Installing vThunder TPS on Microsoft Azure

March, 2022

A10

PATENT PROTECTION

A10 Networks, Inc. products are protected by patents in the U.S. and elsewhere. The following website is provided to satisfy the virtual patent marking provisions of various jurisdictions including the virtual patent marking provisions of the America Invents Act. A10 Networks, Inc. products, including all Thunder Series products, are protected by one or more of U.S. patents and patents pending listed at:

a10-virtual-patent-marking.

TRADEMARKS

A10 Networks, Inc. trademarks are listed at: a10-trademarks

CONFIDENTIALITY

This document contains confidential materials proprietary to A10 Networks, Inc.. This document and information and ideas herein may not be disclosed, copied, reproduced or distributed to anyone outside A10 Networks, Inc. without prior written consent of A10 Networks, Inc..

DISCLAIMER

This document does not create any express or implied warranty about A10 Networks, Inc. or about its products or services, including but not limited to fitness for a particular use and non-infringement. A10 Networks, Inc. has made reasonable efforts to verify that the information contained herein is accurate, but A10 Networks, Inc. assumes no responsibility for its use. All information is provided "as-is." The product specifications and features described in this publication are based on the latest information available; however, specifications are subject to change without notice, and certain features may not be available upon initial product release. Contact A10 Networks, Inc. for current information regarding its products or services. A10 Networks, Inc. products and services are subject to A10 Networks, Inc. standard terms and conditions.

ENVIRONMENTAL CONSIDERATIONS

Some electronic components may possibly contain dangerous substances. For information on specific component types, please contact the manufacturer of that component. Always consult local authorities for regulations regarding proper disposal of electronic components in your area.

FURTHER INFORMATION

For additional information about A10 products, terms and conditions of delivery, and pricing, contact your nearest A10 Networks, Inc. location, which can be found by visiting www.a10networks.com.

Table of Contents

Chapter 1: Overview

vThunder TPS for Microsoft Azure is a fully operational software-only version of the ACOS series running vThunder on TPS release. It can be configured by ACOS CLI, GUI, aXAPI, and aGalaxy management system.

vThunder is a virtual appliance that retains most of the functionality available on the hardware-based ACOS appliances. vThunder can be managed the same way as hardware-based ACOS devices and has similar CLI configurations, networking configurations, and GUI presentation. The maximum throughput of vThunder for Azure depends on vThunder software license that is purchased and the type of instance used to deploy vThunder.

Azure Accelerated Networking enables single root input/output virtualization (SR-IOV) on a virtual machine, which uses a high-performance path to bypass the virtual switch. It improves network throughput and reduces latency and jitter. ACOS 5.3.0 supports Azure Accelerated Networking (SR-IOV) on the vThunder TPS.

The following topics are covered:

About Microsoft Azure

Microsoft Azure is Microsoft's cloud computing platform. Azure is an industry leader for both infrastructure-as-a-service (IaaS) and platform-as-a-service (PaaS). Azure offers a combination of managed and unmanaged services that allows customers to deploy and manage their applications as per their needs.

The Azure cloud computing platform runs on Microsoft data centers and is globally distributed across more than a dozen countries. Such global distribution ensures that the customers receive high performance regardless of their location.

Azure can support virtually any operating system from Windows to Linux, any programming language from Java to C++, and any database from SQL to Oracle. Azure also offers 99.95% uptime and is the platform that Microsoft uses to run many of its popular services, such as Bing, Skype, Xbox, and Office 365.

Microsoft Azure uses the following tools to create and manage resources:

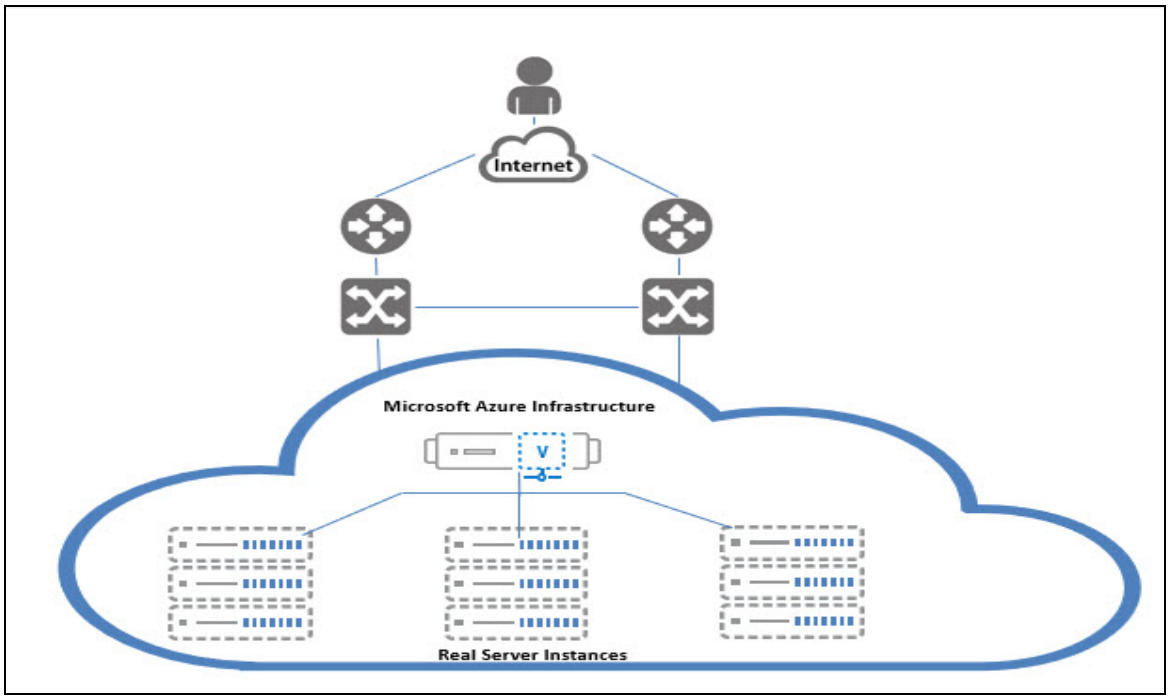
- **Azure Portal** — Azure Portal is a web console to create and monitor Azure resources. For more information, see https://azure.microsoft.com/en-in/features/azure-portal/.
- **Azure PowerShell** — Azure PowerShell is a set of cmdlets used for managing Azure resources from the command line. Azure PowerShell can be launched from a browser within the Azure Cloud Shell or the software can be installed on the system to start a local PowerShell session. For more information, see https://docs.microsoft.com/en-us/powershell/.
- **Azure CLI** — Azure CLI can also be launched from a browser within the Azure Cloud Shell or the software can be installed on the system to start a local CLI session. For more information, see https://docs.microsoft.com/en-us/cli/azure/overview?view=azure-cli-latest.

You can launch Cloud Shell from the top navigation bar of the Azure portal as shown in the figure below:

FIGURE 1-1: Launching Cloud Shell

The following figure shows how vThunder fits into the Microsoft Azure infrastructure:

FIGURE 1-2: vThunder TPS for Microsoft Azure



Microsoft Azure Terminology

Some Azure terminologies that are used in the guide are mentioned below:

- **Azure account** — The Azure account that is created has different support plans for different regions. For more information on different Azure regions and the availability of types of virtual machines in these regions, see https://docs.microsoft.com/en-us/azure/virtual-machines/linux/overview.
- **Resource group** — A resource group is a logical group of all the resources that are related to an Azure solution. Azure offers flexibility in the allocation of resources to the resource groups. For more information, see https://docs.microsoft.com/en-us/azure/azure-resource-manager/resource-group-overview.
- **Availability set** — An availability set is a logical grouping of Azure VM resources so that each VM resource is isolated from other resources when deployed. This hardware isolation ensures that a minimum number of VMs are impacted during a failure. For more information, see https://docs.microsoft.com/en-us/azure/virtual-machines/windows/tutorial-availability-sets.
- **Virtual Machine Scale Sets (VMSS)** — A virtual machine scale set is a group of identical, load balanced VMs. The Azure VMSS can be configured to automatically increase the number of

VM instances or decrease the number of VM instances based on demand or on a predefined schedule. It is used to ensure high availability. For more information, see https://docs.microsoft.com/en-us/azure/virtual-machine-scale-sets/overview.

- **Gateway Load Balancer (GWLB)** — A gateway load balancer is used to easily deploy, scale, and manage your third-party virtual appliances. It provides one gateway for distributing traffic across multiple virtual appliances while scaling them up or down, based on demand. For more information, see https://docs.microsoft.com/en-us/azure/architecture/guide/technology-choices/load-balancing-overview.

NOTE: ACOS 5.3.0-SP2 is required to implement GWLB with TPS.

- **Virtual network** — The Microsoft Azure Virtual Network service enables resources to securely communicate with other resources in an Azure network in the cloud. A virtual network is therefore logical isolation of the Azure cloud for an Azure account. Different virtual networks can be connected to on-premises networks. For more information, see https://docs.microsoft.com/en-us/azure/virtual-network/virtual-networks-overview.
- **Network security group (NSG)** — A network security group (NSG) contains a list of security rules that allow or deny network traffic to the resources that are connected to Azure virtual networks (VNet). The NSGs can be associated with subnets or individual Network Interface Card (NICs) attached to the VMs. When an NSG is associated with a subnet, the rules apply to all the resources connected to the subnet. For more information, see https://docs.microsoft.com/en-us/azure/virtual-network/network-security-groups-overview.

About vThunder Licenses

The GLM is the master licensing system for A10 Networks. The GLM is managed by A10 Networks and is the primary portal for license management for A10 products. The GLM provides GUI where advanced licensing functions can be viewed and managed. Creating a GLM account is optional. The ACOS CLI or GUI can be used to procure licenses for the ACOS devices. A GLM account enables a user to perform advanced licensing functions and also to view and monitor device usage. The GLM portal is available at https://glm.a10networks.com. If you do not have a GLM account, contact A10 Sales.

Without a license, vThunder cannot run production traffic, and the bandwidth is sufficient only for testing network connectivity. After deploying vThunder TPS on Microsoft Azure Cloud, a vThunder license is required to pass live traffic.

A10 Networks offers the following types of licenses to deploy vThunder TPS instances.

- **Trial license** —This mode creates a trial license in the ACOS GUI.
  For more information, see *Global License Manager User Guide*.
- **Capacity Pool (FlexPool) license** — This Bring Your Own License (BYOL) model enables a user to subscribe to a specific bandwidth pool in the Global License Manager (GLM) for a specific period with an additional option of automatically renewing the license before the expiry date.

The capacity pool (FlexPool) license is not node-locked. Multiple ACOS devices can be configured to share the bandwidth from the common license pool. For more information, see *Capacity Pool License User Guide*. For license purchase, contact A10 Sales.

NOTE: When a vThunder license expires, vThunder functionality continues with reduced bandwidth.

To view any of the above license types, features, and the procedure to activate the license, follow the steps mentioned below:

1. Sign In to Global License Manager.
2. Enter your valid A10 **Email**, **Password**, and then click **Sign In**.
   The A10 product documentation page is displayed.
3. On the *A10 Products* page, go to the **Installation Guides for Form Factors** section.
   Choose the product.
4. Click the **View** tab. The Software Installation Guides page is displayed.
5. Click the **View Licensing Guides** option. The portal displays the *Licensing User Guide* section.
6. Click **Download PDF** tab to open the appropriate Global License Manager guide.

Support for NICs

Multi-NIC vThunder TPS deployment is supported on Azure Cloud. The number of interfaces that can be created depends on the VM size provided by Azure. For more information on different VM sizes and the number of NICs supported for each VM size, see https://docs.microsoft.com/en-us/azure/virtual-machines/linux/sizes.

NOTE: vThunder TPS requires 3 or more NICs to function.

To create a Multi-NIC VThunder VM in the Azure portal, first create a single NIC vThunder VM and then use the Azure portal (Azure Power Shell or the Azure CLI) to add more NICs to the VM. For more information, see Creating a vThunder VM.

NOTE: vThunder must be shutdown before adding any additional NICs.

vThunder does not support the hotplug devices. If any hotplug events are detected in a network device, it can result in traffic loss and may require a reboot of the Azure instance.

The following operations are supported for multiple NICs:

- The Azure portal can be used to instantiate a vThunder instance that supports four NICs. If only two NICs are created, two more NICs can be added before shutting down the instance. The Power Shell or Azure CLI can be used to add the remaining NICs. For more information, see Adding More NICs Using the Azure CLI.
- The Azure portal can be used to instantiate an instance that supports only two NICs. To add more NICs, shut down the instance and change the VM size from within the Azure Portal as described in the topic Changing the VM Size. After that, repeat the steps mentioned in the topic Adding More NICs Using the Azure CLI.
- The Azure portal can be used to instantiate an instance with multiple NICs, then shut down the VM and delete NICs as described in the topic Deleting NICs Using the Azure CLI.

NOTE: Users cannot delete all the NICs from a VM.

In the following topic Adding NICs to vThunder VM , a vThunder instance is created with the following interfaces and each interface is associated with a different subnet:

- Management – Dedicated management interface
- Ethernet 1 – Data interface
- Ethernet 2 – Data interface

In a typical deployment, one of the data interfaces is connected to the server farm, and the other data interface is connected to the clients. However, one-arm deployment is also supported which requires one data port and one management port. You also can add additional data interfaces as needed.

Limitations

A user should consider the following limitations while using vThunder for Azure:

- It is recommended that you configure “ip address DHCP” before performing other configurations because there is no predefined DHCP in the start-up configuration file.
  For more information, see Configuring DHCP in vThunder TPS.
- LACP and Static trunk groups are not supported on Azure Cloud.
  For more information, see Configuring DHCP in vThunder TPS.
- Hotplug and Hotplug removal is not supported in Azure instance.
- Port Mirror is not supported.
- vThunder for Azure does not support L3V partition and service partition.
- RIP (v1 and v2), OSPF, and ISIS routing protocols are not supported.
- VLAN, Tagged VLAN, and Virtual Ethernet (VE) interfaces are not supported.
- Layer 2 Switching (VLAN) is not supported.
- Layer 2 deployment is not supported.
- Bridge Protocol Data Unit (BPDU) Forward Group is not supported.

- If the endpoint port number in the Azure portal is changed, the Internet browser's cache should be cleared before attempting to navigate to the vThunder GUI. If the cache is not cleared, the browser uses the previously saved public port and fails to access the vThunder GUI.
- System promiscuous mode is not supported by Microsoft Azure.
- At the interface Ethernet config level, the following commands are disabled:
  - **`mtu`**
  - **`trunk-group`** (command exists, but the function is disabled)
  - **`device-context`**
  - **`duplexity`**
  - **`flow-control`**
  - **`monitor`**
  - **`speed`**
  - **`use-if-ip`**
- The reload command causes kernel panic on Azure due to the limitation imposed by DPDK Netvsc PMD. The reboot command can be used whenever reload is required. For information about the limitation, see https://doc.dpdk.org/guides/rel_notes/known_issues.html#netvsc-driver-and-application-restart.

Chapter 2: Installing vThunder on Microsoft Azure

This chapter describes how to deploy vThunder on Microsoft Azure.

The following topics are covered:

vThunder Images Available on Microsoft Azure

The following is the list of images available for vThunder:

TABLE 2-1 : vThunder SKUs

SKUs	*Offer*	*Publisher Name*	*Location*
`vthunder-tps-byol`	`a10-vthunder-tps`	`a10networks`	`any`

For more information, contact A10 Sales.

System Requirements

Supported Version

Supported version for TPS : *5.0.2, 5.3.0 SP1, and 5.3.0 SP2*

Supported VM Sizes

The supported Azure VM sizes for TPS include VMs from D-series. See the following table:

TABLE 2-2 : Verified VM sizes

Series	*VM Size for TPS*
D series	Standard D8_v3 Standard D8s_v3

For more information, see https://docs.microsoft.com/en-us/azure/virtual-machines/sizes-general.

Creating a vThunder VM

You can create vThunder TPS VM on Microsoft Azure. This topic explains how to create vThunder TPS VM with multiple NICs.

Prerequisites

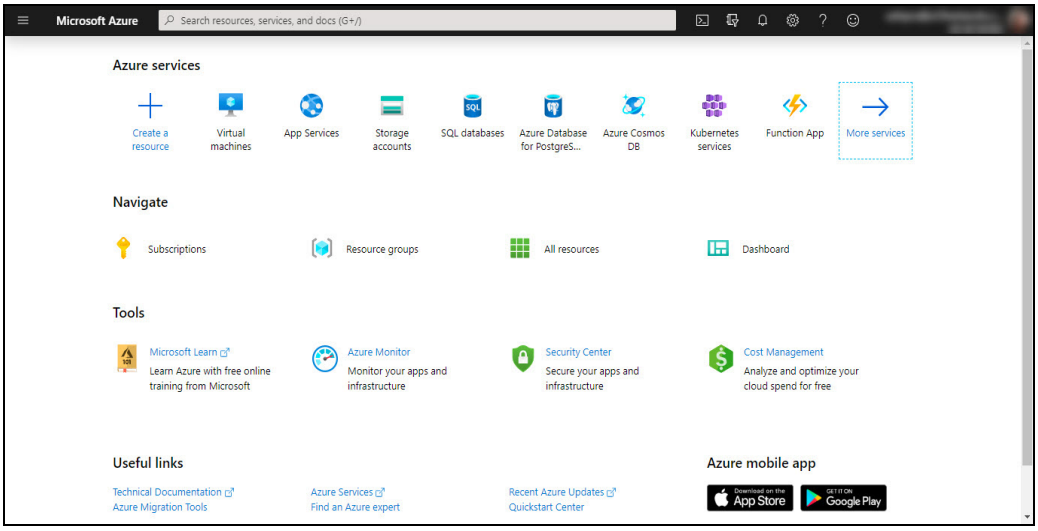
Before deploying vThunder, set up an account with Microsoft Azure or use the MSDN credentials, or use a free trial account from the following location: http://azure.microsoft.com/en-us/-pricing/free-trial/

Deploying a vThunder TPS

To create a vThunder TPS, perform the following steps:

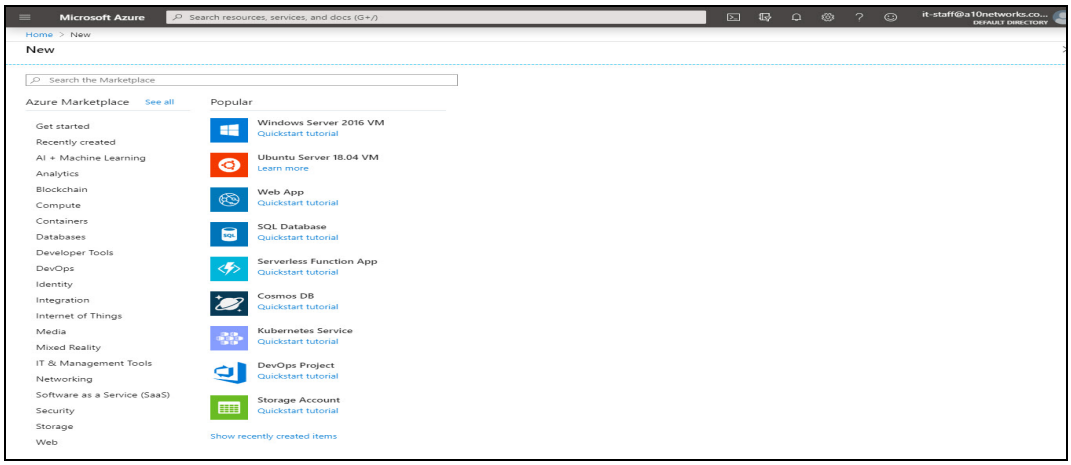
1. Navigate to https://portal.azure.com. The **Microsoft Azure - Services** window is displayed.

   FIGURE 2-3: Microsoft Azure - Services window



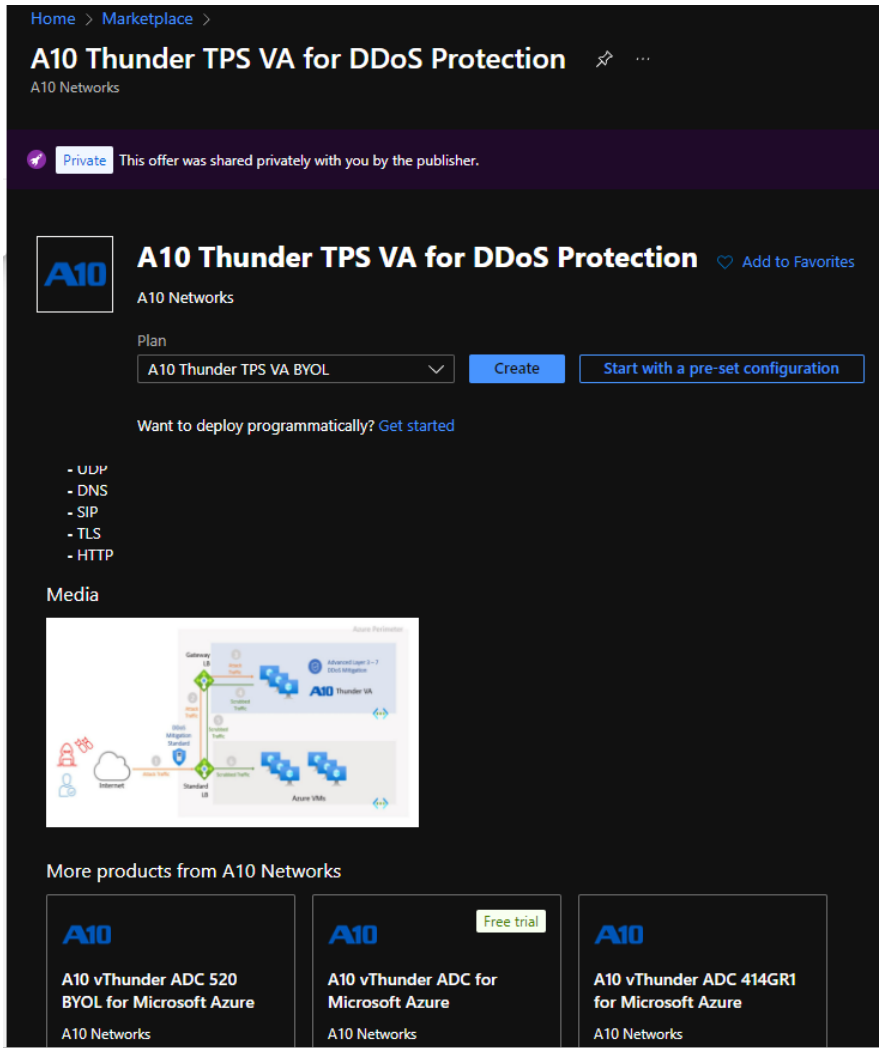
2. Click **Create a resource** from the Microsoft Azure Services menu options.
   The **New** window with **Search the Marketplace** text box is displayed.

   FIGURE 2-4: New window



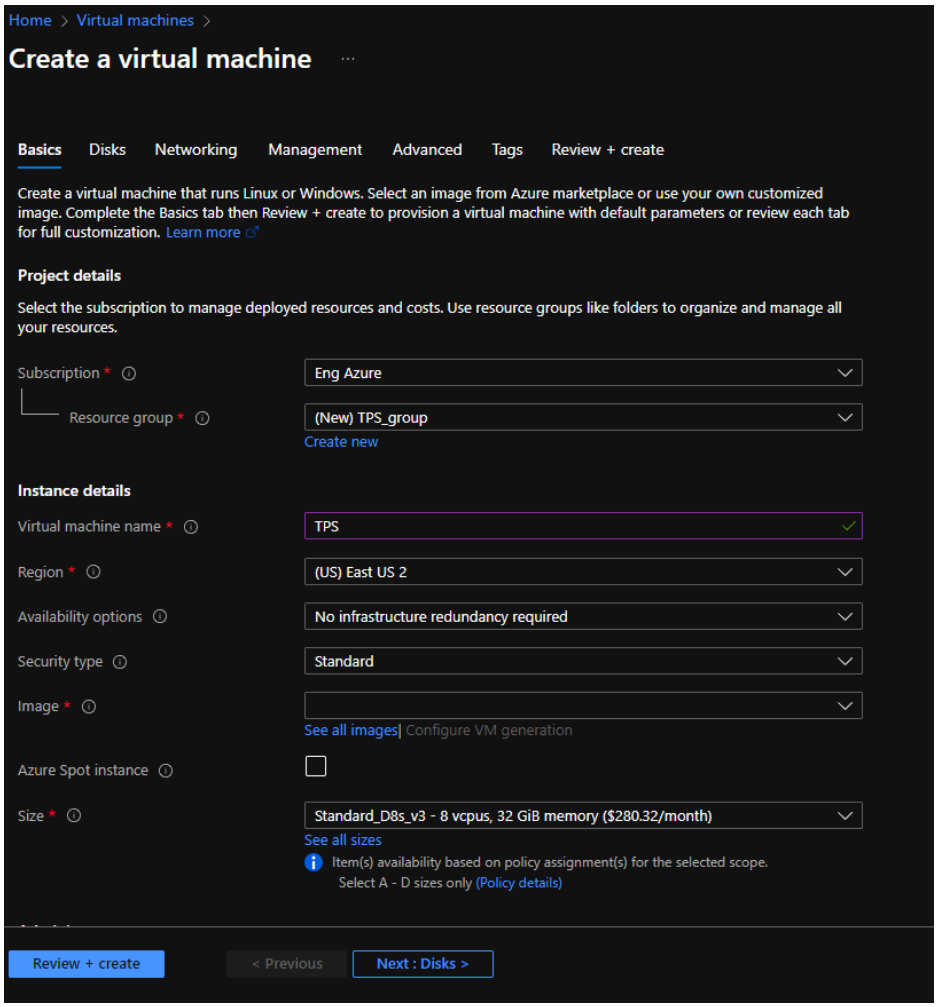
3. Enter the search string 'A10 Networks' and press **Enter**.
   The search displays several types of images that can be grouped into two types, BYOL and fixed throughput images. As the name suggests, for BYOL images, contact A10 Networks Sales for the required license. For fixed throughput images, the license is preinstalled.
4. Select the required image. For example, A10 vThunder TPS for Microsoft Azure. The selected image window is displayed.

FIGURE 2-5: A10 vThunder TPS for Microsoft Azure window



5. Click **Create**. The **Create virtual machine** workflow tabs are displayed.

FIGURE 2-6: Create a virtual machine window

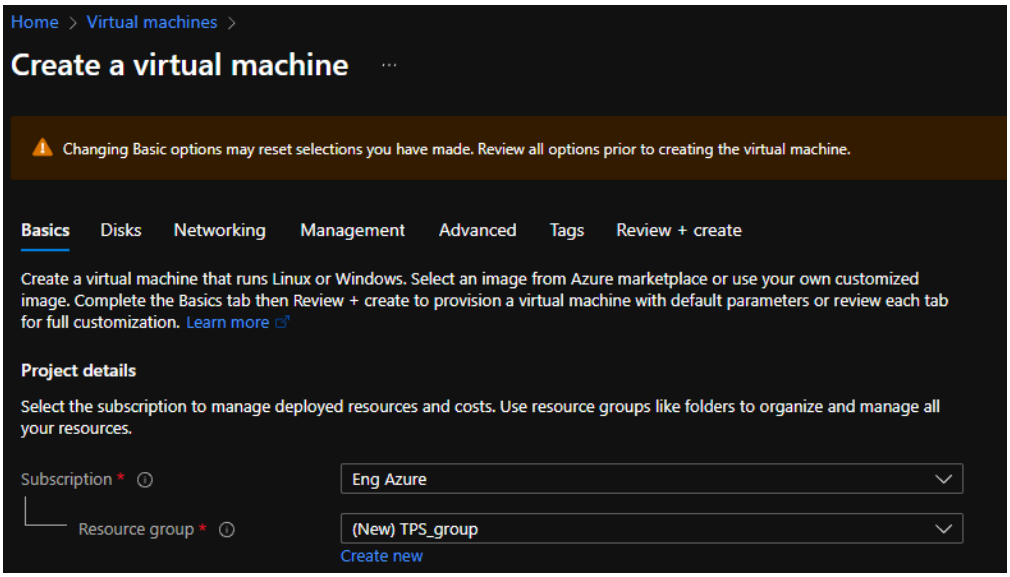


a. Click the **Basics** tab. The Basics window is displayed.

   In the **Basics** window, enter the following details:

   i. Under the Project details section, select the correct **Subscription** and **Resource group**, or choose to **Create new** resource group.

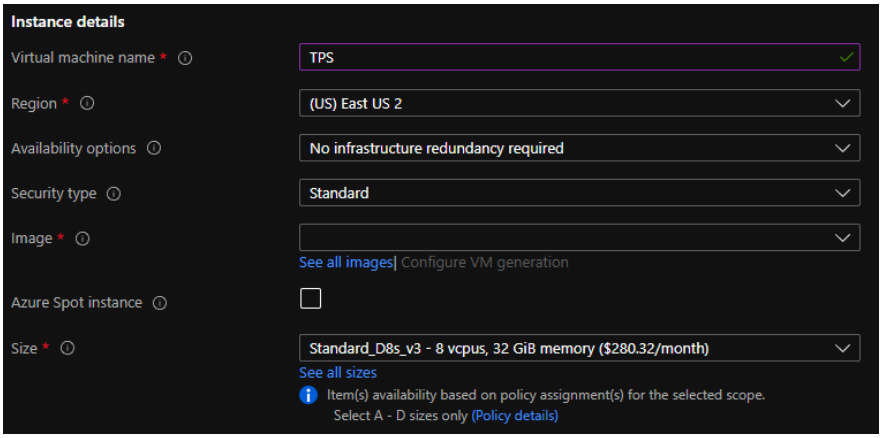
FIGURE 2-7: Basics window- Project details



NOTE: A resource group is a container that holds related resources for an Azure solution.

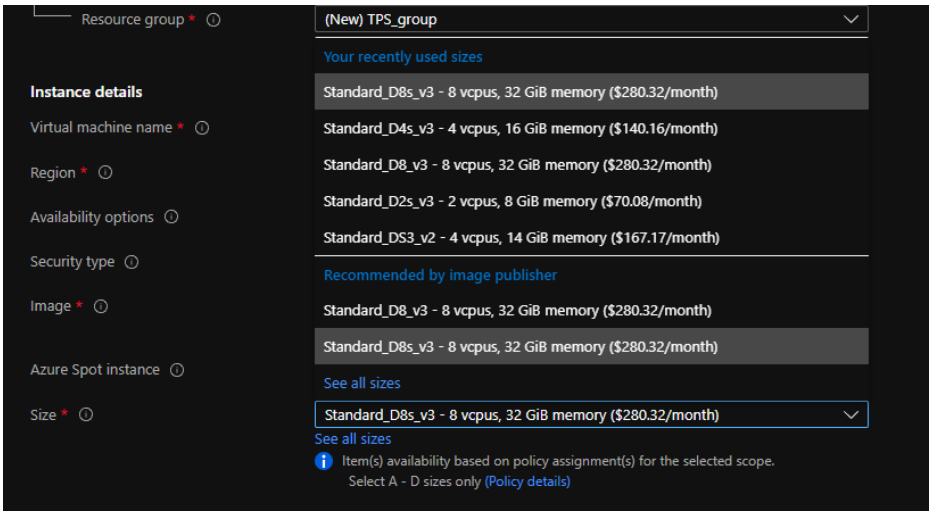
ii. In the **Instance details** section, enter the **Virtual machine name**, select the **Region**, and choose the A10 vThunder **Image** from the drop-down list.

FIGURE 2-8: Basic window- Instance details



iii. Click **Change Size** to select the size of a virtual machine and its features. In the **Select a VM size** window, select any one of the recommended options and click **Select**.

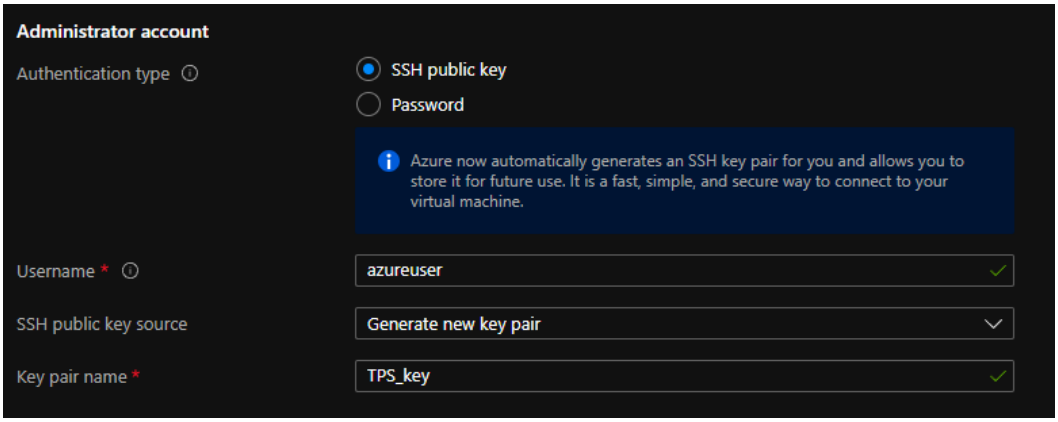
FIGURE 2-9: Selecting a VM Size



NOTE: Each pane displays a combination of Family, vCPUs, RAM size, data disks, IOPS value, and so on. The default size is set to **Standard DS1 v2**.

iv. In the **Administrator account** section, the **Authentication type** is the **Password** or **SSH public key**.

FIGURE 2-10: Basic details - Administrator account and Inbound port rules



i. If **SSH Public Key** is selected, enter the **Username** and the **SSH public key**.

ii. If **Password** is selected, enter the **Username** and **Password**. The entered password must have 12 characters, one lower case, one upper case, a digit, and one

special character.
Note: Re-entered password must match the initially entered Password.

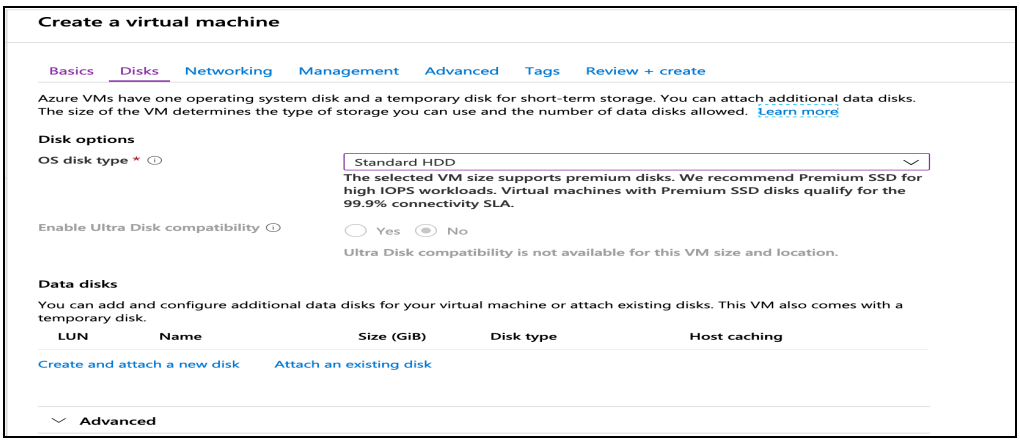
v. Under **Inbound port rules** > **Public inbound ports**, select **Allow selected ports** . Select SSH (22) and HTTP (80) from the drop-down list.

Retain default values for the remaining fields and select **Review + create** at the bottom of the page.

Alternatively, perform the steps mentioned below:

b. Click the **Disks** tab. The Disk option window is displayed.

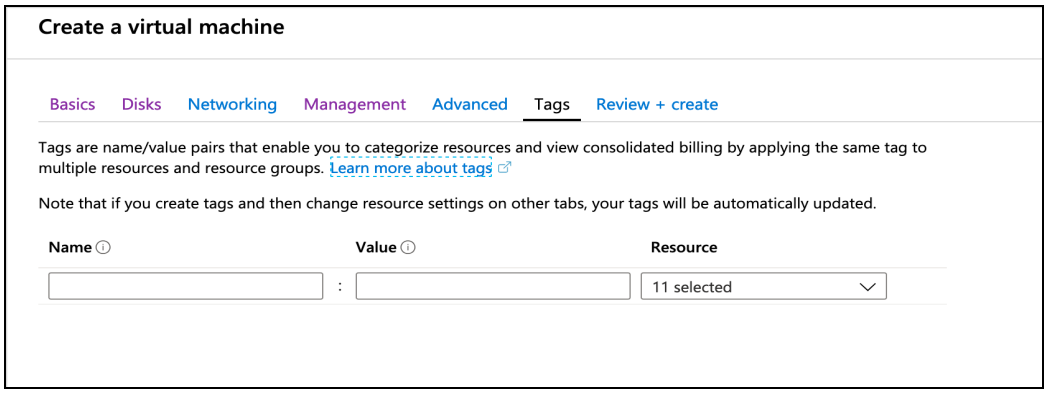
FIGURE 2-11: Disk window



Under **Disk options**, select the OS disk type from the available list of options. Retain default values for the remaining fields.

c. Click the **Tags** tab. The Tags window is displayed.

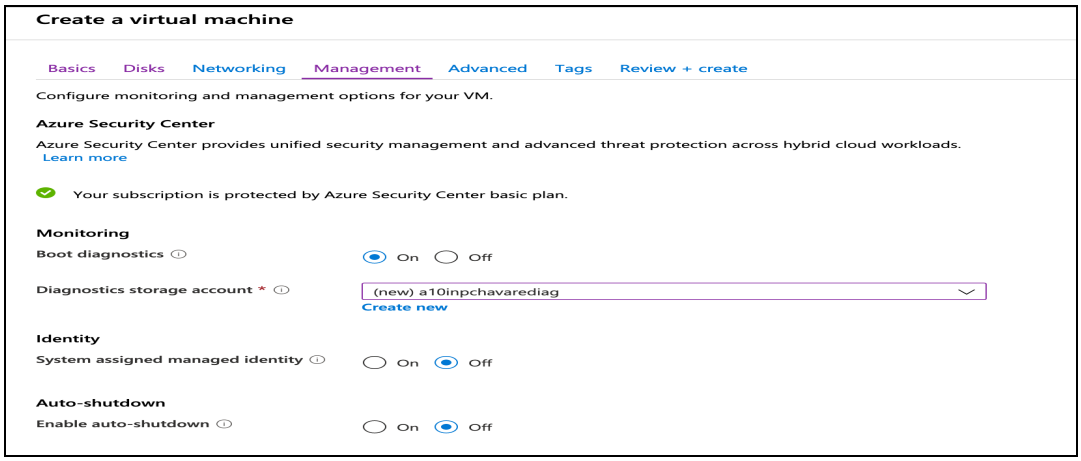
FIGURE 2-12: Tags window



Use tags to categorize resources and view consolidated billing that is paired with name or value.

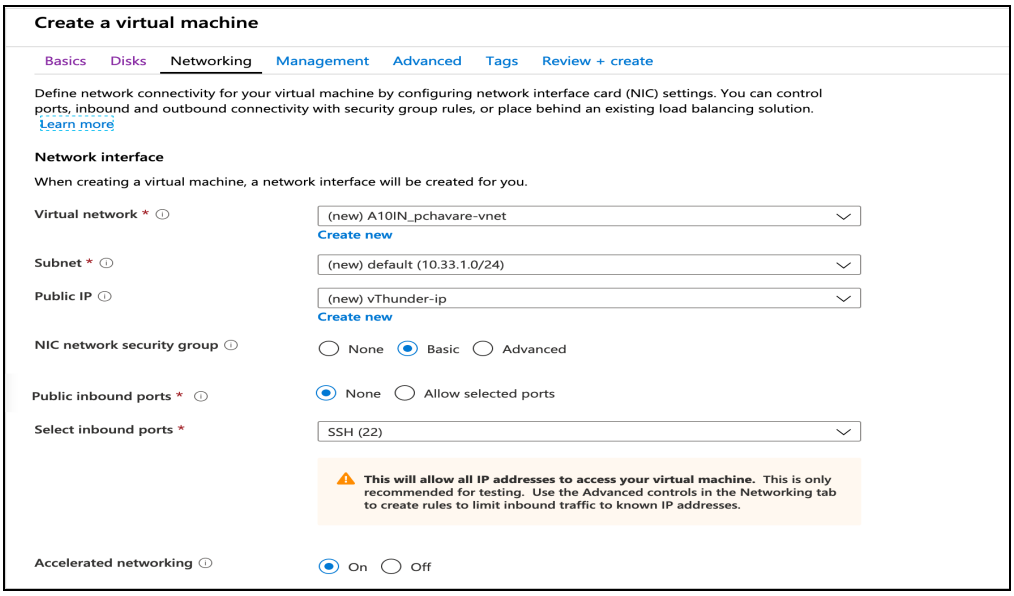
d. Click the **Management** tab to configure monitoring and management options for the VM.

FIGURE 2-13: Management window



e. Click the **Networking** tab. The Networking window is displayed.

FIGURE 2-14: Networking window



i. Select **Virtual Network**, **Subnet**, and **NIC network security group**.

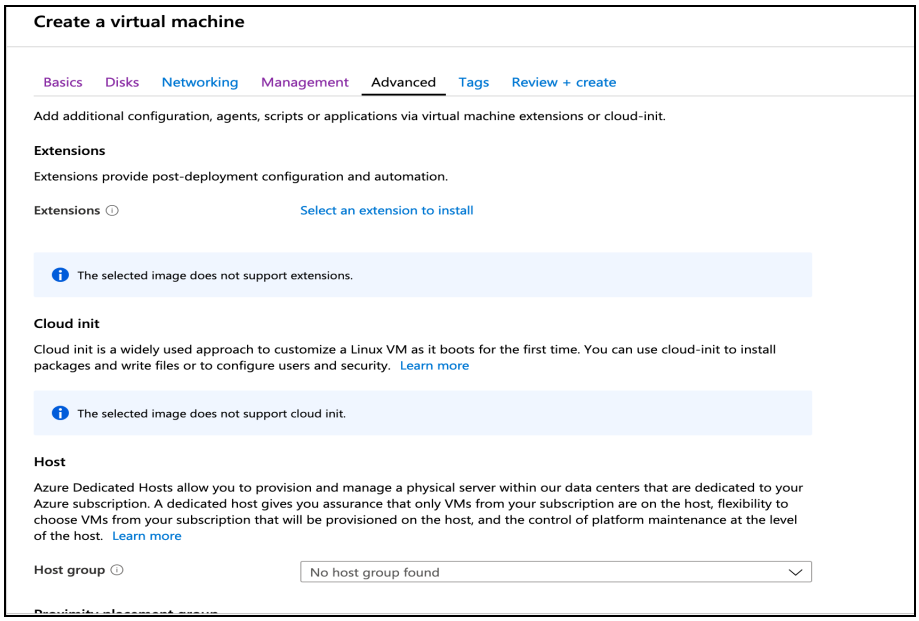
NOTE: To create a new virtual network, subscription, resource group, name, and location must be selected.

ii. Select the **Public inbound ports** as **None**.

iii. Select the **Select inbound ports** from a list of options.

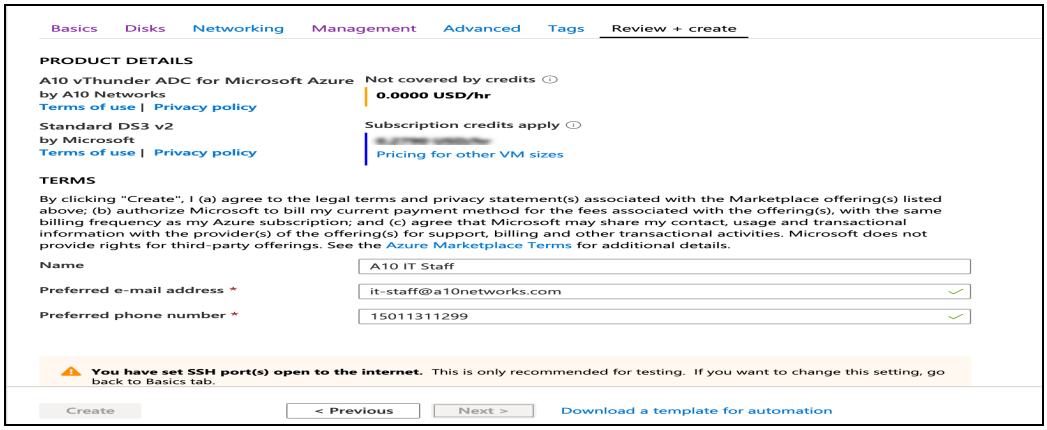
f. Click the **Advanced** tab to add additional details about **Extensions**, **Cloud-init** or **Host.**

FIGURE 2-15: Advance window



6. Click the **Review + create** to view the **Product details, Terms** of use with user details.

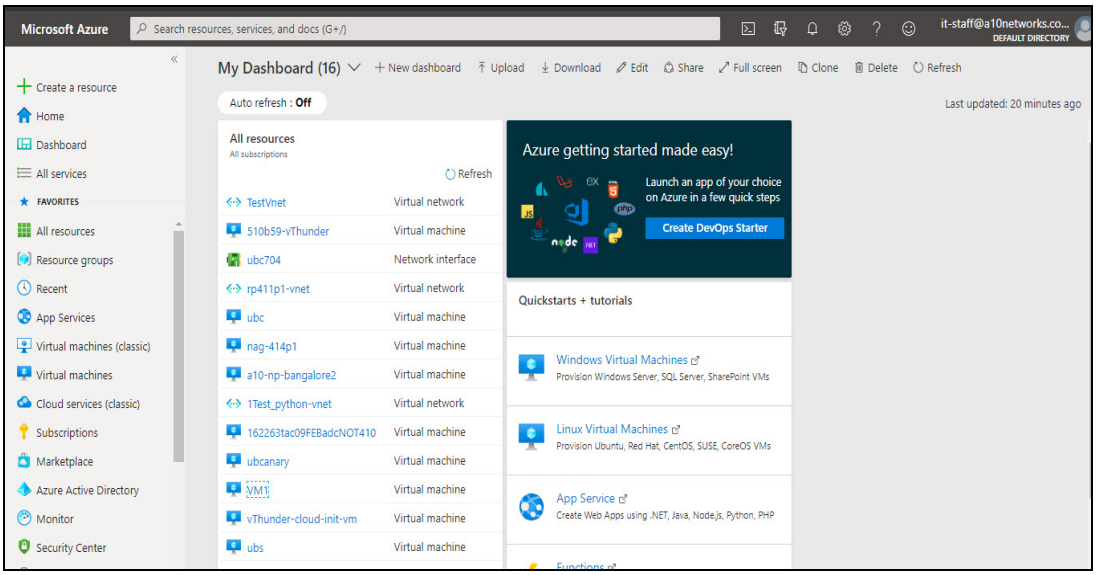
FIGURE 2-16: Review + create window



The preferred e-mail address and phone number display a green check. Click **Create** button to create a virtual machine. In the Azure My Dashboard window, a pane displays the VM just created.

NOTE: Creating the VM may take several minutes depending on several factors.

FIGURE 2-17: My Dashboard - All resources window



Adding NICs to vThunder VM

To create multiple NICs on a vThunder instance, use any one of the following methods:

- Adding NICs on vThunder Using Azure Portal
- Adding NICs on vThunder Using Azure PowerShell

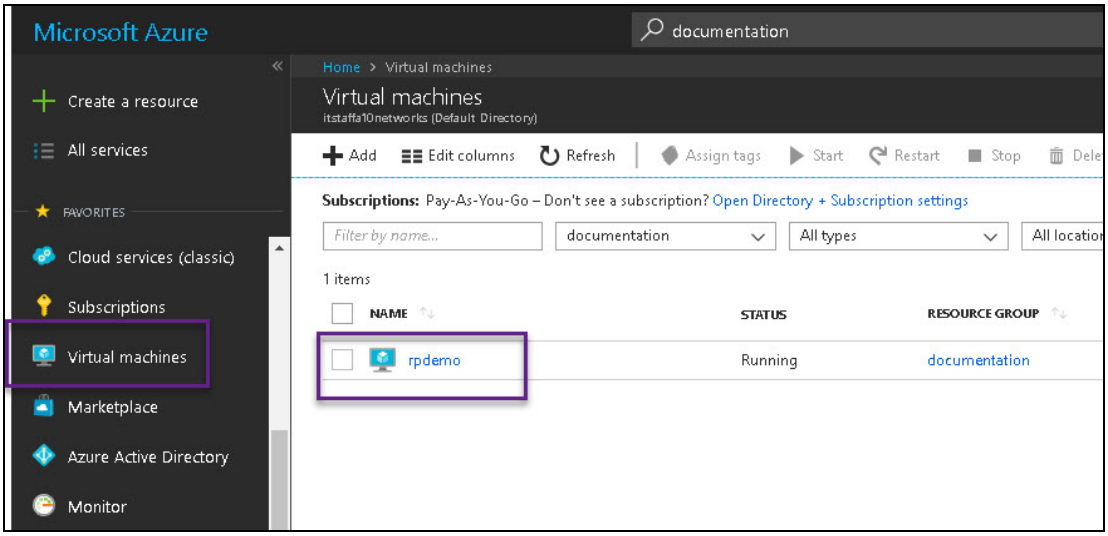
After a VM is created with multiple NICs, use the Azure portal to configure the VM.

Adding NICs on vThunder Using Azure Portal

You can create vThunder TPS VMs with multiple NICs on the Microsoft Azure portal. Perform the steps mentioned in the topic Creating a vThunder VM to create a VM with one interface. After creating a VM, perform the following steps to creating NICs:

1. Click **Virtual machines** and select the VM from the right-pane.

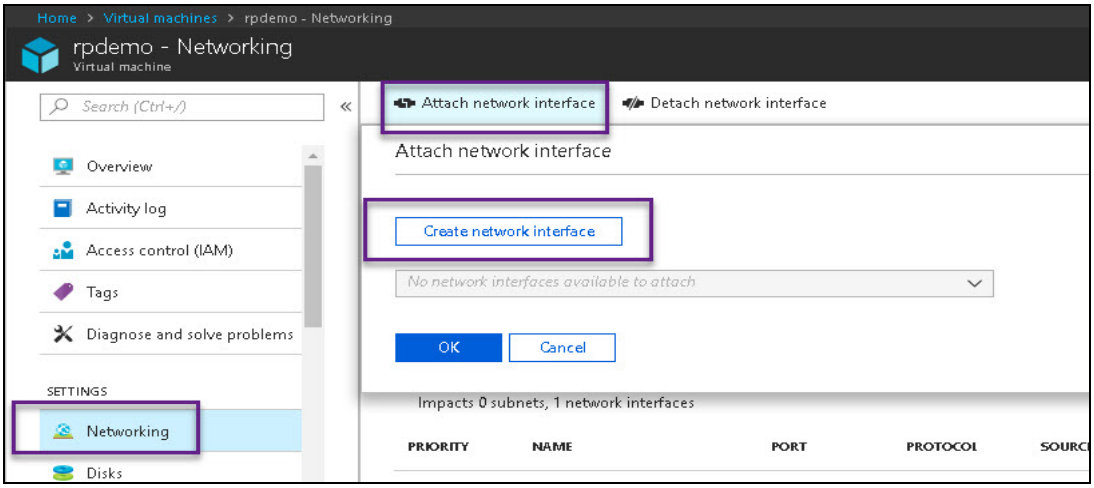
   FIGURE 2-18: Virtual machines window



2. In the **Virtual machines** window, click **Stop** to stop the VM.

3. From the left pane, select **Networking**. From the right pane, select **Attach network interface** > **Create network interface**.

   FIGURE 2-19: Attach network interface

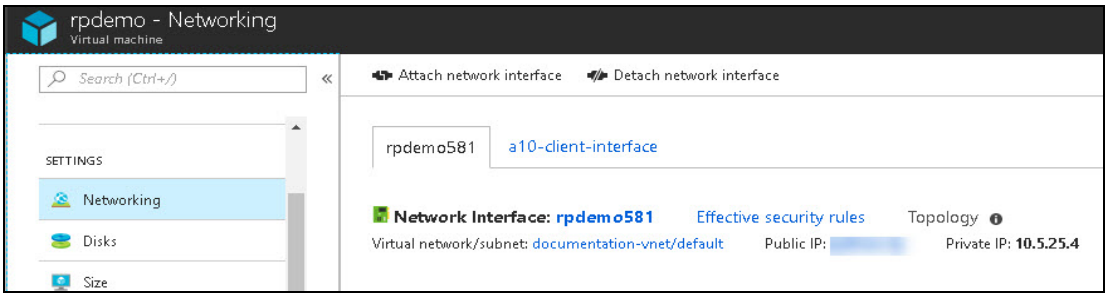


4. On the **Create network interface** page, enter the following information:

- **Name**: a10-client-interface
- **Virtual Network**: Retain the default value.
- **Subnet**: Select one of the existing subnets as appropriate. Each interface must belong to a different subnet.
- **Private IP address assignment**: Dynamic
- **Network security group**: Select one of the existing groups or create a new one.
- **Private IP address (IPv6)**: Not required
- **Subscription**: Retain the default value.
- **Resource group**: Select one of the existing resource groups or create a new one.
- **Location**: Retain the default value.

5. Select the newly created network interface from the drop-down of the right-pane, and select **OK**.

   FIGURE 2-20: VM with Two Network Interfaces



6. Similarly, create and attach another network interface card for the server-side connection.

NOTE: Applicable for ACOS 5.0.2, the Thunder TPS supports Azure Accelerated Networking which improves network performance by using a high-performance path and reducing latency. It is only supported on the data interfaces and not supported on the management interface. See below for details for enabling Accelerated Networking.

7. After the interfaces are created and attached, start the VM.

Adding NICs on vThunder Using Azure PowerShell

In this example, a vThunder VM with three NICs is created by using the Azure PowerShell. One NIC is used for the management interface while the other two NICs are used for the data interfaces.

NOTE: If the inputs provided to the script are not accepted by the Azure cloud portal, the deployment fails.

To deploy Azure VM from the marketplace, perform the steps mentioned below:

1. Deploy the Azure VM from the marketplace:

```
#Deploying azure VM from marketplace
  Login-AzureRmAccount

$location = Read-Host 'Enter the location'
$resourceGroup = Read-Host 'Enter resource group name'
$storageaccount = Read-Host 'Enter storage account name'
$vmName = Read-Host 'VM Name'
$vmSize = Read-Host 'Enter VM size'
```

2. Create a new resource for the deployment:

```
#Create new resource group for deployment
New-AzureRmResourceGroup -Name
 $resourceGroup -Location
 $location
```

3. Create a storage account for the new resource:

```
#Create storage account
New-AzureRmStorageAccount
 -ResourceGroupName $resourceGroup
 -AccountName $storageaccount
 -Location $location
 -SkuName Standard_RAGRS
 -Kind StorageV2
 -AssignIdentity
```

4. Create a virtual network, subnet, and a public IP address. These resources are used to provide network connectivity to the VM and connect it to the internet:

```
# Create a subnet configuration
$mgmtsubnet = New-AzureRmVirtualNetworkSubnetConfig
 -Name "subnet1"
 -AddressPrefix "192.168.1.0/24"
$data1subnet = New-AzureRmVirtualNetworkSubnetConfig
 -Name "subnet2" -AddressPrefix "192.168.2.0/24"
$data2subnet = New-AzureRmVirtualNetworkSubnetConfig
 -Name "subnet3" -AddressPrefix "192.168.3.0/24"

# Create a virtual network
$vnet = New-AzureRmVirtualNetwork
```

```
 -ResourceGroupName $resourceGroup
 -Location $location
 -Name "Vnet"
 -AddressPrefix 192.168.0.0/16
 -Subnet $mgmtsubnet,$data1subnet,$data2subnet

# Create a public IP address and specify a DNS name
$mgmtpip = New-AzureRmPublicIpAddress
 -ResourceGroupName
$resourceGroup
 -Location $location
 -AllocationMethod Dynamic
 -IdleTimeoutInMinutes 4
 -Name "myip$(Get-Random)"
$data1pip = New-AzureRmPublicIpAddress
 -ResourceGroupName $resourceGroup
 -Location $location
 -AllocationMethod Dynamic
 -IdleTimeoutInMinutes 4
 -Name "myip$(Get-Random)"
$data2pip = New-AzureRmPublicIpAddress
 -ResourceGroupName $resourceGroup
 -Location $location
 -AllocationMethod Dynamic
 -IdleTimeoutInMinutes 4
 -Name "myip$(Get-Random)"
```

5. Create an Azure Network Security Group and traffic rule. The Network Security Group secures the VM with inbound and outbound rules. In the following example, an inbound rule is created for TCP port 22 that allows SSH connections. To allow incoming web traffic, an inbound rule for TCP port 80 is also created:

```
# Create an inbound network security group rule for port 22
$nsgRuleSSH = New-AzureRmNetworkSecurityRuleConfig
 -Name "myNetworkSecurityGroupRuleSSH"
  -Protocol "Tcp"
 -Direction "Inbound"
 -Priority 1000 -SourceAddressPrefix *
 -SourcePortRange *
 -DestinationAddressPrefix *
 -DestinationPortRange 22
 -Access "Allow"

# Create an inbound network security group rule for port 80
$nsgRuleWeb = New-AzureRmNetworkSecurityRuleConfig
 -Name "myNetworkSecurityGroupRuleHTTP”
```

```
  -Protocol "Tcp"
 -Direction "Inbound"
 -Priority 1001
 -SourceAddressPrefix *
 -SourcePortRange *
 -DestinationAddressPrefix *
 -DestinationPortRange 80
 -Access "Allow"

# Create a network security group
$nsg = New-AzureRmNetworkSecurityGroup
 -ResourceGroupName $resourceGroup
 -Location $location
 -Name "myNetworkSecurityGroup"
 -SecurityRules $nsgRuleSSH,
$nsgRuleWeb
```

a. Create a virtual network interface card (NIC) with **`New-AzNetworkInterface`**. The virtual NIC connects the VM to a subnet, Network Security Group, and public IP address.

```
# Create a virtual network card and associate with public IP address and
NSG
$mgmtsubnet = $vnet.Subnets | ?{ $_.Name -eq 'subnet1' }
$mgmtnic = New-AzureRmNetworkInterface
 -ResourceGroupName $resourceGroup
 -Name "nic1"
 -Location $location
 -SubnetId $mgmtsubnet.Id
 -PublicIpAddressId $mgmtpip.Id
 -NetworkSecurityGroupId
$nsg.Id
```

NOTE: Applicable for ACOS 5.0.2, Accelerated Networking is only supported on the data interfaces and not supported on the management interface.

```
$data1subnet = $vnet.Subnets | ?{ $_.Name -eq 'subnet2' }
$data1nic = New-AzureRmNetworkInterface
 -ResourceGroupName $resourceGroup
 -Name "nic2"
 -Location $location
 -SubnetId $data1subnet.Id
 -PublicIpAddressId $data1pip.Id
 -NetworkSecurityGroupId $nsg.Id
```

To create data interface 1 and enable Accelerated Networking on data interface 1 (nic2), use the following commands:

```
$data1nic = New-AzureRmNetworkInterface
 -ResourceGroupName $resourceGroup
 -Name "nic2"
 -Location $location
 -SubnetId $data1subnet.Id
 -PublicIpAddressId $data1pip.Id
 -NetworkSecurityGroupId $nsg.Id
 -EnableAcceleratedNetworking

$data2subnet = $vnet.Subnets | ?{ $_.Name -eq 'subnet3' }

$data2nic = New-AzureRmNetworkInterface
 -ResourceGroupName $resourceGroup
 -Name "nic3"
 -Location $location
 -SubnetId $data2subnet.Id
 -PublicIpAddressId $data2pip.Id
 -NetworkSecurityGroupId $nsg.Id
```

Similarly, use the following commands to create data interface 2 (nic3) with Accelerated Networking enabled:

```
$data2nic = New-AzureRmNetworkInterface
 -ResourceGroupName $resourceGroup
 -Name "nic3"
 -Location $location
 -SubnetId $data1subnet.Id
 -PublicIpAddressId $data1pip.Id
 -NetworkSecurityGroupId $nsg.Id
 -EnableAcceleratedNetworking
```

NOTE: For Accelerated Networking support with multiple NICs, Accelerated Networking must be enabled on both data interfaces.

6. To create a VM in PowerShell, firstly create a configuration that has settings like the image to use, size, and the authentication options. Then the configuration is used to build the VM.

```
# Define a credential object
$name= Read-Host 'Enter Username'
$securePassword = Read-Host 'Enter the password' -AsSecureString
```

```
$cred = New-Object System.Management.Automation.PSCredential ($name, $se-
curePassword)

# Start building the VM configuration
$vmConfig = New-AzureRmVMConfig -VMName
$vmName -VMSize
$vmSize

#Create the rest of configuration
$vmConfig = Set-AzureRmVMOperatingSystem -VM
$vmConfig
 -Linux
 -ComputerName
$vmName -Credential
$cred
$vmConfig = Set-AzureRmVMSourceImage -VM
$vmConfig
 -PublisherName "a10networks"
 -Offer "vthunder-414-gr1"
 -skus "vthunder-414gr1-byol"
 -Version "latest"
$vmConfig = Set-AzureRmVMPlan
 -Name "vthunder-414gr1-byol"
 -Product "vthunder-414-gr1"
 -Publisher "a10networks"
 -VM
$vmconfig

# for bootdiag
$vmConfig = Set-AzureRmVMBootDiagnostics -VM
$vmconfig -Enable
 -ResourceGroupName $resourceGroup
 -StorageAccountName $storageaccount

#Attach the NIC that are created
$vmConfig = Add-AzureRmVMNetworkInterface -VM
$vmConfig -Id
$mgmtnic.Id -Primary
$vmConfig = Add-AzureRmVMNetworkInterface -VM
$vmConfig -Id
$data1nic.Id
$vmConfig = Add-AzureRmVMNetworkInterface -VM
$vmConfig -Id
$data2nic.Id

#Creating VM with all configuration
```

```
New-AzureRmVM -ResourceGroupName
$resourceGroup -Location
$location -VM
$vmConfig
```

Assigning IP Addresses to NICs

An Azure VM can have multiple private and public IP addresses. Guidelines for IP addresses are mentioned below:

- A network interface can have one or more static or dynamic public and private IP addresses assigned to it.
- There is a limit to the number of private and public IP addresses that can be assigned to a network interface depending on the type of Azure subscription availed.
- When there are multiple IP addresses assigned to a network interface, only one IP address can be a primary IP address and the other IP addresses are all secondary IP addresses.

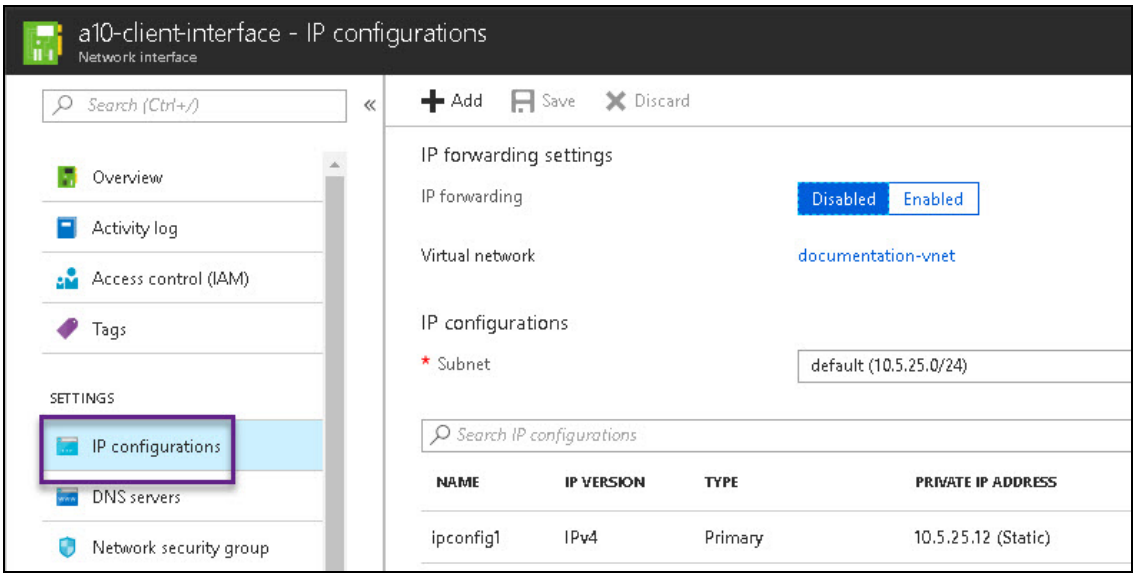
Assigning Primary and Secondary IP Addresses by Using Azure Portal

In this example, the primary IP address is associated with a public IP address, and the secondary IP address is associated with its private IP address.

Perform the following steps to add a primary public IP address to a NIC:

1. From the Microsoft Azure left-most pane, select **Virtual networks**, and then from the list of virtual networks, select the virtual network to which the network interface belongs.
2. Under the virtual network, select the network interface card for which you want to add a public IP address.
3. Under **Settings**, select **IP configurations**.

FIGURE 2-21: Select IP configurations



4. Click the **Public IP address** in the main window.
5. Fill in the following details, and click **Save**:

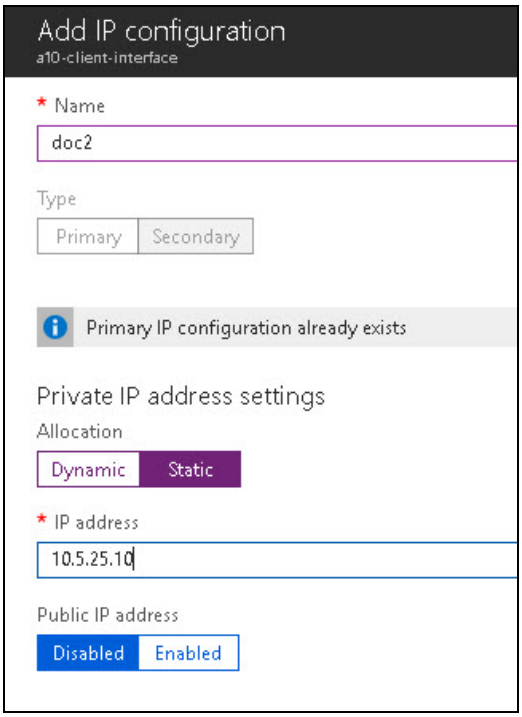
   - **IP forwarding**: Select **Enabled**.
   - **Virtual network**: Select from an existing IP address or create a new one.
   - **IP configurations**: Retain the default value for the **Subnet**.

Perform the following steps to add a secondary IP address to a NIC:

1. From the Microsoft Azure left-most pane, select **Virtual networks**, and then from the list of virtual networks, select the virtual network to which the network interface belongs.
2. Under the virtual network, select the network interface for which you want to add a secondary IP address.
3. Under **Settings**, select **IP configurations** and then **Add** in the main window.
4. In the Add IP configuration window, fill in the following details and click **OK**.

   - **Name**: doc2
   - **Type**: Select **Secondary**. This is the default selection.
   - **Private IP address settings**: Select **Static**. Fill in an IP address.
   - **Public IP address**: Select **Disabled**.

FIGURE 2-22: Add Secondary IP address



The primary and secondary IP addresses are assigned.

Assigning Primary and Secondary IP Addresses by Using Azure CLI

Azure resources cannot receive and send Internet communication without an assigned public IP address. Public IP addresses have a nominal charge. For more information, see https://docs.microsoft.com/en-us/azure/virtual-network/virtual-network-public-ip-address.

To add a primary IP address to a NIC, perform the following steps:

1. Create the public IP address:

```
az network public-ip create -g
testResourceGroup -n testip --dns-name MyLabel --allocation-method dynamic
```

2. Create an IP configuration on the NIC:

```
az network nic ip-config create --name ipconfig2
                                --nic-name data1nic
                                --resource-group testResourceGroup
                                [--application-security-groups]
                                [--lb-address-pools]
                                [--lb-inbound-nat-rules]
```

```
                                    [--lb-name]
                                    [--make-primary]
                                    [--private-ip-address]
                                    [--private-ip-address-version {IPv4, IPv6}]
                                    [--public-ip-address]
                                    [--subnet]
                                    [--vnet-name]
```

The private IP address must be attached to the data interface in Microsoft Azure Portal as a secondary (private) IP address to the interface.

To create a secondary IP address, perform the following steps:

```
az network nic ip-config create --name ipconfigtest
                                --nic-name data1nic
                                --resource-group testResourceGroup
                                [--application-security-groups]
                                [--lb-address-pools]
                                [--lb-inbound-nat-rules]
                                [--lb-name]
                                [--make-primary]
                                [--private-ip-address]
                                [--private-ip-address-version {IPv4, IPv6}]
                                [--subnet]
                                [--vnet-name]
```

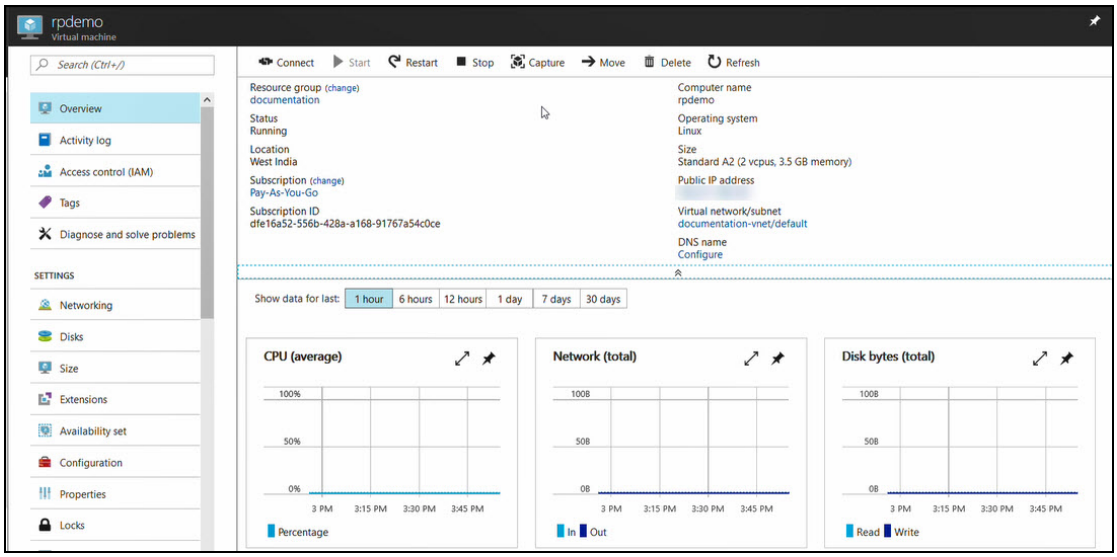
Accessing vThunder

Accessing vThunder Using ACOS CLI

To connect to the VM, perform the steps mentioned below:

1. After the VM is created, type the VM name in the Azure search box and click Enter.
   The search results display the VM.
2. Click on the link to launch the VM details page.
3. Wait until the Status column for the VM changes to **Running**.
   When the status changes to **Running**, a PuTTY session with the virtual machine can be established.
4. Select the public IP address from the VM Overview page.

FIGURE 2-23: VM Overview Page



5. Open an SSH client and access the IP address on the client.
6. Enter the following credentials to access the VM:
   User name: **`admin`**
   Password: **`a10`**
   The vThunder prompt is displayed.

Accessing vThunder Using ACOS GUI

If the vThunder VM uses Network Security Group, then configure endpoint mapping to access the VM by using the ACOS GUI.

For single NIC VMs, launch a web browser and enter the following URL `https://public IP: 8443`. The public IP portion of this URL can be obtained by looking up the public IP address, as described in FIGURE 2-23.

For multi-NIC VMs, enter the URL `https://public IP`. When accessing the web GUI, the default value is port 80.

Configuring Endpoint Mapping

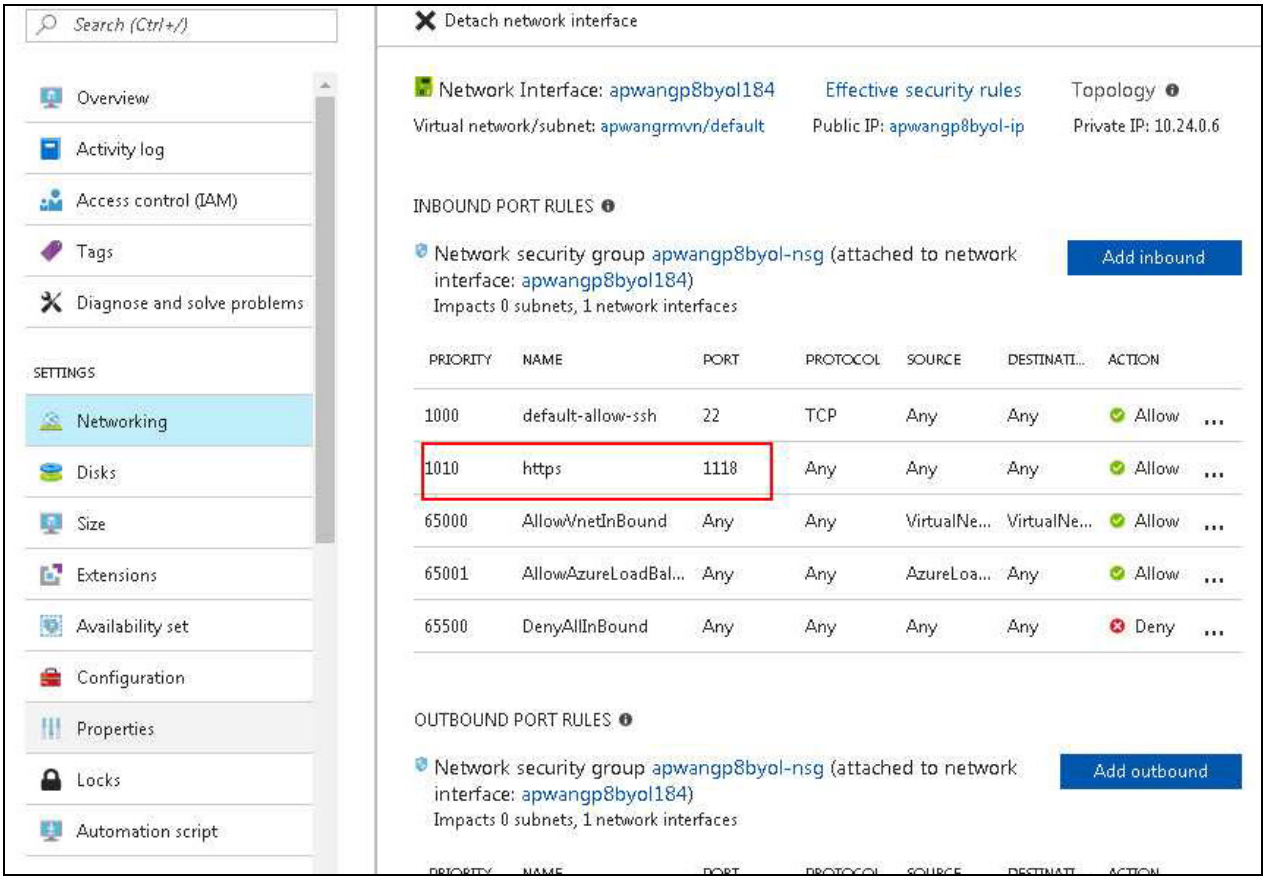
To access the web GUI for configured VM images, configure endpoint mapping in the Azure management portal. The public IP address for the web GUI will not work unless this is set up as per the procedure mentioned below:

1. Navigate to **Virtual Machines**.
2. Click on the configured VM and select **Networking**.
3. Select the management interface and add an inbound HTTPS rule as follows:
   a. A high priority.
   b. Name as HTTPS.
   c. A designated port such as 1113.

   You can now access the ACOS GUI at `https://<azure_public_ip>:1113`.
4. Select the management interface and add an inbound HTTP rule as follows:
   a. A high priority.
   b. Name as HTTP.
   c. A designated port such as 1115.

   You can now access the ACOS GUI at `http://<azure_public_ip>:1115`.

FIGURE 2-24: Editing Endpoint Mapping within the Azure Management Portal



Chapter 3: Initial vThunder Configuration for Azure

This chapter describes how to configure vThunder for Azure.

The following topics are covered:

Changing the VM Size

The size of a vThunder VM can be changed by using either the Windows Azure Management Portal or Power Shell commands. The size of a virtual machine determines the vCPUs, RAM size, data disks, IOPS value, and so on for the VM.

For information on changing VM sizes, see the 'Resize a Linux virtual machine using CLI 2.0' on page https://docs.microsoft.com/en-us/azure/virtual-machines/linux/change-vm-size.

Changing the Disk Size

The existing data storage of a vThunder VM can be expanded. The default virtual hard disk size is 30 GB. It can be expanded up to 2048 GB.

NOTE: Once the disk is expanded, it cannot be reduced.

For information on changing disk size, see https://docs.microsoft.com/en-us/azure/virtual-machines/windows/expand-os-disk

Adding More NICs Using the Azure CLI

More NICs can be added to a vThunder VM if the VM size supports the NICs. If the vThunder VM does not support more NICs, the VM size can be changed as described in Changing the VM Size and then more NICs can be added. For more information, see https://docs.microsoft.com/en-us/azure/virtual-network/virtual-network-network-interface-vm.

Follow the steps mentioned below:

1. To add a NIC to an existing vThunder instance, first deallocate and shutdown the VM:

```
az vm deallocate --resource-group testResourceGroup --name vThunderVM
az vm stop --resource-group testResourceGroup --name vThunderVM
```

2. Add the NIC using the command **`az vm nic add`**:

```
az vm nic add \
    --resource-group testResourceGroup \
    --vm-name vThunderVM\
    --nics myNic3
```

3. Start the VM with the following command:

```
az vm start --resource-group testResourceGroup --name vThunderVM
```

Deleting NICs Using the Azure CLI

Before deleting a NIC from a vThunder instance, ensure that the VM is stopped and there are at least two network interfaces attached to the VM. If you remove a primary network interface, Azure assigns the primary attribute to the network interface that is connected for the longest period to the VM. For more information, see https://docs.microsoft.com/en-us/azure/virtual-network/virtual-network-network-interface-vm.

1. To remove a NIC from a vThunder VM, first deallocate and stop the VM as follows:

```
az vm deallocate --resource-group testResourceGroup --name vThunderVM
az vm stop --resource-group testResourceGroup --name vThunderVM
```

2. Remove the NIC using the command **`az vm nic remove`**:

```
az vm nic remove \
    --resource-group testResourceGroup \
    --vm-name vThunderVM \
    --nics myNic3
```

3. Start the VM with the following command:

```
az vm start --resource-group testResourceGroup --name vThunderVM
```

Initial vThunder Configuration

This section describes how to configure IP connectivity on the vThunder management and data interfaces.

NOTE: To display a list of commands for a level of the CLI, enter a question mark as (?), and press **Enter**. It displays the list separately for each level. For syntax help, enter a command or keyword followed by a "space", then enter (?) and press **Enter**.

Logging in with ACOS CLI

Follow the steps mentioned below:

1. Log in to vThunder with the default **Username** and **Password** or the **ssh key-pair associated** with the instance.
2. Enable the Privileged EXEC level by typing `enable` and pressing the **Enter** key. There is no default password for Privileged EXEC mode; just press Enter.

```
vThunder>enable
Password:(just press Enter on a new system)
```

```
vThunder#
```

3. Enable the configuration mode by typing `config` and pressing **Enter**.

```
vThunder#config
vThunder(config)#
```

It is strongly suggested that a Privileged EXEC enable password be set up as follows:

```
vThunder(config)#enable-password newpassword
```

Changing the Admin Password

A10 Networks recommends that you change the admin password immediately for security as mentioned below:

```
vThunder(config)#admin admin password newpassword
vThunder(config-admin:admin)#
```

The vThunder is now network accessible for configuration under the new IP address and admin password.

Saving the Configuration Changes – write memory

Configuration changes must be saved to system memory to take effect the next time the vThunder is powered on. Otherwise, the changes are lost if the vThunder virtual machine or its host machine is powered down.

To write the current configuration to system memory, run the following command:

```
vThunder(config)# write memory
Building configuration...
[OK]
```

Configuring DHCP in vThunder TPS

Follow the steps to configure DHCP in vThunder TPS:

1. Access the IP address of the vThunder instance through SSH.
2. Use the following CLI commands to force the interface to use the IP assigned by DHCP.

```
interface ethernet mgmt
 ip address dhcp
```

NOTE: Do not use the **"no ip address dhcp"** command. This may result into losing the SSH connection to vThunder. The workaround for a lost connection is to restart the vThunder instance.

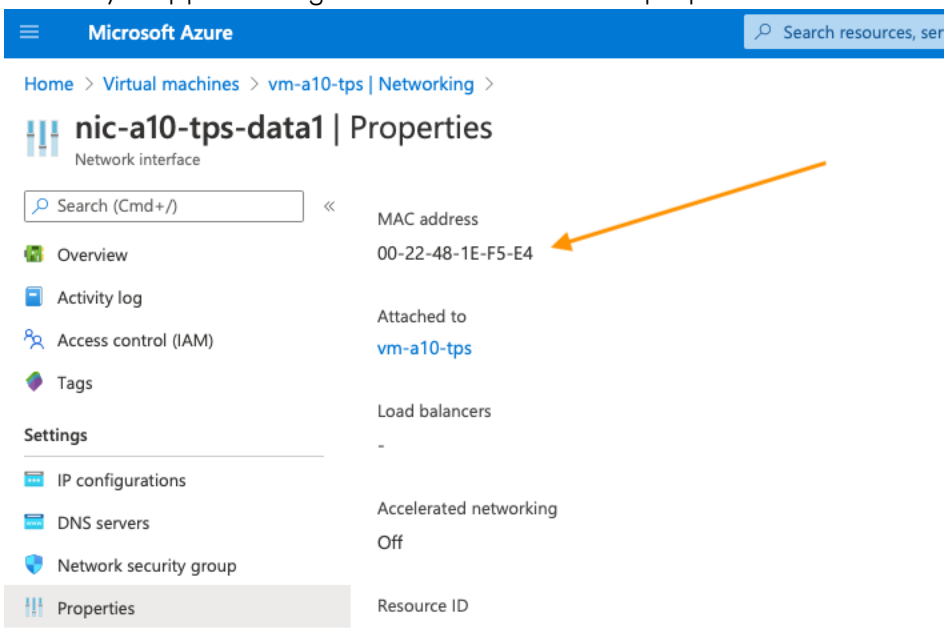
Configuring Multiple NICs on vThunder TPS

1. Log in to vThunder TPS VM using SSH credentials.
2. Run the `show interface brief` command to see the interfaces on the vThunder TPS.

```
vThunder(NOLICENSE)#show interface brief
Port     Link  Dupl  Speed  Trunk Vlan   MAC            IP Address       IPs  Name
-----------------------------------------------------------------------------------
mgmt     Up    auto  auto   N/A   N/A    000d.3a1e.6ed6 10.1.0.200/24     1
1        Disb  None  None   None  1      0022.481e.f5e4 0.0.0.0/0         0
2        Disb  None  None   None  1      0022.481e.fbfc 0.0.0.0/0         0
```

In this example, a management IP address and two data interface IP addresses are shown.

3. On the Azure Portal interface, check the MAC addresses to verify that the data interfaces are correctly mapped. Navigate to network interface properties to see the MAC address.



4. Run the `config` command to enter the configuration mode.

```
vThunder# config
vThunder(config)#
```

5. Run the following commands:

```
!
interface ethernet 1
 enable
 ddos outside
 ip address 10.1.10.200 255.255.255.0
!
interface ethernet 2
 enable
 ddos inside
 ip address 10.1.20.200 255.255.255.0
!
```

The `ddos outside` and `ddos inside` commands indicate which interfaces the traffic will arrive on (ddos outside), and which interfaces the protected objects will lay behind (ddos inside). When you configure these commands the first time, the following prompt appears.

```
ddos mode change will come into effect next time you write memory and
reload/reboot the software
```

As the prompt suggests, the device is reload proactively. This is a one-time operation.

6. Before the device is reload, run the following command to write the configuration to memory.

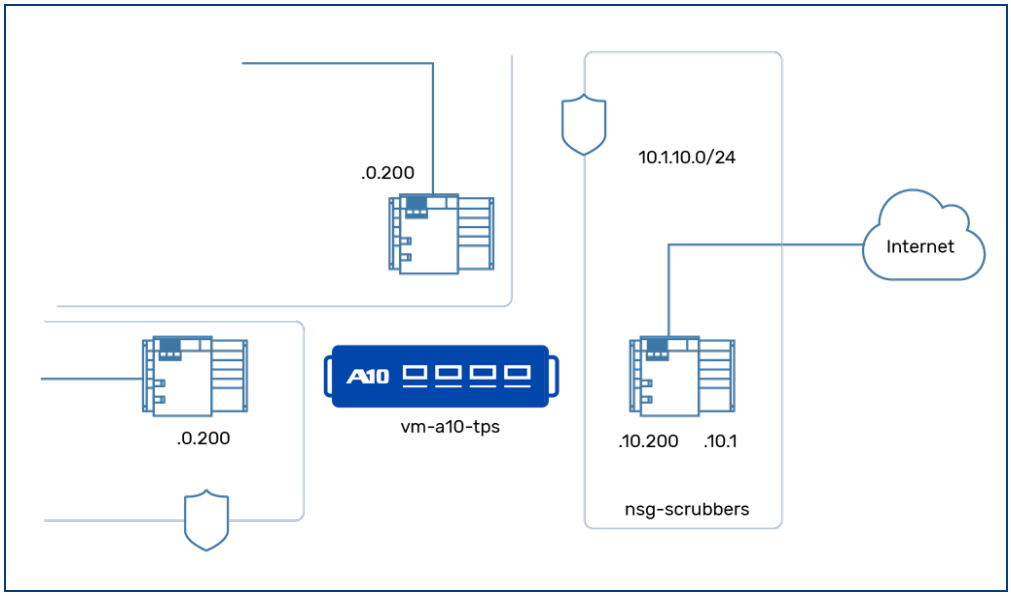
```
vThunder(config-if:ethernet:2)(NOLICENSE)#write memory
Building configuration...
Write configuration to default primary startup-config
[OK]
```

7. Enter the exit commands to get into privileged mode before running the reload command.

```
vThunder(config-if:ethernet:2)(NOLICENSE)#exit
vThunder(config)(NOLICENSE)#exit
vThunder(NOLICENSE)#reload
Do you wish to proceed with reload? [yes/no]:yes System is reloading now.
Please wait ....
System has reloaded successfully. vThunder(NOLICENSE)#
Session closed
Session closed
Connection to 10.1.0.200 closed.
[azureuser@vm-jumpserver ^]$ []
```

Reload is a faster process than upgrading the device so the vThunder TPS will be up in few minutes. Interfaces are now configured.

8. Configure the default routes on the vThunder TPS :
   a. Go into privileged mode
   b. Go into config mode
   c. Execute the command to configure route, for example,`ip route 0.0.0.0 /0 10.1.10.1`



9. Enable DDoS protection by running the `ddos protection enable` command.
10. Configure the protected object on the vThunder TPS as shown. In this example, the private IP address is 10.1.10.201 and the public IP address is 52.152.230.140.

```
!
ddos dst zone vm-appserver
 operational-mode monitor
 ip 10.1.10.201
 description "Pub-52.152.230.140, Priv-10.1.10.201, App-10.1.20.11"
 dest-nat 10.1.20.11
 ip-proto icmp-v4
 port other tcp
 port other udp
!
```

Code explanation

- `ddos dst zone vm-appserver`

  vm-appserver is just a name given to the protected object. For ease of tracking, you can keep the same name in the code as the VM that you are trying to protect.

- `operational-mode monitor`

  The vThunder TPS lets you configure idle protected objects. Such objects take effect when the object is put into `monitor`mode. Other operational modes are `idle` and `learning`.

- `ip 10.1.10.201`

This is the private IP address. This IP is configured to help the client traffic land at the vThunder TPS. In a physical TPS deployment, the IP configured here is the actual IP on the protected servers. However, in the case of Azure where NATting is required at the vThunder TPS, this is simply the IP address you create to associate them with the actual IP that resides on the protected server (vm-appserver)

- `description "Pub-52.152.230.140, Priv-10.1.10.201, App-10.1.20.11"`

  General description or comments.

- `dest-nat 10.1.20.11`

  After the client traffic lands at the vThunder TPS, traffic is mitigated and then NAT is implemented on the packet, so that it reaches the actual appserver. The destination NAT address is an input in this configuration.

- `port other tcp, port other udp, ip-proto icmp-v4`

  When DDoS protection is enabled on the TPS device, the device starts acting like a firewall. It will drop all packets arriving at the vThunder TPS unless ports are explicitly opened. If you want to restrict the ports that can be accessed, use the `no port other udp` command, and then open up a specific port with a Deploying and Directing Traffic through a vThunder TPS in Azure command, such as `port 10000 udp`. Similarly, the command `ip-proto icmp-v4` enables pinging the vm-appserver through the TPS VM.

Chapter 4: Advanced vThunder TPS Configuration on Microsoft Azure

This chapter describes advanced vThunder TPS configurations for Microsoft Azure.

The following topics are covered:

About Microsoft Azure Gateway Load Balancer

Microsoft's Azure Gateway Load Balancer (GWLB) is a fully managed service enabling you to deploy, scale, and manage third-party Network Virtual Appliances (NVAs) such as firewalls, inline DDoS appliance, deep packet inspection system, or any custom appliance. It provides you with a single gateway load balancer for distributing traffic among various virtual appliances ensuring high availability and scalability based on demand.

NOTE: ACOS 5.3.0-SP2 is required to implement GWLB with TPS.

Implementing Azure Gateway LB with TPS

A10 Networks, Inc. has partnered with Microsoft Azure to support its DDoS mitigation solution, Thunder TPS VA, with Azure's Gateway LB (GWLB). Thunder TPS scales to defend against the DDoS of Things and traditional zombie botnets and Azure's Gateway LB provides an option to add inline DDoS protection through Thunder TPS. The Gateway LB ensures that relevant Thunder TPS are

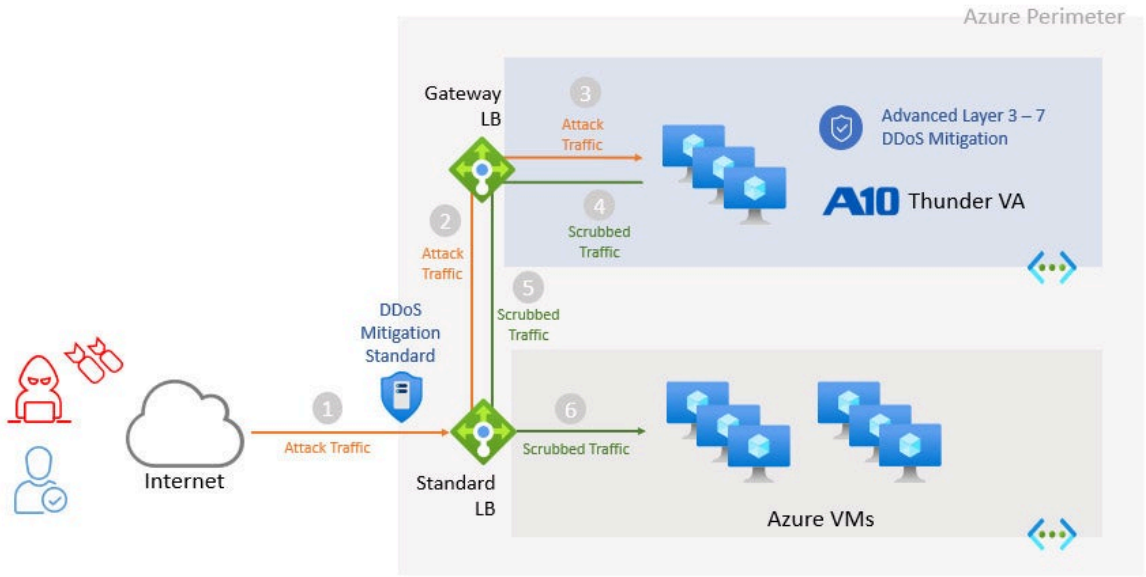
injected into the ingress and the egress path of the internet traffic as it heads towards Azure-hosted applications, services, and the sender.

NOTE: For high availability and scalable DDoS protection, multiple vThunder TPS devices can be deployed in a cluster that share the same protected object (IP address and service), where all vThunder TPS devices are active for mitigating DDoS traffic and forwarding legitimate traffic.

The Gateway LB will also periodically send health checks to confirm if the Thunder TPS is up and running. It expects the response of the health check to follow the same interface that the health check was sent on.

When Thunder TPS is combined with Azure's DDoS Protection Standard, the solution provides comprehensive protection against various L3 to L7 DDoS attacks.

FIGURE 4-1: Gateway LB Topology



Configuring Gateway LB TCP/HTTP Health Check on TPS

```
vThunder#config
vThunder(config)#ddos interface-http-health-check enable

  challenge-method javascript
```

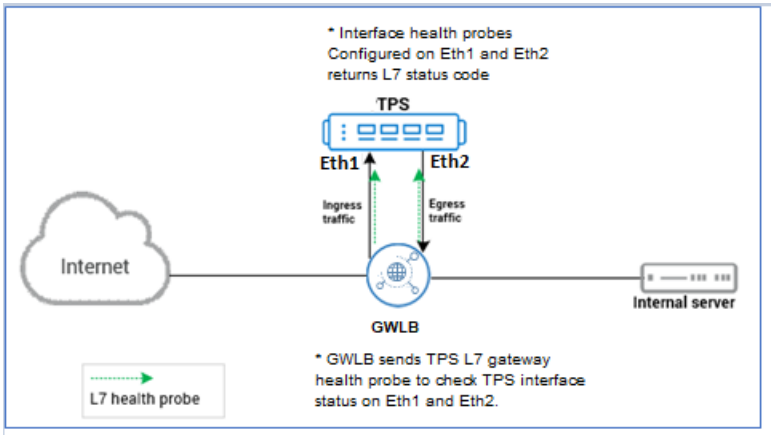
```
!
ddos dst interface-ip 10.29.1.4
  port 80 http-probe
!
ddos dst interface-ip 10.29.2.4
  port 80 http-probe
```

Gateway LB Health Check Traffic Flow

Gateway LB will send the health checks to the Thunder TPS VA over its eth1 interface and the TPS is expected to respond over the same eth1 interface. Gateway LB will send health checks to the TPS over its eth2 interface and the TPS is expected to respond to these health checks over eth2.

```
GWLB -> (eth1) TPS (eth1) -> GWLB
GWLB -> (eth2) TPS (eth2) -> GWLB
```

FIGURE 4-2: Gateway LB Health Probe Traffic Flow



Configuring Gateway LB Data Traffic on TPS

```
interface ethernet 1
enable
  ddos inside
  ip address 10.29.2.4 255.255.255.0
!

interface ethernet 2
  enable
  ddos outside
  ip address 10.29.1.4 255.255.255.0
```

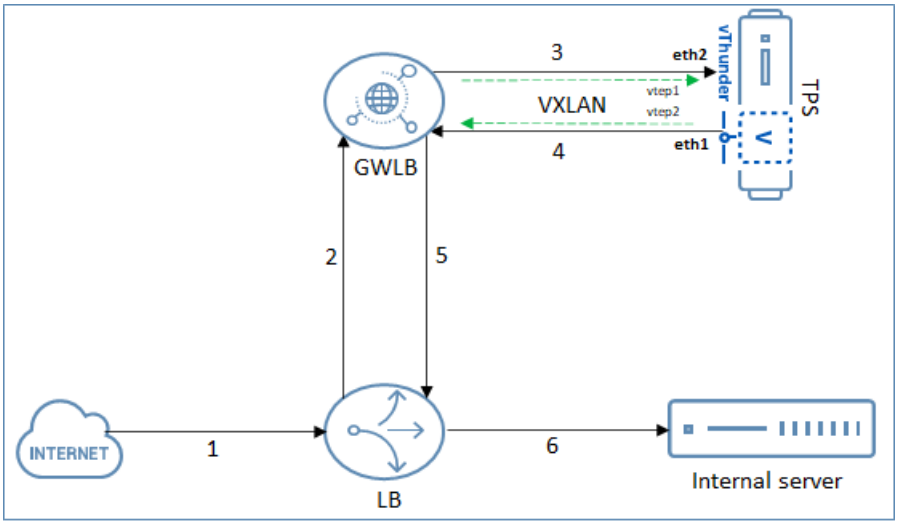
```
!
interface lif clean
  ip address 172.16.2.1 255.255.255.252
!

interface lif dirty
  ip address 172.16.1.1 255.255.255.252
!
overlay-tunnel vtep 1
  encap vxlan
  dest-port 10801
  local-ip-address 10.29.1.4
    vni 801 lif dirty
  remote-ip-address 10.29.3.5
    vni 801
  host 172.16.1.2 aaaa.aaaa.0267 vni 801 remote-vtep 10.29.3.5
!
overlay-tunnel vtep 2
  encap vxlan
  dest-port 10800
  local-ip-address 10.29.2.4
    vni 800 lif clean
  remote-ip-address 10.29.3.5
    vni 800
  host 172.16.2.2 aaaa.aaaa.0266 vni 800 remote-vtep 10.29.3.5
```

Gateway LB Data Traffic Flow

Inbound Client

Gateway LB will send client-side data traffic to the Thunder TPS over the VxLAN tunnel on vtep1/eth2. The Thunder TPS will forward the client-side data traffic out the VxLAN tunnel on vtep2/eth1 back to the Gateway LB which will be routed to the server.

FIGURE 4-3: Inbound Client

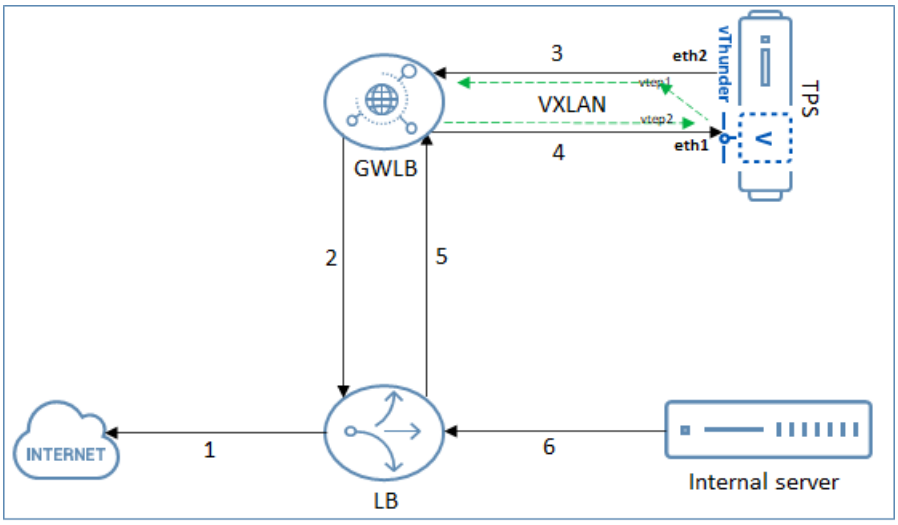


Outbound Server

When the server responds, Gateway LB will forward the traffic back to the Thunder TPS VA through VxLAN tunnel on vtep2/eth1. The Thunder TPS VA will then send the server-side traffic out of the VxLAN tunnel on vtep1/eth1 to the Gateway LB which will eventually be routed back to the client.

NOTE: vThunder TPS sends the server response traffic over VxLAN tunnel vtep1 (dirty tunnel) but interface eth1 (inside interface).

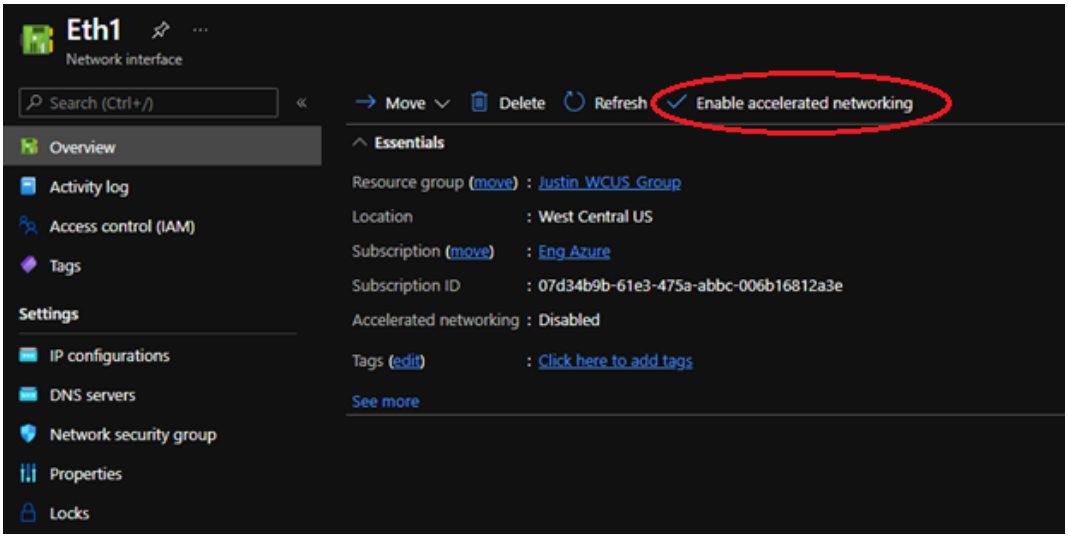
FIGURE 4-4: Outbound Client



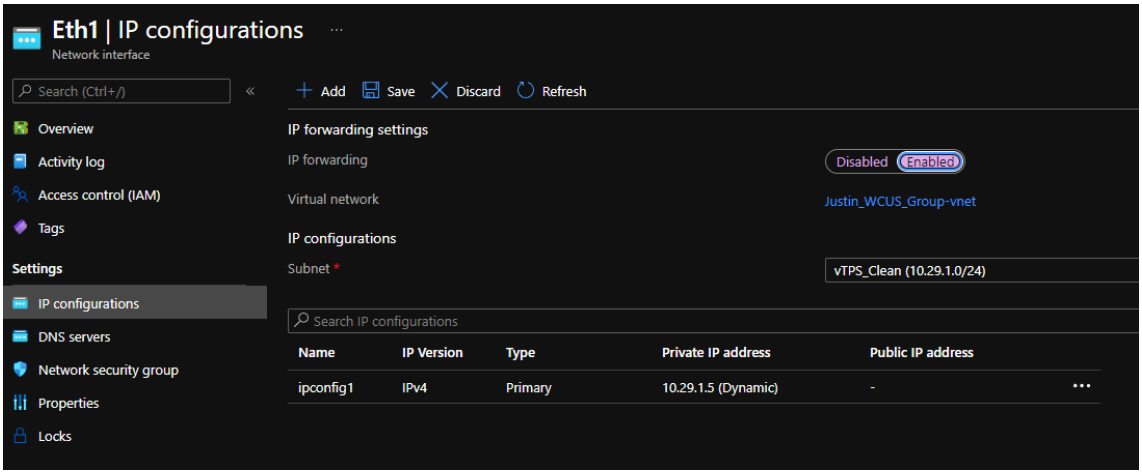
Prerequisites

Before deploying a new Gateway LB, perform the following steps:

- If you are using a Virtual machine setup, ensure a TPS virtual machine <*TPS_virtual_machine_name*> is already created. For more information, refer Creating a vThunder VM. This TPS VM will be referenced while creating the new Gateway LB instance.
- Similarly, if you are using a virtual machine scale set, ensure a TPS VMSS is already created. For more information, refer https://docs.microsoft.com/en-us/azure/virtual-machine-scale-sets/overview.
  This TPS VMSS will be referenced while creating the new GWLB instance.
- If the vThunder TPS supports accelerated networking, accelerated networking should be enabled on all the data NICs (data ports) of the TPS device but not on any management NICs.
  In the FIGURE 4-2, eth1 and eth2 of the TPS device should have accelerated networking enabled.
  The marketplace image does not support network acceleration. Hence, this option should be enabled on a case-to-case basis.



- IP forwarding should be enabled on all the data NICs (data ports) of the TPS device but not on any management NICs.
  In the FIGURE 4-2, eth1 and eth2 of the TPS device should have accelerated networking enabled.



Enabling IP Forwarding for VMSS backend pool

To enable IP Forwarding on all data NICs for a VMSS instance, run the following commands in Azure Powershell:

```
$vmss = Get-AzVmss -ResourceGroupName "<Resource_group>" -VMScaleSetName
"<VMSS_name>"
$vmss.VirtualMachineProfile.NetworkProfile.NetworkInterfaceConfigurations
[1].EnableIPForwarding = 1
$vmss.VirtualMachineProfile.NetworkProfile.NetworkInterfaceConfigurations
[2].EnableIPForwarding = 1
Update-AzVmss -ResourceGroupName "<Resource_group>" -VMScaleSetName
"<VMSS_name>" -VirtualMachineScaleSet $vmss
```

Example

```
$vmss = Get-AzVmss -ResourceGroupName "Justin_WCUS_Group" -VMScaleSetName
"Justin-WCUS-VMSS"
$vmss.VirtualMachineProfile.NetworkProfile.NetworkInterfaceConfigurations
[1].EnableIPForwarding = 1
$vmss.VirtualMachineProfile.NetworkProfile.NetworkInterfaceConfigurations
[2].EnableIPForwarding = 1
Update-AzVmss -ResourceGroupName "Justin_WCUS_Group" -VMScaleSetName
"Justin-WCUS-VMSS" -VirtualMachineScaleSet $vmss
```

Deploying Azure Gateway LB with TPS using Azure Portal

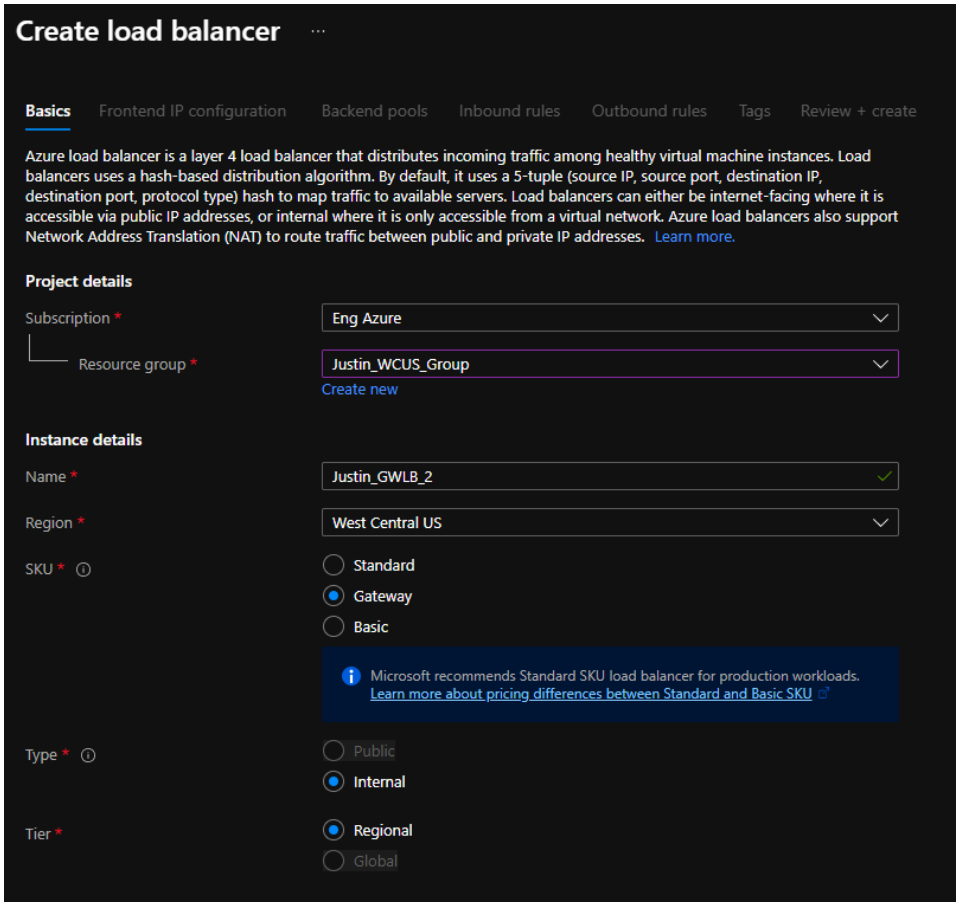
To deploy Azure Gateway LB with TPS using Azure Portal, perform the following steps:

1. Log in to Azure portal https://portal.azure.com as a Global Administrator.
   The **Microsoft Azure - Services** window is displayed.
2. Under **Azure Services**, click **Load balancers** or enter Load balancers in the search field of the **Microsoft Azure** homepage.
   The **Load balancers** window is displayed.
3. Click **Create load balancer** to create a new load balancer.
   The **Create load balancer** window with **Basics** tab is displayed.
4. Under **Basics** tab, provide the following details:
   Project details section
   a. Select the correct **Subscription** from the drop-down list.
   b. Select the existing **Resource group** from the drop-down list or choose to **Create new** resource group if an existing resource group is unavailable in the selected subscription.

   Instance details section

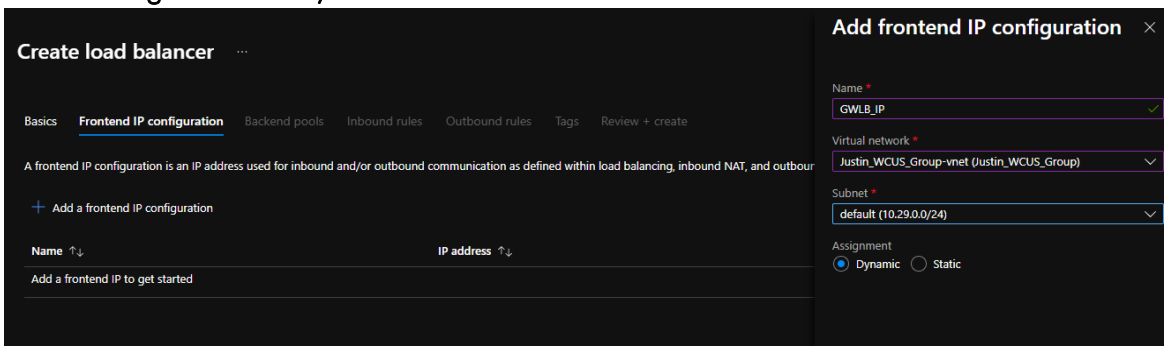
   a. Enter <load_balancer_name> as the **Name** for the load balancer.
   b. Select the **Region**.
   c. Select **SKU** as **Gateway**.
   d. Select **Type** as **Internal**.
   e. Select **Tier** as **Regional**.

   The new Gateway LB is ready to get assigned to the selected resource group.



5. Click **Next : Frontend IP configuration >**.
   The **Frontend IP configuration** tab is displayed.
6. Click **Add a frontend IP configuration**.
   The **Add frontend IP configuration** pane is enabled to the right-side of the window.
7. In the **Add frontend IP configuration** pane, provide the following details:
   a. Enter frontend IP **Name**.
   b. Select <*TPS_virtual_machine_name*> from the drop-down list as the **Virtual network**. This VM was created as the prerequisite step.
   c. Select **Subnet** from the drop-down list.

d. Select **Assignment** as **Dynamic**.



e. Click **Add**.
The frontend IP is created and it appears in the list on the **Frontend IP configuration** tab. The private IP address is assigned to Gateway LB.

8. Click **Next : Backend pools >**.
The **Backend pools** tab is displayed.

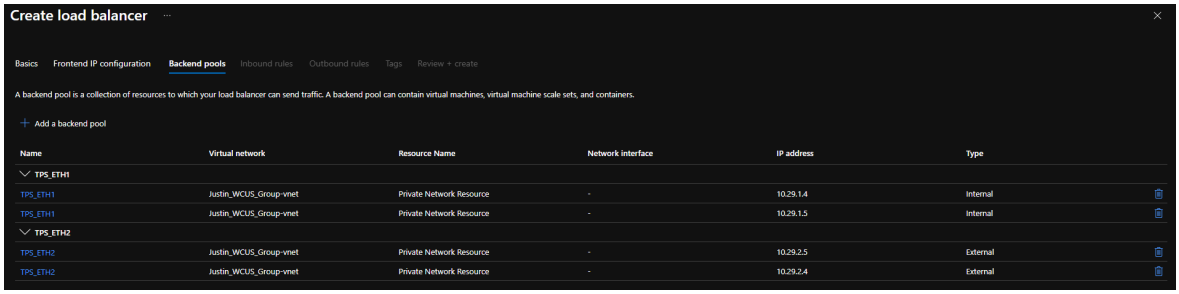
A backend pool can either be created using IP Address or NIC. In the procedure below, two backend pools are created for each TPS VM, one for internal use and another for external use. Before creating backend pools, refer Creating Backend pools.

NOTE: The port should be same for these two backend pools and the TPS VM.

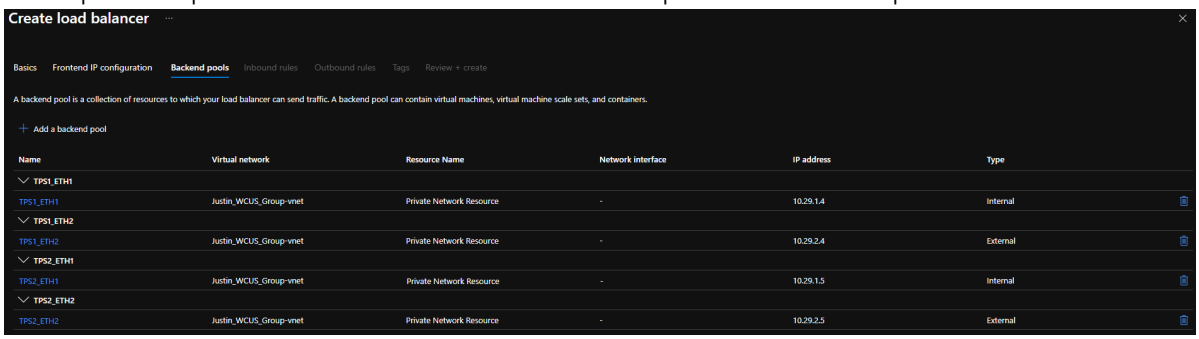
Creating Backend pools

While implementing Gateway LB with TPS, backend pools are required to be created. These backend pools can be created in three different ways:

a. Create two backend pools with multiple IP address or NIC for each TPS VM.
This option requires one backend pool for eth1 of TPS and the other backend pool for eth2 of TPS.



b. Create multiple backend pool with single IP address or NIC for each TPS VM.
This option requires twice the number of backend pools created in option 1.



c. Create two backend pools with VMSS NIC.
This option allows easy scalability.

NOTE: If you are using NIC-based backend pool, ensure that the TPS does not have any public IPs associated to it.

To create backend pools using IP Address, perform the following steps:

a. Click **Add a backend pool** to create internal backend pool.
The **Add backend pool** window is displayed.
b. Provide the following details:

Main section

i. Enter <*TPS_EHT1*> as the **Name** of the backend pool.
ii. Virtual network should be auto-populated to <*TPS_virtual_machine_name*>.
This VM was created as the prerequisite step.
iii. Select **Backend Pool Configuration** as **IP Address**.
iv. Select **IP Version** as **IPv4**.

Gateway load balancer configuration section

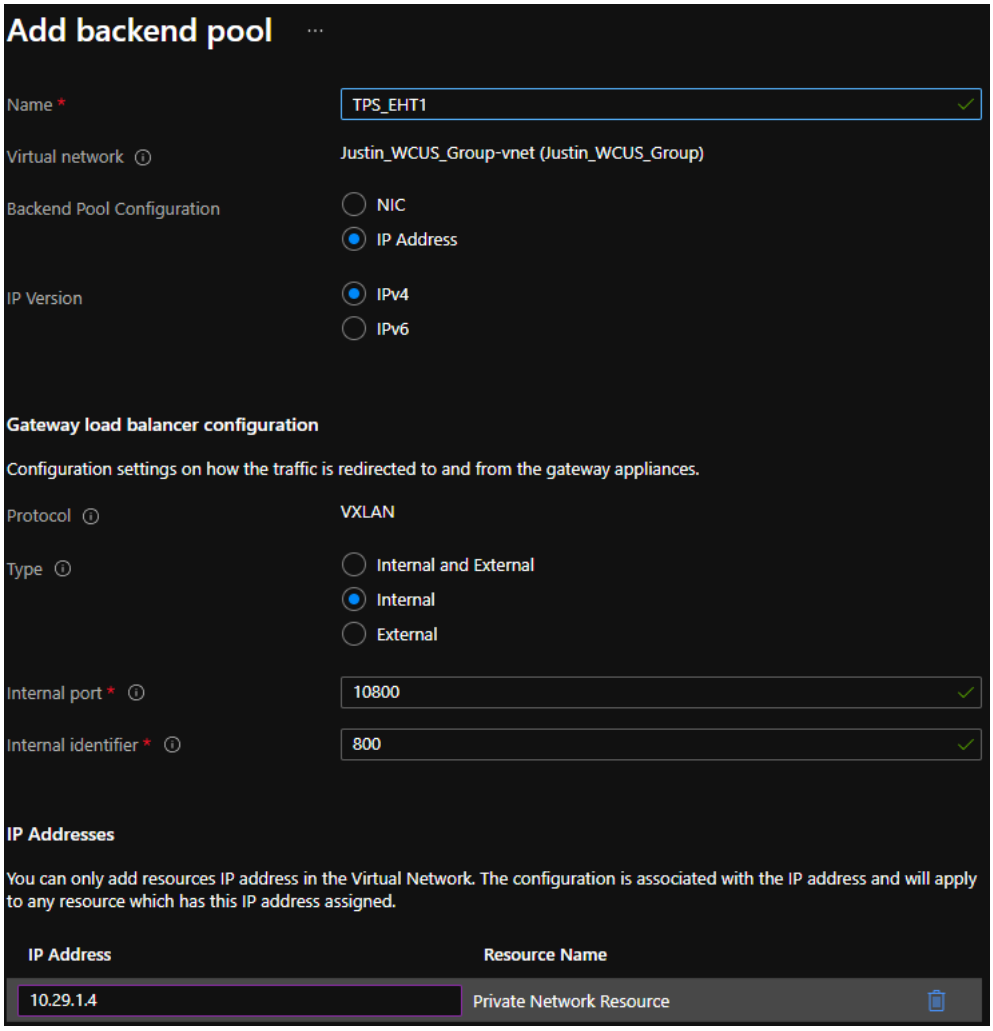
i. **Protocol** should be auto-populated to **VXLAN**.
ii. Select **Type** as **Internal**.
iii. Enter the **Internal port** same as that of **TPS device port**.
iv. Enter **the Internal identifier**.

IP Addresses section

i. Enter the **IP Address**.
ii. Select the **Resource Name**.

c. Click **Add**.
The <*TPS_EHT1*> backend pool is created to link TPS VM with Gateway LB internally and it

appears in the list on the **Backend pools** tab.



d. Click **Add a backend pool** to create external backend pool.
The **Add backend pool** window is displayed.

e. Provide the following details:

Main section

i. Enter <*TPS_EHT2*> as the **Name** of the backend pool.
ii. **Virtual network** should be auto-populated to the <*TPS_virtual_machine_name*>. This VM was created as the prerequisite step.
iii. Select **Backend Pool Configuration** as **IP Address**.
iv. Select **IP Version** as **IPv4**.

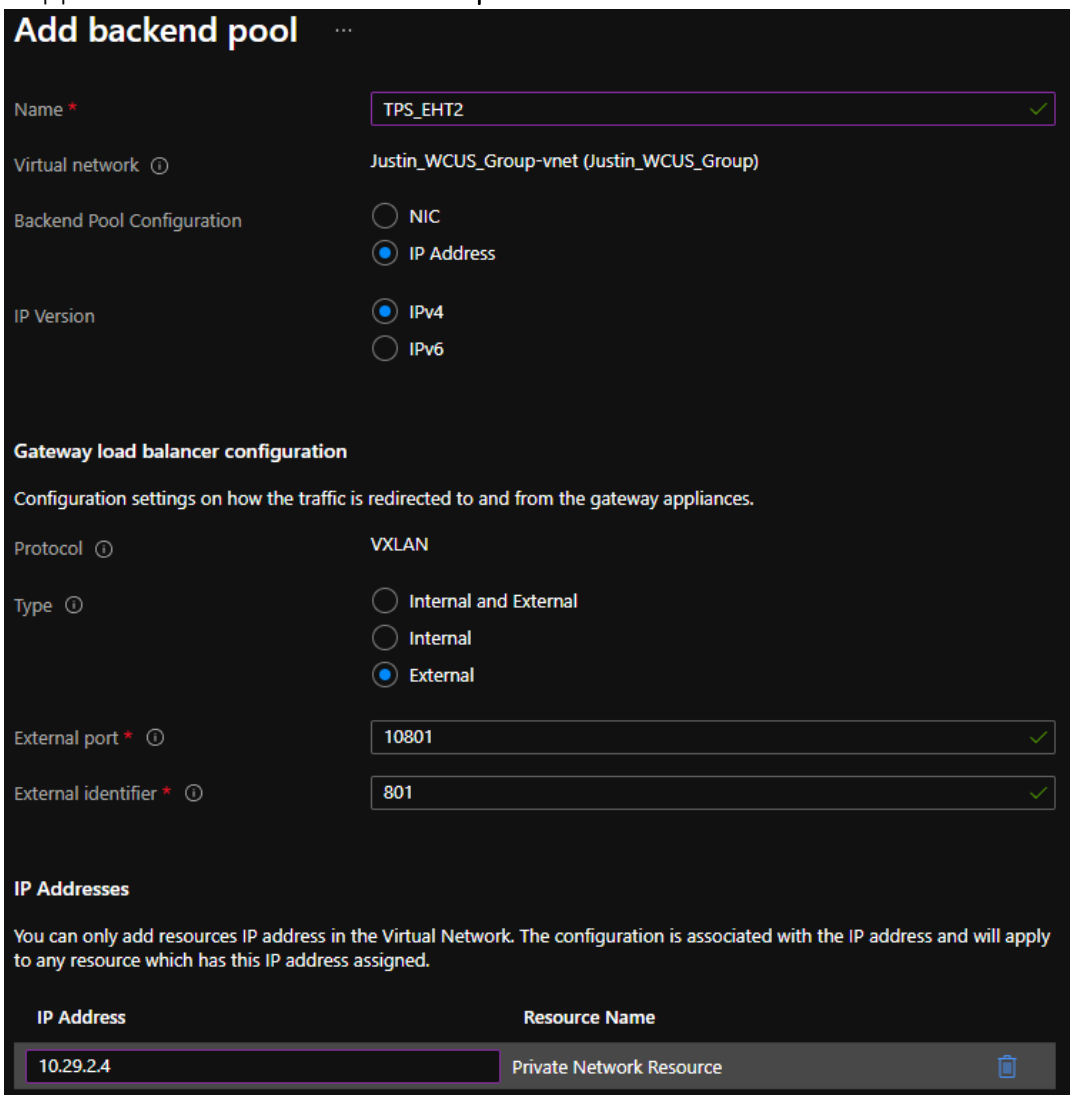
Gateway load balancer configuration section

i. **Protocol** should be auto-populated to **VXLAN**.
ii. Select **Type** as **External**.
iii. Enter the **External port** same as that of TPS device port.
iv. Enter the **External identifier**.

IP Addresses section

i. Enter the **IP Address**.
ii. Select the **Resource Name**.

f. Click **Add**.
The <*TPS_EHT2*> backend pool is created to link TPS VM with Gateway LB externally and it appears in the list on the **Backend pools** tab.



To create backend pools using VMSS NIC, perform the following steps:

a. Click **Add a backend pool** to create internal backend pool.

   The **Add backend pool** window is displayed.

b. Provide the following details:

   Main section

   i. Enter <*TPS_VMSS_EHT1*> as the **Name** of the backend pool.
   ii. **Virtual network** should be auto-populated to the <*TPS_virtual_machine_name*>. This VM was created as the prerequisite step.
   iii. Select **Backend Pool Configuration** as **NIC**.
   iv. Select **IP Version** as **IPv4**.

   Gateway load balancer configuration section

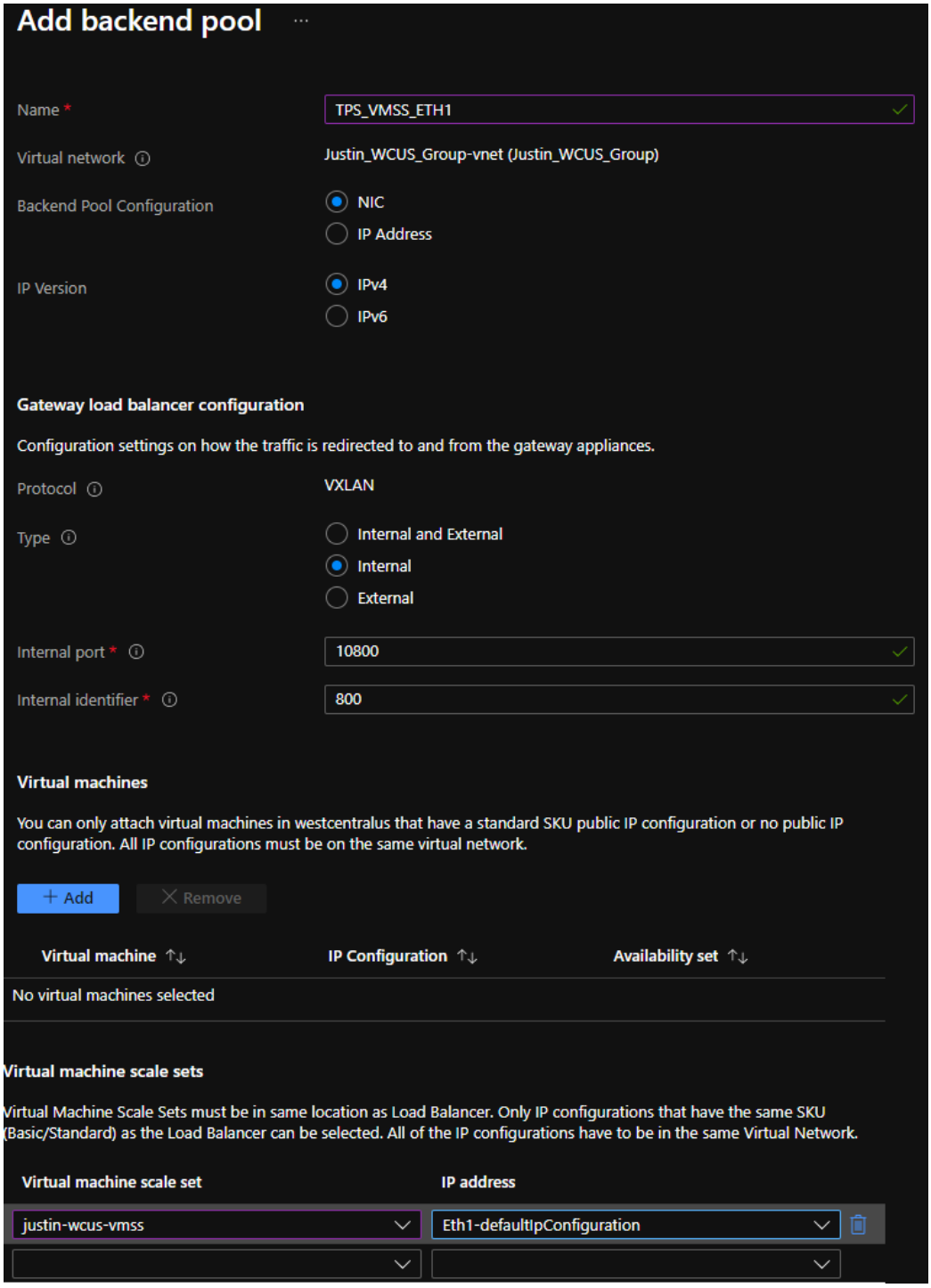
   i. **Protocol** should be auto-populated to **VXLAN**.
   ii. Select **Type** as **Internal**.
   iii. Enter the **Internal port** same as that of TPS device port.
   iv. Enter the **Internal identifier**.

   Virtual machine scale sets section

   i. Select the **Virtual machine scale set.**
   ii. Select the **IP address**.

c. Click **Add**.
   The <*TPS_VMSS_EHT1*> backend pool is created for one interface of the virtual machine

scale set and it appears in the list on the **Backend pools** tab.



d. Click **Add a backend pool** to create external backend pool.
The **Add backend pool** window is displayed.

e. Provide the following details:

Main section

i. Enter <*TPS_VMSS_EHT2*> as the **Name** of the backend pool.
ii. **Virtual network** should be auto-populated to the <*TPS_virtual_machine_name*>. This VM was created as the prerequisite step.
iii. Select **Backend Pool Configuration** as **NIC**.
iv. Select **IP Version** as **IPv4**.

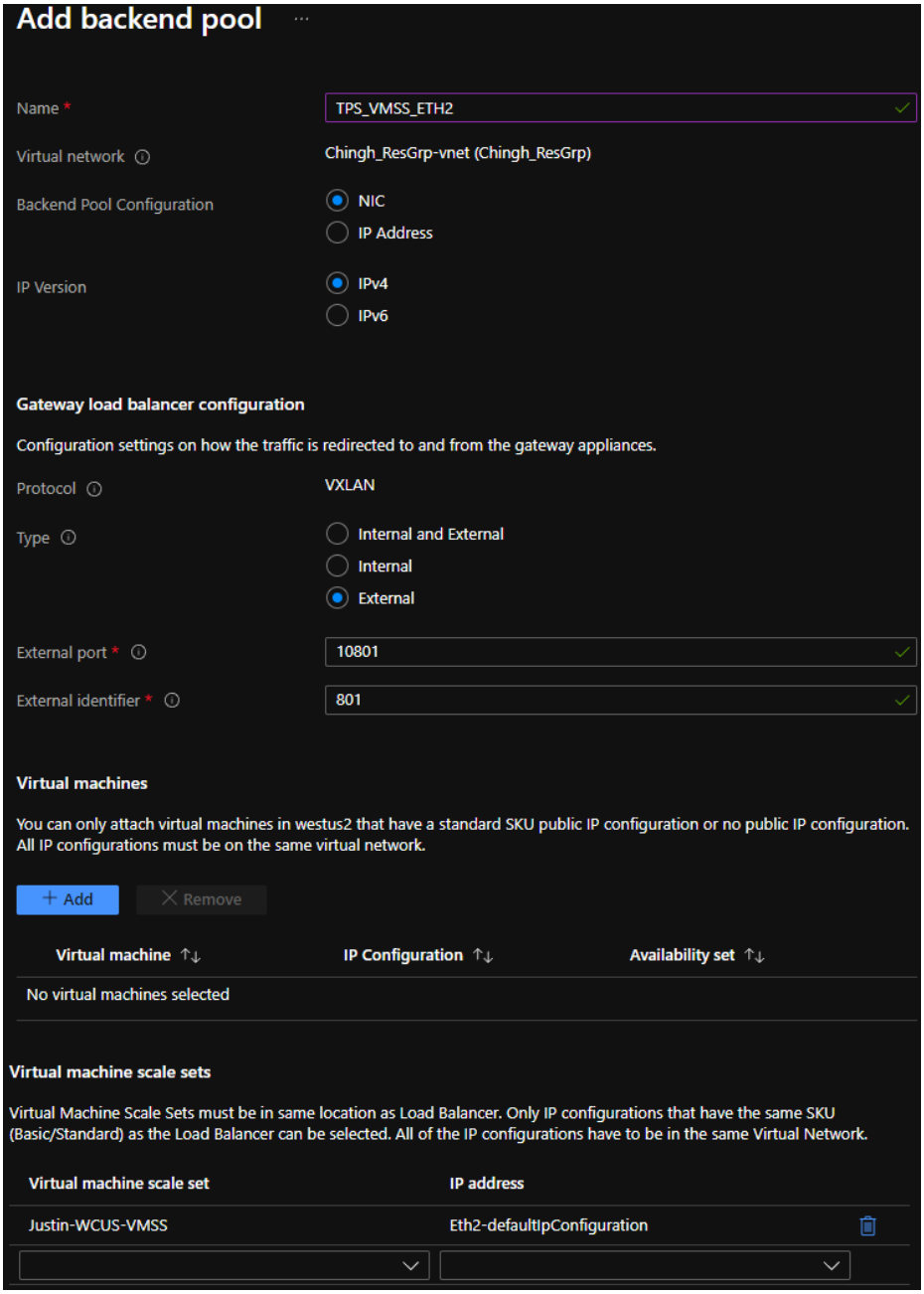
Gateway load balancer configuration section

i. **Protocol** should be auto-populated to **VXLAN**.
ii. Select **Type** as **External**.
iii. Enter the **External port** same as that of TPS device port.
iv. Enter the **External identifier**.

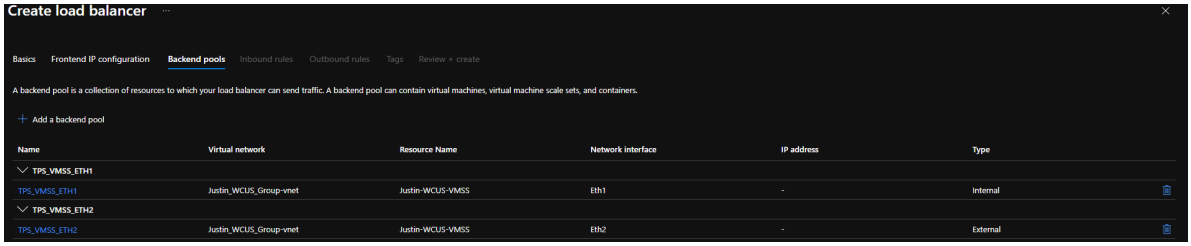
Virtual machine scale sets section

i. Select the same **Virtual machine scale set** as for ETH1.
ii. Select the **IP address**.
iii. Click **Add**.
The <*TPS_VMSS_EHT2*> backend pool is created for second interface of the virtual

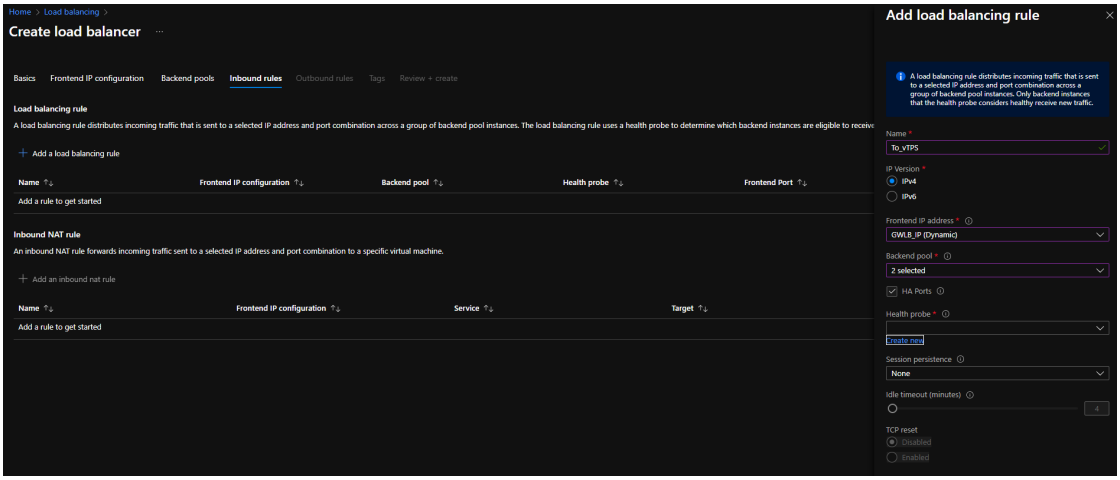
machine scale set and it appears in the list on the **Backend pools** tab.



f. Verify if the recently created backend pools appears in the list on the **Backend pools** tab.

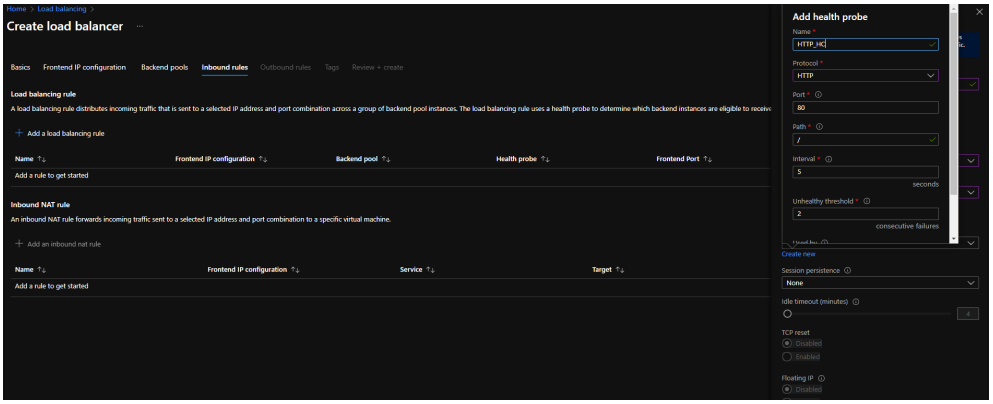


9. Click **Next : Inbound rules >**.
   The **Inbound rules** tab is displayed.
   To create the inbound rule, perform the following steps:
   a. Click **Add a load balancing rule**.
      The **Add load balancing rule** pane is enabled to the right-side of the window.
   b. In the **Add load balancing rule** pane, provide the following details:
      i. Enter load balancing inbound rule **Name**.
      ii. Select **IP version** as **IPv4**.
      iii. Select the **Frontend IP address** from the drop-down list.
      iv. Select the count of **Backend pool** from the drop-down list.
      v. Select the **HA Ports**.



      vi. Select an existing **Health probe** or click **Create new** if an existing health probe is unavailable.
      To create a new health probe, perform the following steps:
         A. Enter health probe **Name**.
         B. Select **Protocol** as **HTTP**.
         C. Enter **Port**.
         D. Enter **Path**.
         E. Enter **Interval**.
         F. Enter **Unhealthy threshold**.

G. Leave all other values as is.
H. Click **Create new**.
A new health probe is created and appears in the **Health probe** field.



vii. Leave all other values as is.
v. Click **Add**.
The new Inbound rule is created with health probe and appears in the list on **Inbound rules** tab.

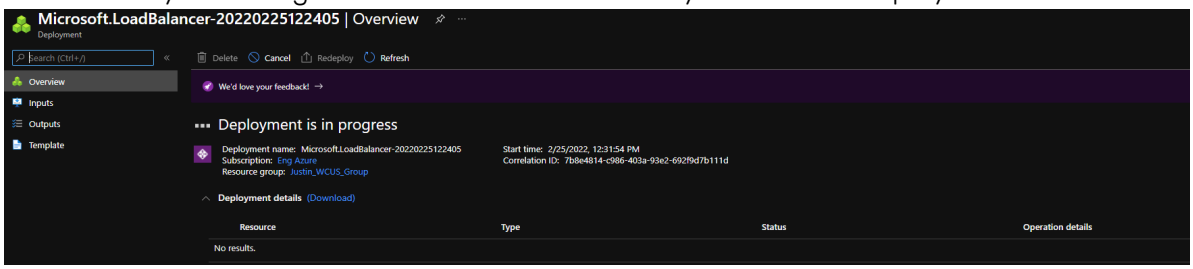
10. Click **Next : Outbound rules >**.
The **Outbound rules** tab is displayed.
To create the outbound rule, perform the following steps:

a. Click **Add a load balancing rule**.
The **Add load balancing rule** pane is enabled to the right-side of the window.

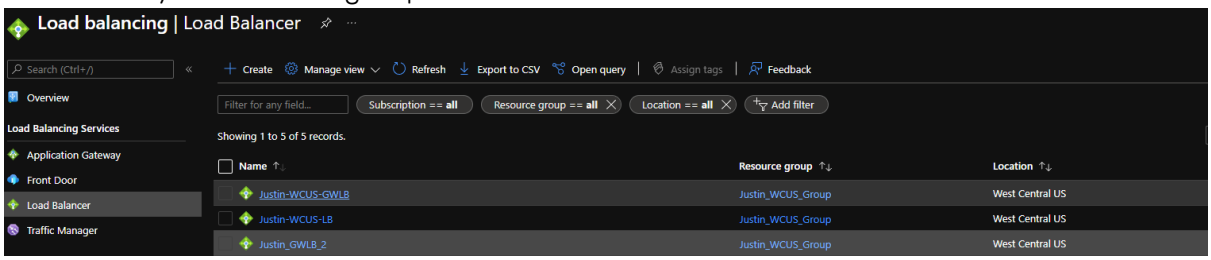
b. In the **Add load balancing rule** pane, provide the following details:
i. Enter load balancing outbound rule **Name**.
ii. Select **IP version** as **IPv4**.
iii. Select the **Frontend IP address** from the drop-down list.
iv. Select the count of **Backend pool** from the drop-down list.
Ensure to select both or all the backend pools created in 'Creating Backend pool' step.
v. Select the **HA Ports**.
vi. Select the existing **Health probe** created for inbound rule or create a new health probe.
vii. Leave all other values as is.
viii. Click **Add**.
The new Outbound rule is created.

11. Click **Next : Review + create >**.
The **Review + create** tab is displayed.
12. Review the configuration.

13. Click **Create**.
    The Gateway LB configuration is saved and Gateway LB starts to deploy.



14. Once the Gateway LB deployment is done, the Gateway LB will always be on and should be available in your resource group.



Deploying Azure Gateway LB with TPS using Azure CLI

To deploy Azure Gateway LB with TPS using Azure CLI, perform the following steps:

1. Create a Gateway LB with a backend pool.

```
az network lb create -g <Resource_group> --name <GWLB_name> --sku Gateway --
frontend-ip-name <Frontend_IP_name> --vnet-name <Virtual_network> --subnet
ETC  --backend-pool-name <Backend_pool_name>
```

Example This example creates a Gateway LB with a backend pool called TPS_ETH1. This is an internal backend pool.

```
az network lb create -g Justin_WCUS_Group --name Justin-WCUS-GWLB2 --sku
Gateway --frontend-ip-name GWLB_IP --vnet-name Justin_WCUS_Group-vnet --sub-
net ETC  --backend-pool-name TPS_ETH1
```

2. Assign an IP address to the internal backend pool.

```
az network lb address-pool address add -g <Resource_group> -- lb-name <GWLB_
name> --pool-name <Backend_pool_name> -n <NIC_name> --vnet <Virtual_network>
--ip-address <IP_address>
```

Example This example assigns IP to the backend pool TPS_ETH1.

```
az network lb address-pool address add -g Justin_WCUS_Group --lb-name
Justin-WCUS-GWLB2 --pool-name TPS_ETH1 -n Eth1 --vnet Justin_WCUS_Group-vnet
--ip-address 10.29.2.4
```

3. Update the internal backend pool to have the correct configuration.

```
az network lb address-pool tunnel-interface update -g <Resource_group> --lb-
name <GWLB_name> --address-pool <Backend_pool_name> --type Internal --index
0 -port <port> -identifier <ID> -protocol VXLAN
```

Example

```
az network lb address-pool tunnel-interface update -g Justin_WCUS_Group --
lb-name Justin-WCUS-GWLB2 --address-pool TPS_ETH1 --type Internal --index 0
-port 10800 -identifier 800 -protocol VXLAN
```

4. Create the external backend pools

```
az network lb address-pool create -g <Resource_group> --lb-name <GWLB_name>
-n <Backend_pool_name>
```

Example

```
az network lb address-pool create -g Justin_WCUS_Group --lb-name Justin-
WCUS_GWLB2 -n TPS_ETH2
```

By default, this creates the backend pool as internal. Update the type to external.

```
az network lb address-pool tunnel-interface update -g <Resource_group> --lb-
name <GWLB_name> --address-pool <Backend_pool_name> --type External --index
0 -port <port> -identifier <ID> -protocol VXLAN
```

Example

```
az network lb address-pool tunnel-interface update -g Justin_WCUS_Group --
lb-name Justin-WCUS-GWLB2 --address-pool TPS_ETH2 --type External --index 0
-port 10801 -identifier 801 -protocol VXLAN
```

5. Assign IP address to the external backend pool.

```
az network lb address-pool address add -g <Resource_group> --lb-name <GWLB_
name> --pool-name <Backend_pool_name> -n <NIC> --vnet <Virtual_network> --
ip-address <IP_address>
```

Example

```
az network lb address-pool address add -g Justin_WCUS_Group --lb-name
Justin-WCUS-GWLB2 --pool-name TPS_ETH2 -n Eth2 --vnet Justin_WCUS_Group-vnet
--ip-address 10.29.1.4
```

6. Create the Health check.

```
az network lb address-pool tunnel-interface update -g <Resource_group> --lb-
name <GWLB_name> --name <Health_Check_Name> --port <Port> --protocol
<HTTP/HTTPS/TCP> --path <Path>
```

Example

```
az network lb address-pool tunnel-interface update -g Justin_WCUS_Group --
lb-name Justin-WCUS-GWLB2 --name HTTP_HC --port 80 --protocol http --path /
```

7. Create the Inbound Rule.

```
az network lb rule create -g <Resource_group> --lb-name <GWLB_name> -n
<Inbound_Rule_Name> --protocol <TCP/UDP/ALL> --frontend-port <Port> --
backend-port <Port> --probe-name <Health_Check_Name> --backend-pools-name
<Backend_pool_name> <Backend_pool_name>
```

Example

```
az network lb rule create -g Justin_WCUS_Group --lb-name Justin-WCUS-GWLB2 -
n To_TPS --protocol All --frontend-port 0 --backend-port 0 --probe-name
HTTP_HC --backend-pools-name TPS_ETH1 TPS_ETH2
```

Verifying the Gateway LB deployment

To verify the Gateway LB deployment, perform the following steps:

1. Launch the web browser.
2. Access the Public IP address of the load balancer.

Chapter 5: Additional Resources – Where to go from here?

After logging into the vThunder GUI or CLI, you may need some assistance to configure the device. More information can be found in the latest ACOS Release Notes. This document has a list of new features, known issues, and other information to help you get started.

It is recommended to use the basic deployment instructions that appear in the 'System Configuration and Administration Guide' that is available on the A10 Networks support site.