authentication in case of migration failures under Live Migration in Hyper-V Settings at both source and destination.



To configure this delegation, use the Active Directory Users and Computer management tool and the computer account properties of the server that will be allowed to delegate. As below Figure shows, the Delegation tab contains settings for the allowed level of delegation. For most computers, the

configuration that this figure shows -- allowing delegation only for ***specific services*** and only for the Kerberos protocol -- is optimal.

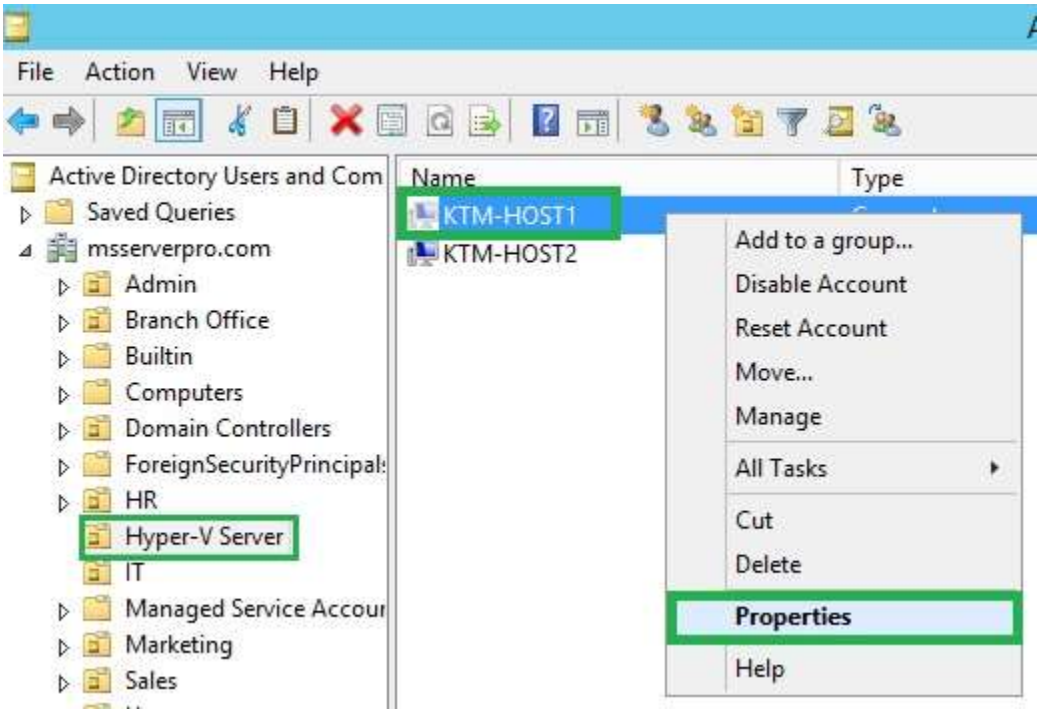


The only service that requires delegation is the ***Microsoft Virtual System Migration Service & CIFS***, which should be enabled for the target Hyper-V server. You *must* set authentication to *Use Kerberos only*

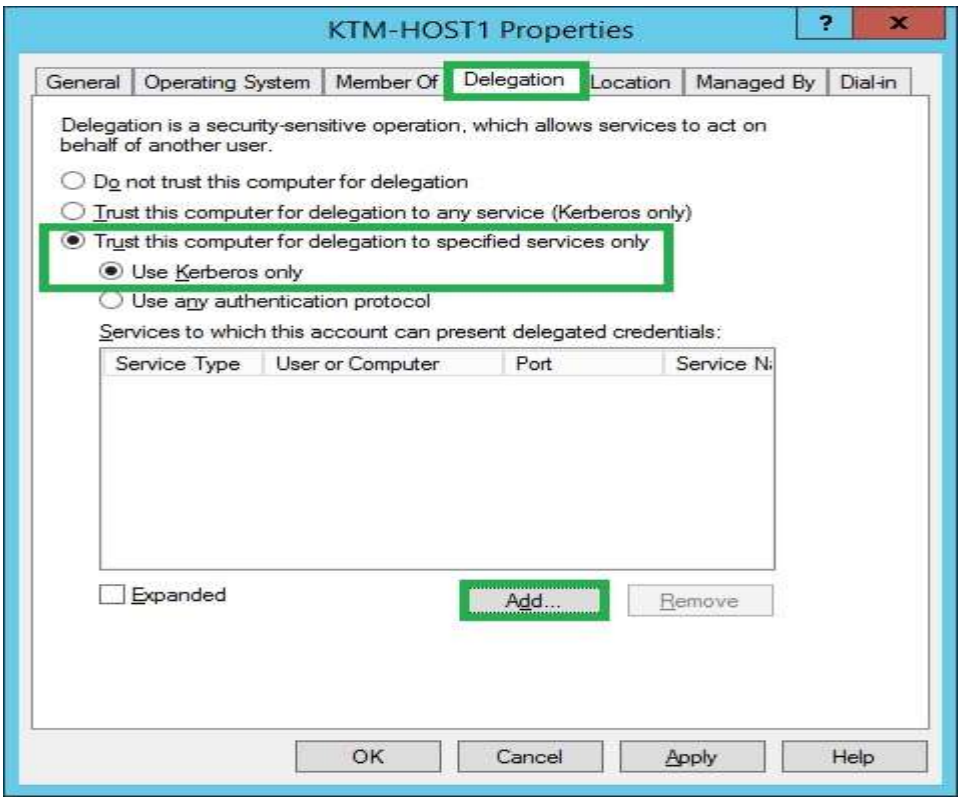
Let us take the example of Hosts KTM-HOST1 & KTM-HOST2 to show the SNLM steps

Task 1: Configure Constrained Delegation

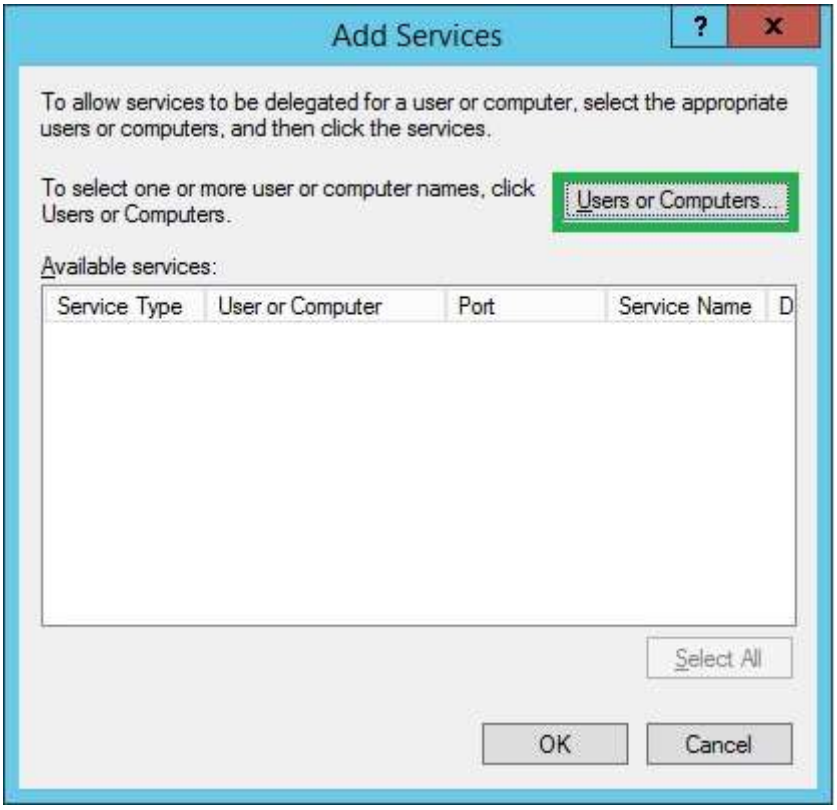
1. On KTM-HOST1, log in as **Domain Administrator (Administrator)**. Open **Active Directory Users and Computers**, in the navigation pane, expand **msserverpro.com** and click then **Hyper-V Server OU**. **Right-Click** on Hyper-V Host, **KTM-HOST1** and select **Properties**.



2. In the KTM-HOST1 **Properties**, Click the **Delegation** tab. On the Delegation tab, click the **Trust this computer for delegation to specify services only** option, then click **Use Kerberos only** and Click **Add**.

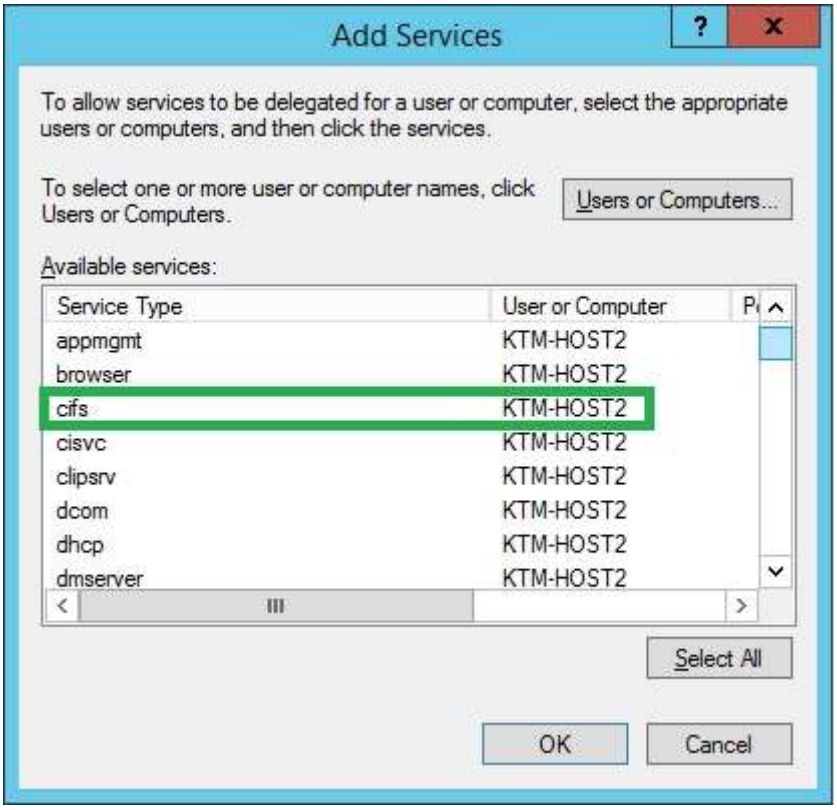


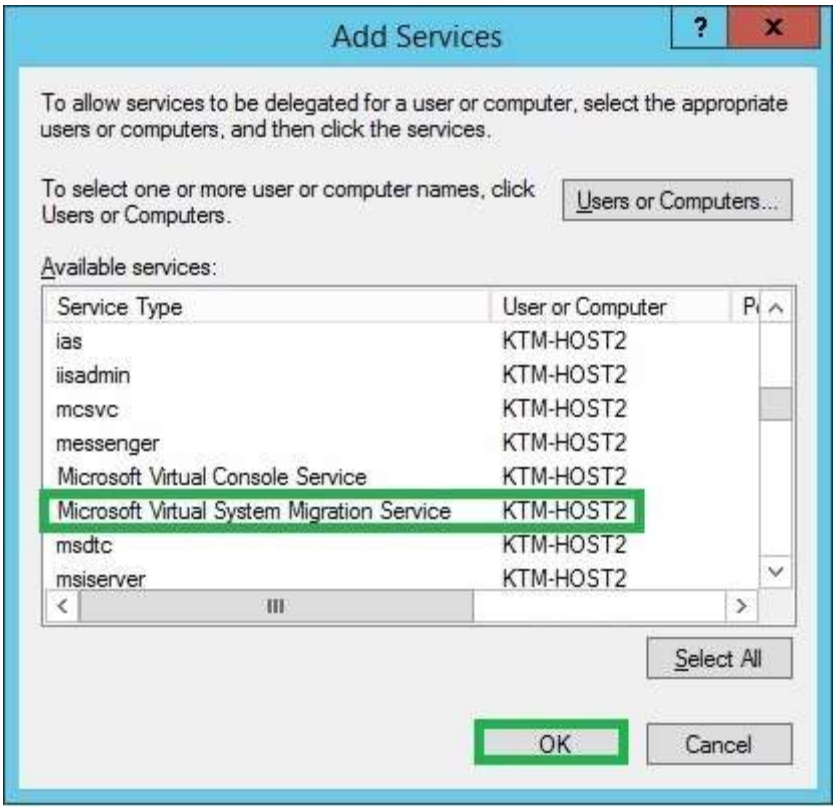
3. In **Add Services**, click **Users or Computers**.



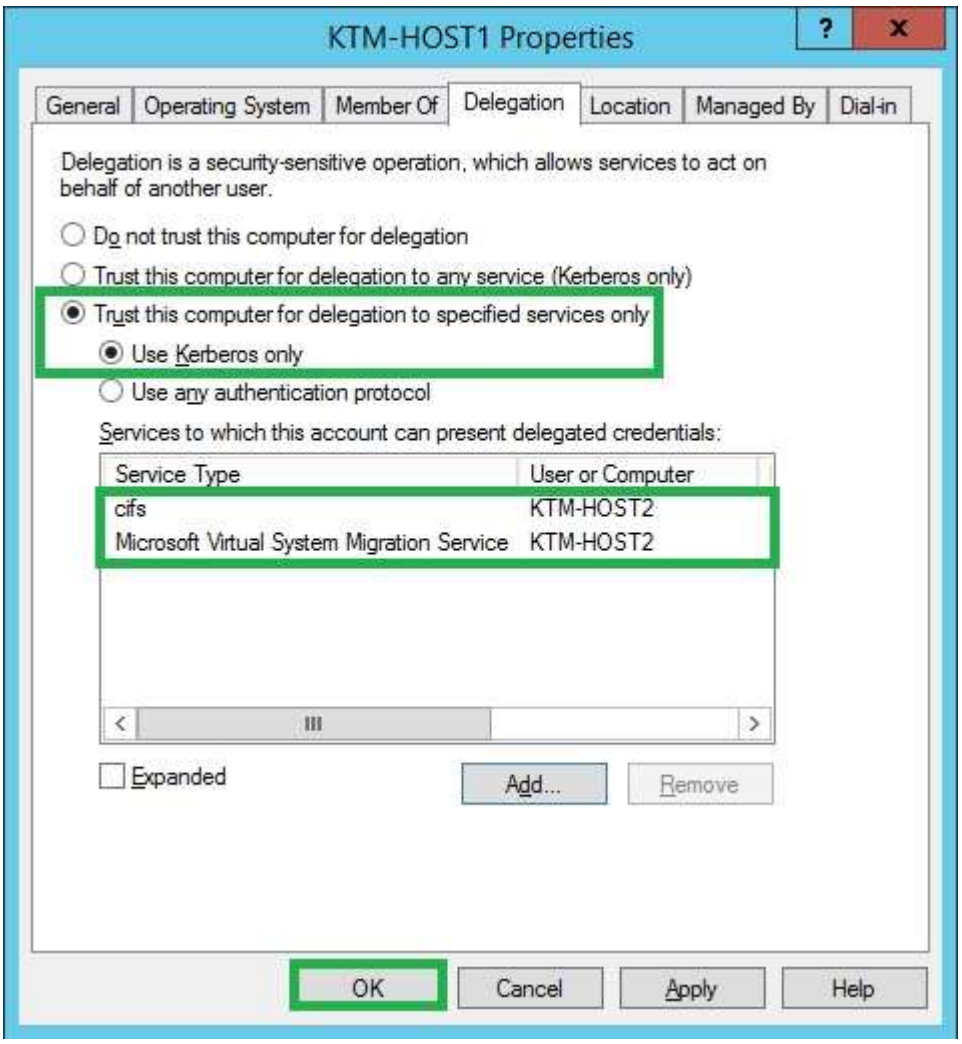
4. In **Select Users or Computers**, in the **Enter the object names to select** field, type **KTM-HOST2**, and then click **OK.**

5. In **Add Services**, select both **cifs** and **Microsoft Virtual System Migration Services service** types, and then click **OK**.

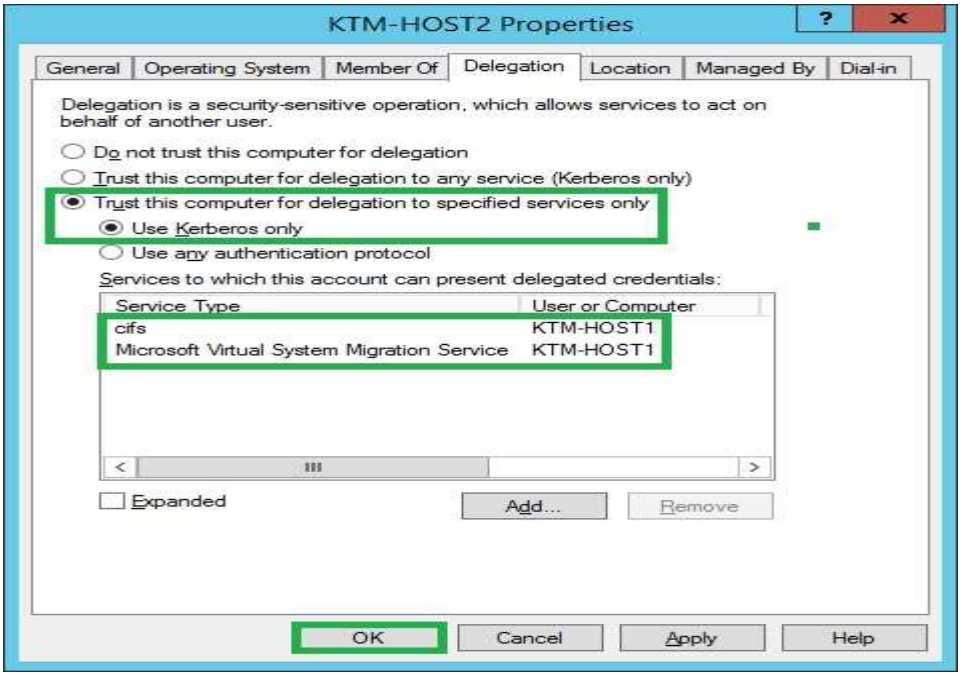

Add Services
To allow services to be delegated for a user or computer, select the appropriate users or computers, and then click the services.
To select one or more user or computer names, click Users or Computers.
Users or Computers...
Available services:
Service Type
User or Computer
appmgmt KTM-HOST2
browser KTM-HOST2
cifs KTM-HOST2
cisvc KTM-HOST2
clipsrv KTM-HOST2
dcom KTM-HOST2
dhcp KTM-HOST2
dmserver KTM-HOST2
Select All
OK
Cancel


Add Services
To allow services to be delegated for a user or computer, select the appropriate users or computers, and then click the services.
To select one or more user or computer names, click Users or Computers.
Users or Computers...
Available services:
Service Type
User or Computer
ias KTM-HOST2
iisadmin KTM-HOST2
mcsvc KTM-HOST2
messenger KTM-HOST2
Microsoft Virtual Console Service KTM-HOST2
Microsoft Virtual System Migration Service KTM-HOST2
msdtc KTM-HOST2
msiserver KTM-HOST2
Select All
OK
Cancel

6. In the **KTM-HOST1 Properties** dialog box, on the **Delegation** tab, confirm that both services are listed and then click **OK**.

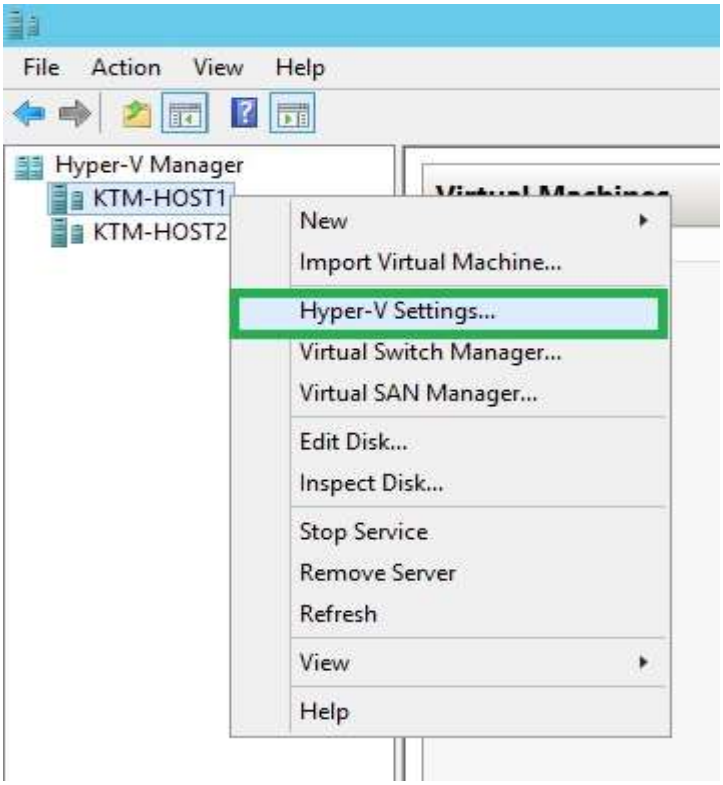


7. Do the same process on **KTM-HOST2**. Here, In **Select Users and Computers**, in the **Enter the object names to select** field, type **KTM-HOST2**, and then click **OK**. In the **KTM-HOST2 Properties** dialog box, on the **Delegation** tab, confirm that both services are listed and then click **OK**.

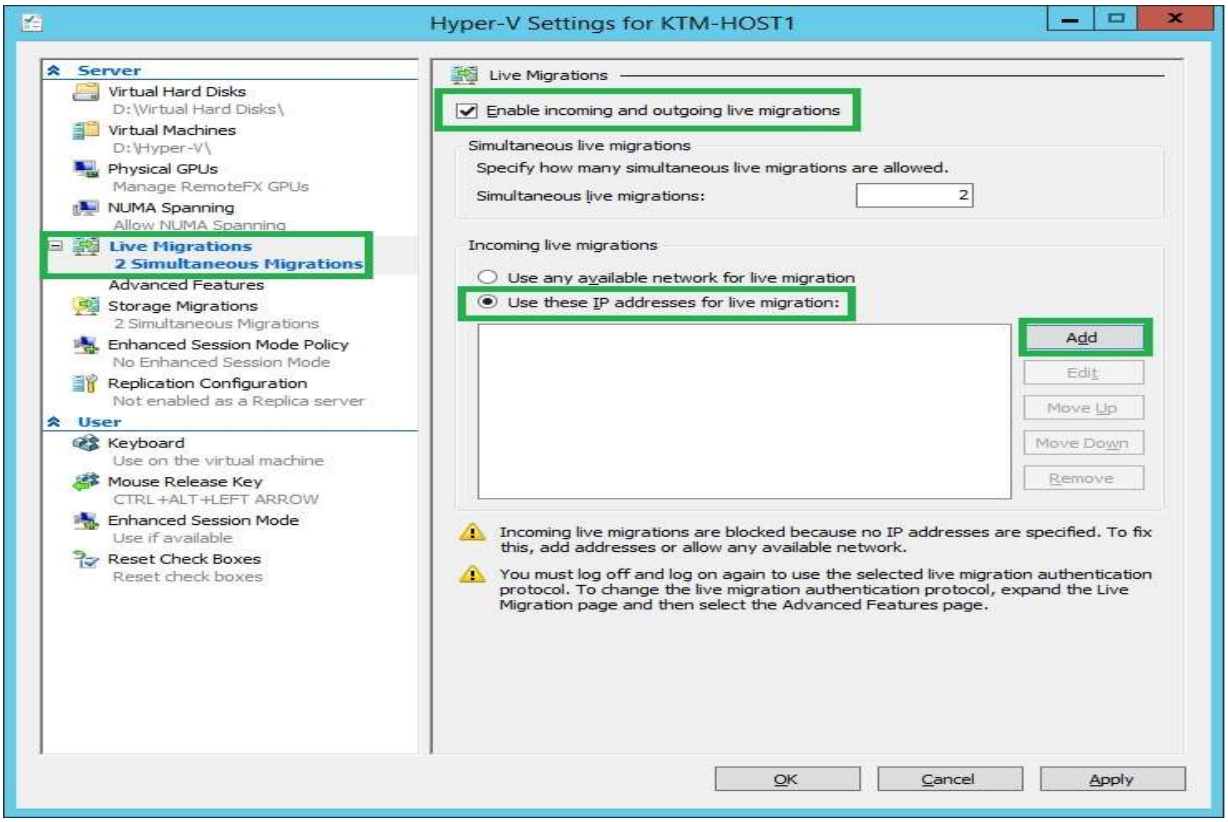


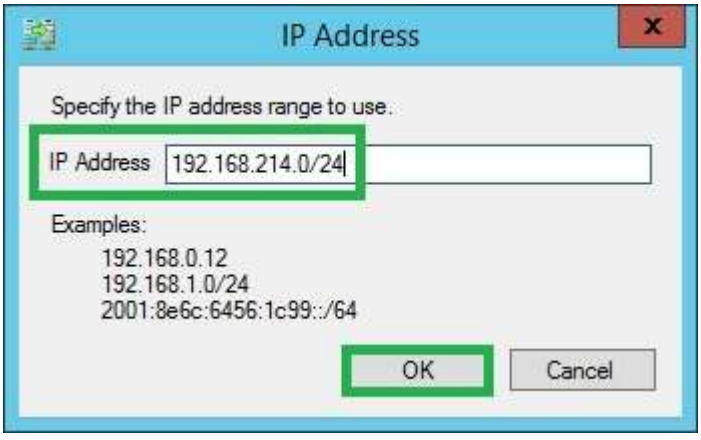
Task 2: Configure Live Migrations Hyper-V Host settings:

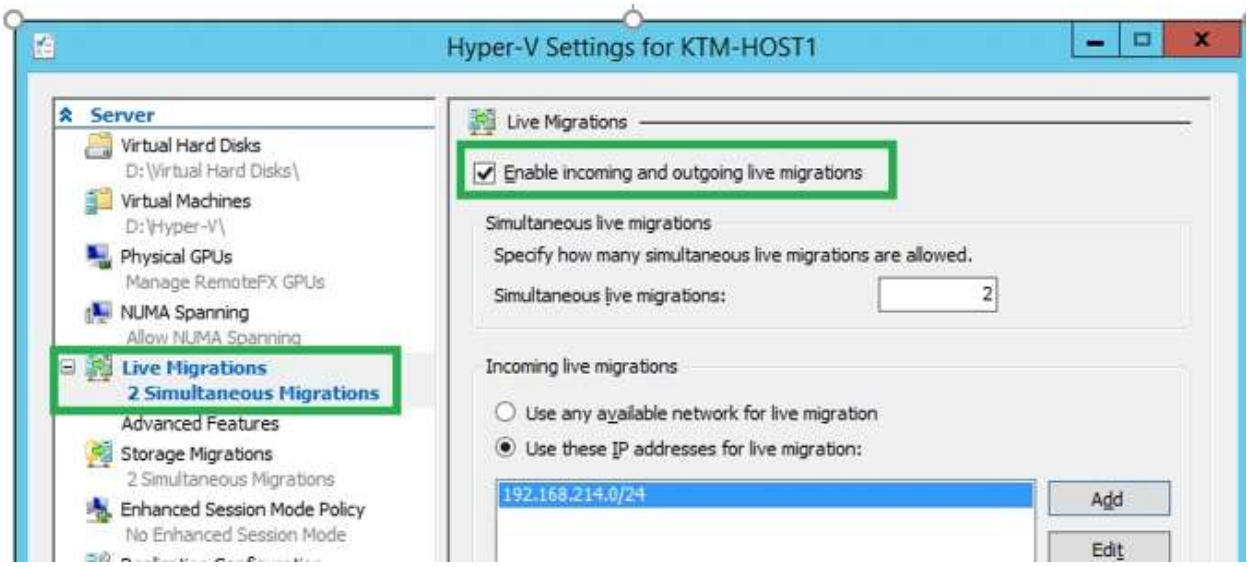
1. In Hyper-V Manager, right-click **KTM-HOST1**, Click **Hyper-V Settings**.



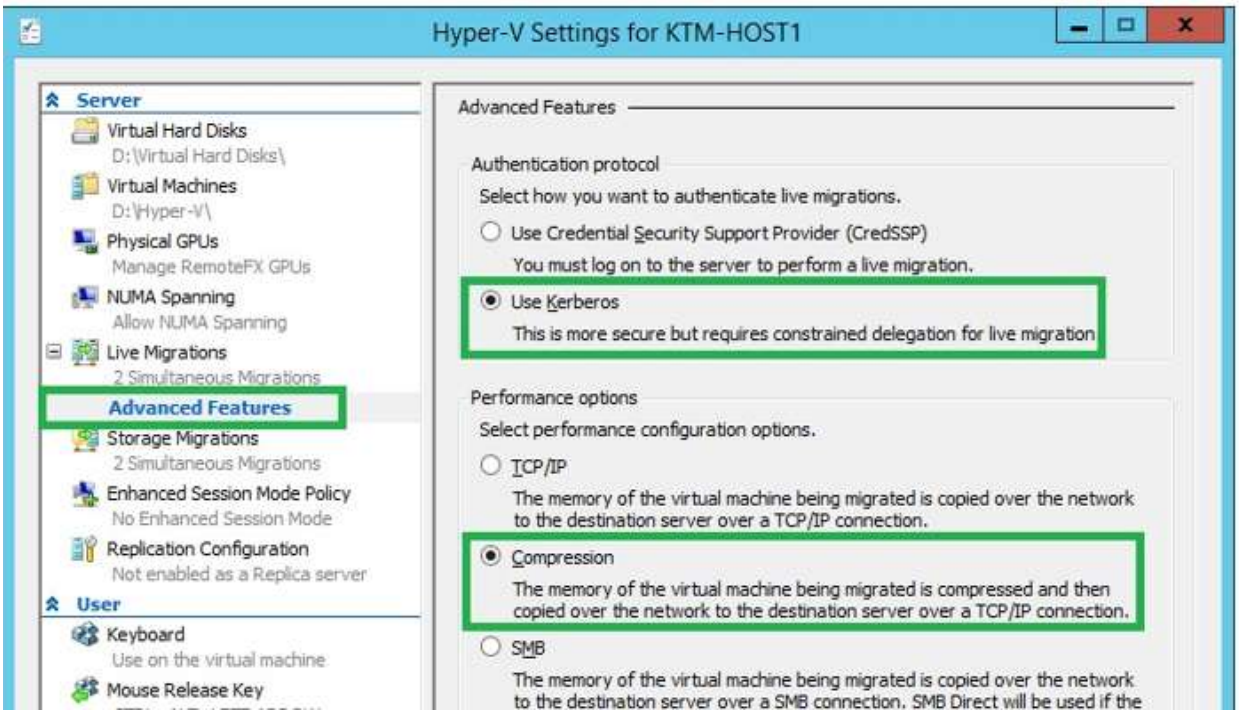
2. In Hyper-V Settings, in the left pane, click **Live Migrations**. In the right pane, in the **Live Migrations** section, click **Enable incoming and outgoing live migrations.** Select the option for Incoming live migration to **Use these IP addresses for live migration.** Click **Add.** In the IP Address dialog box, **specify the IP address range** to use and click **OK**. Click **Apply** and Click **OK**.





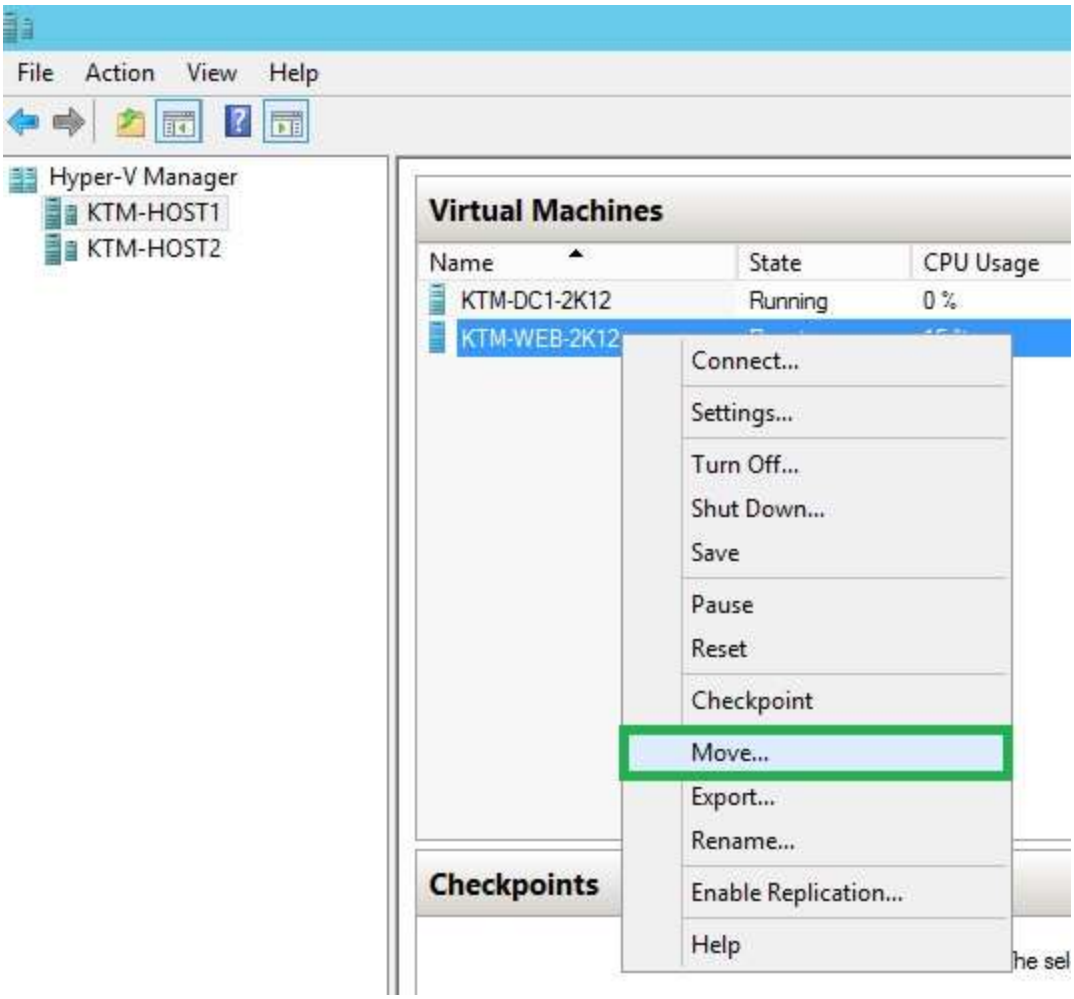


3. In the Hyper-V Settings, expand **Live Migrations**, click Advanced Features, and select **Use Kerberos** to be used as Authentication Protocol and select **Compression** to be used as Performance Options. Then Click **OK**.

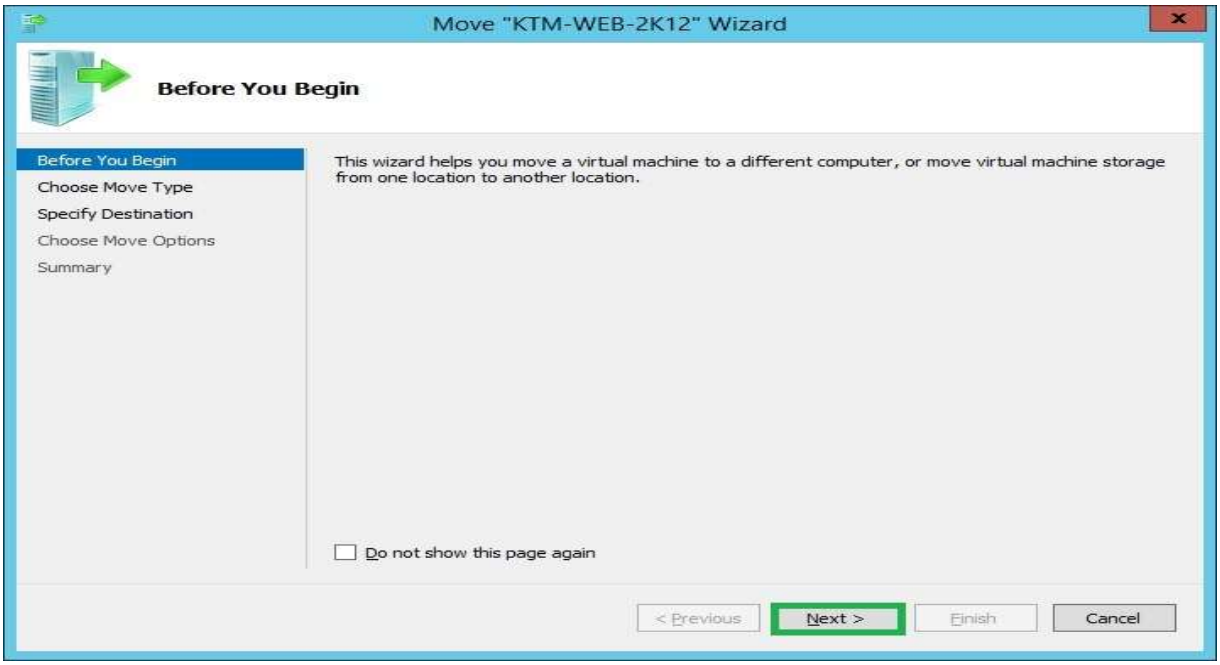


Task 3: Performing Live Migration Move a Virtual Machine:

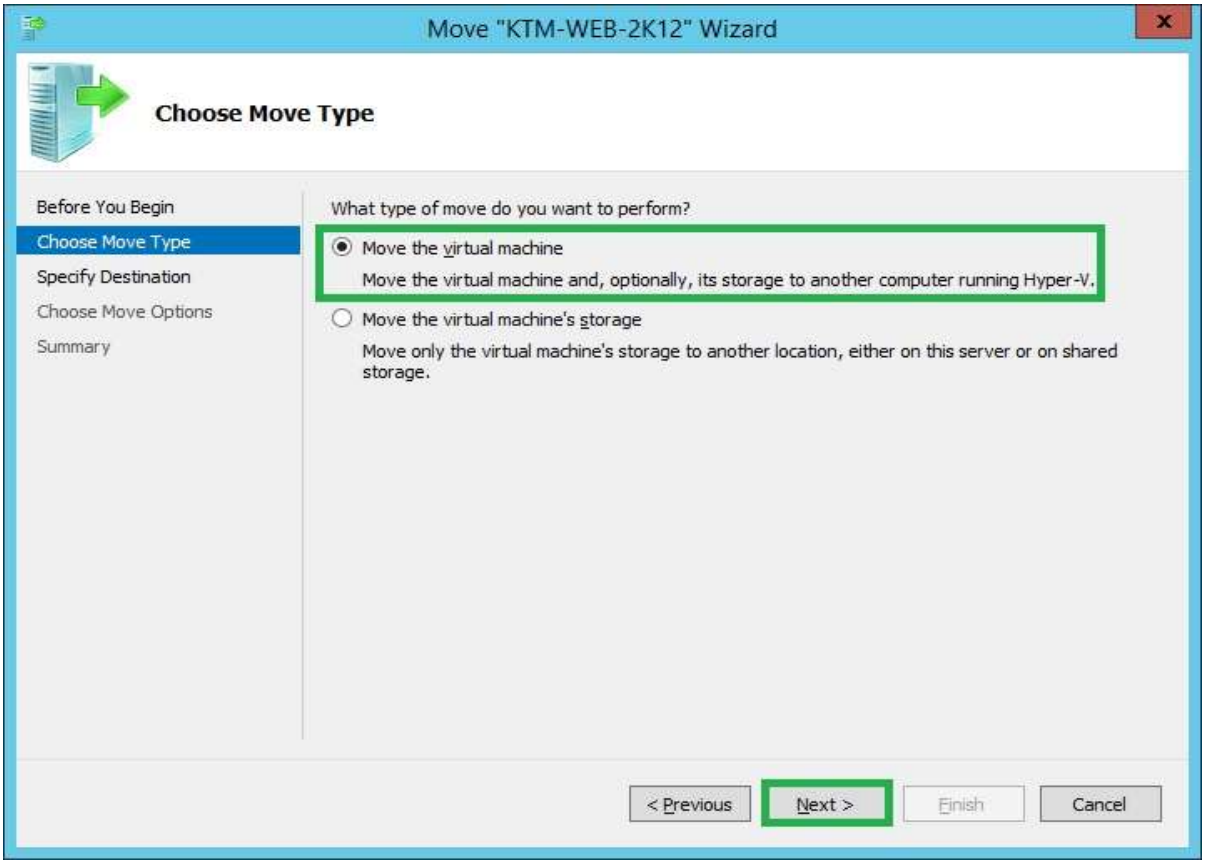
1. On **KTM-HOST1**, In Hyper-V Manager, right-click **KTM-WEB-2K12**, and then click **Move**.



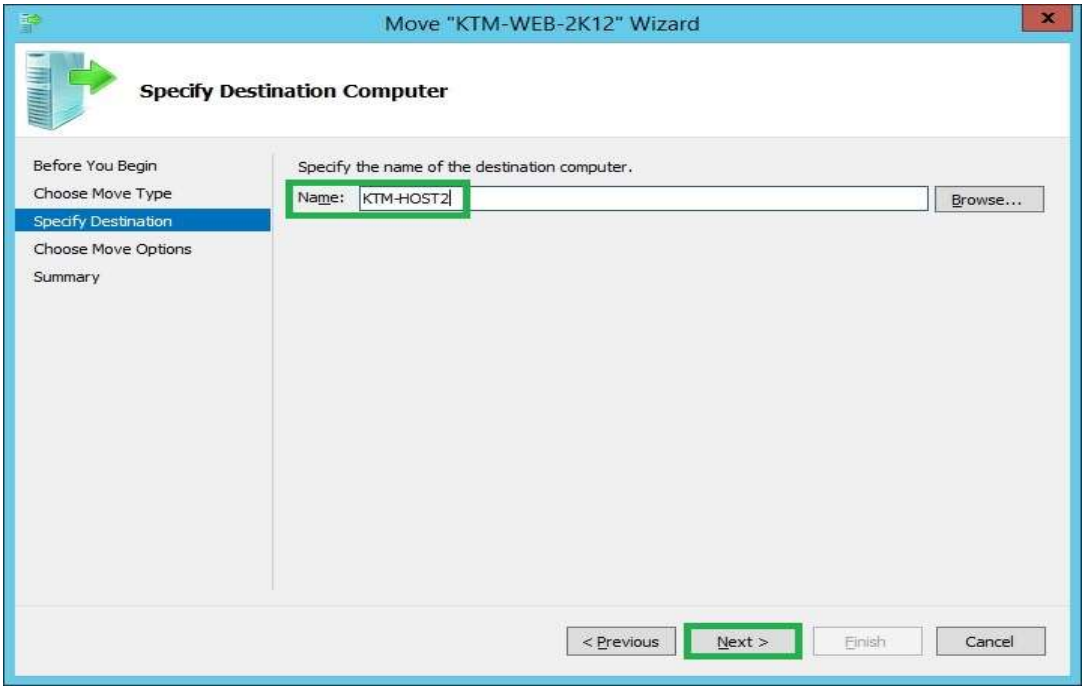
2. In the **Move "KTM-WEB-2K12"** Wizard, On **Before You Begin** page, click **Next**.



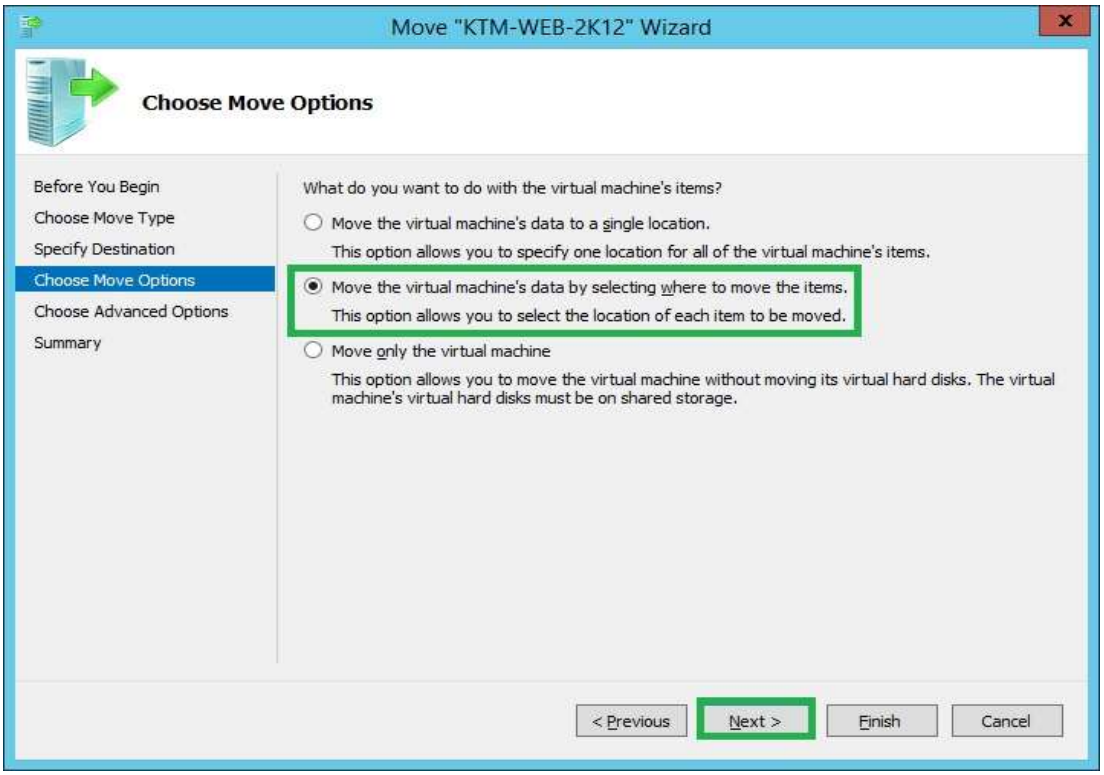
3. On the **Choose Move Type** page, select **Move the virtual machine** and then click **Next**.



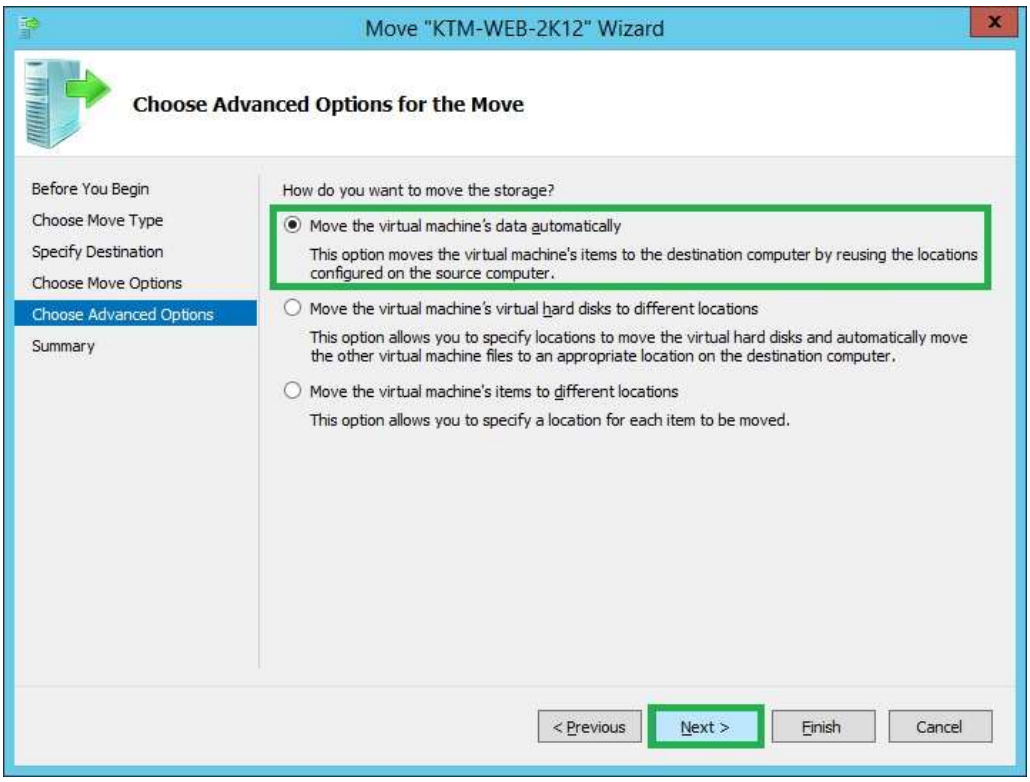
4. On the **Specify Destination Computer** page, in the **Name** field, type **KTM-HOST2**, and then click **Next**.



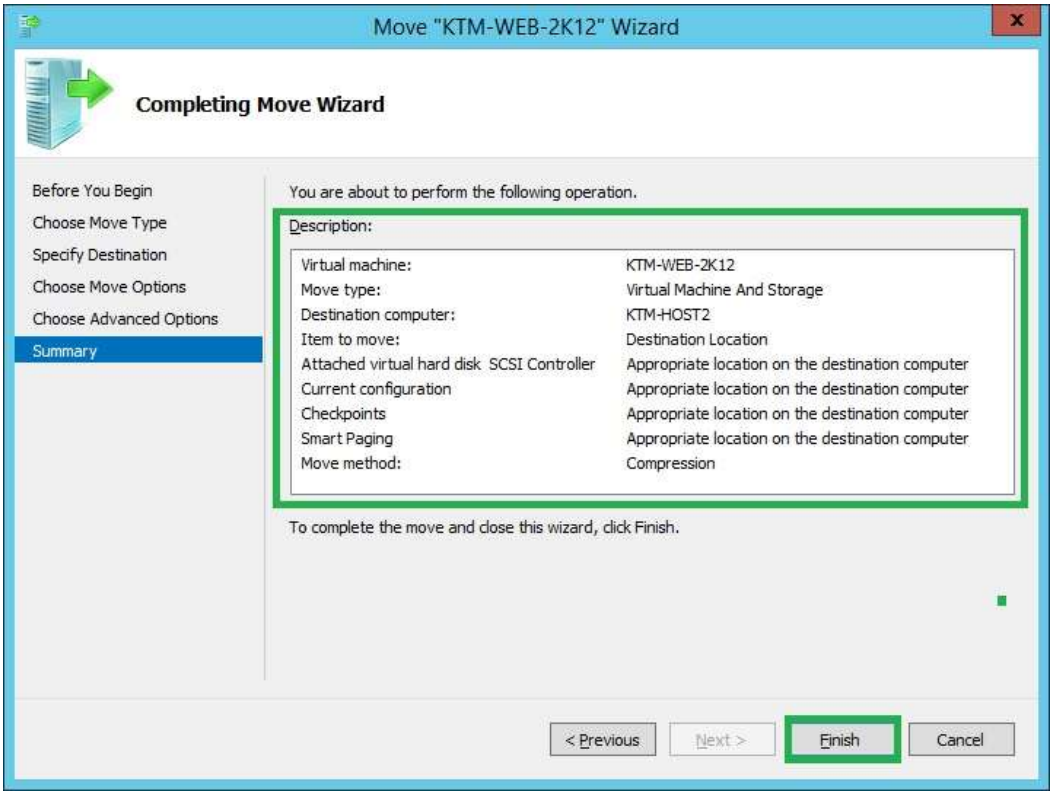
5. On the **Choose Move Options** page, select **Move the virtual machine's data by selecting where to move the items** and then click **Next**.



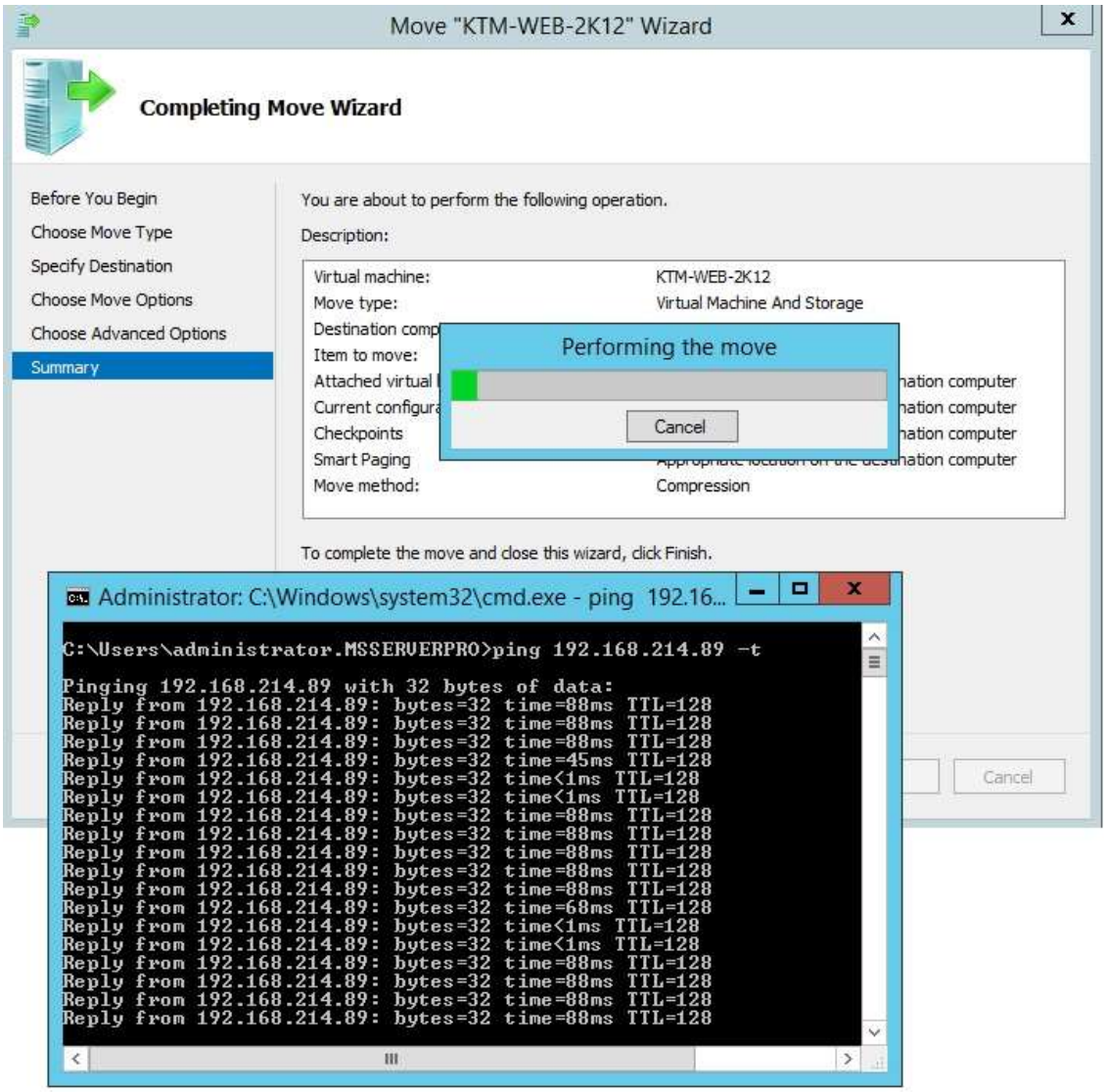
6. On the **Choose Advanced Options for the Move** page, select **Move the virtual machine's data automatically** and then click **Next**.

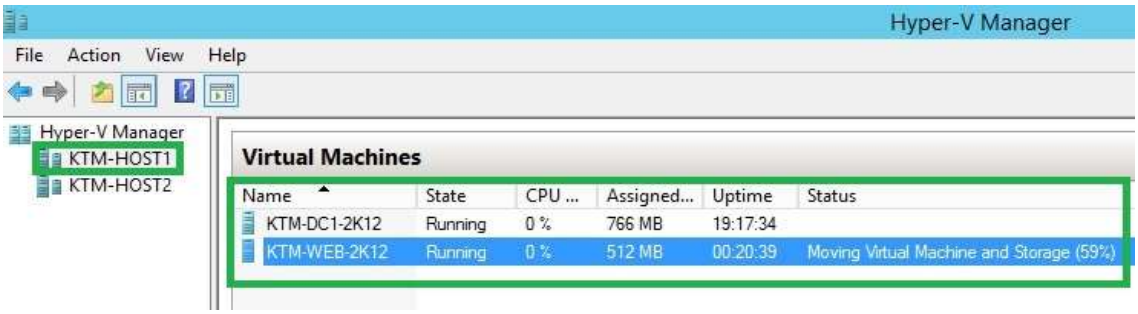


7. On the **Completing Move Wizard** page, Click **Finish**.



8.In the **Move "KTM-WEB-2K12"** Wizard, On **Completing Move Wizard** page, Performing the live migration. You can monitor the progress of the live migration in Hyper-V Manager, in the **Status column**. During the live migration, you can also able to **ping KTM-WEB-2K12**.





9. After **live migration completes**, in Hyper-V Manager, confirm that **KTM-WEB-2K12** is **no longer** running on **KTM-HOST1.**

Shared Nothing Live Migration Using PowerShell

Shared Nothing Live Migration is enabled by using Move-VM. It works almost exactly like Move-VMStorage, but adds a DestinationHost parameter. In fact, everything you learned from Move-VMStorage above (in the Storage Live Migrations Using PowerShell section) applies to Move-VM.

Move-VM svtest -ComputerName svhv2 -DestinationHost svhv1 -VirtualMachinePath C:\ClusterStorage\CSV1 -Vhds @(@{'SourceFilePath' = '\\svstore\vms\Virtual Hard Disks\svtest.vhdx'; 'DestinationFilePath' = 'C:\ClusterStorage\CSV1\Virtual Hard Disks\svtest.vhdx' }

Shared-nothing live migration is the most extreme type of zero-downtime migration. However, there are other types of zero-downtime migration, such as storing VMs on an SMB file share that both Hyper-V hosts can access. If you're moving a VM between failover clusters or into or out of a failover cluster from a standalone Hyper-V host, you'll need to remove the VM from the cluster before migrating the VM