

Deploying Hyper-V on Oracle Cloud Infrastructure

ORACLE WHITE PAPER | NOVEMBER 2018



Disclaimer

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Revision History

The following revisions have been made to this white paper since its initial publication:

Date	Revision
November 9, 2018	<ul style="list-style-type: none">• Added information about configuring guests that can use vNIC directly.• Made distinctions between guests that directly use vNICs (direct access) and guests that use the helper guests (indirect access).• Removed requirements to configure a second physical NIC on a Hyper-V host.• Added information that enables guests using the direct access method to communicate with each other using the SRIOV bridge. This includes the hvnat and hvrouter guests, if installed.• Removed the requirement for a second VCN subnet for hvnat/hvrouter connections to the VCN.• Removed the requirement for a Broadcom driver for hvnat/hvrouter guests.• Updated diagrams to reflect new information.• Revised the name of the paper.
August 3, 2018	<ul style="list-style-type: none">• Updated introduction to include background information about Hyper-V.• Added an appendix illustrating the implementation of Hyper-V Replica within Oracle Cloud Infrastructure.
April 20, 2018	<ul style="list-style-type: none">• Updated diagrams to be more readable.• Added an internal network adapter based on the Microsoft Loopback Adapter.• Added instructions that indicate that second subnets should be used for the Hyper-V access to the VCN (in support of eventual SCVMM, Hyper-V Replica, and Hyper-V Clustering deployment guides).• Added instructions for installing the Microsoft Loopback Adapter.• Added instructions and information about guest OS licensing and procedures.

You can find the most recent versions of the Oracle Cloud Infrastructure white papers at <https://cloud.oracle.com/iaas/technical-resources>.



Table of Contents

Overview	5
Purpose of This Paper	6
Guest Deployment Methods	6
Direct Access	6
Indirect Access	8
Prerequisites	12
Set Up the Oracle Cloud Infrastructure VCN and Deploy the Bare Metal Instance for Hyper-V	12
Configure Windows Server, Hyper-V, and Supporting Network Services	15
Install Hyper-V	15
Configure Hyper-V	16
Configure Indirect Access	16
Install and Configure Windows DHCP, DNS, and the Microsoft Loopback Adapter	16
Create the Internal Network for the Hyper-V Guests	21
Configure DNS and DHCP for the Hyper-V Guests	22
Create the Helper Hyper-V Guests	25
Install and Configure Windows for the hvnat and hvrouter Hyper-V guests	29
Configure the hvnat Guest	30
Configure the hvrouter Guest	33
Install Guests	33
Windows Automatic Virtual Machine Activation	33
Direct Access Guests	34
Indirect Access Guests	36
Enable Connections Between Guests on Different Hyper-V Servers	37



Conclusion	38
Appendix A: Deploying Hyper-V Replica	39
Prerequisites	39
Configure the Hyper-V Target Instance	40
Configure the Hyper-V Guests	43
Limitations of Guest Replication from On-Premises to Oracle Cloud Infrastructure	43
Configure a Hyper-V Guest	44
Fail Over Between Source and Target Hyper-V Instances	48



Overview

Oracle Cloud Infrastructure provides a robust, highly configurable way of deploying individual guest instances that are highly flexible, have a range of different shape configurations, and provide a well-defined CPU-to-RAM relationship that meets the needs of most customers. When you are considering the deployment of operating system platforms, using either bare metal or virtual instances within Oracle Cloud Infrastructure is considered the best method of achieving operational excellence for a particular application stack.

In certain situations, however, it's not possible to use the capabilities of the various instance types within Oracle Cloud Infrastructure. Following are some of these situations:

- Instances or applications that require a specific relationship between the number of CPUs and the amount of RAM that is different than the ratios provided by the standard virtual instances
- Requirements for legacy or other operating systems that aren't provided by Oracle Cloud Infrastructure and can't be run in a virtual instance
- Operating systems that aren't contained within an existing hypervisor, or provided in a format that is incompatible with the Oracle Cloud Infrastructure import image process
- Operational requirements to maintain existing operational standards, tools, and standards of execution

These situations and others might require the use of a customer-installed hypervisor to provide the necessary environment in which applications can run within the cloud and gain the associated benefits. To support these situations, Oracle Cloud Infrastructure supports the installation of various hypervisors: Oracle VM, KVM, and Microsoft Hyper-V.

Microsoft Hyper-V is the hypervisor of choice within many environments. It provides the ability to leverage the Windows Administrator experience found in many organizations, but it also provides a method for running disparate operating systems on a common hardware platform. Many toolsets and operational practices have been developed for the use of Hyper-V in the on-premises environment, and the process illustrated in this paper extends those practices to the Oracle Cloud Infrastructure environment.



Purpose of This Paper

This paper describes how to deploy Hyper-V in Oracle Cloud Infrastructure. The deployment method differs from the method used on-premises, but it results in a configuration that is compatible with standard on-premises management tools, such as Microsoft SCVMM.

Additionally, this paper describes two different methods for deploying guests: direct and indirect. Guests can be deployed using either method on a single Hyper-V host, depending on the requirements of the guest. Rationales for the selection of method are provided in the description of each method. You don't have to select the guest deployment method before you deploy Hyper-V itself, but you should identify it for each guest after the basic Hyper-V deployment is complete.

Finally, this paper covers a basic deployment of Hyper-V Replica—a method of replicating live guests between different Hyper-V hosts—in a limited configuration. As detailed in the description, the deployment of Hyper-V Replica has a specific set of limitations and can be applied only to guests that have been deployed using the indirect method of deployment.

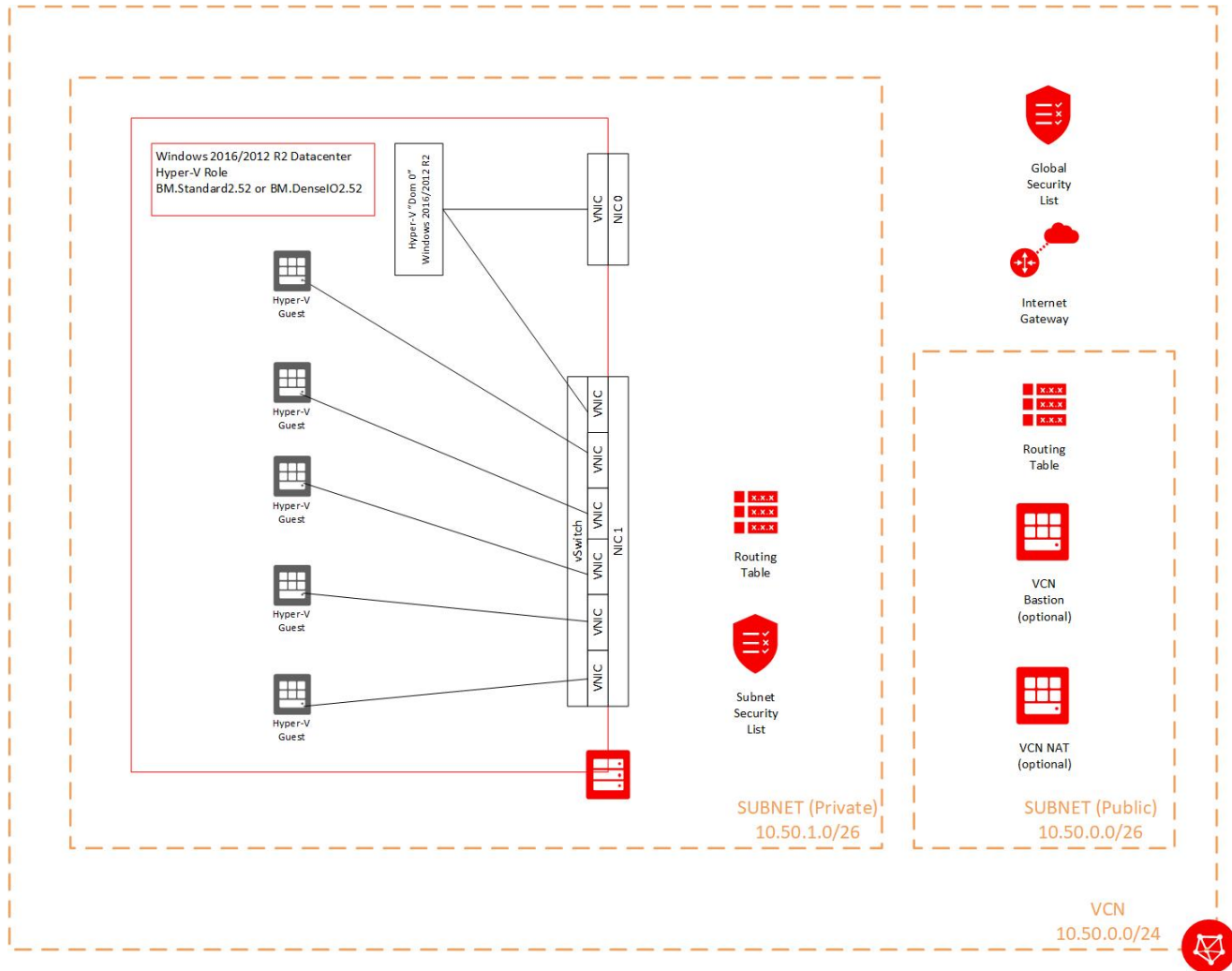
Guest Deployment Methods

Two methods for deploying guests are currently possible when you deploy Hyper-V. Using either method, combined with services provided by Oracle Cloud Infrastructure, Windows Server, and Hyper-V, creates an environment that seamlessly gives guests running on Hyper-V the ability to interact with instances and services provided by Oracle Cloud Infrastructure, and gives guests running on Oracle Cloud Infrastructure the ability to interact with instances and services provided by Hyper-V.

Direct Access


The direct access method assigns a discrete vNIC to each Hyper-V guest. The guest is granted an IP address on the subnet to which the vNIC is attached and must use the MAC address of the vNIC as its own. Guests must be manually configured with their IP address information outside of the deployment process.

The following diagram illustrates an example architecture associated with the direct access method:



The architecture has the following elements:

- A VCN with a CIDR of 10.50.0.0/24
- A private subnet with CIDR 10.50.1.0/26
- A public subnet with CIDR 10.50.0.0/26 (optional)
- A bare metal Oracle Cloud Infrastructure Compute instance running Windows 2016 Datacenter, with the Hyper-V role installed
- An internet gateway, local and global security lists, and a route table that supports the VCN

- 
- A NAT gateway and bastion host for private subnets, located on a public subnet (optional)

The bare metal instance running Hyper-V has the following elements:

- A vNIC provisioned and assigned to the second physical NIC (NIC1)
- External vSwitch with single root I/O virtualization (SR-IOV) enabled that uses NIC1

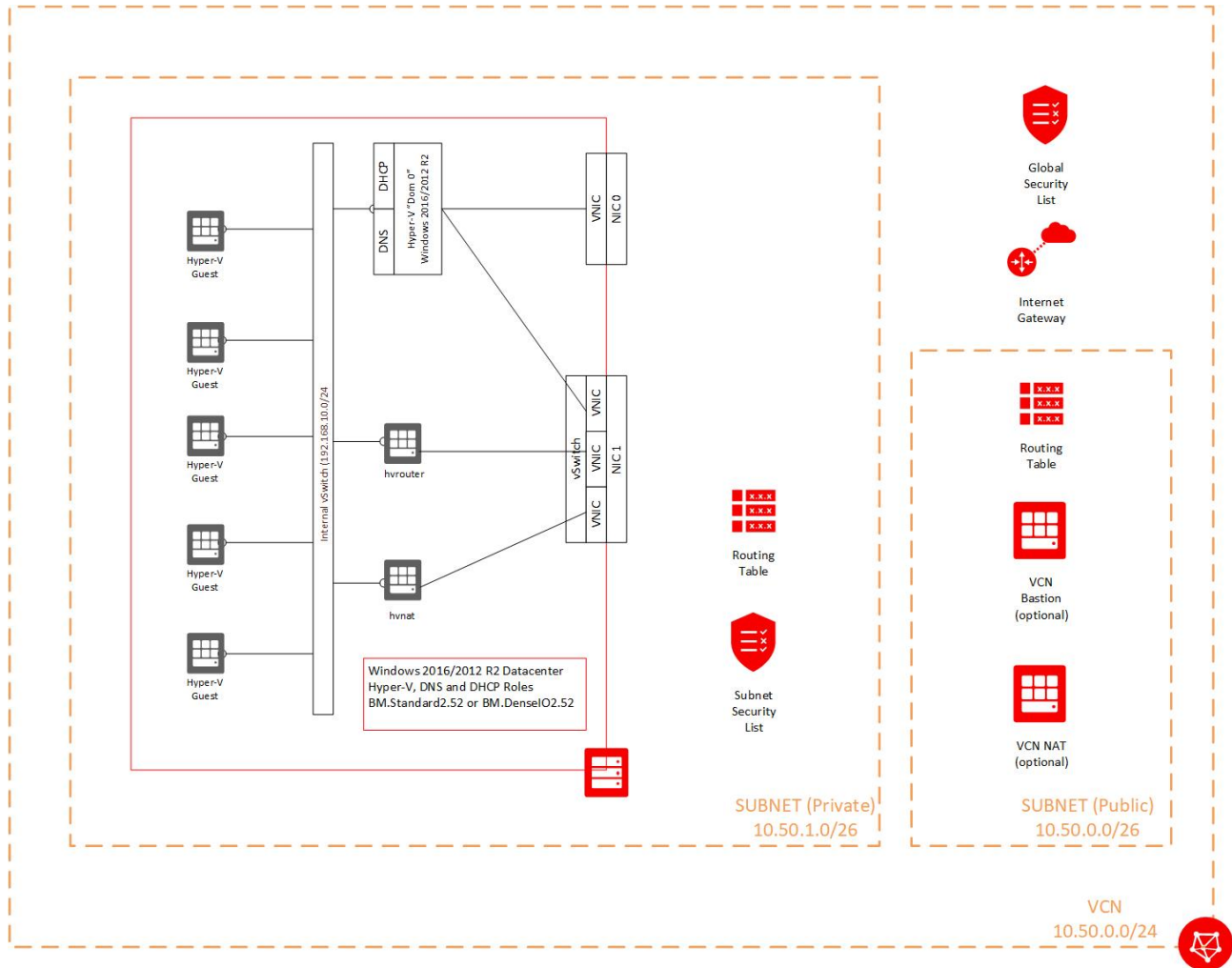
The advantage of direct access is the ability of guests to access Oracle Cloud Infrastructure resources directly, without passing through “helper” guests, which might limit performance. Each guest uses an individual connection, using SR-IOV, to the VCN subnet and is granted the highest possible performance through the use of the shared hardware connection. Guests can also be accessed directly from on-premises via private connections (FastConnect and VPN access to the VCN), or via the internet via public IP addresses.

However, there are some limitations to consider. The number of direct access guests that can be deployed on a single Hyper-V host is currently limited to 24 guests. If you need to deploy more than 24 guests, you can’t use this method. In addition, guests can’t be migrated by using the Hyper-V Replica method detailed later in this paper. The current implementation of the vNIC technology doesn’t allow vNICs and their associated IP addresses to be ported between Hyper-V hosts, which restricts the ability to create target guests as part of the Replica process.

Indirect Access


The indirect access method was detailed in earlier versions of this paper. This method uses two “helper” Hyper-V guests, in conjunction with the VCN route-to-IP feature, to allow traffic to be routed in and out of a private subnet located within the Hyper-V host itself. Guests are provided with IP addresses via a private DHCP server, addresses that aren’t part of the VCN address space, and use locally generated MAC addresses for their configuration.

The following diagram shows an example architecture of the indirect access method:



The architecture has the following elements:

- A VCN with a CIDR of 10.50.0.0/24
- A private subnet within the VCN with CIDR 10.50.1.0/26
- A public subnet within the VCN with CIDR 10.50.0.0/26 (optional)
- A bare metal Oracle Cloud Infrastructure Compute instance running Windows 2016 Datacenter, with the Hyper-V role installed
- An internet gateway, local and global security lists, and a route table that support the VCN

- 
- A NAT gateway and bastion host for private subnets, located on a public subnet (optional)
 - Representative resources on the subnet that the guests within Hyper-V can access

The bare metal instance running Hyper-V has the following elements:

- External type vSwitch with single root I/O virtualization (SR-IOV) enabled
- External type vSwitch for the guests (labeled as “Internal”), which does not require SR-IOV enablement
- Two special-purpose Hyper-V helper guests to act as a NAT gateway and router for the internal guests (hvnat/hvrouter)
- A DHCP server running on the Windows 2016 Hyper-V “Dom0”, bound to the internal vSwitch, providing address/network information to the Hyper-V guests
- A DNS server running on the Windows 2016 Hyper-V “Dom0”, bound to the internal vSwitch, providing name resolution services to the Hyper-V guests

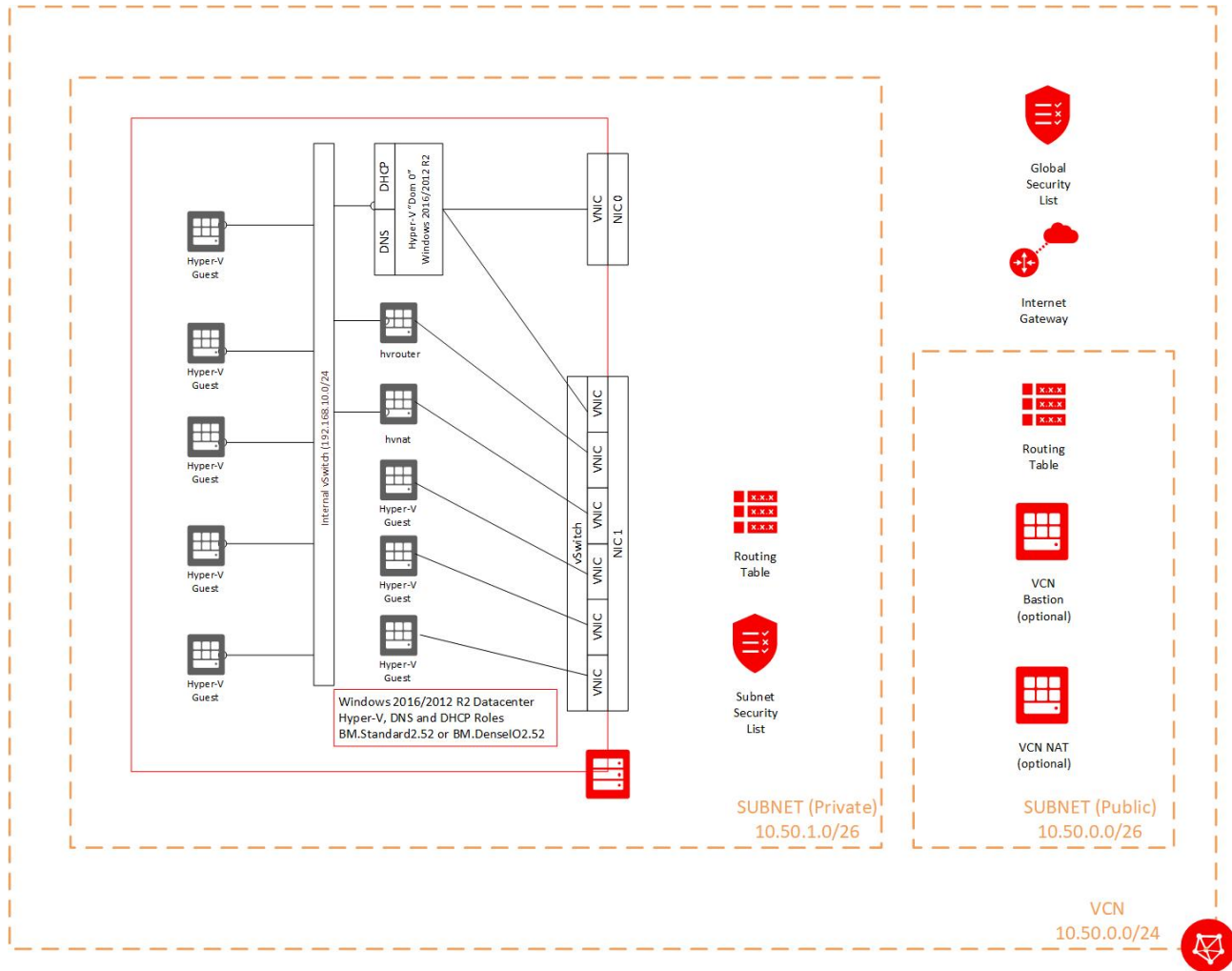
This method has several advantages:

- Deploying guests in this environment is relatively simple. Because internal DHCP and DNS servers are provided as part of the method, guests can be installed in the environment with a minimum of interaction. IP addresses, routes, and DNS information is provided as part of the guest deployment process for guests that are configured for DHCP.
- The number of allowed guests is significantly higher than with the direct access method. Because there is no limitation based on vNIC count, and depending on the configuration of the guests, you can deploy hundreds of guests within a single Hyper-V host.
- Guests can be migrated between Hyper-V hosts by using the Hyper-V Replica method.

However, the advantages of the direct access method become the disadvantages of the indirect access method:

- The network performance of the guests is limited by the ability of the two “helper” guests to efficiently route traffic between the private Hyper-V subnet and the VCN subnet.
- Although guests within the Hyper-V host using this method can get out to external resources and targets, there currently is no method to reach the guests inside Hyper-V from sources outside of the local VCN. If you use the indirect method to host guests that need to be accessed from on-premises networks, you must deploy a terminal server, proxy, or other type of “jump host” within the VCN to reach the guests.

The methods can be intermixed. For example, if you have a series of instances that need to be accessed by non-VCN based locations, you can create those instances by using direct access. Then, on the same Hyper-V host, you can create the higher-density instances required for application purposes and have them access or be accessed via the helper guests. This setup is shown in the following diagram:



Prerequisites

Before you begin the deployment, have the following items ready:

- Have at least one public subnet available, or be connected to a private subnet with a NAT gateway. If you want your guests to connect to the internet, the secondary VNICs used for routing must also be on a public subnet, or be connected to a private subnet with a NAT gateway.
 - Instructions for configuring a NAT gateway in Oracle Cloud Infrastructure are located in the [NAT Gateway topic](#) in the documentation.
 - The subnet used for the secondary VNIC should be on a separate subnet from the primary interface if clustering is to be used in the future. If the installation is a standalone installation, then the primary and secondary subnets can be colocated on the same subnet.
- (*Indirect access only*) Select an IP range, not part of the VCN IP range, that you want to use for the internal Hyper-V network. Identify the first three addresses of the IP range for use by Hyper-V:
 - Default gateway (hvnat)
 - VCN gateway (hvrouter)
 - DNS/DHCP

Set Up the Oracle Cloud Infrastructure VCN and Deploy the Bare Metal Instance for Hyper-V

The first two steps apply to both the direct access and the indirect access method.

1. Create a V2 (BM 2.x type) bare metal instance, and attach at least one block storage volume to it. For instructions, see the following topics in the Oracle Cloud Infrastructure documentation:
 - [Creating an Instance](#)
 - [Overview of Block Volume](#)
2. After the instance is provisioned, provision a VNIC for the secondary NIC (NIC 1) on the bare metal instance, preferably in a different subnet than that used for the bare metal instance. For instructions on how to deploy secondary VNICs, see the [Virtual Network](#)

[Interface Cards \(VNICs\) topic](#) in the Oracle Cloud Infrastructure Networking service documentation.

PHYSICAL NIC

NIC 1 (3 VNICs allocated)

Skip Source/Destination Check

The source/destination check causes this VNIC to drop any network traffic whose source or destination is not this VNIC. Only check the checkbox if you want this VNIC

Do *not* select the **Skip Source/Destination Check** option. This VNIC can be on either a public or private subnet. If it's on a private subnet, we recommend that you configure a NAT gateway as noted in the "Prerequisites" section. We also recommend manually assigning the private IP address in order to maintain consistency. Note the private IP address of the NIC.

If you are using indirect access, complete the following steps. If you are using the direct access method, skip to the next section.

3. Create two secondary VNICs for the secondary NIC (NIC 1).

NAME (Optional)

hvnat

VIRTUAL CLOUD NETWORK

c4-vcn1

SUBNET

c4-vcn1-ad3-sn3

PHYSICAL NIC

NIC 1 (3 VNICs allocated)

Skip Source/Destination Check

The source/destination check causes this VNIC to drop any network traffic whose

Be sure to *select* the **Skip Source/Destination Check** option for both VNICs created in this step. The VNICs can be on either a public or private subnet, but should be either on the same subnet as the one assigned to the second physical NIC, or one that is different than that used for the primary physical NIC. If it's on a private subnet, we recommend that you configure a NAT gateway as noted in the "Prerequisites" section. We also recommend that you manually assign the private IP addresses, in order to maintain consistency, but it's not required. Note the MAC and private IP addresses, and the VLAN ID of these secondary VNICs.

- Select one of the secondary VNICs as the route target (hvrouter), and note the IP address of the selected target.
- Open the route table of the VCN where the Hyper-V server is being configured, and add a route rule. Enter the private IP address that you just selected as the route target for the IP address range selected for the Hyper-V deployment.

For example, 192.168.11.0/24 was selected as the address space for Hyper-V. A secondary VNIC on the Hyper-V server was selected with an IP address of 10.50.0.69. The resulting route table entry would look as follows:

For information about managing route tables, see the [Route Tables topic](#) in the Networking service documentation.

- Open the security list for the subnet. If you have configured a global security list for the VCN, consider using that security list instead.
- Add an entry that allows traffic from the Hyper-V internal network to be accepted by instances within the VCN. For example, if the VCN has a global security list that covers the entire VCN, the entry added to the global security list would look as follows:

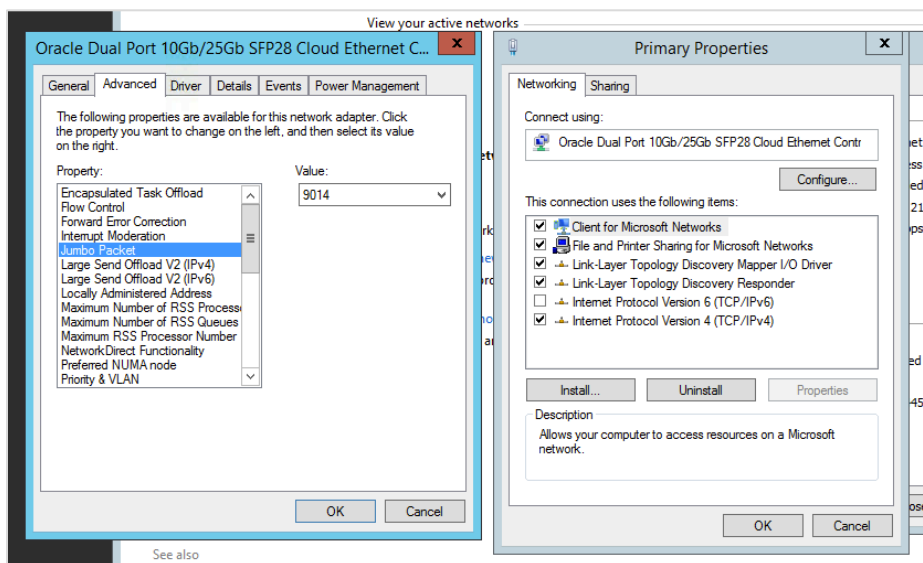
For information about how to manage and update security lists, see the [Security Lists topic](#) in the Networking service documentation.

Configure Windows Server, Hyper-V, and Supporting Network Services

This section provides instructions for installing and configuring Hyper-V, configuring DNS and DHCP for Hyper-V guests, creating the initial Hyper-V guests, and installing and configuring Windows for the Hyper-V guests.

Install Hyper-V

1. Log in to the provisioned Windows bare metal instance.
2. In Windows Firewall, enable ICMP for the bare metal instance. The firewall rule for ICMP is labeled `File and Printer Sharing (Echo Request - ICMPv4-In)`.
3. Mount and format the block volume on the bare metal instance.
4. Verify that the MTU for each NIC is set to 9014.
 - A. Access the properties of each network adapter.
 - B. Click **Configure**.
 - C. On the **Advanced** tab, ensure that the value of **Jumbo Packet** is 9014.

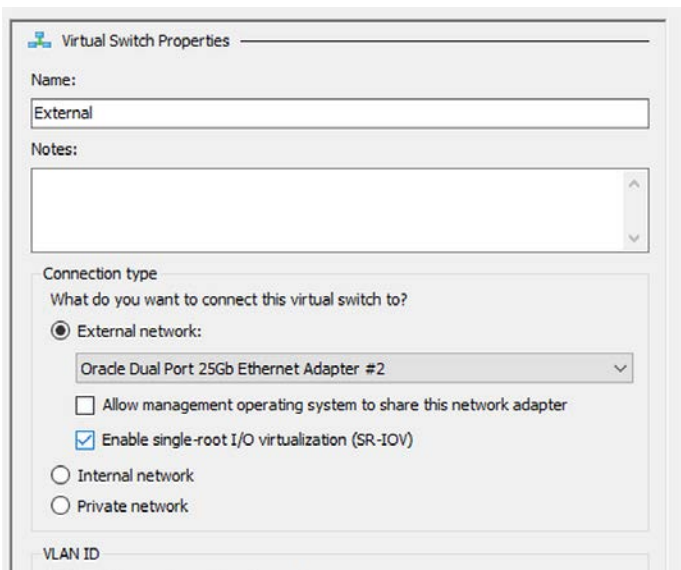


Note: Do *not* restart the system yet.

5. Install the Hyper-V role on the server.
6. *(Direct access only)* Reboot the bare metal instance.

Configure Hyper-V

1. Open the Hyper-V Manager.
2. Open the Virtual Switch Manager and create a new external vSwitch named **External**. Select the **Enable single-root IOV virtualization (SR-IOV)** option only.



Configure Indirect Access

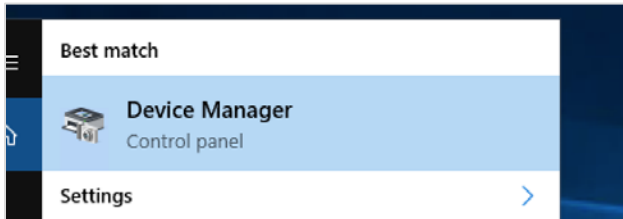
The following steps are required only if you are using the indirect access method to deploy guests within Hyper-V. If you are not using this method, skip to the next section.

Install and Configure Windows DHCP, DNS, and the Microsoft Loopback Adapter

1. Install the following Windows features:
 - DHCP
 - DNS

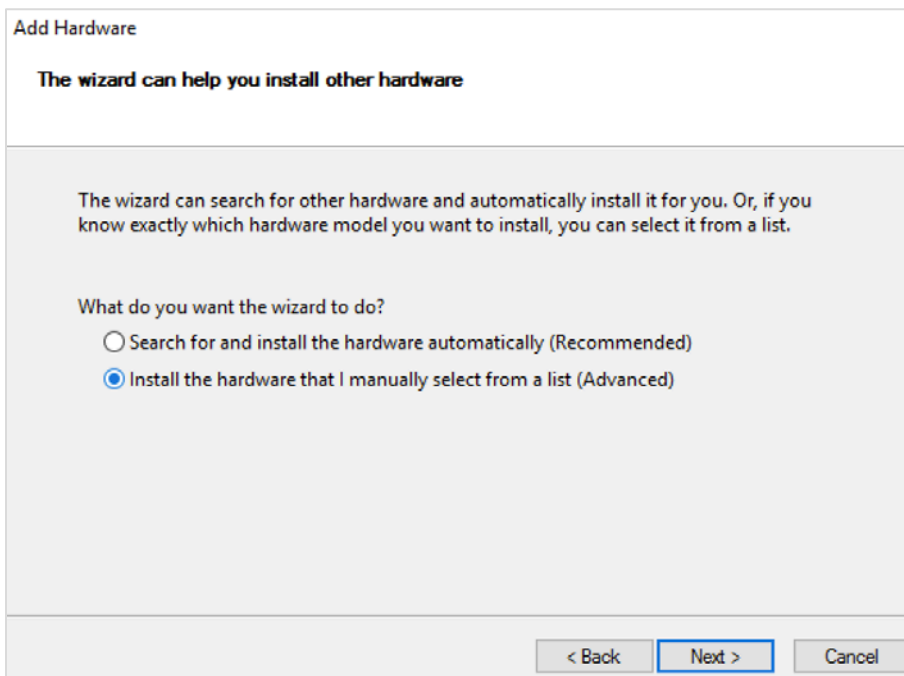
2. Install the Microsoft KM-TEST Loopback Adapter, which is a localized loopback network adapter that will be used for guest connectivity.

- A. Open Device Manager.

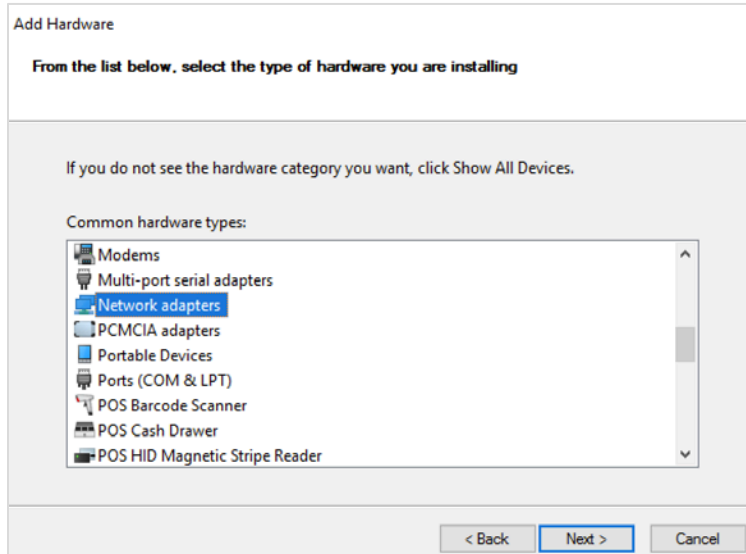


- B. Right-click the server (top of the device tree) and select **Add legacy hardware**.

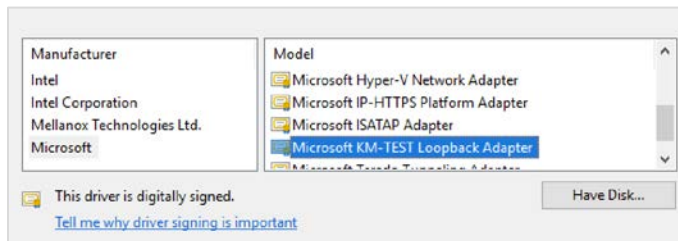
- C. In the wizard, select the **Advanced** option.



D. In the list of options, select **Network adapters**.



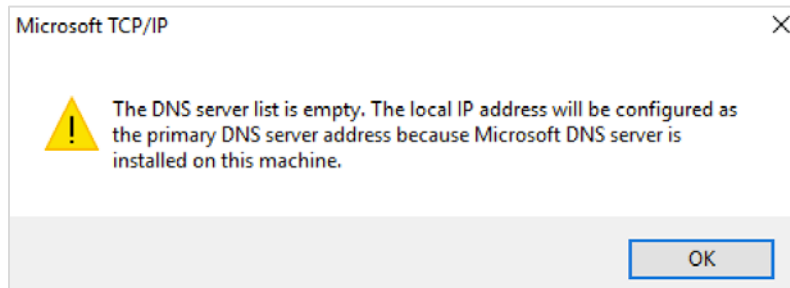
E. In the box on the left, select **Microsoft**, and in the box on the right, select **Microsoft KM-TEST Loopback Adapter**.



F. Finish the installation wizard.

3. Configure the loopback adapter with the IP address that you intend to assign for the DNS/DHCP interface in Hyper-V. Do not assign a default gateway or DNS address at this time.

In the process of performing this configuration, you will get the following warning. You can safely ignore it:



You can perform this installation by using PowerShell. A script to perform these operations might look like the following one:

```
$internalIpBase="<IP address range with CIDR>"
$hvIp="<IP Address of Internal Hyper-V adapter>"

$instance=$(ConvertFrom-Json $(Invoke-WebRequest
("http://169.254.169.254/opc/v1/instance/")).Content)
$vnics=$(ConvertFrom-Json $(Invoke-WebRequest
("http://169.254.169.254/opc/v1/vnics/")).Content)

$secondVnic=""
$primaryVnic=""
$secondAdapter=""
$primaryAdapter=""
$loopAdapter=""

foreach ($vnic in $vnics) {
    $vnic.macAddr = $vnic.macAddr.replace(":", "-").toupper()
    if ($vnic.nicIndex -eq 1 -and $vnic.vlanTag -eq 0) {
        $secondVnic = $vnic
    } elseif ($vnic.nicIndex -eq 0 -and $vnic.vlanTag -eq 0) {
        $primaryVnic = $vnic
    }
}

foreach ($phys in Get-NetAdapter) {

    if ($secondVnic.macAddr -eq $phys.MacAddress) {
        $secondAdapter=$phys
        Rename-NetAdapter -Name $phys.Name -NewName "External-HV"
        continue
    }

    if ($primaryVnic.macAddr -eq $phys.MacAddress) {
        $primaryAdapter=$phys
        Rename-NetAdapter -Name $phys.Name -NewName "Primary"
```

```

        continue
    }

    $loopAdapter=$phys
    Rename-NetAdapter -Name $phys.Name -NewName "Internal-HV"
}

Update-Help -Force:$true
$intPrefix=$internalIpBase.split("/")[1]
$netPrefix=$secondVnic.subnetCidrBlock.split("/")[1]
New-NetIPAddress -InterfaceIndex $secondAdapter.ifIndex -IPAddress
$secondVnic.privateIp -PrefixLength $netPrefix -DefaultGateway
$secondVnic.virtualRouterIp
New-NetIPAddress -InterfaceIndex $loopAdapter.ifIndex -IPAddress $hvIp -
PrefixLength $intPrefix -Confirm:$false
Start-Sleep 10
Set-DnsClientServerAddress -InterfaceIndex $secondAdapter.ifIndex -
ServerAddresses "169.254.169.254"
Set-NetAdapterAdvancedProperty -Name * -RegistryKeyword "*JumboPacket" -
RegistryValue 9014 -NoRestart
Enable-NetFirewallRule -DisplayName "File and Printer Sharing (Echo
Request - ICMPv4-In)"

$nrTargetPortal=$(New-IscsiTargetPortal -TargetPortalAddress 169.254.2.2)
Get-IscsiTarget -IscsiTargetPortal $nrTargetPortal | Connect-IscsiTarget -
IsPersistent $true
$nrDisk=Get-Disk | Where-Object {$_.IsSystem -eq $false}
Initialize-Disk -Number $nrDisk.Number
New-Partition -DiskNumber $nrDisk.Number -UseMaximumSize -
AssignDriveLetter
$nrPart=$(Get-Partition -DiskNumber $nrDisk.Number | where-object {$_.Type
-match "Basic"})
Format-Volume -Partition $nrPart -NewFileSystemLabel "hvsystem" -
Confirm:$false

Install-WindowsFeature -Name "DHCP" -IncludeManagementTools
Install-WindowsFeature -Name "DNS" -IncludeManagementTools
Install-WindowsFeature -Name "Hyper-V" -IncludeManagementTools -
IncludeAllSubFeature -Restart

```

Note: This script does not install the loopback adapter. You *must* install the loopback adapter separately by using the regular GUI, and before you run this script. Do not attempt to run a script like this without installing the loopback adapter first.

4. Reboot the instance, and then log back in.

Create the Internal Network for the Hyper-V Guests

1. Open the Hyper-V Manager.
2. Open the Virtual Switch Manager and create a new internal vSwitch named **Internal**. Select the **External network** option, and select the Microsoft Loopback Adapter previously configured as the DNS/DHCP adapter. Select the **Allow management operating system to share this network adapter** option.

Virtual Switch Properties

Name:
Internal

Notes:

Connection type
What do you want to connect this virtual switch to?

External network:
Microsoft KM-TEST Loopback Adapter

Allow management operating system to share this network adapter

Enable single-root I/O virtualization (SR-IOV)

Internal network

Private network

VLAN ID
 Enable virtual LAN identification for management operating system

The VLAN identifier specifies the virtual LAN that the management operating system will use for all network communications through this network adapter. This setting does not affect virtual machine networking.

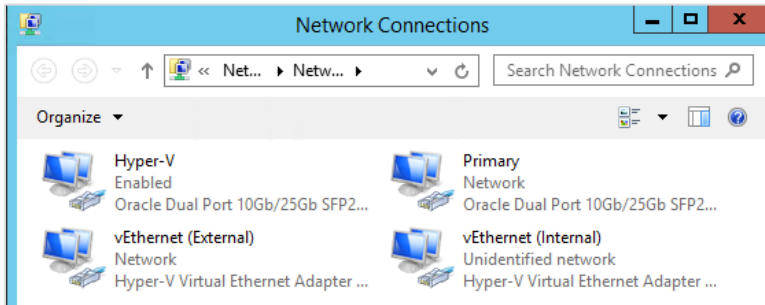
2

Remove

i SR-IOV can only be configured when the virtual switch is created. An external virtual switch with SR-IOV enabled cannot be converted to an internal or private switch.

OK Cancel Apply

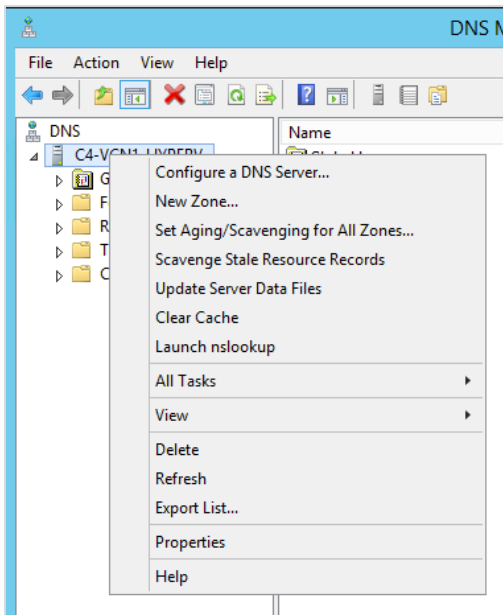
3. Open the network configuration for the instance.
- There should be three network interfaces, and one of them should be a vEthernet interface. Open the configuration for the vEthernet interface labeled **Internal**.



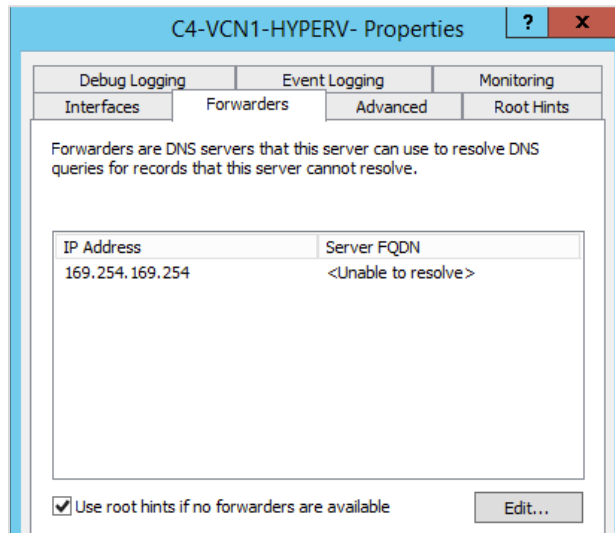
4. Change the MTU on both vEthernet adapters to 9014, using the same procedure that you did for the NICs. Do not reboot at this time.

Configure DNS and DHCP for the Hyper-V Guests

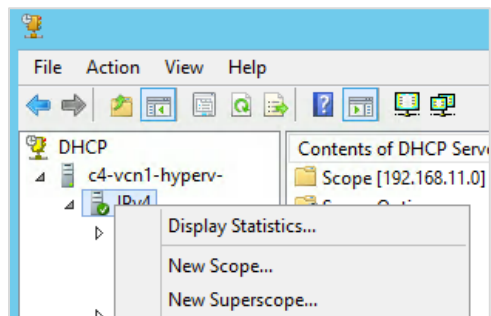
1. Open the DNS management application, right-click the server, and select **Properties**.



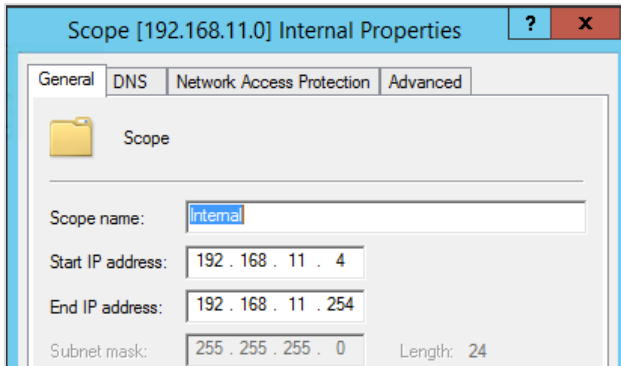
2. On the **Forwarders** tab, click **Edit** and create an entry for 169.254.169.254. The FQDN does not resolve, which is a normal condition.



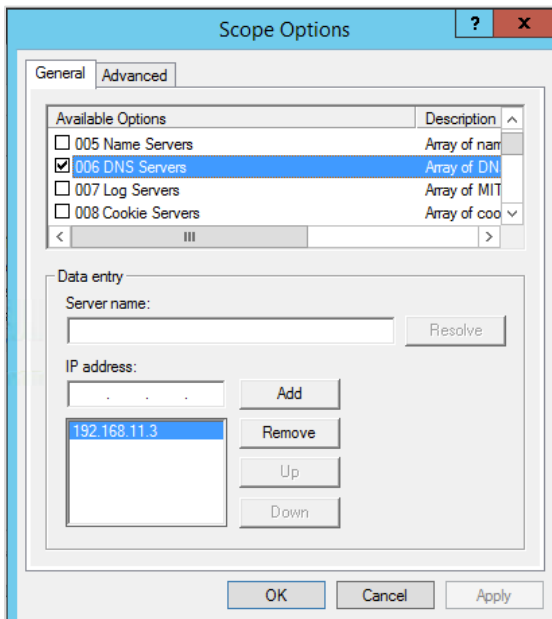
3. Save the configuration and restart DNS.
4. Open the DHCP management application.
5. Expand the server, right-click **IPv4**, and select **New Scope**.



6. Create the scope by using the IP address range selected for Hyper-V. Start with the address that immediately follows the three that you are using for this process.

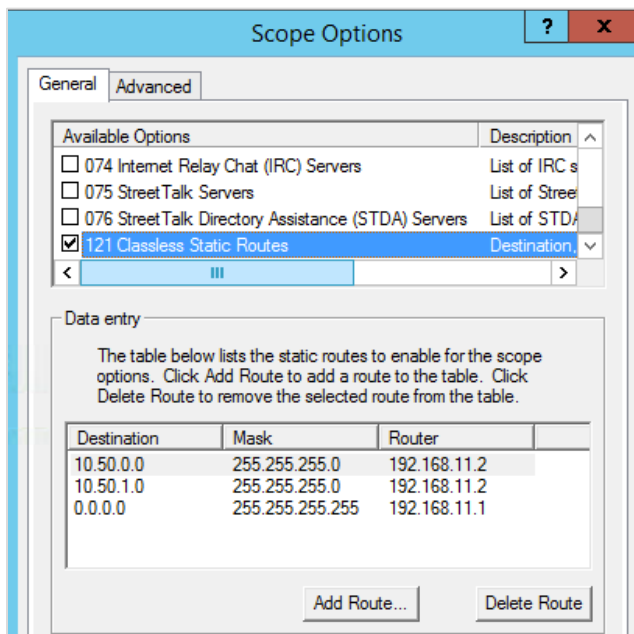


7. After you create the scope, expand the scope in the left navigation pane, right-click the **Scope Options** folder, and select **Configure Options**.
8. In the **Scope Options** dialog box, select the DNS option and enter the IP address that you selected for the DHCP/DNS.



- Under **Available Options**, select **121 Classless Static Routes**, and add the following routes (at a minimum). If you have other VCNs or networks via dynamic routing gateways (DRGs), also enter them here by using the VCN gateway or router address.

Destination	Mask	Router
0.0.0.0	0.0.0.0	IP address previously selected for the default gateway/NAT
VCN network space	Netmask of the VCN	IP address previously selected for the VCN gateway/router

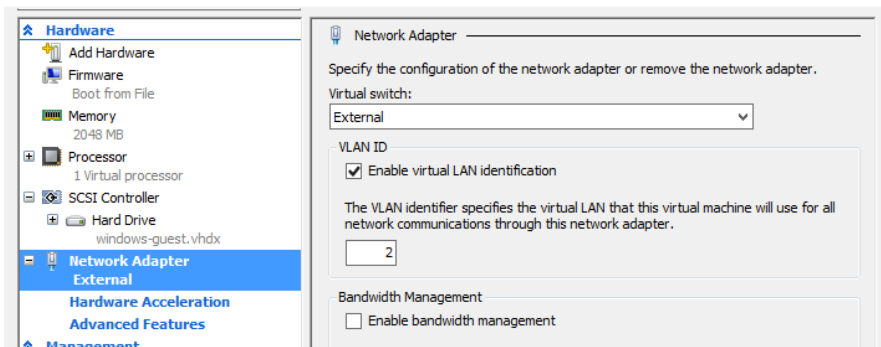


- Save the configuration and restart DHCP.

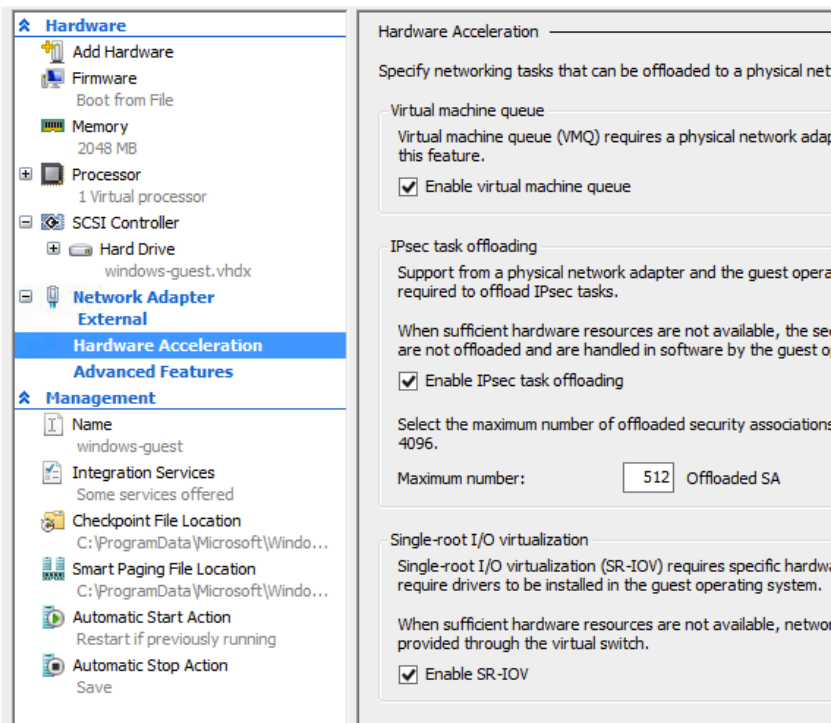
Create the Helper Hyper-V Guests

- Open Hyper-V Manager and create two new VMs, one named `hvnat` and one named `hvrouter`. Define the following characteristics for the VMs:
 - Generation 2 VM
 - Minimum memory of 4196 MB (dynamic memory)
 - Network connection to the external vSwitch
 - 100G of space for the VHDX, located on the block storage device (typically the D: drive)

2. After it is created, select the hvnat VM and select **Settings**.
3. Get the information for the secondary VNIC that is *not* selected as the route target, and then perform the following steps in the **Settings** dialog box:
 - A. In the left navigation pane, select **Network Adapter External**.
 - B. Select the **Enable virtual LAN identification** check box.
 - C. Enter the VLAN ID of the secondary VNIC.



4. In the left navigation pane, select **Hardware Acceleration** (under **Network Adapter External**), and then select the **Enable SR-IOV** check box.



- In the left navigation pane, click **Advanced Features** (under **Network Adapter External**). In the **MAC Address** section on the right, select the **Static** option, and then enter the MAC address associated with the secondary vNIC being used. Select the **Enable MAC address spoofing** check box.

Advanced Features

MAC address

Dynamic

Static

00 - 00 - 17 - 00 - 17 - 26

MAC address spoofing allows virtual machines to change the source MAC address in outgoing packets to one that is not assigned to them.

Enable MAC address spoofing

- In the left navigation pane, select **Add Hardware** and then select **Network Adapter**.
- Add an internal network adapter, keeping all of the default settings.
- Save the configuration.
- Repeat the preceding steps for the `hvrouter` VM.

The configuration items for the preceding three sections can also be completed using PowerShell. The following sample script (`hvrouting.ps1`) provides a template for running these steps:

```
Function ConvertTo-NetMask($cidr) {
    $mask=[Convert]::ToString((([Math]::Pow(2,$cidr) - 1) -shl (32 - $cidr), 2)
    $netmask = @()
    for($x = 0; $x -lt 4; $x++) {
        $netmask += [Convert]::ToUInt32($mask.Substring((8 *
    $x),8),2).ToString()
    }
    return [String]::Join(".", $netmask)
}

$instance=$(ConvertFrom-Json $(Invoke-WebRequest
("http://169.254.169.254/opc/v1/instance/")).Content)
$vnics=$(ConvertFrom-Json $(Invoke-WebRequest
("http://169.254.169.254/opc/v1/vnics/")).Content)
$secondVnic=""
$adapter=""
$hvVnics=@()

$instance.IPAddresses | ForEach-Object {
    $internalIpBase=$_
}
$hvIp=$instance.IPAddresses[0]
$beginIpRange=$instance.IPAddresses[1]
```

```

$endIpRange = "192.168.11.254"

foreach ($vnic in $vnics) {
    if ($vnic.nicIndex -eq 1 -and $vnic.vlanTag -eq 0) {
        $secondVnic = $vnic
        foreach ($phys in Get-NetAdapter) {
            $vnicMac = $secondVnic.macAddr.replace(":", "-").toupper()
            if ($vnicMac -eq $phys.MacAddress) {
                $adapter=$phys
                break
            }
        }
    } elseif ($vnic.nicIndex -eq 1 -and $vnic.vlanTag -gt 0) {
        $hvVnics+=$vnic
    }
}

New-VMSwitch -Name "External" -AllowManagementOS $true -NetAdapterName
$adapter.InterfaceAlias -EnableIov $true
New-VMSwitch -SwitchType Internal -Name "Internal"
$intAdapter=$(Get-NetAdapter | Where-Object {$_ .Name -match "internal"})
$extAdapter=$(Get-NetAdapter | Where-Object {$_ .Name -match "external"})

New-NetIPAddress -InterfaceIndex $intAdapter.ifIndex -IPAddress $hvIp -
PrefixLength $internalIpBase.split("/")[1]
Set-DnsServerForwarder -IPAddress 169.254.169.254
sleep 30
Add-DhcpServerv4Scope -Name "Internal" -EndRange $endIpRange -StartRange
$beginIpRange -SubnetMask $(ConvertTo-NetMask
([Convert]::ToInt16($internalIpBase.split("/")[1])))
Set-DhcpServerv4OptionValue -ScopeId $internalIpBase.split("/")[0] -DnsServer
$hvIp
Set-DhcpServerv4Binding -InterfaceAlias $intAdapter.InterfaceAlias -BindingState
$true
Set-DhcpServerv4Binding -InterfaceAlias $extAdapter.InterfaceAlias -BindingState
$false

New-Item -ItemType Directory -Path "D:\Virtual Machines"
$hvNetGuests=@("hvnat", "hvrouter")
$count=0
foreach ($guest in $hvNetGuests) {
    New-Item -ItemType Directory -Path ("D:\Virtual Machines\" + $guest)
    New-VM -Name $guest -MemoryStartupBytes (4* [Math]::Pow(1024,3)) -NewVHDPATH
("D:\Virtual Machines\" + $guest + "\" + $guest + ".vhdx") -NewVHDSIZEBytes (100
* [Math]::Pow(1024, 3)) -Generation 2 -SwitchName "External"
    Add-VMDvdDrive -VMName $guest
    Set-VMFirmware -VMName $guest -BootOrder @($(Get-VMDvdDrive -VMName $guest),
$(Get-VMHardDiskDrive -VMName $guest))
    Set-VM -Name $guest -DynamicMemory -MemoryMinimumBytes (4*
[Math]::Pow(1024,3))
}

```

```
Set-VMNetworkAdapter -VMName $guest -StaticMacAddress
$hvVnics[$count].macAddr -IovWeight 100
Set-VMNetworkAdapterVlan -VMName $guest -Access -VlanId
$hvVnics[$count].vlanTag
Add-VMNetworkAdapter -VMName $guest -SwitchName "Internal"
$count++
}

Set-NetAdapterAdvancedProperty -Name * -RegistryKeyword "*JumboPacket" -
RegistryValue 9014
```

Install and Configure Windows for the hvnat and hvrouter Hyper-V guests

1. Install Windows 2012 R2 on both the `hvnat` VM and the `hvrouter` VM.
2. Perform the following steps on each guest:
 - A. Identify the interface associated with the secondary VNIC. To do this, look at the MAC address of the virtual NIC in the instance.
 - B. Configure the IP address, subnet mask, and default gateway of the secondary VNIC with the information identified for the particular instance.
 - C. Configure the second interface with the IP address selected for the function.
 - The `hvnat` guest should *not* be the route target for the VCN and should get the Hyper-V address associated with the default gateway/NAT.
 - The `hvrouter` guest should have the secondary VNIC associated with the route target address and should get the Hyper-V address associated with the VCN gateway/router.

The interface with the Hyper-V address should *not* get a default gateway configured, but should have the DNS address assigned. The DNS address should be the internal address associated with the DNS/DHCP function for Hyper-V.
 - D. Verify that each instance can ping the subnet default gateway on the VCN and the internal DNS/DHCP address, and can get to the internet.
 - E. Apply all current Windows patches to the operating system.

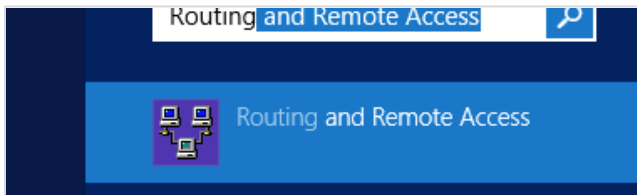
Configure the hvnat Guest

Before performing these steps, ensure that all networking on the `hvnat` guest has been configured and is functional. Tests of the network should include being able to communicate with both instances on the VCN side of the `hvnat` guest and network targets on the internal Hyper-V side. A simple ping test should suffice.

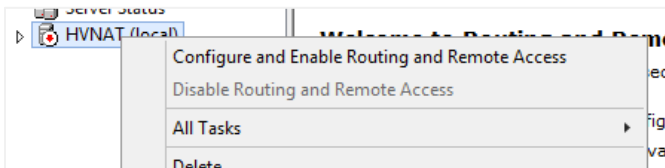
1. On the `hvnat` instance, install the Routing and Remote Access service (RRAS), routing only. Perform this step by using the following PowerShell command, run as Administrator:

```
Install-WindowsFeature -Name "Routing" -IncludeSubFeature -  
IncludeManagementTools -Confirm:$false
```

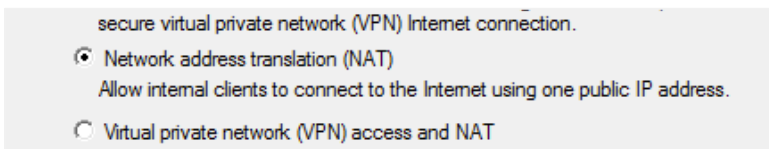
2. Open the Routing and Remote Access tool.



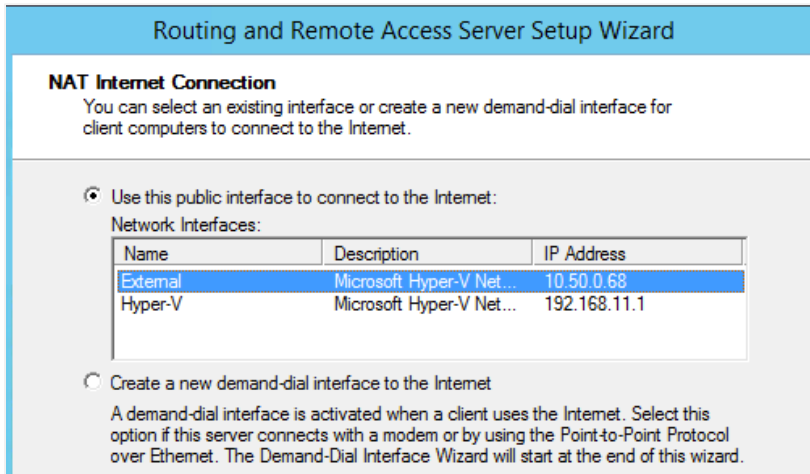
3. Right-click the server and select **Configure and Enable Routing and Remote Access**.



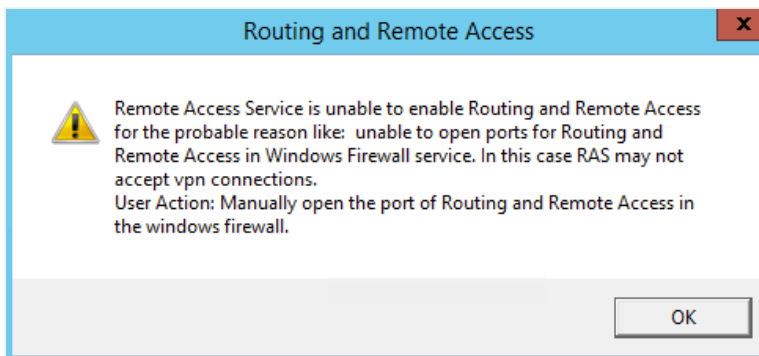
4. On the welcome page of the wizard, click **Next**.
5. On the **Configuration** page, select **Network address translation (NAT)**, and then click **Next**.



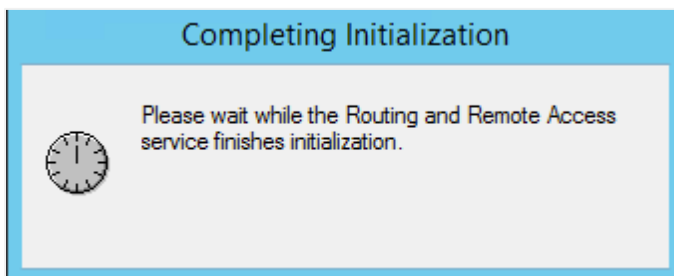
6. On the **NAT Internet Connection** page, select the interface that has the VCN IP address for the internet-facing interface, and then click **Next**.



7. Click **Finish**.
8. If you get the following warning, you can ignore it and click **OK**.



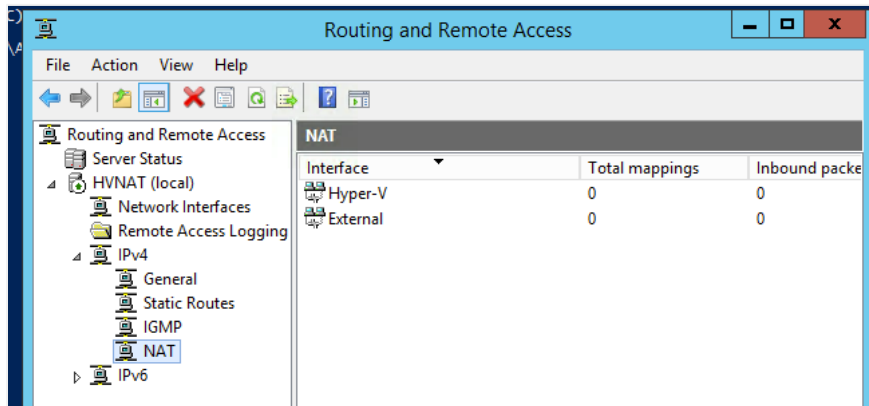
A dialog box is displayed, indicating that the configuration is being committed.



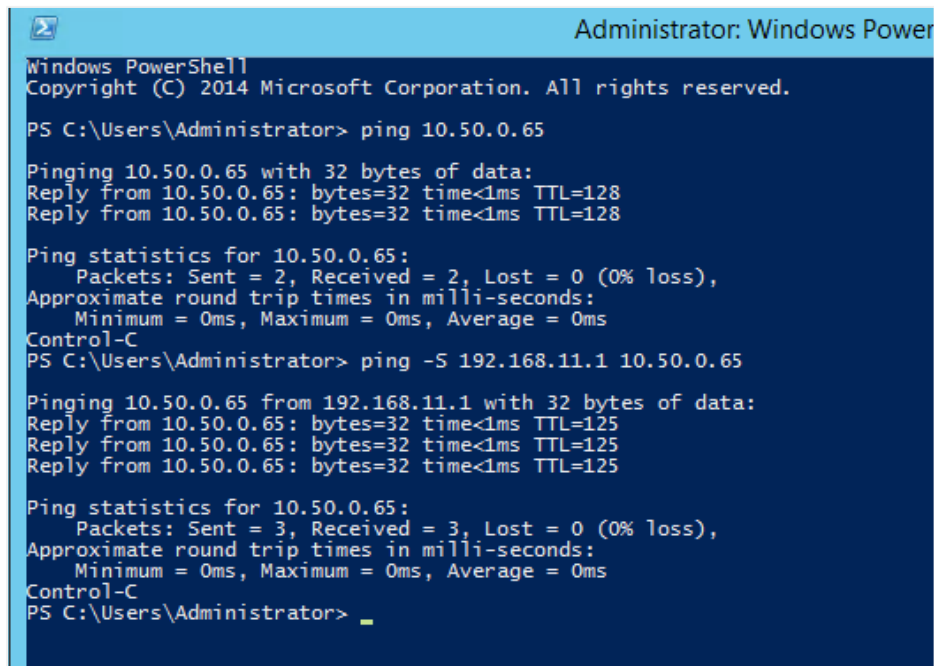
This process might stop responding. If the dialog box does not disappear after 10 or so minutes, restart the instance.

9. In the Routing and Remote Access tool, expand the server and click **NAT**.

The right side of the window should display entries similar to the following ones:



10. Verify that you can still ping the VCN subnet default gateway via the Hyper-V address. In the following example, the VCN gateway is 10.50.0.65 and the Hyper-V address for hvnat is 192.168.11.1. You can open a PowerShell window to issue the commands.



The `ping -S` command sends the ping from the designated interface.

Configure the hvrouter Guest

Before performing these steps, ensure that all networking on the `hvrouter` guest has been configured and is functional. Tests of the network should include being able to communicate with both instances on the VCN side of the `hvrouter` guest and network targets on the internal Hyper-V side. A simple ping test should suffice.

1. Log in to the `hvrouter` instance.
2. Open a PowerShell window as Administrator.
3. Identify the interface index numbers by running the following command:

```
Get-NetAdapter
```

Note the `ifIndex` numbers for each of the Hyper-V interfaces.

4. Configure forwarding on each interface by running the following command for each interface number identified:

```
Set-NetIPInterface -InterfaceIndex <ifIndex_number> -Forwarding Enabled
```

5. Test to ensure that you can ping the Hyper-V interface from an instance on the VCN subnet.

The configuration is now complete.

Install Guests

You can install guest instances by using the normal process associated with building Hyper-V guests, with the following changes.

Windows Automatic Virtual Machine Activation

Windows guests listed in the following table can participate in the Automatic Virtual Machine Activation (AVMA) process, which allows guests to be installed in the Windows 2016 Datacenter Hyper-V role and still be licensed. Linux and other operating systems do not require Windows licenses to run. Note that Windows Desktop licenses are not covered in this configuration.

Operating System Version	AVMA Key
Windows Server 2012 R2 Essentials	K2XGM-NMBT3-2R6Q8-WF2FK-P36R2
Windows Server 2012 R2 Standard	DBGBW-NPF86-BJVTX-K3WKJ-MTB6V
Windows Server 2012 R2 Datacenter	Y4TGP-NPTV9-HTC2H-7MGQ3-DV4TW

Operating System Version	AVMA Key
Windows Server 2016 Essentials	B4YNW-62DX9-W8V6M-82649-MHBKQ
Windows Server 2016 Standard	C3RCX-M6NRP-6CXC9-TW2F2-4RHYD

These keys are publicly available and valid for all installations of these operating systems under Hyper-V. You can activate guests by using the following procedure on each guest:

1. Launch an administrative command prompt.
2. Type the following command:

```
slmgr /ipk <AVMA_key>
```

3. Close the command prompt.

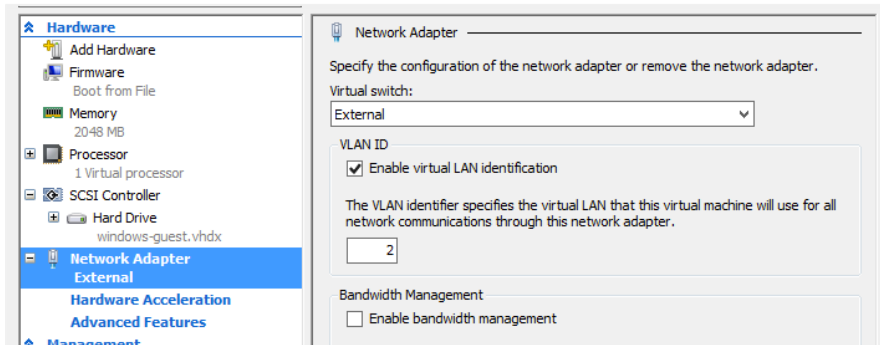
For details about AVMA, see the [Hyper-V Automatic Virtual Machine Activation in Windows Server 2016](#) blog post at altaro.com.

Direct Access Guests

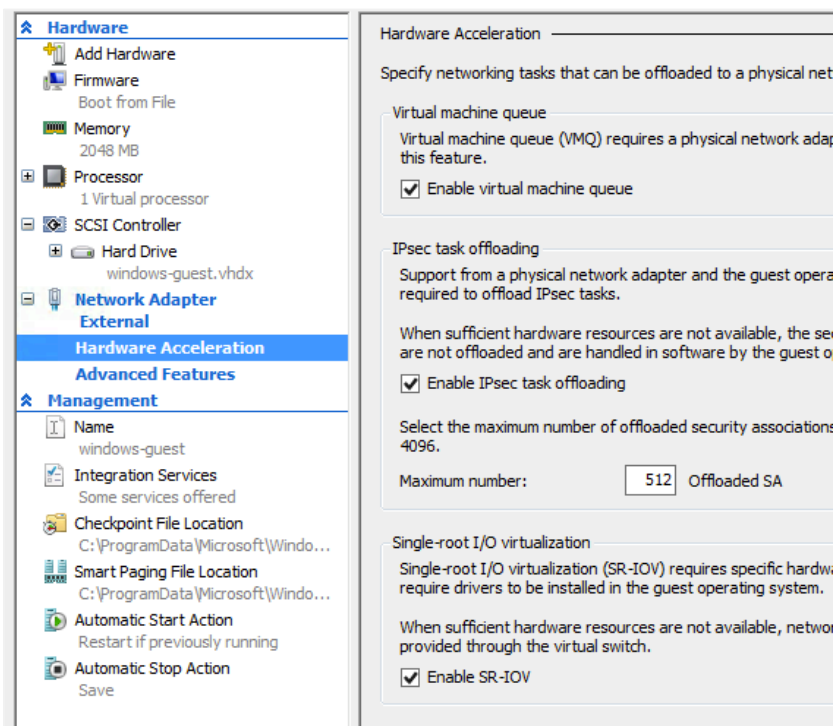
Direct access guests have some networking configuration requirements in addition to the normal Hyper-V guest configuration:

1. Because direct access guests use VNICs, you must assign a VNIC to each guest before installation. The VNIC *must* be assigned to NIC1 on the Hyper-V host.
2. After the VNIC is attached, gather the following information about the VNIC:
 - o IP address, subnet default gateway, and subnet netmask
 - o MAC address
 - o VLAN tag
3. Install the guest operating system, or import the guest as normal. Connect the guest to the external vSwitch.
4. Before starting the guest, select the guest and click **Settings**.
5. With the information gathered in step 1, perform the following steps in the **Settings** dialog box:
 - A. In the left navigation pane, click **Network Adapter External**.
 - B. Select the **Enable virtual LAN identification** check box.

C. Enter the VLAN ID.

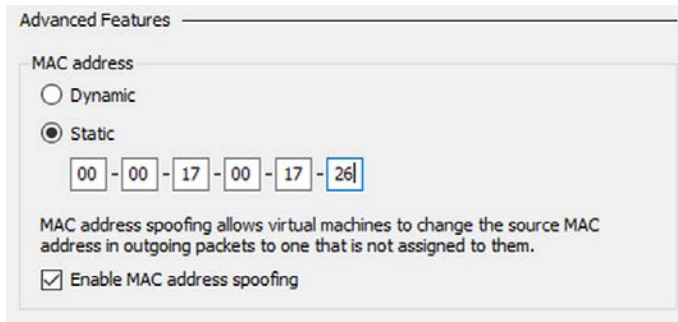


6. In the left navigation pane, click **Hardware Acceleration** (under **Network Adapter External**), and then select the **Enable SR-IOV** check box.



7. In the left navigation pane, click **Advanced Features** (under **Network Adapter External**). In the **MAC Address** section on the right, select the **Static** option, and then

enter the MAC address associated with the secondary vNIC being used. Select the **Enable MAC address spoofing** check box.



Advanced Features

MAC address

Dynamic

Static

00 - 00 - 17 - 00 - 17 - 26

MAC address spoofing allows virtual machines to change the source MAC address in outgoing packets to one that is not assigned to them.

Enable MAC address spoofing

8. Add an internal network adapter, keeping all of the default settings.
9. Save the configuration.
10. Start the guest. If this is a new installation, configure the guest to use a static IP configuration as directed by your operating system. If this is an import, reconfigure your network interface to have a static IP address configuration.
11. If you are installing a Linux operating system, you might have to load the Hyper-V specific drivers. For information about your specific Linux distribution, see [Supported Linux and FreeBSD virtual machines for Hyper-V on Windows](#) on the Microsoft website.

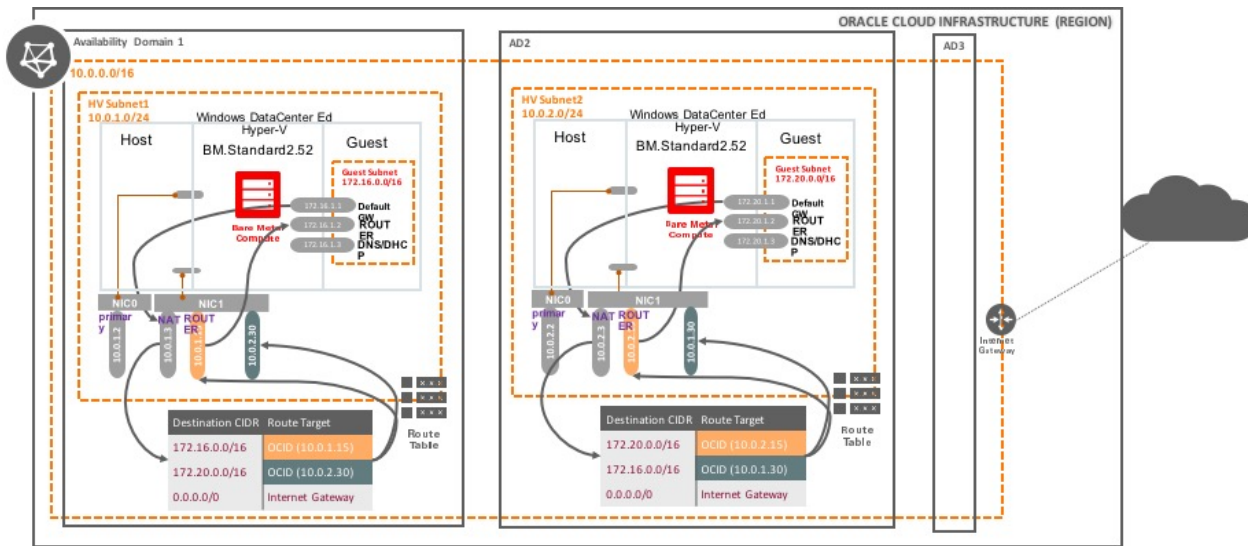
Indirect Access Guests

Indirect access guests have a simpler network configuration requirement. Follow these guidelines when configuring indirect access guests:

- Guests should be configured with access to *only* the internal Hyper-V network vSwitch, and *not* the external vSwitch.
- Guests should always use DHCP. If static addressing is required, either register the MAC address of the guest as a DHCP reservation within the Hyper-V based DHCP server, or configure the guest with the following information and add the exclusion to DHCP:
 - **IP address:** Assigned IP address
 - **Netmask:** Netmask for IP subnet
 - **Default gateway:** Hyper-V address of the hvnat instance
 - **DNS:** Hyper-V address of the hypervisor (DNS/DHCP configured earlier)
 - Manually add a route to the VCN subnets and other internal networks, via the VCN gateway (hvrouter) address


Enable Connections Between Guests on Different Hyper-V Servers

This process enables guests that were installed using the indirect access method on one Hyper-V server to interact fully with guests on another. This process is not required for direct access method guests because the configuration is implied by the overall Oracle Cloud Infrastructure environment. The architecture of this configuration is illustrated in the following diagram:



This process can be applied to all Hyper-V servers running within the same VCN. The instructions assume that you have two Hyper-V servers, Svr1 and Svr2, with two different IP address ranges for their guests. Both Hyper-V servers must be configured by using the process described earlier in this paper before you attempt this procedure. You should identify the guest IP subnet, associated netmask, and `hvrrouter` IP address for each Hyper-V server before starting this process.

1. Open the DHCP server on Svr1.
2. Select the scope that you created in "Configure DNS and DHCP for the Hyper-V Guests."
3. Open the **Scope Options** dialog box and select **121 Classless Static Routes**.
4. Enter the following route information:
 - o **Destination:** IP subnet for the guests located on Svr2
 - o **Mask:** IP subnet mask for the guests on Svr2
 - o **Router:** IP address of `hvrrouter` created on Svr1
5. Save the configuration and restart DHCP.

- 
6. Renew the DHCP leases on all guests running on Svr1.
 7. Open the DHCP server on Svr2.
 8. Select the scope that you created in “Configure DNS and DHCP for the Hyper-V Guests.”
 9. Open the **Scope Options** dialog box and select **121 Classless Static Routes**.
 10. Enter the following route information:
 - **Destination:** IP subnet for the guests on Svr1
 - **Mask:** IP subnet mask for the guests on Svr1
 - **Router:** IP address of `hvrrouter` created on Svr2
 11. Save the configuration and restart DHCP.
 12. Renew the DHCP leases on all guests running on Svr2.

Assuming that you have created the security lists and entries in the route table for the VCN for each Hyper-V server, you should now be able to communicate between guests residing in each Hyper-V server. You can extend this model to additional servers by adding their routing information to each other Hyper-V server that needs to participate in the relationship.

Conclusion

Although the bare metal and virtual instance types remain the recommended method for deploying operating systems and applications, there are a number of situations in which using them is simply not possible. In environments where Windows Server is prevalent, operational practices have evolved to provide robust support for management of guests within Hyper-V. The method of deploying Hyper-V detailed in this paper provides the ability to extend those operations into the cloud with a minimum of disruption, as well as providing a method of deploying legacy operating systems, and the ability to adjust the ratio of CPU to RAM, all while continuing the journey to cloud operations.

Appendix A: Deploying Hyper-V Replica

NOTE: As indicated earlier, Hyper-V Replica can be applied only to guests that are installed using the Indirect access method.

Although you can use Microsoft Hyper-V to create guests in the Oracle Cloud Infrastructure environment, like any hypervisor, they act as a single point of failure. Although cloud environments such as Oracle Cloud Infrastructure provide a resilient operational environment, many customers find it necessary to maintain strong high availability (HA) to ensure workload availability and business continuity. To help maintain strong HA, Oracle Cloud Infrastructure has enabled the ability to use Hyper-V Replica to establish replication connectivity between two Hyper-V instances.

Hyper-V Replica is the built-in functionality of Hyper-V that enables the online replication of a guest operating system to a target system without requiring a shared storage resource. Hyper-V Replica uses change tracking within the source guest virtual hard disks (VHDs) and replicates those changes via a TCP network connection to an arbitrary Hyper-V server acting as a replica target.

This appendix describes the process for establishing Hyper-V Replica connectivity between two Hyper-V instances built within Oracle Cloud Infrastructure. The process described here can also be applied to replicas deployed from on-premises instances to Oracle Cloud Infrastructure instances.

Prerequisites

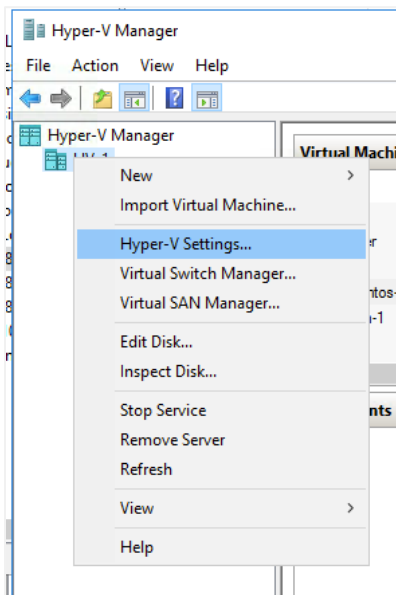
For this process to work as designed, the following prerequisites must be satisfied:

- For intra-region replicas, deploy two Hyper-V servers as explained in the main part of this paper. If Oracle Cloud Infrastructure is to act as a Hyper-V Replica target from on-premises, only one Hyper-V server is required.
- Ensure that connectivity exists bidirectionally between the source and target Hyper-V instances. A simple ping suffices for the connectivity test.
- Ensure that hostname resolution works between the source and target Hyper-V instances.
- Make TCP ports 80 and 443 available within the subnet. Configure your Oracle Cloud Infrastructure [security lists](#) appropriately.
- Attach additional [Oracle Cloud Infrastructure Block Volumes storage volumes](#), or identify a directory within an existing attached block storage volume on the Hyper-V instances, to serve as a repository for the replica Hyper-V guests.

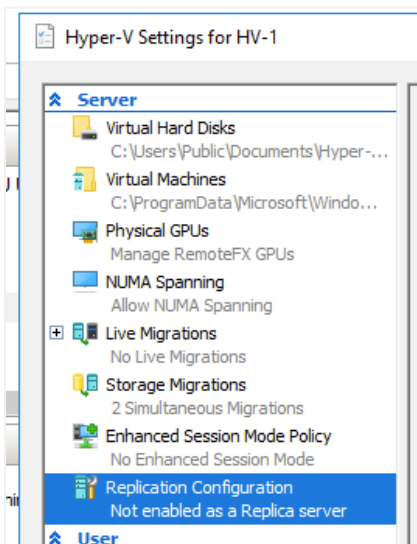
Configure the Hyper-V Target Instance

After the Hyper-V instances are deployed and tested, use the following procedure to configure Hyper-V Replica on the instance that is the target for replication.

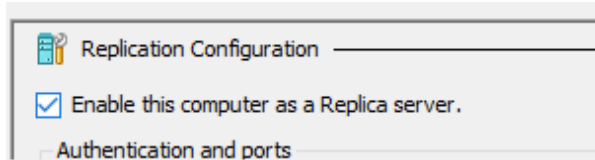
1. Open Hyper-V Manager.
2. Right-click the Hyper-V host in the left-side pane and select **Hyper-V Settings**.



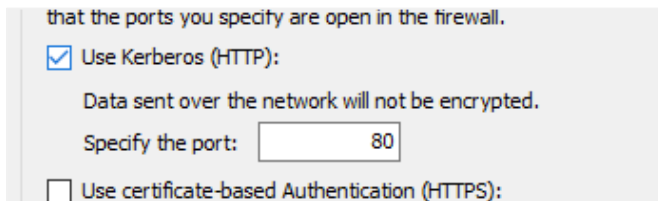
3. In the left-side pane of the **Hyper-V Settings** dialog box, click **Replication Configuration**.



4. In the top-right pane, select **Enable this computer as a Replica server**.

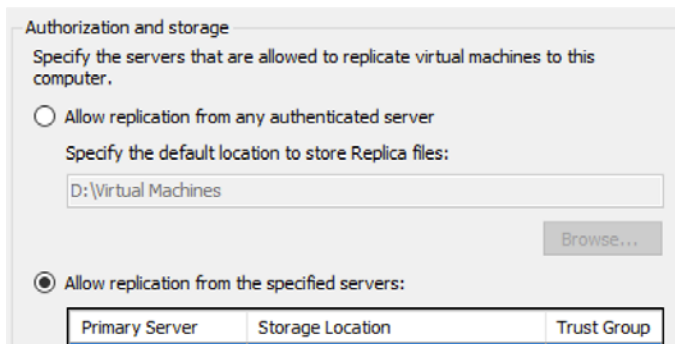


5. In the **Authentication and ports** section, select **Use Kerberos (HTTP)**. Keep the TCP port value of 80.



Note: Certificate-based authentication (HTTPS) is possible in this configuration, but the setup of that method is beyond the scope of this paper. See the appropriate Microsoft documentation for the configuration of a certificate provider within Windows Server 2016.

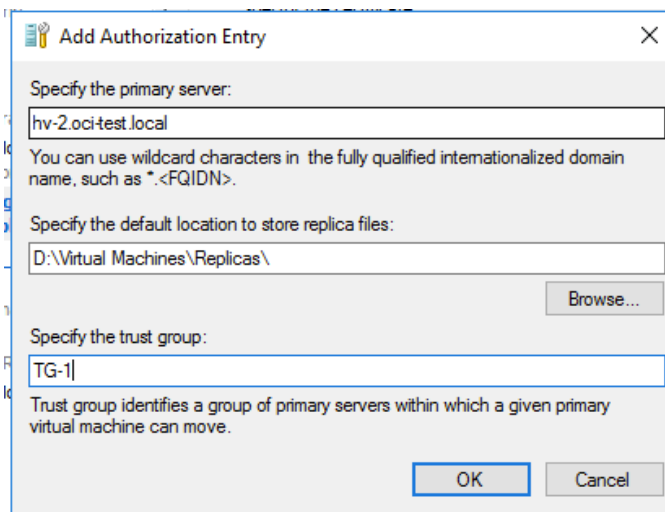
6. In the **Authorization and storage** section, select the **Allow replication from the specified servers** option.



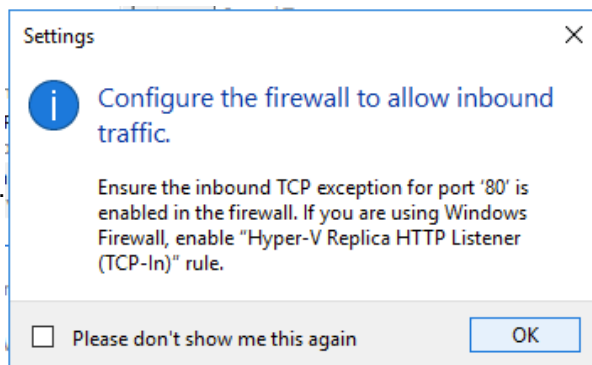
Note: The **Allow replication from any authenticated server** function works, but is not recommended for security reasons.

7. Click **Add**.

8. In the **Add Authorization Entry** dialog box, add the following information:
 - **Primary server:** This is the source server for the replica information. This is the host name that you verified for hostname resolution as part of the prerequisites.
 - **Location for replica files:** This is the full path to either the additional block storage volume attached to the target Hyper-V instance for replicas or the block storage volume attached for primary Hyper-V guests on this instance. Do not use the boot volume as the replica location.
 - **Trust group:** This is an arbitrary group name that identifies the participant Hyper-V instances within a replication relationship. This name *must* be consistent between the source and target instances if you want to enable bidirectional replication.

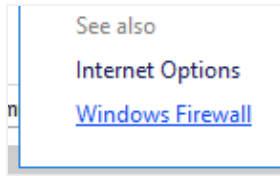


9. Click **OK**, and then click **OK** in the **Replication Configuration** section.
10. Click **OK** in the settings message that is displayed.

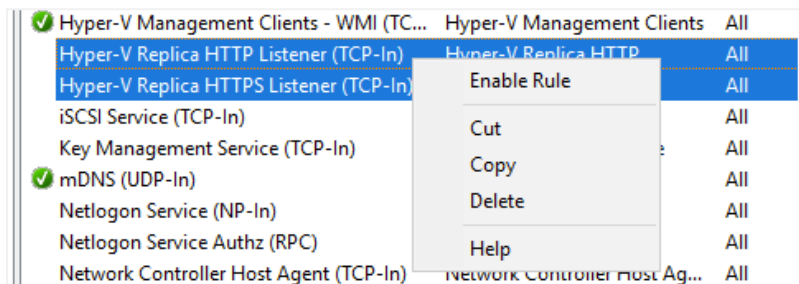


11. In the Windows Control Panel, open the Network and Sharing Center.

12. On the bottom-left side of the window, click **Windows Firewall**.



13. In the **Windows Firewall** window, click **Advanced Options**.
14. In the **Windows Firewall with Advanced Security** window, click **Inbound Rules**.
15. Scroll down and find the two Hyper-V Replica rules, as shown in the following screenshot. Select both rules, right-click, and then select **Enable Rule**.



16. Close both of the Windows Firewall windows.

The instance is now set up to receive replicas from the named Hyper-V instance. If you want to enable bidirectional replication, repeat this process on the source Hyper-V instance.

Configure the Hyper-V Guests

After you configure the target Hyper-V instance, you need to configure individual guests within the source Hyper-V instance to replicate to the target Hyper-V instance.

Limitations of Guest Replication from On-Premises to Oracle Cloud Infrastructure

Guests replicated from on-premises to Oracle Cloud Infrastructure might require some configuration changes, depending on their network configuration, domain status, and DNS registration. In the default configuration, guests created within the Oracle Cloud Infrastructure environment have access to only a single vSwitch for their networking. The same limitation applies to guests that are replicated into the Oracle Cloud Infrastructure Hyper-V instances from an on-premises instance. On-premises guests that have more than one vSwitch attached will have only a single network interface when replicated to Oracle Cloud Infrastructure, and might require configuration as a result.

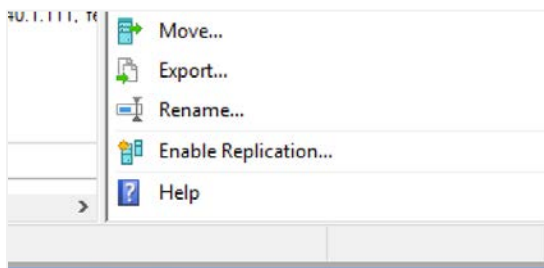
Domain-joined guests might experience authentication issues if they can't reach a domain controller after replication to Oracle Cloud Infrastructure. If domain-joined guests are to be replicated into an Oracle Cloud Infrastructure Hyper-V instance, we recommend establishing a primary domain controller, backup domain controller, or read-only domain controller within Oracle Cloud Infrastructure to prevent authentication issues resulting from long-latency or disconnected domains.

Guest hostname resolution might fail, particularly on non-domain-joined or non-Windows guests that use DNS services. Resolution might also fail if a Windows DNS service is not established in support of domain services established within Oracle Cloud Infrastructure. To maintain connectivity with guests after failover, we recommend manually verifying DNS to ensure that the current IP address assigned to the replica has been correctly inserted into the appropriate DNS zone or zones. A manual update of DNS might be required to ensure correct hostname resolution. Oracle Cloud Infrastructure DNS services do not currently recognize or accept updates from Hyper-V or any other external DNS service.

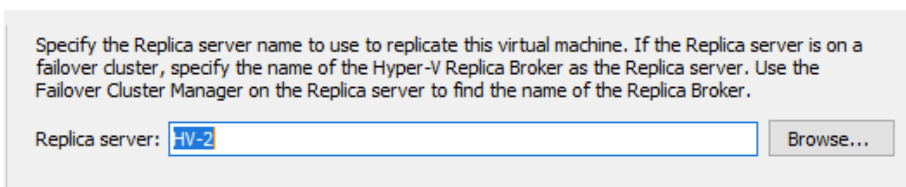
Configure a Hyper-V Guest

Follow this procedure to configure each guest.

1. Open Hyper-V Manager and select the guest to replicate.
2. In the lower-right pane, click **Enable Replication**.



3. On the first page of the wizard, click **Next**.
4. On the next page, enter the hostname of the target Hyper-V instance in the **Replica server** box and then click **Next**.



NOTE: Communication failures between source and target Hyper-V instances are mainly caused by a failure of hostname resolution. If a failure occurs, ensure that the hostname entered in the box can be resolved by using that exact hostname. An FQDN might be required for correct connectivity, and the source Hyper-V instance name must match that set within the Add Authorization Entry step detailed in the “Configure the Hyper-V Target Instance” procedure.

If communication is successful, the **Specify Connection Parameters** page is displayed.

Enable Replication for test-centos-1

Specify Connection Parameters

Before You Begin
Specify Replica Server
Specify Connection Parameters
Choose Replication VHDs
Configure Replication Frequency
Configure Additional Recovery Points
Choose Initial Replication Method
Summary

Replica server: test-centos-1
Replica server port: 80

Authentication Type

Use Kerberos authentication (HTTP)
Data will not be encrypted while being transmitted over the network.

Use certificate-based authentication (HTTPS)
Data will be encrypted while being transmitted over the network.

Issued To:
Issued By:
Expiration Date:
Intended Purpose:

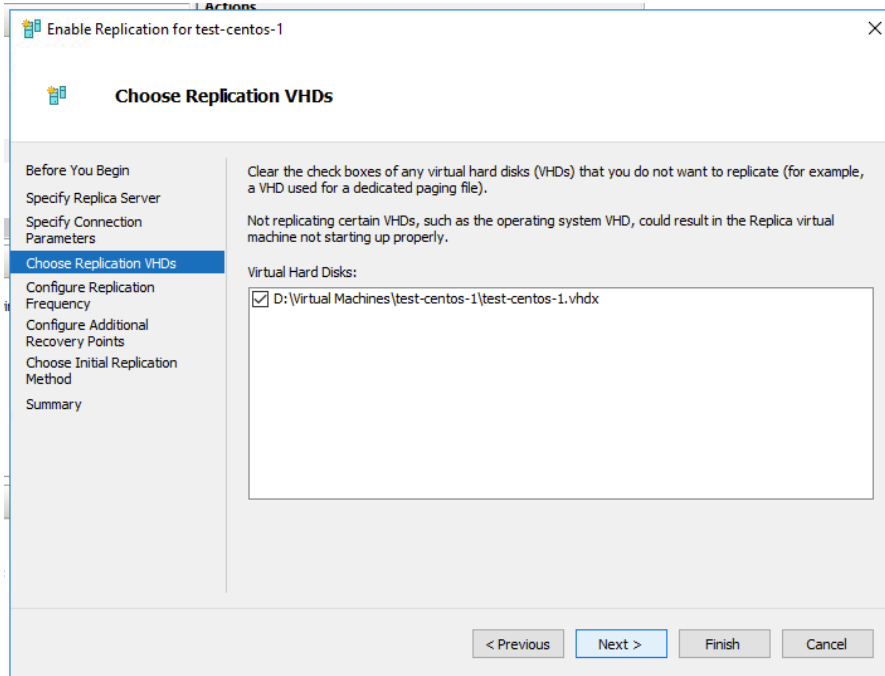
Select Certificate...

Compress the data that is transmitted over the network.

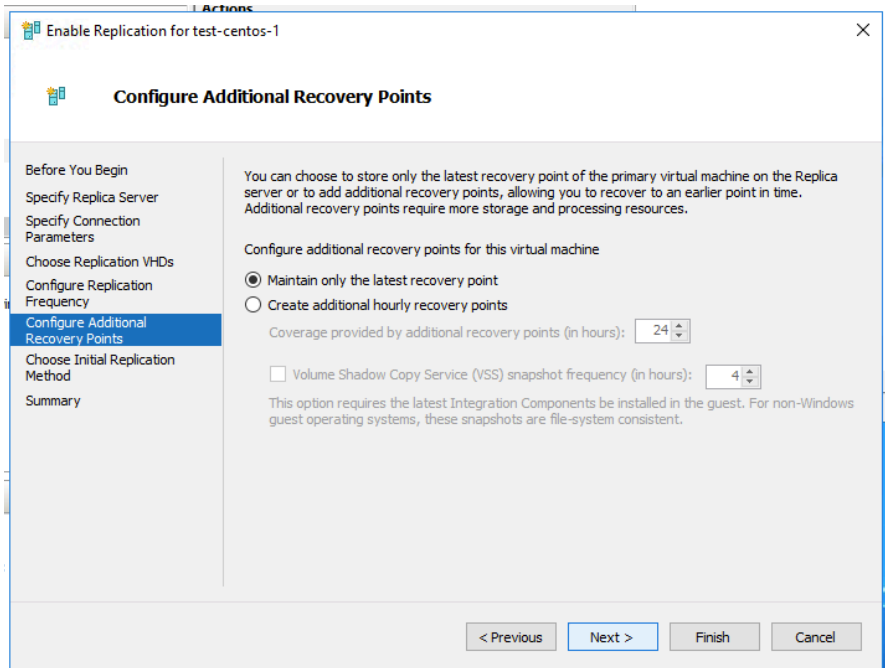
< Previous Next > Finish Cancel

5. Select the **Use Kerberos authentication (HTTP)** option, select the **Compress the data that is transmitted over the network** check box, and then click **Next**.

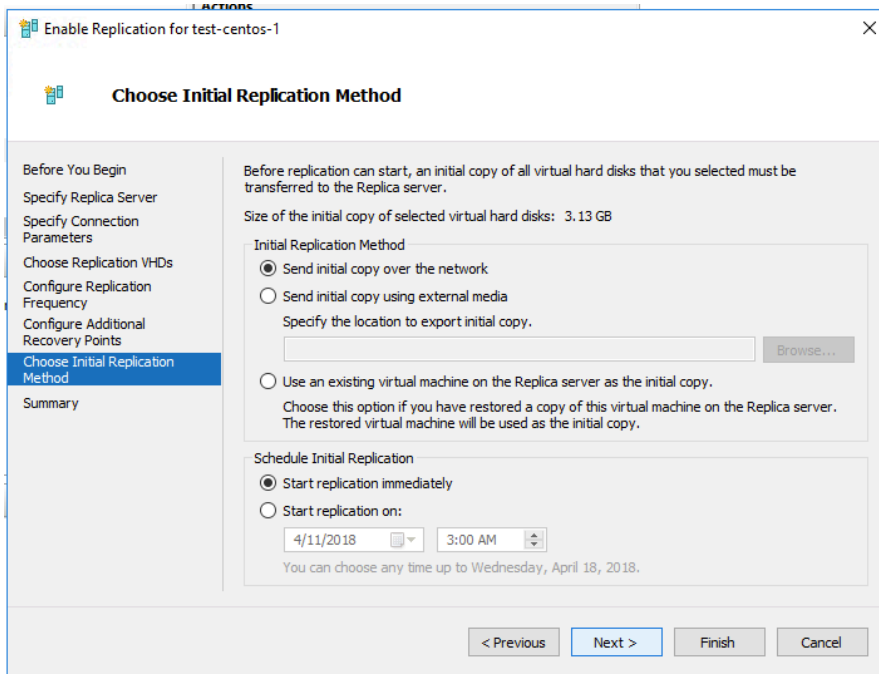
6. Select all the VHDs associated with this guest, and then click **Next**.



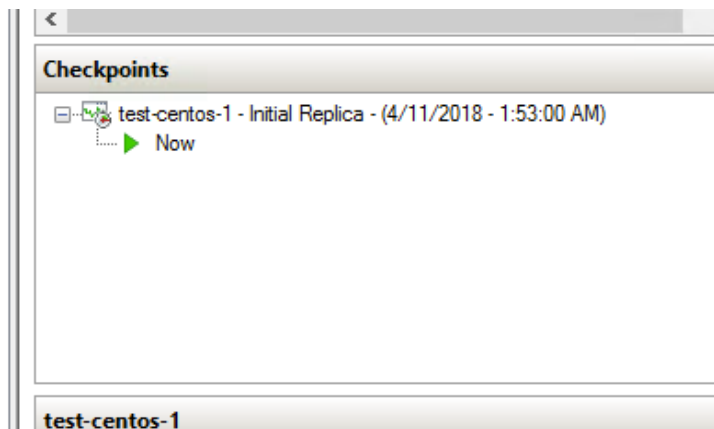
7. Select the frequency of replication, and then click **Next**.
8. Select the number of recovery points that you want to maintain. If you select multiple recovery points, *do not* select the VSS option.



9. Select **Send initial copy over the network** as the initial replication method, and then select the schedule for the initial replication as appropriate for the environment. Click **Finish**.



If you selected an immediate replication, you should see an entry similar to the following one in the Hyper-V Manager for the guest being replicated.



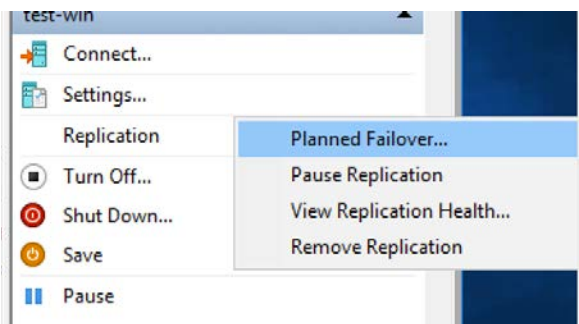
Fail Over Between Source and Target Hyper-V Instances

Using this configuration, instances must be manually failed to the replica target before use.

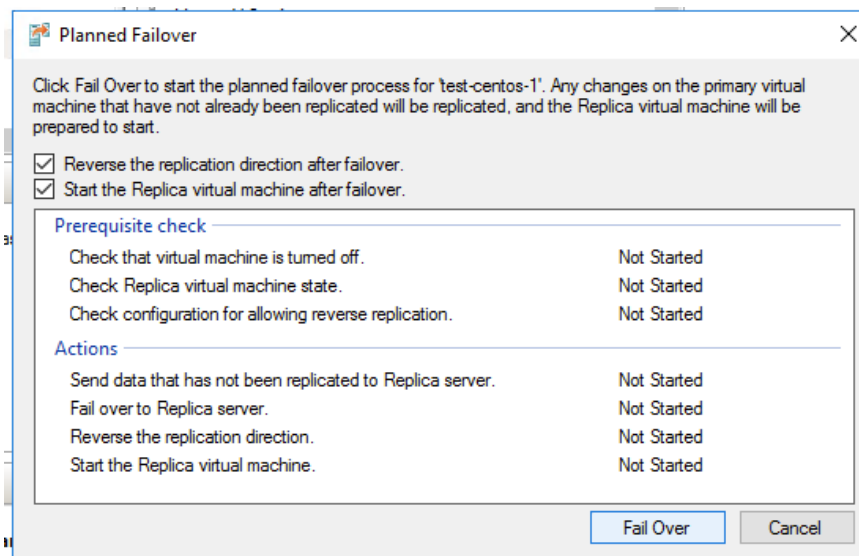
Note: Guests failed over to target Hyper-V instances, particularly those from on-premises environments, are subject to the limitations listed in the “Limitations of Guest Replication from On-Premises to Oracle Cloud Infrastructure” section.

Use this procedure to fail over Hyper-V guests to the replica instance.

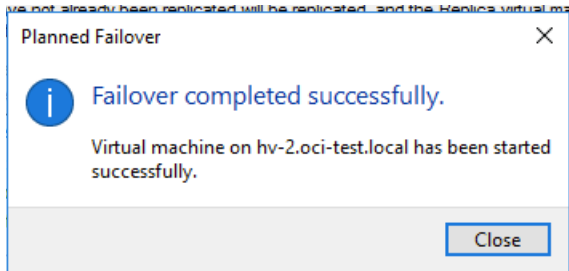
1. Open the Hyper-V Manager on the source instance and stop the Hyper-V guest.
2. With the guest selected in Hyper-V Manager, right-click **Replication** in the lower-right pane, and then select **Planned Failover** from the menu.



3. In the **Planned Failover** dialog box, select the **Reverse the replication direction after failover** check box, the **Start the Replica virtual machine after failover** check box, or both, as needed for the operation. Then click **Fail Over**.



The replication runs through the steps listed in the box and provides feedback about the success of each step. If the replication is successful, the following message appears:







To fail the guest back to the original Hyper-V instance, log in to the target Hyper-V instance and reverse this process.



Oracle Corporation, World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065, USA

Worldwide Inquiries
Phone: +1.650.506.7000
Fax: +1.650.506.7200

CONNECT WITH US

-  blogs.oracle.com/oracle
-  facebook.com/oracle
-  twitter.com/oracle
-  oracle.com

Integrated Cloud Applications & Platform Services

Copyright © 2018, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only, and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group. 1118

Deploying Hyper-V on Oracle Cloud Infrastructure
November 2018
Author: Steven B. Nelson



Oracle is committed to developing practices and products that help protect the environment.