Secure Router Manual

Microsoft Azure

© Copyright 2021 Rubicon Communications LLC

Jan 01, 2021

CONTENTS

1 Learn the Basics 2

2 Launch an Instance 3

3 Connect to the Instance 6

4 Configure Interface Addresses in TNSR 7

5 Configure Default Route in TNSR 10

6 Ping TNSR WAN Interface from your Network 11

7 Regional Market Availability 12

8 Additional Resources 14

9 Limitations 15

This zero-to-ping setup guide will explain how to get started using TNSR to route network traffic in an Azure Virtual Network environment.

Note: Visit the TNSR product page for additional information on purchasing access to TNSR on Azure.

The steps involved are:

© Copyright 2021 Rubicon Communications LLC

CHAPTER

ONE

LEARN THE BASICS

TNSR utilizes an optimized userspace data plane to forward packets at very high rates. On Azure, TNSR runs on a customized CentOS 7 Linux VM instance and is managed by connecting to a command-line interface (CLI) over SSH.

There are many different network designs possible in Azure. This guide assumes a TNSR instance will sit in a Virtual Network connected to a private subnet and a public subnet (one which has access to the Internet).

This guide will show how to bring up a TNSR instance with 3 Virtual Network Interfaces attached:

Management Interface The primary network interface on the instance is used for management of the TNSR instance. This is the interface reached via SSH to connect to the CLI on the TNSR instance. Packets received on this interface will not be forwarded to another interface. The interface is used for system functions such as DNS resolution and downloading software updates.

The management interface is required but it doesn't need to have **IP Forwarding** and **Accelerated Networking** options set.

TNSR WAN/Internet Interface The TNSR WAN interface is used by TNSR to connect to the Internet. A WAN interface will have a **Public IP Address** assigned and it will be attached to a subnet that has a route to an **Internet Gateway** in its **Route Table**.

TNSR LAN/Private Interface The TNSR LAN interface connects TNSR to a private Subnet in the Virtual Network. The instances in the private subnet do not have their own **Public IP Addresses** and the **Route Table** for the subnet does not have a route to an **Internet Gateway**, but instead has a route to the **TNSR LAN interface**.

Instances on the private subnet will use TNSR as their gateway to the Internet.

Each of the three network interfaces resides on a distinct subnet.

The examples in this guide use the following configuration:

Table 1: Example Azure Network Configuration

Item	Value
Virtual Network Address Space	10.5.0.0/16
WAN Subnet	10.5.0.0/24
LAN Subnet	10.5.1.0/24
Management Subnet	10.5.2.0/24

In a real production Virtual Network, the TNSR instance may have more than one WAN interface and/or more than one LAN interface. The concepts covered in this guide can be extended to additional interfaces.

There are some needed flags that cannot be configured using Azure Portal. This guide will cover all necessary steps using **azure-cli**.

CHAPTER

TWO

LAUNCH AN INSTANCE

Now launch an instance of TNSR:

1. Install `azure-cli`. Instructions can be found at https://docs.microsoft.com/en-us/cli/azure/install-azure-cli?view=azure-cli-latest
2. Login to your Azure account running:

```
$ az login
```

3. Configure the default location.

```
$ az configure --defaults location=centralus
```

4. Create a resource group to be used to store all TNSR related objects if you don't have it already.

```
$ az group create -n TNSR-Resource-Group
```

5. Create Virtual Network and Subnets.

```
$ az network vnet create \
    -n TNSR-VNet \
    -g TNSR-Resource-Group \
    --address-prefixes 10.5.0.0/16

$ az network vnet subnet create \
    -g TNSR-Resource-Group \
    --vnet-name TNSR-VNet \
    -n TNSR-WAN-Subnet \
    --address-prefixes 10.5.0.0/24

$ az network vnet subnet create \
    -g TNSR-Resource-Group \
    --vnet-name TNSR-VNet \
    -n TNSR-LAN-Subnet \
    --address-prefixes 10.5.1.0/24

$ az network vnet subnet create \
    -g TNSR-Resource-Group \
    --vnet-name TNSR-VNet \
    -n TNSR-MGMT-Subnet \
    --address-prefixes 10.5.2.0/24
```

6. Create Public IPs to be used by WAN and Management interfaces.

```
$ az network public-ip create \
    -g TNSR-Resource-Group \
    -n TNSR-WAN-IP

$ az network public-ip create \
    -g TNSR-Resource-Group \
    -n TNSR-MGMT-IP
```

7. Create a Network Security Group (NSG).

```
$ az network nsg create -n TNSR-MGMT-NSG -g TNSR-Resource-Group
$ az network nsg rule create \
    --name MGMT_Allow_SSH \
    --nsg-name TNSR-MGMT-NSG \
    -g TNSR-Resource-Group \
    --priority 100 \
    --access Allow \
    --destination-port-ranges 22 \
    --direction Inbound \
    --protocol Tcp
```

8. Create the Management Network Interface.

```
$ az network nic create \
    -g TNSR-Resource-Group \
    --vnet-name TNSR-VNet \
    --subnet TNSR-MGMT-Subnet \
    -n TNSR-MGMT-nic \
    --public-ip-address TNSR-MGMT-IP \
    --network-security-group TNSR-MGMT-NSG
```

9. Create the WAN Network Interface.

```
$ az network nic create \
    -g TNSR-Resource-Group \
    --vnet-name TNSR-VNet \
    --subnet TNSR-WAN-Subnet \
    -n TNSR-WAN-nic \
    --public-ip-address TNSR-WAN-IP \
    --ip-forward \
    --accelerated-network
```

10. Create the LAN Network Interface.

```
$ az network nic create \
    -g TNSR-Resource-Group \
    --vnet-name TNSR-VNet \
    --subnet TNSR-LAN-Subnet \
    -n TNSR-LAN-nic \
    --ip-forward \
    --accelerated-network
```

11. Choose the VM Size to be used. To get a list of sizes that are able to run TNSR, run the following command and export a variable called **TNSR_SIZE** with it.

```
$ az vm list-sizes \
    --query "[?numberOfCores >= \`4\`] | [?memoryInMb >= \`8192\`].name | sort(@)
↪" \
```

(continues on next page)

© Copyright 2021 Rubicon Communications LLC

(continued from previous page)

```
    --output tsv

$ export TNSR_SIZE=<FILL DESIRED SIZE HERE>

# EXAMPLE:
$ export TNSR_SIZE="Standard_DS4_v2"
```

12. Choose the TNSR image **URN** to be used from the list obtained with the following command and export a variable called **TNSR_URN** with it.

```
$ az vm image list \
    --publisher Netgate \
    --all \
    --query "[?contains(offer,'tnsr')].{Sku:sku, Version:version Urn:urn}" \
    --output table

$ export TNSR_URN="netgate:netgate-tnsr-azure-fw-vpn-router:netgate-tnsr:20.02.2"
```

13. Export a variable called **TNSR_SSH_KEY** containing a path to a valid SSH public key.

```
$ export TNSR_SSH_KEY="~/.ssh/id_rsa.pub"
```

14. Accept Azure Marketplace terms so that the image can be used to create VMs.

```
$ az vm image terms accept --urn ${TNSR_URN}
```

Note: Previous versions of `Azure CLI` used the command `$ az vm image accept-terms --urn ${TNSR_URN}`

---

15. Create a Storage Account.

```
$ az storage account create -n tnsrsa -g TNSR-Resource-Group
```

16. Create the TNSR Virtual Machine.

```
$ az vm create \
    --admin-username tnsr \
    --image ${TNSR_URN} \
    --name TNSR-Instance1 \
    --nics TNSR-MGMT-nic TNSR-WAN-nic TNSR-LAN-nic \
    --os-disk-size-gb 20 \
    --resource-group TNSR-Resource-Group \
    --size ${TNSR_SIZE} \
    --ssh-key-value ${TNSR_SSH_KEY} \
    --boot-diagnostics tnsrsa
```

© Copyright 2021 Rubicon Communications LLC

CHAPTER

THREE

CONNECT TO THE INSTANCE

The TNSR instance does not have a default password. SSH connections to this instance require key-based authentication using an SSH key selected when launching the instance.

The default account is named `tnsr`.

The Management interface Public IP can be discovered from the Azure CLI by running:

```
$ az network public-ip show \
    -n TNSR-MGMT-IP \
    -g TNSR-Resource-Group \
    --query "{ipAddress:ipAddress}" \
    --output tsv
```

To connect from a shell prompt in a Unix/Linux terminal, type the following:

```
$ ssh -i <my_key_file> tnsr@<MGMT_public_ip_addr>
```

Substitute the actual key file name instead of typing `<my_key_file>` and the management interface Public IP Address instead of typing `<mgmt_public_ip_addr>`.

The ssh client will print a warning similar to:

```
The authenticity of host 'x.x.x.x' can't be established.
ECDSA key fingerprint is SHA256:6/LDXVPpD2v6hnWdFHFWZhkCbSpMcaH4tBgTuDLAa40.
Are you sure you want to continue connecting (yes/no)?
```

This warning only appears the first time connecting using SSH on a given system and user account. Type `yes` to continue connecting.

If all went well, the TNSR CLI will automatically be launched, resulting in output similar to the following:

```
Netgate TNSR
Version: tnsr-v19.02.1-2
Build timestamp: Mon Apr  8 15:16:48 2019 CDT
Git Commit: 0x8b47d140

This TNSR instance is not configured for package updates.
For information see http://www.netgate.com/docs/tnsr/updating/index.html

TNSR-Instance1 tnsr#
```

CHAPTER

FOUR

CONFIGURE INTERFACE ADDRESSES IN TNSR

Now that the TNSR CLI is open, start configuring the TNSR instance. First, configure the network interfaces and bring them up.

In TNSR, type `show interface` to view the interface configurations. Here's an example of what will appear:

```
TNSR-Instance1 tnsr# show interface

Interface: FortyGigabitEthernet2
    Admin status: down
    Link down, 100 Mbit/sec, full duplex
    Link MTU: 9206 bytes
    MAC address: 00:0d:3a:41:f6:b1
    IPv4 Route Table: ipv4-VRF:0
    IPv6 Route Table: ipv6-VRF:0
    counters:
      received: 0 bytes, 0 packets, 0 errors
      transmitted: 0 bytes, 0 packets, 0 errors
      0 drops, 0 punts, 0 rx miss, 0 rx no buffer

Interface: FortyGigabitEthernet3
    Admin status: down
    Link down, 100 Mbit/sec, full duplex
    Link MTU: 9206 bytes
    MAC address: 00:0d:3a:41:f7:20
    IPv4 Route Table: ipv4-VRF:0
    IPv6 Route Table: ipv6-VRF:0
    counters:
      received: 0 bytes, 0 packets, 0 errors
      transmitted: 0 bytes, 0 packets, 0 errors
      0 drops, 0 punts, 0 rx miss, 0 rx no buffer
```

The interface order follows the same order NICs were passed to parameter `--nics` to `az vm create` at *Launch an Instance*. In this guide, we will have `FortyGigabitEthernet2` as **WAN** and `FortyGigabitEthernet3` as **LAN**.

During the process of creating Network Interfaces, a private IP address was assigned to each interface. We will configure those addresses on the interfaces in TNSR in order to communicate with other instances in the Virtual Network.

Configure WAN interface:

1. Discover assigned IP address in the Azure CLI.

```
$ az network nic show \
    -g TNSR-Resource-Group \
```

(continues on next page)

(continued from previous page)

```
    -n TNSR-WAN-nic \
    --query "ipConfigurations[].privateIpAddress" \
    -o tsv
10.5.0.4
```

2. Configure the interface in the TNSR CLI.

```
TNSR-Instance1 tnsr# configure
TNSR-Instance1 tnsr(config)# interface FortyGigabitEthernet2
TNSR-Instance1 tnsr(config-interface)# ip address 10.5.0.4/24
TNSR-Instance1 tnsr(config-interface)# enable
TNSR-Instance1 tnsr(config-interface)# description TNSR-Instance1 WAN
TNSR-Instance1 tnsr(config-interface)# exit
```

This sets an address, brings up the interface, and sets a description to serve as a reminder of the interface identity & purpose.

Substitute a different Private IP address/mask and description as needed.

Configure LAN interface:

1. Discover the assigned IP address from the Azure CLI.

```
$ az network nic show \
    -g TNSR-Resource-Group \
    -n TNSR-LAN-nic \
    --query "ipConfigurations[].privateIpAddress" \
    -o tsv
10.5.1.4
```

2. Configure the interface in the TNSR CLI.

```
TNSR-Instance1 tnsr(config)# interface FortyGigabitEthernet3
TNSR-Instance1 tnsr(config-interface)# ip address 10.5.1.4/24
TNSR-Instance1 tnsr(config-interface)# enable
TNSR-Instance1 tnsr(config-interface)# description TNSR-Instance1 LAN
TNSR-Instance1 tnsr(config-interface)# exit
```

Again, substitute the interface Private IP address/mask and description as needed.

Check the interface status in TNSR again by typing `show interface`.

```
TNSR-Instance1 tnsr# show interface

Interface: FortyGigabitEthernet2
    Description: TNSR-Instance1 WAN
    Admin status: up
    Link up, 100 Mbit/sec, full duplex
    Link MTU: 9206 bytes
    MAC address: 00:0d:3a:41:f6:b1
    IPv4 Route Table: ipv4-VRF:0
    IPv4 addresses:
        10.5.0.4/24
    IPv6 Route Table: ipv6-VRF:0
    counters:
      received: 480 bytes, 8 packets, 0 errors
      transmitted: 822 bytes, 9 packets, 0 errors
      8 drops, 0 punts, 0 rx miss, 0 rx no buffer
```

(continues on next page)

(continued from previous page)

```
Interface: FortyGigabitEthernet3
    Description: TNSR-Instance1 LAN
    Admin status: up
    Link up, 100 Mbit/sec, full duplex
    Link MTU: 9206 bytes
    MAC address: 00:0d:3a:41:f7:20
    IPv4 Route Table: ipv4-VRF:0
    IPv4 addresses:
        10.5.1.4/24
    IPv6 Route Table: ipv6-VRF:0
    counters:
      received: 0 bytes, 0 packets, 0 errors
      transmitted: 892 bytes, 10 packets, 0 errors
      0 drops, 0 punts, 0 rx miss, 0 rx no buffer
```

The output shows that the interfaces are up and configured, and the counters show that a few packets have been received.

It is now possible to verify connectivity with the `ping` command from the TNSR CLI.

```
TNSR-Instance1 tnsr# ping www.netgate.com
PING www.netgate.com (208.123.73.73) 56(84) bytes of data.
64 bytes from www.netgate.com (208.123.73.73): icmp_seq=1 ttl=49 time=19.6 ms
64 bytes from www.netgate.com (208.123.73.73): icmp_seq=2 ttl=49 time=19.5 ms
64 bytes from www.netgate.com (208.123.73.73): icmp_seq=3 ttl=49 time=19.4 ms
64 bytes from www.netgate.com (208.123.73.73): icmp_seq=4 ttl=49 time=20.1 ms
64 bytes from www.netgate.com (208.123.73.73): icmp_seq=5 ttl=49 time=19.5 ms
64 bytes from www.netgate.com (208.123.73.73): icmp_seq=6 ttl=49 time=19.5 ms
64 bytes from www.netgate.com (208.123.73.73): icmp_seq=7 ttl=49 time=19.6 ms
64 bytes from www.netgate.com (208.123.73.73): icmp_seq=8 ttl=49 time=19.6 ms
64 bytes from www.netgate.com (208.123.73.73): icmp_seq=9 ttl=49 time=19.5 ms
64 bytes from www.netgate.com (208.123.73.73): icmp_seq=10 ttl=49 time=19.5 ms

--- www.netgate.com ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9014ms
rtt min/avg/max/mdev = 19.435/19.616/20.136/0.262 ms
```

© Copyright 2021 Rubicon Communications LLC

CHAPTER

FIVE

CONFIGURE DEFAULT ROUTE IN TNSR

In order for the TNSR data plane to forward packets outside of the VPC to the Internet, a default route needs to be configured which sets a next hop of the VPC gateway for the WAN subnet using the TNSR CLI.

Configure a default route by typing the commands in TNSR as shown below.

```
TNSR-Instance1 tnsr# configure
TNSR-Instance1 tnsr(config)# route ipv4 table ipv4-VRF:0
TNSR-Instance1 tnsr(config-route-table-v4)# route 0.0.0.0/0
TNSR-Instance1 tnsr(config-rttbl4-next-hop)# next-hop 1 via 10.5.0.1␣
↪FortyGigabitEthernet2
TNSR-Instance1 tnsr(config-rttbl4-next-hop)# exit
TNSR-Instance1 tnsr(config-route-table-v4)# exit
TNSR-Instance1 tnsr(config)# exit
TNSR-Instance1 tnsr#
```

CHAPTER

SIX

PING TNSR WAN INTERFACE FROM YOUR NETWORK

The instance should now be reachable via ICMP echo request (ping) using the Public IP Address associated to the TNSR WAN Interface.

To find the Public IP address associated to the TNSR WAN Interface, run:

```
$ az network public-ip show \
    -n TNSR-WAN-IP \
    -g TNSR-Resource-Group \
    --query "{ipAddress:ipAddress}" \
    --output tsv
```

Now, try to ping the **Public IP Address** of the TNSR WAN Interface.

```
$ ping -c 5 40.122.49.143
PING 40.122.49.143 (40.122.49.143) 56(84) bytes of data.
64 bytes from 40.122.49.143: icmp_seq=1 ttl=49 time=19.9 ms
64 bytes from 40.122.49.143: icmp_seq=2 ttl=49 time=19.8 ms
64 bytes from 40.122.49.143: icmp_seq=3 ttl=49 time=19.8 ms
64 bytes from 40.122.49.143: icmp_seq=4 ttl=49 time=19.6 ms
64 bytes from 40.122.49.143: icmp_seq=5 ttl=49 time=19.9 ms

--- 40.122.49.143 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 19.649/19.862/19.940/0.140 ms
```

Warning: Once the Host OS is capable of reaching the Internet, check for updates (Updating TNSR) before proceeding. This ensures the security and integrity of the router before TNSR interfaces are exposed to the Internet.

References

- *Regional Market Availability*
- *Additional Resources*
- Resource Library

CHAPTER

SEVEN

REGIONAL MARKET AVAILABILITY

The tables below represent the current availability by regional market. If the desired regional market is not listed, refer to the Microsoft Regions availability or submit a support ticket directly to Microsoft Azure.

Table 1: Microsoft Azure Available Regions

Market	pfSense
Armenia	Available
Australia	*
Austria	Available
Belarus	Available
Belgium	Available
Brazil	Available
Canada	Available
Croatia	Available
Cyprus	Available
Czechia	Available
Denmark	Available
Estonia	Available
Finland	Available
France	Available
Germany	Available
Greece	Available
Hungary	Available
India	Available
Ireland	Available
Italy	Available
Korea	Available
Latvia	Available
Liechtenstein	Available
Lithuania	Available
Luxembourg	Available
Malta	Available
Monaco	Available
Netherlands	Available
New Zealand	Available
Norway	Available
Poland	Available
Portugal	Available
Puerto Rico	Available

continues on next page

Table 1 – continued from previous page

Market	pfSense
Romania	Available
Russia	Available
Saudi Arabia	Available
Serbia	Available
Slovakia	Available
Slovenia	Available
South Africa	Available
Spain	Available
Sweden	Available
Switzerland	Available
Taiwan	Available
Turkey	Available
United Arab Emirates	Available
United Kingdom	Available
United States	Available

* Australia is a Microsoft Managed Country for sales through all customer purchase scenarios except the Enterprise Agreement customer purchase scenario.

© Copyright 2021 Rubicon Communications LLC

CHAPTER

EIGHT

ADDITIONAL RESOURCES

8.1 Professional Services

Support does not cover more complex tasks such as network design and conversion from other firewalls. These items are offered as professional services and can be purchased and scheduled accordingly.

https://www.netgate.com/our-services/professional-services.html

8.2 Netgate Training

Netgate training offers training courses for increasing your knowledge of Netgate products and services. Whether you need to maintain or improve the security skills of your staff or offer highly specialized support and improve your customer satisfaction; Netgate training has got you covered.

https://www.netgate.com/training/

8.3 Resource Library

To learn more about how to use your Netgate appliance and for other helpful resources, make sure to browse our Resource Library.

https://www.netgate.com/resources/

CHAPTER

NINE

LIMITATIONS

There are issues running TNSR on Azure which can lead to problems when communicating using public IP addresses between multiple TNSR instances all running on Azure.

When a batch of packets is read using the DPDK `netvsc` PMD, the driver occasionally populates an invalid buffer address. Attempting to process that packet results in a segmentation fault in the dataplane (VPP).

This issue only occurs when sending packets to a public IP address that is associated with a NIC that is managed by TNSR on Azure. Sending to the NIC private address from another VM in the same vnet does not result in a crash. Sending to the public IP address from a host outside of Azure also does not result in a crash.

The issue is being investigated by Microsoft.