SOFTWARE DOCUMENTATION

1  Tested Platforms  
2  Untested Platforms

2
50
TNSR software is available pre-installed on Netgate hardware or as a Bare Metal Image (BMI) for use on Commercial Off the Self (COTS) hardware or in virtual environments. For quotes, updates, and more information about TNSR, please visit tnsr.com or contact TNSR sales.

**General System Requirements**

- 64-bit x86 processor with SSE4.2 extensions
- Minimum of 4GB RAM for basic routing functionality
  
  RAM requirements increase significantly if the installation will need to handle large volumes of connections, large volumes of routing data (e.g. full BGP feeds), or multiple worker threads.

  **See also:**
  
  See Memory Usage and Tuning for information on how to determine optimal memory sizes based on what the router must handle.

- Minimum of 20GB Storage
- Minimum of two network interface ports, three network interface ports recommended (two for dataplane, one for management)
- Option to disable Secure Boot if using UEFI, as it is incompatible with the TNSR dataplane

**Compatibility**

TNSR is a platform for high-speed packet processing, delivered as services that run on top of an operating system. TNSR 21.11 and later are based on Ubuntu, currently the 20.04 LTS release for x86 processors. This platform has been tested by Netgate, so that means most compatibility questions can be resolved by checking whether the hardware can run Ubuntu Linux 20.04.

- **Tested Platforms**
- **Untested Platforms**
These systems and components have been tested extensively by Netgate and are known to work well with TNSR. In addition to the support included with the TNSR license, Netgate warranty and technical support is available for purchase for Netgate-branded appliances.

1.1 Netgate Appliances

- Netgate 5100
- Netgate 1537-1U
- Netgate 1541-1U

1.1.1 Netgate 5100 Secure Router Manual

This Quick Start Guide covers the first time connection procedures for the Netgate 5100 Secure Router and also provides information needed to stay up and running.
Getting Started

Use the following steps to configure the TNSR Secure Router.

1. To configure the Network Interfaces and gaining access to the Internet, follow the instructions provided in the Zero-to-Ping documentation.

   **Note:** Not all steps in the Zero-to-Ping documentation will be necessary for every configuration scenario.

2. 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.

3. Finally, configure the TNSR instance to meet the specific use case. The topics are listed on the left column of the TNSR Documentation site. There are also TNSR Configuration Example Recipes that might be of assistance when configuring TNSR.

## Input and Output Ports

### Rear Side

![Netgate 5100 Network Interface Layout](image)

### Network Ports

<table>
<thead>
<tr>
<th>Port Label</th>
<th>Linux Label</th>
<th>TNSR Label</th>
<th>Port Type</th>
<th>Port Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>IGB0</td>
<td>enp3s0</td>
<td>GigabitEthernet3/0/0</td>
<td>RJ-45</td>
<td>1 Gbps</td>
</tr>
<tr>
<td>IGB1</td>
<td>enp4s0</td>
<td>GigabitEthernet4/0/0</td>
<td>RJ-45</td>
<td>1 Gbps</td>
</tr>
<tr>
<td>IX0</td>
<td>enp6s0f0</td>
<td>GigabitEthernet6/0/0</td>
<td>RJ-45</td>
<td>1 Gbps</td>
</tr>
<tr>
<td>IX1</td>
<td>enp6s0f1</td>
<td>GigabitEthernet6/0/1</td>
<td>RJ-45</td>
<td>1 Gbps</td>
</tr>
<tr>
<td>IX2</td>
<td>enp8s0f0</td>
<td>GigabitEthernet8/0/0</td>
<td>RJ-45</td>
<td>1 Gbps</td>
</tr>
<tr>
<td>IX3</td>
<td>enp8s0f1</td>
<td>GigabitEthernet8/0/1</td>
<td>RJ-45</td>
<td>1 Gbps</td>
</tr>
</tbody>
</table>

**Note:** Default Host OS Interface is enp3s0. The Host OS Interface is one network interface that is only available to
the host OS and not available in TNSR. Though technically optional, the best practice is to have one for accessing and updating the host OS.

<table>
<thead>
<tr>
<th>Status LED</th>
<th>State</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Left LED (Link Status)</td>
<td>Solid Amber</td>
<td>Link has been established and there is no activity on this port</td>
</tr>
<tr>
<td></td>
<td>Blinking Amber</td>
<td>Link has been established and there is activity on this port</td>
</tr>
<tr>
<td></td>
<td>Off</td>
<td>No link has been established</td>
</tr>
<tr>
<td>Right LED (Speed)</td>
<td>Solid Green</td>
<td>Operating as a 100 Mbps connection</td>
</tr>
<tr>
<td></td>
<td>Blinking Amber</td>
<td>Operating as a Gigabit connection (1000 Mbps)</td>
</tr>
<tr>
<td></td>
<td>Off</td>
<td>No link has been established</td>
</tr>
</tbody>
</table>

**Other Ports and Indicators**

- Console (Mini-USB)
- Status LEDs
- 2x USB 3.0

**Front Side**

1. Recessed Reset Button
2. Power Button
3. Power (12VDC with threaded locking connector)

Note: The power button on the Netgate 5100 has been programmed to perform a graceful shutdown when depressed.
Connecting to Console Port

Simple Configuration

Below are the simple instructions for connecting to the console port with Microsoft Windows. If these steps do not work or if the client is using an operating system other than Windows, then please skip forward to Advanced Configuration.

Serial Terminal Emulation Client

A serial terminal emulation program is required to access the Netgate appliance console through the serial interface. Microsoft Windows no longer includes HyperTerminal in Versions 7 and up. PuTTY is free and can be downloaded from:

http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html

Configuring Serial Terminal Emulator

PuTTY must be configured to communicate with the Netgate appliance. In order to do so, first identify which COM Port the computer has assigned to the serial port. Even if the serial port is assigned to COM1 in the BIOS, Windows may remap it to a different COM Port.

To determine this, open Windows Device Manager and view the COM port assignment:

Before Driver

After Driver
**Note:** The first time the computer is connected to the SG-5100, it may take up to 3 minutes for the driver to install. It should install automatically for Windows 7 and above.

Open PuTTY and locate the **Session** display as shown below. For the **Connection type**, select **Serial**. Set **Serial line** to the COM Port that is displayed in Windows Device Manager, **COM3** for this example, and the **Speed** to 115200 bits per second, the speed of the BIOS in this case.

Select **Open** and the console screen will be displayed.
Advanced Configuration

A Prolific PL2303 USB-to-UART bridge is used to provide access to the serial port that acts as a system console. This is exposed via a **USB Mini-B (5-pin)** port on the front of the case. There are several steps required to access the system console via this port.

Install the Driver

Install an appropriate PL2303 USB to UART Bridge VCP (virtual COM port) driver on the workstation used to connect with the system if needed. There are drivers available for Windows, macOS, and Linux available in the Download Software section of the Prolific Website.

**Note:** Recent versions of FreeBSD and many Linux distributions include this driver and will not require manual installation.

Connect a USB Cable

Next, locate an appropriate USB cable. The type of cable required for the serial console has a **USB Mini-B (5-pin)** connector on one end and a regular USB (Type A) plug on the other end. These cables are commonly used with smaller USB peripherals such as GPS units, cameras, and so on.

Attach the USB cable between a workstation and the system. Gently push the Mini-B plug end into the console port on the system and connect the USB type A plug into an available USB port on the workstation.

**Tip:** Be certain to gently push in the Mini-B connector on the system side completely. With most cables there will be a tangible “click”, “snap”, or similar indication when the cable is fully engaged.

Locate the Console Port Device

The appropriate device to attach the terminal program to each platform varies by platform and must be located before attempting to connect to the console.

Windows

To locate the device name on Windows, open **Device Manager** and expand the section for **Ports (COM & LPT)**. Look for an entry with a title such as **Prolific USB-to-Serial Comm Port**. If there is a label in the name that contains “COMX” where X is a decimal digit (e.g. COM1), that value is what would be used as the port in the terminal program.
macOS

The device associated with the system console is likely to show up as /dev/cu.usbserial-<id>.

Linux

The device associated with the system console is likely to show up as /dev/ttyUSB0. Look for messages about the device attaching in the system log files or by running dmesg.

**Note:** If the device does not appear in /dev/, see the note above in the driver section about manually loading the Linux driver and then try again.

FreeBSD

The device associated with the system console is likely to show up as /dev/cuaU0. Look for messages about the device attaching in the system log files or by running dmesg.

**Launch a Terminal Program**

Use a terminal program to connect to the system console port. PuTTY is a popular terminal program that is available on various operating systems. Some other choices of terminal programs:

- Linux: screen, PuTTY, minicom, dterm
- macOS: screen, ZTerm, cu
- Windows: PuTTY, SecureCRT, **Do not use Hyperterminal**
- FreeBSD: screen, cu

The settings to use within the terminal program are:

- **Speed** 115200 baud
- **Data bits** 8
- **Parity** none
- **Stop bits** 1
- **Flow Control** Off or XON/OFF.

  Hardware flow control (RTS/CTS) **must** be disabled.

**Client-Specific Examples**

**PuTTY**

Launch PuTTY and configure it for a **Serial** type connection with a speed of **115200** using the port name located previously.
PuTTY generally handles most cases OK but can have issues with line drawing characters on certain platforms. These settings seem to work best (tested on Windows):

- **Window** Columns x Rows = 80x24
- **Window > Appearance** Font = Courier New 10pt or Consolas 10pt
- **Window > Translation** Remote Character Set = Use font encoding or UTF-8
- **Window > Translation** Handling of line drawing characters = Use font in both ANSI and OEM modes or Use Unicode line drawing code points
- **Window > Colours** Indicate bolded text by changing = The colour

### GNU screen

In many cases screen may be invoked simply by using the proper command line:

- **macOS**
  
  ```bash
  sudo screen /dev/cu.usbserial-<id> 115200
  ```

- **Linux**

  ```bash
  sudo screen /dev/ttyUSB0 115200
  ```

- **FreeBSD**

  ```bash
  sudo screen /dev/cuaU0 115200
  ```

If portions of the text are unreadable but appear to be properly formatted, the most likely culprit is a character encoding mismatch in the terminal. For example, on macOS this is commonly required:

```bash
sudo screen -U /dev/cu.usbserial-<id> 115200
```

Adding the -U parameter to the `screen` command line arguments forces it to use UTF-8 for character encoding.
Troubleshooting

No Serial Output

If there is no output at all, check the following items:

- Ensure the cable is correctly attached and fully inserted
- Ensure the terminal program is using the correct port
- Ensure the terminal program is configured for the correct speed. The default BIOS speed is 115200, and many other modern operating systems use that speed as well. Some older operating systems or custom configurations may use slower speeds such as 9600 or 38400.
- Ensure the operating system is configured for the proper console (e.g. ttyS1 in Linux). Consult the various operating install guides on this site for further information.

Garbled Serial Output

If the serial output appears to be garbled, binary, or random characters check the following items:

- Ensure the terminal program is configured for the correct speed. (See “No Serial Output” above)
- Ensure the terminal program is configured for the proper character encoding, such as UTF-8 or Latin-1, depending on the operating system. (See the previous entry under “GNU screen”)

Serial Output Stops After the BIOS

If serial output is shown for the BIOS but stops afterward, check the following items:

- Ensure the terminal program is configured for the correct speed for the installed operating system. (See “No Serial Output” above)
- Ensure the installed operating system is configured to activate the serial console.
- Ensure the installed operating system is configured for the proper console (e.g. ttyS1 in Linux). Consult the various operating install guides on this site for further information.
- If booting from a USB flash drive, ensure that the drive was written correctly and contains a bootable operating system image.

Additional Resources

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
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/

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/

Warranty and Support

- One year manufacturer’s warranty.
- Please contact Netgate for warranty information or view the Product Lifecycle page.
- All Specifications subject to change without notice.

Enterprise Support is included with an active software subscription, for more information view the Netgate Global Support page.

See also:

For more information on how to use pfSense® software, see the TNSR Documentation and Resource Library.

1.1.2 Netgate 1537 Secure Router Manual

This Quick Start Guide covers the first time connection procedures for the Netgate 1537 1U Secure Router and also provides information needed to stay up and running.
Getting Started

Use the following steps to configure the TNSR Secure Router.

1. To configure the Network Interfaces and gaining access to the Internet, follow the instructions provided in the Zero-to-Ping documentation.

   Note: Not all steps in the Zero-to-Ping documentation will be necessary for every configuration scenario.

2. 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.

3. Finally, configure the TNSR instance to meet the specific use case. The topics are listed on the left column of the TNSR Documentation site. There are also TNSR Configuration Example Recipes that might be of assistance when configuring TNSR.

Input and Output Ports

Contents

- Input and Output Ports
  - Default Configuration
    - Network Ports
    - Other I/O
    - Status LEDs
  - With CPIC-8955 Accelerator Card with Intel® QuickAssist Technology
  - With 2 Port Intel 1 Gbps Ethernet Expansion Card
  - With 2 Port Intel 10 Gbps X710 SFP+ Expansion Card
  - With 4 Port Intel 1 Gbps Ethernet Expansion Card
  - With 4 Port Intel 10 Gbps X710 SFP+ Expansion Card

Default Configuration
Network Ports

<table>
<thead>
<tr>
<th>Port</th>
<th>Linux Label</th>
<th>TNSR Label</th>
<th>Port Type</th>
<th>Port Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>eno1</td>
<td>GigabitEthernet6/0/0</td>
<td>RJ-45</td>
<td>1 Gbps</td>
</tr>
<tr>
<td>2</td>
<td>eno2</td>
<td>GigabitEthernet6/0/1</td>
<td>RJ-45</td>
<td>1 Gbps</td>
</tr>
<tr>
<td>3</td>
<td>eno3</td>
<td>TenGigabitEthernet3/0/0</td>
<td>SFP+</td>
<td>10 Gbps</td>
</tr>
<tr>
<td>4</td>
<td>eno4</td>
<td>TenGigabitEthernet3/0/1</td>
<td>SFP+</td>
<td>10 Gbps</td>
</tr>
</tbody>
</table>

**Note:** Default Host OS Interface is eno1. The Host OS Interface is one network interface that is only available to the host OS and not available in TNSR. Though technically optional, the best practice is to have one for accessing and updating the host OS.

Other I/O

<table>
<thead>
<tr>
<th>Port</th>
<th>I/O Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>2x USB 3.0</td>
</tr>
<tr>
<td>6</td>
<td>IPMI</td>
</tr>
<tr>
<td>7</td>
<td>VGA</td>
</tr>
<tr>
<td>8</td>
<td>Reset &amp; Power buttons</td>
</tr>
<tr>
<td>9</td>
<td>Status LEDs (see table below)</td>
</tr>
</tbody>
</table>

Status LEDs

![Status LEDs Image]
<table>
<thead>
<tr>
<th>LED</th>
<th>State</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>9a</td>
<td>Continuously on and red</td>
<td>An overheat condition has occurred. (This may be caused by cable congestion.)</td>
</tr>
<tr>
<td></td>
<td>Blinking red (1Hz)</td>
<td>Fan failure, check for an inoperative fan.</td>
</tr>
<tr>
<td></td>
<td>Blinking red (0.25Hz)</td>
<td>Power failure, check for a non-operational power supply.</td>
</tr>
<tr>
<td></td>
<td>Solid blue</td>
<td>Local UID has been activated. Use this function through IPMI to locate the server in a rack mount environment.</td>
</tr>
<tr>
<td></td>
<td>Blinking blue</td>
<td>Remote UID is on. Use this function through IPMI to identify the server from a remote location.</td>
</tr>
<tr>
<td>9b</td>
<td>Flashing</td>
<td>Indicates network activity on igb1 (upper left port).</td>
</tr>
<tr>
<td>9c</td>
<td>Flashing</td>
<td>Indicates network activity on igb0 (lower left port).</td>
</tr>
<tr>
<td>9d</td>
<td>Flashing</td>
<td>Indicates IDE channel activity on the hard drive.</td>
</tr>
<tr>
<td>9e</td>
<td>Illuminated</td>
<td>Indicates power is being supplied to the system power supply units. This LED should normally be illuminated when the system is operating.</td>
</tr>
<tr>
<td></td>
<td>Off</td>
<td>Indicates no power is being supplied to the system power supply. System is powered off.</td>
</tr>
</tbody>
</table>
With CPIC-8955 Accelerator Card with Intel® QuickAssist Technology

<table>
<thead>
<tr>
<th>Port</th>
<th>Linux Label</th>
<th>TNSR Label</th>
<th>Port Type</th>
<th>Port Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>eno1</td>
<td>GigabitEthernet6/0/0</td>
<td>RJ-45</td>
<td>1 Gbps</td>
</tr>
<tr>
<td>2</td>
<td>eno2</td>
<td>GigabitEthernet6/0/1</td>
<td>RJ-45</td>
<td>1 Gbps</td>
</tr>
<tr>
<td>3</td>
<td>eno3</td>
<td>TenGigabitEthernet3/0/0</td>
<td>SFP+</td>
<td>10 Gbps</td>
</tr>
<tr>
<td>4</td>
<td>eno4</td>
<td>TenGigabitEthernet3/0/1</td>
<td>SFP+</td>
<td>10 Gbps</td>
</tr>
</tbody>
</table>

**Note:** Default Host OS Interface is eno1. The Host OS Interface is one network interface that is only available to the host OS and not available in TNSR. Though technically optional, the best practice is to have one for accessing and updating the host OS.

With 2 Port Intel 1 Gbps Ethernet Expansion Card

<table>
<thead>
<tr>
<th>Port</th>
<th>Linux Label</th>
<th>TNSR Label</th>
<th>Port Type</th>
<th>Port Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ens2f0</td>
<td>GigabitEthernet5/0/0</td>
<td>RJ-45</td>
<td>1 Gbps</td>
</tr>
<tr>
<td>2</td>
<td>ens2f1</td>
<td>GigabitEthernet5/0/1</td>
<td>RJ-45</td>
<td>1 Gbps</td>
</tr>
<tr>
<td>3</td>
<td>eno1</td>
<td>GigabitEthernet7/0/0</td>
<td>RJ-45</td>
<td>1 Gbps</td>
</tr>
<tr>
<td>4</td>
<td>eno2</td>
<td>GigabitEthernet7/0/1</td>
<td>RJ-45</td>
<td>1 Gbps</td>
</tr>
<tr>
<td>5</td>
<td>eno3</td>
<td>TenGigabitEthernet3/0/0</td>
<td>SFP+</td>
<td>10 Gbps</td>
</tr>
<tr>
<td>6</td>
<td>eno4</td>
<td>TenGigabitEthernet3/0/1</td>
<td>SFP+</td>
<td>10 Gbps</td>
</tr>
</tbody>
</table>

**Note:** Default Host OS Interface is eno1. The Host OS Interface is one network interface that is only available to the host OS and not available in TNSR. Though technically optional, the best practice is to have one for accessing and updating the host OS.
### With 2 Port Intel 10 Gbps X710 SFP+ Expansion Card

<table>
<thead>
<tr>
<th>Port</th>
<th>TNSR Label</th>
<th>Port Type</th>
<th>Port Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>TenGigabitEthernet5/0/1</td>
<td>SFP+</td>
<td>10 Gbps</td>
</tr>
<tr>
<td>2</td>
<td>TenGigabitEthernet5/0/0</td>
<td>SFP+</td>
<td>10 Gbps</td>
</tr>
<tr>
<td>3</td>
<td>GigabitEthernet7/0/0</td>
<td>RJ-45</td>
<td>1 Gbps</td>
</tr>
<tr>
<td>4</td>
<td>GigabitEthernet7/0/1</td>
<td>RJ-45</td>
<td>1 Gbps</td>
</tr>
<tr>
<td>5</td>
<td>TenGigabitEthernet3/0/0</td>
<td>SFP+</td>
<td>10 Gbps</td>
</tr>
<tr>
<td>6</td>
<td>TenGigabitEthernet3/0/1</td>
<td>SFP+</td>
<td>10 Gbps</td>
</tr>
</tbody>
</table>

**Note:** Default Host OS Interface is `eno1`. The Host OS Interface is one network interface that is only available to the host OS and not available in TNSR. Though technically optional, the best practice is to have one for accessing and updating the host OS.

### With 4 Port Intel 1 Gbps Ethernet Expansion Card

<table>
<thead>
<tr>
<th>Port</th>
<th>Linux Label</th>
<th>TNSR Label</th>
<th>Port Type</th>
<th>Port Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ens2f0</td>
<td>GigabitEthernet5/0/0</td>
<td>RJ-45</td>
<td>1 Gbps</td>
</tr>
<tr>
<td>2</td>
<td>ens2f1</td>
<td>GigabitEthernet5/0/1</td>
<td>RJ-45</td>
<td>1 Gbps</td>
</tr>
<tr>
<td>3</td>
<td>ens2f2</td>
<td>GigabitEthernet5/0/2</td>
<td>RJ-45</td>
<td>1 Gbps</td>
</tr>
<tr>
<td>4</td>
<td>ens2f3</td>
<td>GigabitEthernet5/0/3</td>
<td>RJ-45</td>
<td>1 Gbps</td>
</tr>
<tr>
<td>5</td>
<td>eno1</td>
<td>GigabitEthernet7/0/0</td>
<td>RJ-45</td>
<td>1 Gbps</td>
</tr>
<tr>
<td>6</td>
<td>eno2</td>
<td>GigabitEthernet7/0/1</td>
<td>RJ-45</td>
<td>1 Gbps</td>
</tr>
<tr>
<td>7</td>
<td>eno3</td>
<td>TenGigabitEthernet3/0/0</td>
<td>SFP+</td>
<td>10 Gbps</td>
</tr>
<tr>
<td>8</td>
<td>eno4</td>
<td>TenGigabitEthernet3/0/1</td>
<td>SFP+</td>
<td>10 Gbps</td>
</tr>
</tbody>
</table>

**Note:** Default Host OS Interface is `eno1`. The Host OS Interface is one network interface that is only available to the host OS and not available in TNSR. Though technically optional, the best practice is to have one for accessing and updating the host OS.
With 4 Port Intel 10 Gbps X710 SFP+ Expansion Card

<table>
<thead>
<tr>
<th>Port</th>
<th>TNSR Label</th>
<th>Port Type</th>
<th>Port Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>TenGigabitEthernet5/0/3</td>
<td>SFP+</td>
<td>10 Gbps</td>
</tr>
<tr>
<td>2</td>
<td>TenGigabitEthernet5/0/2</td>
<td>SFP+</td>
<td>10 Gbps</td>
</tr>
<tr>
<td>3</td>
<td>TenGigabitEthernet5/0/1</td>
<td>SFP+</td>
<td>10 Gbps</td>
</tr>
<tr>
<td>4</td>
<td>TenGigabitEthernet5/0/0</td>
<td>SFP+</td>
<td>10 Gbps</td>
</tr>
<tr>
<td>5</td>
<td>GigabitEthernet7/0/0</td>
<td>RJ-45</td>
<td>1 Gbps</td>
</tr>
<tr>
<td>6</td>
<td>GigabitEthernet7/0/1</td>
<td>RJ-45</td>
<td>1 Gbps</td>
</tr>
<tr>
<td>7</td>
<td>TenGigabitEthernet3/0/0</td>
<td>SFP+</td>
<td>10 Gbps</td>
</tr>
<tr>
<td>8</td>
<td>TenGigabitEthernet3/0/1</td>
<td>SFP+</td>
<td>10 Gbps</td>
</tr>
</tbody>
</table>

**Note:** Default Host OS Interface is eno1. The Host OS Interface is one network interface that is only available to the host OS and not available in TNSR. Though technically optional, the best practice is to have one for accessing and updating the host OS.

**Connecting to the Console Port**

Connecting to the VGA console is identical to connecting any computer to a monitor. Connect the VGA cable (DB-15) between the Netgate appliance and the monitor. Use a USB or PS/2 keyboard and mouse as applicable to the hardware.

**Additional Resources**

**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
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/

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/

Warranty and Support

- One year manufacturer’s warranty.
- Please contact Netgate for warranty information or view the Product Lifecycle page.
- All Specifications subject to change without notice.

Enterprise Support is included with an active software subscription, for more information view the Netgate Global Support page.

See also:

For more information on how to use pfSense® software, see the TNSR Documentation and Resource Library.

1.1.3 Netgate 1541 Secure Router Manual

This Quick Start Guide covers the first time connection procedures for the Netgate 1541 1U Secure Router and also provides information needed to stay up and running.
Getting Started

Use the following steps to configure the TNSR Secure Router.

1. To configure the Network Interfaces and gaining access to the Internet, follow the instructions provided in the Zero-to-Ping documentation.

   **Note:** Not all steps in the Zero-to-Ping documentation will be necessary for every configuration scenario.

2. 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.

3. Finally, configure the TNSR instance to meet the specific use case. The topics are listed on the left column of the TNSR Documentation site. There are also TNSR Configuration Example Recipes that might be of assistance when configuring TNSR.

Input and Output Ports

## Contents

- Input and Output Ports
  - Default Configuration
    - Network Ports
    - Other I/O
    - Status LEDs
      - With CPIC-8955 Accelerator Card with Intel® QuickAssist Technology
      - With 2 Port Intel 1 Gbps Ethernet Expansion Card
      - With 2 Port Intel 10 Gbps X710 SFP+ Expansion Card
      - With 4 Port Intel 1 Gbps Ethernet Expansion Card
      - With 4 Port Intel 10 Gbps X710 SFP+ Expansion Card

## Default Configuration
Network Ports

<table>
<thead>
<tr>
<th>Port</th>
<th>Linux Label</th>
<th>TNSR Label</th>
<th>Port Type</th>
<th>Port Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>eno1</td>
<td>GigabitEthernet6/0/0</td>
<td>RJ-45</td>
<td>1 Gbps</td>
</tr>
<tr>
<td>2</td>
<td>eno2</td>
<td>GigabitEthernet6/0/1</td>
<td>RJ-45</td>
<td>1 Gbps</td>
</tr>
<tr>
<td>3</td>
<td>eno3</td>
<td>TenGigabitEthernet3/0/0</td>
<td>RJ-45</td>
<td>10 Gbps</td>
</tr>
<tr>
<td>4</td>
<td>eno4</td>
<td>TenGigabitEthernet3/0/1</td>
<td>RJ-45</td>
<td>10 Gbps</td>
</tr>
</tbody>
</table>

Note: Default Host OS Interface is eno1. The Host OS Interface is one network interface that is only available to the host OS and not available in TNSR. Though technically optional, the best practice is to have one for accessing and updating the host OS.

Other I/O

<table>
<thead>
<tr>
<th>Port</th>
<th>I/O Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>IPMI</td>
</tr>
<tr>
<td>6</td>
<td>2x USB 3.0</td>
</tr>
<tr>
<td>7</td>
<td>VGA</td>
</tr>
<tr>
<td>8</td>
<td>Reset &amp; Power buttons</td>
</tr>
<tr>
<td>9</td>
<td>Status LEDs (see table below)</td>
</tr>
</tbody>
</table>

Status LEDs
<table>
<thead>
<tr>
<th>LED</th>
<th>State</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>9a</td>
<td>Continuously on and red</td>
<td>An overheat condition has occurred. (This may be caused by cable congestion.)</td>
</tr>
<tr>
<td></td>
<td>Blinking red (1Hz)</td>
<td>Fan failure, check for an inoperative fan.</td>
</tr>
<tr>
<td></td>
<td>Blinking red (0.25Hz)</td>
<td>Power failure, check for a non-operational power supply.</td>
</tr>
<tr>
<td></td>
<td>Solid blue</td>
<td>Local UID has been activated. Use this function through IPMI to locate the server in a rack mount environment.</td>
</tr>
<tr>
<td></td>
<td>Blinking blue</td>
<td>Remote UID is on. Use this function through IPMI to identify the server from a remote location.</td>
</tr>
<tr>
<td>9b</td>
<td>Flashing</td>
<td>Indicates network activity on igb1 (upper left port).</td>
</tr>
<tr>
<td>9c</td>
<td>Flashing</td>
<td>Indicates network activity on igb0 (lower left port).</td>
</tr>
<tr>
<td>9d</td>
<td>Flashing</td>
<td>Indicates IDE channel activity on the hard drive.</td>
</tr>
<tr>
<td>9e</td>
<td>Illuminated</td>
<td>Indicates power is being supplied to the system power supply units. This LED should normally be illuminated when the system is operating.</td>
</tr>
<tr>
<td></td>
<td>Off</td>
<td>Indicates no power is being supplied. System is powered off.</td>
</tr>
</tbody>
</table>
With CPIC-8955 Accelerator Card with Intel® QuickAssist Technology

<table>
<thead>
<tr>
<th>Port</th>
<th>Linux Label</th>
<th>TNSR Label</th>
<th>Port Type</th>
<th>Port Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>eno1</td>
<td>GigabitEthernet6/0/0</td>
<td>RJ-45</td>
<td>1 Gbps</td>
</tr>
<tr>
<td>2</td>
<td>eno2</td>
<td>GigabitEthernet6/0/1</td>
<td>RJ-45</td>
<td>1 Gbps</td>
</tr>
<tr>
<td>3</td>
<td>eno3</td>
<td>TenGigabitEthernet3/0/0</td>
<td>RJ-45</td>
<td>10 Gbps</td>
</tr>
<tr>
<td>4</td>
<td>eno4</td>
<td>TenGigabitEthernet3/0/1</td>
<td>RJ-45</td>
<td>10 Gbps</td>
</tr>
</tbody>
</table>

**Note:** Default Host OS Interface is eno1. The Host OS Interface is one network interface that is only available to the host OS and not available in TNSR. Though technically optional, the best practice is to have one for accessing and updating the host OS.

With 2 Port Intel 1 Gbps Ethernet Expansion Card

<table>
<thead>
<tr>
<th>Port</th>
<th>Linux Label</th>
<th>TNSR Label</th>
<th>Port Type</th>
<th>Port Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ens2f0</td>
<td>GigabitEthernet5/0/0</td>
<td>RJ-45</td>
<td>1 Gbps</td>
</tr>
<tr>
<td>2</td>
<td>ens2f1</td>
<td>GigabitEthernet5/0/1</td>
<td>RJ-45</td>
<td>1 Gbps</td>
</tr>
<tr>
<td>3</td>
<td>eno1</td>
<td>GigabitEthernet7/0/0</td>
<td>RJ-45</td>
<td>1 Gbps</td>
</tr>
<tr>
<td>4</td>
<td>eno2</td>
<td>GigabitEthernet7/0/1</td>
<td>RJ-45</td>
<td>1 Gbps</td>
</tr>
<tr>
<td>5</td>
<td>eno3</td>
<td>TenGigabitEthernet3/0/0</td>
<td>RJ-45</td>
<td>10 Gbps</td>
</tr>
<tr>
<td>6</td>
<td>eno4</td>
<td>TenGigabitEthernet3/0/1</td>
<td>RJ-45</td>
<td>10 Gbps</td>
</tr>
</tbody>
</table>

**Note:** Default Host OS Interface is eno1. The Host OS Interface is one network interface that is only available to the host OS and not available in TNSR. Though technically optional, the best practice is to have one for accessing and updating the host OS.
With 2 Port Intel 10 Gbps X710 SFP+ Expansion Card

<table>
<thead>
<tr>
<th>Port</th>
<th>TNSR Label</th>
<th>Port Type</th>
<th>Port Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>TenGigabitEthernet5/0/1</td>
<td>SFP+</td>
<td>10 Gbps</td>
</tr>
<tr>
<td>2</td>
<td>TenGigabitEthernet5/0/0</td>
<td>SFP+</td>
<td>10 Gbps</td>
</tr>
<tr>
<td>3</td>
<td>GigabitEthernet7/0/0</td>
<td>RJ-45</td>
<td>1 Gbps</td>
</tr>
<tr>
<td>4</td>
<td>GigabitEthernet7/0/1</td>
<td>RJ-45</td>
<td>1 Gbps</td>
</tr>
<tr>
<td>5</td>
<td>TenGigabitEthernet3/0/0</td>
<td>RJ-45</td>
<td>10 Gbps</td>
</tr>
<tr>
<td>6</td>
<td>TenGigabitEthernet3/0/1</td>
<td>RJ-45</td>
<td>10 Gbps</td>
</tr>
</tbody>
</table>

Note: Default Host OS Interface is eno1. The Host OS Interface is one network interface that is only available to the host OS and not available in TNSR. Though technically optional, the best practice is to have one for accessing and updating the host OS.

With 4 Port Intel 1 Gbps Ethernet Expansion Card

<table>
<thead>
<tr>
<th>Port</th>
<th>Linux Label</th>
<th>TNSR Label</th>
<th>Port Type</th>
<th>Port Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ens2f0</td>
<td>GigabitEthernet5/0/0</td>
<td>RJ-45</td>
<td>1 Gbps</td>
</tr>
<tr>
<td>2</td>
<td>ens2f1</td>
<td>GigabitEthernet5/0/1</td>
<td>RJ-45</td>
<td>1 Gbps</td>
</tr>
<tr>
<td>3</td>
<td>ens2f2</td>
<td>GigabitEthernet5/0/2</td>
<td>RJ-45</td>
<td>1 Gbps</td>
</tr>
<tr>
<td>4</td>
<td>ens2f3</td>
<td>GigabitEthernet5/0/3</td>
<td>RJ-45</td>
<td>1 Gbps</td>
</tr>
<tr>
<td>5</td>
<td>eno1</td>
<td>GigabitEthernet7/0/0</td>
<td>RJ-45</td>
<td>1 Gbps</td>
</tr>
<tr>
<td>6</td>
<td>eno2</td>
<td>GigabitEthernet7/0/1</td>
<td>RJ-45</td>
<td>1 Gbps</td>
</tr>
<tr>
<td>7</td>
<td>eno3</td>
<td>TenGigabitEthernet3/0/0</td>
<td>RJ-45</td>
<td>10 Gbps</td>
</tr>
<tr>
<td>8</td>
<td>eno4</td>
<td>TenGigabitEthernet3/0/1</td>
<td>RJ-45</td>
<td>10 Gbps</td>
</tr>
</tbody>
</table>

Note: Default Host OS Interface is eno1. The Host OS Interface is one network interface that is only available to
the host OS and not available in TNSR. Though technically optional, the best practice is to have one for accessing and updating the host OS.

**With 4 Port Intel 10 Gbps X710 SFP+ Expansion Card**

<table>
<thead>
<tr>
<th>Port</th>
<th>TNSR Label</th>
<th>Port Type</th>
<th>Port Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>TenGigabitEthernet5/0/3</td>
<td>SFP+</td>
<td>10 Gbps</td>
</tr>
<tr>
<td>2</td>
<td>TenGigabitEthernet5/0/2</td>
<td>SFP+</td>
<td>10 Gbps</td>
</tr>
<tr>
<td>3</td>
<td>TenGigabitEthernet5/0/1</td>
<td>SFP+</td>
<td>10 Gbps</td>
</tr>
<tr>
<td>4</td>
<td>TenGigabitEthernet5/0/0</td>
<td>SFP+</td>
<td>10 Gbps</td>
</tr>
<tr>
<td>5</td>
<td>GigabitEthernet7/0/0</td>
<td>RJ-45</td>
<td>1 Gbps</td>
</tr>
<tr>
<td>6</td>
<td>GigabitEthernet7/0/1</td>
<td>RJ-45</td>
<td>1 Gbps</td>
</tr>
<tr>
<td>7</td>
<td>TenGigabitEthernet3/0/0</td>
<td>RJ-45</td>
<td>10 Gbps</td>
</tr>
<tr>
<td>8</td>
<td>TenGigabitEthernet3/0/1</td>
<td>RJ-45</td>
<td>10 Gbps</td>
</tr>
</tbody>
</table>

**Note:** Default Host OS Interface is `eno1`. The Host OS Interface is one network interface that is only available to the host OS and not available in TNSR. Though technically optional, the best practice is to have one for accessing and updating the host OS.

**Connecting to the Console Port**

Connecting to the VGA console is identical to connecting any computer to a monitor. Connect the VGA cable (DB-15) between the Netgate appliance and the monitor. Use a USB or PS/2 keyboard and mouse as applicable to the hardware.

**Additional Resources**

**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](https://www.netgate.com/our-services/professional-services.html)
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/

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/

Warranty and Support

- One year manufacturer’s warranty.
- Please contact Netgate for warranty information or view the Product Lifecycle page.
- All Specifications subject to change without notice.

Enterprise Support is included with an active software subscription, for more information view the Netgate Global Support page.

See also:

For more information on how to use pfSense® software, see the TNSR Documentation and Resource Library.

1.2 Cloud Platforms

TNSR Secure Networking software is available in the following Cloud Platforms.

- Amazon Web Services - Instances with Enhanced Network Adapters (ENA)
- Microsoft Azure - Instances with Accelerated Networking

1.2.1 AWS TNSR Instance Setup

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

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

The steps involved are:
Learn the Basics

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

There are many different network designs possible in AWS. This guide assumes a TNSR instance will sit in a VPC 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 Elastic Network Adapter 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.

**TNSR WAN/Internet Interface** The TNSR WAN interface is used by TNSR to connect to the Internet. A WAN interface will have an Elastic 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 VPC. The instances in the private subnet do not have their own Elastic 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>
<thead>
<tr>
<th>Item</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>VPC Address Space</td>
<td>10.5.0.0/16</td>
</tr>
<tr>
<td>WAN Subnet</td>
<td>10.5.0.0/24</td>
</tr>
<tr>
<td>LAN Subnet</td>
<td>10.5.1.0/24</td>
</tr>
<tr>
<td>Management Subnet</td>
<td>10.5.2.0/24</td>
</tr>
</tbody>
</table>

In a real production VPC, 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.

Launch an Instance

Now launch an instance of TNSR:

1. Navigate to [https://console.aws.amazon.com/ec2/](https://console.aws.amazon.com/ec2/)
2. Click **Instances**
3. Click **Launch Instance** to enter the **Launch Instance Wizard**
4. Click the **AWS Marketplace** heading
5. Type Netgate in the search box and press Enter
6. Find the entry for TNSR and click **Select**
7. Click **Continue** on the information page
8. Choose an **Instance Type**, then click **Next**

**Note:** The available instance types are those that support ENA network adapters. These include all C5 and M5 instance types. The type of C5 or M5 instance depends on the needs of a given network. For networks with a large number of subnets in the VPC or for networks that expect high throughput, one of the larger instance types is likely to be more appropriate.

For information on bandwidth limits and limits on the number of Network Interfaces and IP addresses for different instance types, see the following links:

- [https://aws.amazon.com/ec2-instance-types/](https://aws.amazon.com/ec2-instance-types/)

In environments where the requirements are unclear, start with **c5.xlarge** and migrate to a different instance type later as necessary.

9. Configure Instance Details:
   - Select the VPC in which to launch the instance
   - Under **Network Interfaces**:
     - Select the **Management subnet** as the subnet for the **eth0** interface
     - Click the **Add Device** button
     - Select the **WAN subnet** as the subnet for the **eth1** interface

Only two Network Interfaces may be added to an instance at launch time. The LAN interface can be added after the instance is launched.

Click **Next** after completing the choices

10. Add storage if this instance will require more than the default 10GB disk, then click **Next**

11. Add **Tags** to identify this instance if desired, then click **Next**

12. Configure Security Group

   Default rules should appear to allow SSH and ICMP. These rules can be used to create a new security group, and to add access for other ports if needed.

   **Warning:** The best practice is to limit the allowed **Source** to a specific address or network that will be used connect to the TNSR instance.

   Give the security group a name such as “TNSR management”.

13. Verify the settings selected in earlier steps, then click **Launch**

   Select an ssh key or create a new key in the popup. Click **Launch Instances**
Add TNSR LAN Interface to the Instance

The Management and WAN Interfaces were created while launching the instance. Now create the LAN interface. If this instance requires additional interfaces, either public or private, create those now.


The subnet connected to the TNSR LAN interface is a private network which is using the TNSR instance as its Internet gateway. It can have a much less restrictive **Security Group** set so that traffic from the LAN can reach the TNSR instance. Select the default **Security Group** for the VPC, which should allow all inbound traffic.

**Note:** The Description field is optional when creating a **Network Interface** but the best practice is to enter **Description** text that identifies the interface so it can be easily identified when it is attached to an instance. To attach the LAN Network Interface to the instance, follow the instructions at [https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-eni.html#attach_eni_running_stopped](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-eni.html#attach_eni_running_stopped)

**Prepare TNSR Network Interfaces**

The TNSR WAN and LAN interfaces should have **Source/Destination Check** disabled in order to allow the TNSR instance to route packets. If these settings are not disabled, packets from the LAN subnet to the Internet will be dropped before reaching the TNSR LAN interface.


**Connect Management and WAN Interfaces to the Internet**

The Management Interface and the TNSR WAN interface must be assigned public Elastic IP Addresses by AWS.


Make a note of the allocated Elastic IP Address.

Before associating an Elastic IP Address to a **Network Interface**, make a note of the ID of the **Network Interface** to use. To find the **Network Interface ID**:

1. Navigate to [https://console.aws.amazon.com/ec2/](https://console.aws.amazon.com/ec2/)
2. Click **Instances**
3. Click the button next to the TNSR interface to select it
4. Look at the bottom of the page, under the **Description** tab to see **Network Interfaces**
5. Click on the interface names to display information about the **Network Interface**:  
   - eth0 for management interface
   - eth1 for WAN interface
6. Write down the **Interface ID** for each interface

After allocating the Elastic IP Addresses and finding the Network Interface IDs for eth0 and eth1, associate the Elastic IP Addresses to the Network Interfaces by following the instructions at [https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-eni.html#associate_eip](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-eni.html#associate_eip)
Connect to the instance

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

The default account is named tnsr.

To connect from a shell prompt in a Unix/Linux terminal:

```
$ ssh -i <my_key_file> tnsr@<eth0_public_ip_addr>
```

Substitute the actual key file name instead of typing `<my_key_file>` and the management interface Elastic IP
Address instead of typing `<eth0_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/LDXVPpD2v6hnWdFHW2hkCSpMcaH4tB9uDLAa40.
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: v0.1.0-567-g0967ac3
Build timestamp: Fri Apr 20 16:16:48 2018 CDT
Git Commit: 0x967ac3d
ip-10-5-2-225.ec2.internal tnsr#
```

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.

Since the TNSR LAN interface was added to the instance after launching the instance, it will not be visible yet to the
TNSR data plane unless the instance has been rebooted. Check which interfaces are visible to TNSR by typing `show
interface` at the CLI prompt.

Here’s an example of what will appear:

```
tnsr# show interface
Interface: VirtualFunctionEthernet0/6/0
  Admin status: down
  Link down, 100 Gbit/sec, full duplex
  Link MTU: 9216 bytes
  MAC address: 0a:54:d0:7c:df:c0
  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, 2 rx miss, 0 rx no buffer
```

Only one interface is visible on this instance: the WAN interface which was attached at the time the instance launched.

If all of the TNSR instances, other than the Management Interface, are not displayed by `show interface`, restart
the data plane and the missing interfaces will appear:
Product Manual, TNSR 19.02

```
tnsr# configure
tnsr(config)# service dataplane restart
Success
tnsr(config)# exit
```

Check the interfaces again:

```
tnsr# show interface
Interface: VirtualFunctionEthernet0/6/0
    Admin status: down
    Link down, 100 Gbit/sec, full duplex
    Link MTU: 9216 bytes
    MAC address: 0a:54:d0:7c:df:c0
    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: VirtualFunctionEthernet0/7/0
    Admin status: down
    Link down, 100 Gbit/sec, full duplex
    Link MTU: 9216 bytes
    MAC address: 0a:0a:7b:cd:89:6e
    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
```

After the restart a second interface is visible: the TNSR LAN interface.

When all of the interfaces that are attached are present, the instance is ready to start enabling and configuring IP addresses on interfaces.

During the process of creating Network Interfaces, a private IP address was assigned to each interface. The next step is to configure those addresses on the interfaces in TNSR in order to communicate with other instances in the VPC.

Configure WAN interface:

1. Navigate to https://console.aws.amazon.com/ec2/
2. Click Instances
3. Click the button next to the TNSR interface to select it
4. Look at the bottom of the page, under the Description tab to see Network Interfaces
5. Click on eth1
6. Find the field named “Private IP address” in the popup
7. Configure the interface in the CLI:

```
tnsr# configure
tnsr(config)# interface VirtualFunctionEthernet0/6/0
tnsr(config-interface)# ip address 10.5.0.222/24
tnsr(config-interface)# enable
```

(continues on next page)
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. Navigate to https://console.aws.amazon.com/ec2/
2. Click Instances
3. Click the button next to the TNSR interface to select it
4. Look at the bottom of the page, under the Description tab to see Network Interfaces
5. Click on eth2
6. Find the field named “Private IP address” in the popup
7. Configure the interface in the CLI:

```
    tnsr(config)# interface VirtualFunctionEthernet0/7/0
    tnsr(config-interface)# ip address 10.5.1.218/24
    tnsr(config-interface)# enable
    tnsr(config-interface)# description eth2 eni-6fa572f0 LAN
    tnsr(config-interface)# exit
    tnsr(config)# exit
```

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

Check interface status again:

```
    tnsr# show interface
    Interface: VirtualFunctionEthernet0/6/0
        Description: eth1 eni-beaa7c21 WAN
        Admin status: up
        Link up, unknown, unknown duplex
        Link MTU: 9216 bytes
        MAC address: 0a:54:d0:7c:df:c0
        IPv4 Route Table: ipv4-VRF:0
        IPv4 addresses:
            10.5.0.222/24
        IPv6 Route Table: ipv6-VRF:0
        counters:
            received: 798 bytes, 19 packets, 0 errors
            transmitted: 1604 bytes, 28 packets, 0 errors
            0 drops, 0 punts, 5 rx miss, 0 rx no buffer

    Interface: VirtualFunctionEthernet0/7/0
        Description: eth2 eni-6fa572f0 LAN
        Admin status: up
        Link up, unknown, unknown duplex
        Link MTU: 9216 bytes
        MAC address: 0a:0a:7b:cd:89:6e
        IPv4 Route Table: ipv4-VRF:0
        IPv4 addresses:
            10.5.1.218/24
```

(continues on next page)
IPv6 Route Table: ipv6-VRF:0

counters:
received: 1925 bytes, 30 packets, 0 errors
transmitted: 1226 bytes, 19 packets, 0 errors
20 drops, 0 punts, 27 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 to the VPC gateway on each subnet with the ping command. The VPC gateway address is the base address of a subnet + 1. e.g.:

- VPC gateway IP address for 10.5.0.0/24:
  Base address 10.5.0.0 + 1 = 10.5.0.1

- VPC gateway IP address for 10.5.1.0/24: 10.5.1.1
  Base address 10.5.1.0 + 1 = 10.5.1.1

```plaintext
tnsr# ping 10.5.0.1 source 10.5.0.222 count 3
PING 10.5.0.1 (10.5.0.1) 56(84) bytes of data.
64 bytes from 10.5.0.1: icmp_seq=1 ttl=64 time=0.096 ms
64 bytes from 10.5.0.1: icmp_seq=2 ttl=64 time=0.231 ms
64 bytes from 10.5.0.1: icmp_seq=3 ttl=64 time=0.220 ms

--- 10.5.0.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1999ms
rtt min/avg/max/mdev = 0.096/0.182/0.231/0.062 ms

tnsr# ping 10.5.1.1 source 10.5.1.218 count 3
PING 10.5.1.1 (10.5.1.1) 56(84) bytes of data.
64 bytes from 10.5.1.1: icmp_seq=1 ttl=64 time=0.071 ms
64 bytes from 10.5.1.1: icmp_seq=2 ttl=64 time=0.123 ms
64 bytes from 10.5.1.1: icmp_seq=3 ttl=64 time=0.157 ms

--- 10.5.1.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1999ms
rtt min/avg/max/mdev = 0.071/0.117/0.157/0.035 ms```

**Interface Naming**

The names that are displayed for Network Interfaces on an instance in the EC2 Web Console are different than the names that appear in the TNSR CLI. The interfaces in TNSR are named using the PCI bus/slot/function of the device. The names in the EC2 Web Console use the traditional Linux naming scheme and display as eth0, eth1, and so on.

In this example, it is straightforward to determine which interface in TNSR corresponds to an AWS Network Interface in the EC2 Web Console because there are only 2 interfaces and one of them will be present at boot time.

If this instance has more Network Interfaces than in the example, or if it is unclear which interface in the TNSR CLI matches up with which Network Interface in the EC2 Web Console, the two can be correlated by checking the MAC addresses. The TNSR CLI command show interface will display all of the interfaces present and the output includes the MAC address of each. The MAC address of each TNSR interface can be checked in the EC2 Web Console to find the right Network Interface.

To find the MAC address of a Network Interface in the EC2 Web Console:

1. Navigate to https://console.aws.amazon.com/ec2/
2. Click **Instances**
3. Click the button next to the TNSR interface to select it
4. Look at the bottom of the page, under the **Description** tab to see **Network Interfaces**
5. The eth0 interface is the management interface and won’t appear in the TNSR CLI. Look at eth1, eth2, etc.
6. Click on the interface name to display information about the **Network Interface**
7. Click on the **Interface ID** to open the **Network Interfaces** page
   Only the **Network Interface** matching the selected ID will be displayed.
8. Look at the bottom of the page, under the **Details** tab, to find the “MAC address” field.
9. Match this MAC address to one of the MAC addresses printed from the `show interface` output in the CLI

**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:

```
tnsr# configure
tnsr(config)# route ipv4 table ipv4-VRF:0
tnsr(config-route-table-v4)# route 0.0.0.0/0
tnsr(config-route-table-v4)# next-hop 1 via 10.5.0.1 VirtualFunctionEthernet0/6/0
tnsr(config-route-table-v4)# exit
tnsr(config)# exit
tnsr#
```

**Ping TNSR WAN Interface from local network**

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

To find the Elastic IP address associated to the TNSR WAN Interface, use the EC2 Web Console:

1. Navigate to [https://console.aws.amazon.com/ec2/](https://console.aws.amazon.com/ec2/)
2. Click **Instances**
3. Click the button next to the TNSR interface to select it
4. Look at the bottom of the page, under the **Description** tab to see **Network Interfaces**
5. Click on **eth1**
6. Find the **Elastic IP Address** field in the popup

Now, try to ping the **Elastic IP Address** of the TNSR WAN Interface:

```
bash-3.2$ ping -c 5 52.7.26.219
PING 52.7.26.219 (52.7.26.219): 56 data bytes
64 bytes from 52.7.26.219: icmp_seq=0 ttl=45 time=48.781 ms
64 bytes from 52.7.26.219: icmp_seq=1 ttl=45 time=49.232 ms
64 bytes from 52.7.26.219: icmp_seq=2 ttl=45 time=49.238 ms
64 bytes from 52.7.26.219: icmp_seq=3 ttl=45 time=48.632 ms
```

(continues on next page)
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

Regional Market Availability

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

<table>
<thead>
<tr>
<th>Market</th>
<th>Availability</th>
</tr>
</thead>
<tbody>
<tr>
<td>us-east-1 N. Virginia</td>
<td>Available</td>
</tr>
<tr>
<td>us-east-2 Ohio</td>
<td>Available</td>
</tr>
<tr>
<td>us-gov-east-1 GovCloud East</td>
<td>Available</td>
</tr>
<tr>
<td>us-gov-west-1 GovCloud West</td>
<td>Available</td>
</tr>
<tr>
<td>us-west-1 N. California</td>
<td>Available</td>
</tr>
<tr>
<td>us-west-2 Oregon</td>
<td>Available</td>
</tr>
<tr>
<td>af-south-1 Cape Town</td>
<td>Available</td>
</tr>
<tr>
<td>ap-east-1 Hong Kong</td>
<td>Available</td>
</tr>
<tr>
<td>ap-northeast-1 Tokyo</td>
<td>Available</td>
</tr>
<tr>
<td>ap-northeast-2 Seoul</td>
<td>Available</td>
</tr>
<tr>
<td>ap-south-1 Mumbai</td>
<td>Available</td>
</tr>
<tr>
<td>ap-southeast-1 Singapore</td>
<td>Available</td>
</tr>
<tr>
<td>ap-southeast-2 Sydney</td>
<td>Available</td>
</tr>
<tr>
<td>ca-central-1 Quebec</td>
<td>Available</td>
</tr>
<tr>
<td>eu-central-1 Frankfurt</td>
<td>Available</td>
</tr>
<tr>
<td>eu-north-1 Stockholm</td>
<td>Available</td>
</tr>
<tr>
<td>eu-south-1 Milan</td>
<td>Available</td>
</tr>
<tr>
<td>eu-west-1 Ireland</td>
<td>Available</td>
</tr>
<tr>
<td>eu-west-2 London</td>
<td>Available</td>
</tr>
<tr>
<td>eu-west-3 Paris</td>
<td>Available</td>
</tr>
<tr>
<td>sa-east-1 São Paulo</td>
<td>Available</td>
</tr>
</tbody>
</table>
Additional Resources

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

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/

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/

1.2.2 Azure TNSR Instance Setup

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:

Learn the Basics

TNSR utilizes an optimized userspace data plane to forward packets at very high rates. On Azure, TNSR runs on a customized 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>
<thead>
<tr>
<th>Item</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Virtual Network Address Space</td>
<td>10.5.0.0/16</td>
</tr>
<tr>
<td>WAN Subnet</td>
<td>10.5.0.0/24</td>
</tr>
<tr>
<td>LAN Subnet</td>
<td>10.5.1.0/24</td>
</tr>
<tr>
<td>Management Subnet</td>
<td>10.5.2.0/24</td>
</tr>
</tbody>
</table>

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**.

**Launch an Instance**

Now launch an instance of TNSR:


2. Login to your Azure account running:

   ```bash
   $ az login
   ```

3. Configure the default location.

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

4. Create a resource group to be used to store all TNSR related objects if it does not already exist.

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

5. Create Virtual Network and Subnets.

   ```bash
   $ 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 \
   ```

   (continues on next page)
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
```


```
$ 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
```
10. Create the LAN Network Interface.

```bash
$ 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.

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

```bash
$ az vm image list \
  --publisher Netgate \
  --all \
  --query '[?contains(offer,"tnsr").{Sku:s 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.

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

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

```bash
$ 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.

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

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/LDXVPpD2v6hnWdFHW2zkCbSpMcaH4tBgTuDLAa40. 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#
```

1.2. Cloud Platforms
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: NetVSC2
  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: NetVSC3
  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, the VM has NetVSC2 as WAN and NetVSC3 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 \
   -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 NetVSC2
   TNSR-Instance1 tnsr(config)# ip address 10.5.0.4/24
   TNSR-Instance1 tnsr(config)# enable
   TNSR-Instance1 tnsr(config)# description TNSR-Instance1 WAN
   TNSR-Instance1 tnsr(config)# 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 NetVSC3
   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: NetVSC2
  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

Interface: NetVSC3
  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
```

**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-route-table-v4)# next-hop 1 via 10.5.0.1 NetVSC2
TNSR-Instance1 tnsr(config-route-table-v4)# exit
TNSR-Instance1 tnsr(config)# exit
TNSR-Instance1 tnsr#
```

**Ping TNSR WAN Interface from local 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
```

(continues on next page)
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

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 5: Microsoft Azure Available Regions

<table>
<thead>
<tr>
<th>Market</th>
<th>pfSense</th>
</tr>
</thead>
<tbody>
<tr>
<td>Armenia</td>
<td>Available</td>
</tr>
<tr>
<td>Australia</td>
<td>*</td>
</tr>
<tr>
<td>Austria</td>
<td>Available</td>
</tr>
<tr>
<td>Belarus</td>
<td>Available</td>
</tr>
<tr>
<td>Belgium</td>
<td>Available</td>
</tr>
<tr>
<td>Brazil</td>
<td>Available</td>
</tr>
<tr>
<td>Canada</td>
<td>Available</td>
</tr>
<tr>
<td>Croatia</td>
<td>Available</td>
</tr>
<tr>
<td>Cyprus</td>
<td>Available</td>
</tr>
<tr>
<td>Czechia</td>
<td>Available</td>
</tr>
<tr>
<td>Denmark</td>
<td>Available</td>
</tr>
<tr>
<td>Estonia</td>
<td>Available</td>
</tr>
<tr>
<td>Finland</td>
<td>Available</td>
</tr>
<tr>
<td>France</td>
<td>Available</td>
</tr>
<tr>
<td>Germany</td>
<td>Available</td>
</tr>
<tr>
<td>Greece</td>
<td>Available</td>
</tr>
<tr>
<td>Hungary</td>
<td>Available</td>
</tr>
<tr>
<td>India</td>
<td>Available</td>
</tr>
<tr>
<td>Ireland</td>
<td>Available</td>
</tr>
<tr>
<td>Italy</td>
<td>Available</td>
</tr>
<tr>
<td>Korea</td>
<td>Available</td>
</tr>
<tr>
<td>Latvia</td>
<td>Available</td>
</tr>
</tbody>
</table>

continues on next page
<table>
<thead>
<tr>
<th>Market</th>
<th>pfSense</th>
</tr>
</thead>
<tbody>
<tr>
<td>Liechtenstein</td>
<td>Available</td>
</tr>
<tr>
<td>Lithuania</td>
<td>Available</td>
</tr>
<tr>
<td>Luxembourg</td>
<td>Available</td>
</tr>
<tr>
<td>Malta</td>
<td>Available</td>
</tr>
<tr>
<td>Monaco</td>
<td>Available</td>
</tr>
<tr>
<td>Netherlands</td>
<td>Available</td>
</tr>
<tr>
<td>New Zealand</td>
<td>Available</td>
</tr>
<tr>
<td>Norway</td>
<td>Available</td>
</tr>
<tr>
<td>Poland</td>
<td>Available</td>
</tr>
<tr>
<td>Portugal</td>
<td>Available</td>
</tr>
<tr>
<td>Puerto Rico</td>
<td>Available</td>
</tr>
<tr>
<td>Romania</td>
<td>Available</td>
</tr>
<tr>
<td>Russia</td>
<td>Available</td>
</tr>
<tr>
<td>Saudi Arabia</td>
<td>Available</td>
</tr>
<tr>
<td>Serbia</td>
<td>Available</td>
</tr>
<tr>
<td>Slovakia</td>
<td>Available</td>
</tr>
<tr>
<td>Slovenia</td>
<td>Available</td>
</tr>
<tr>
<td>South Africa</td>
<td>Available</td>
</tr>
<tr>
<td>Spain</td>
<td>Available</td>
</tr>
<tr>
<td>Sweden</td>
<td>Available</td>
</tr>
<tr>
<td>Switzerland</td>
<td>Available</td>
</tr>
<tr>
<td>Taiwan</td>
<td>Available</td>
</tr>
<tr>
<td>Turkey</td>
<td>Available</td>
</tr>
<tr>
<td>United Arab Emirates</td>
<td>Available</td>
</tr>
<tr>
<td>United Kingdom</td>
<td>Available</td>
</tr>
<tr>
<td>United States</td>
<td>Available</td>
</tr>
</tbody>
</table>

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

### Additional Resources

### 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](https://www.netgate.com/our-services/professional-services.html)

### 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/](https://www.netgate.com/training/)
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/

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.

1.3 Virtual Machines

A TNSR software Bare Metal Image is available and will run in most virtual x86 environments.

- **VMware** - vSphere/ESXi 6.0 and later, running VMXNET3 virtual network adapters.

  **Note:** VMware VMXNET3 interfaces are not currently compatible with VRRP. The VMXNET3 interface driver does not support setting virtual MAC addresses, which is required by VRRP.

  **Warning:** Secure boot is not compatible with TNSR, it prevents interfaces from being added to the dataplane when it is enabled. If a VMware virtual machine has **Firmware** set to **EFI** under **VM Options > Boot Options**, then **Enable UEFI secure boot** must be **unchecked**.

- **KVM** - Kernel Virtual Machine on Linux.

- **virtio-net** - Paravirtualized network adapter used by other hypervisors such as QEMU and VirtualBox.

1.3.1 Using TNSR on KVM

TNSR can be run on a Linux Kernel Virtual Machine (KVM) hypervisor host. The advice on this page is specifically geared toward KVM managed by **libvirt**.
Installing TNSR on KVM

When creating the virtual machine, use the requirements on Supported Platforms as a guide for determining configuration parameters before starting. For example:

- Number of CPUs, Cores, and their topology
- Amount of RAM
- Storage size
- Network connectivity type, number of interfaces, networks to which interfaces connect

Creating a VM

Before starting, obtain the installation ISO and place it in a known location.

The following command will create a new virtual machine from the KVM CLI with the following configuration:

- 2 virtual CPUs (1 socket, 2 cores per CPU, 1 thread per core)
  - Set CPU to `host` or `qemu64,+ssse3,+sse4.1,+sse4.2,+x2apic`
- 4GB RAM
- A new 32GB virtio disk named `tnsr.qcow2`
- 3 virtio-based network interfaces

```
# virt-install --name TNSR --vcpus=2,sockets=1,cores=2,threads=1 \
--os-type linux --os-variant ubuntu20.04 --cpu host --ram 4096 \
--disk /var/lib/libvirt/images/tnsr.qcow2,size=32,device=disk, \
--bus=virtio \
--network=default,model=virtio --network bridge=br0,model=virtio \
--network bridge=br1,model=virtio \
--nographics --noautoconsole \
--location /root/TNSR-Ubuntu.iso,kernel=casper/vmlinuz,initrd=casper/
--initrd \
--extra-args 'console=ttyS0,115200n8 quiet fsck.mode=skip \ 
   network-config=disabled autodinstall ds=nccloud;=s=/cdrom/server/'
```

Note: Replace the parameters as needed to conform to the local KVM environment. In particular, the `--disk` path, ISO `--location` path, and bridge device or network names will likely be different.

Access the management console with the following command:

```
# virsh console TNSR
```

From the console, follow the standard TNSR installation procedure and the VM will shut down afterward. Start it again and reconnect to the console:

```
# virsh start TNSR
# virsh console TNSR
```

KVM Frontends/GUIs can also accomplish the same goal in different ways. Use whichever method is preferred by the hypervisor administrator.
**KVM Optimization**

Virtio interfaces use tap as a backend, which requires a `memcpy()` of each packet forwarded. Due to this design, the stock configuration can result in poor performance. The tuning suggestions in this section will help obtain higher performance in these environments.

**Note:** Though these suggested changes have been found to improve performance in testing, every installation and workload is different. Real-world results may vary depending on the environment. Generally speaking, values should only be changed from the defaults in cases where performance is lower than expected.

- Set the `vhost` backend driver `rx_queue_size` and `tx_queue_size` values to 1024 instead of the default 256.
  
  To set these values in the `libvirt` xml configuration for a VM, see *Changing VM Parameters*.

- Increase the number of queues in the `vhost` backend driver configuration, especially if TNSR is configured to use worker threads. This information is also in the section linked above.

- Try using SR-IOV VFs instead of Virtio interfaces.

- Try using a DPDK accelerated OpenVSwitch (OVS-DPDK) instead of a standard linux bridge.

**Changing VM Parameters**

Some values must be changed by editing the VM settings XML directly. This includes the receive and transmit ring queue sizes and the number of queues.

When setting the receive and transmit ring queue sizes, keep in mind that some environments impose specific requirements on the values. For example, they may only work with certain drivers, or may have value restrictions such as being a power of 2 (256, 512, 1024, etc.).

To edit the VM XML parameters, use the following command:

```
# virsh edit TNSR
[...]
```

Find the `interface` tag(s) and the `driver` tags inside. In the `driver` tag, edit or add the desired attributes and values. For example, to set 5 queues, and 1024 size transmit and receive ring queue sizes:

```
<interface [...]>  
[...]  
<driver name='vhost' txmode='iothread' ioeventfd='on' event_idx='off'
   queues='5' rx_queue_size='1024' tx_queue_size='1024'>  
[...]  
</driver>  
[...]  
</interface>
```

**Note:** Details of the above XML block have been omitted for brevity and generality. Interfaces will vary in their specific settings.

Start the VM, and check the `qemu` command line, which should contain `rx_queue_size=1024`, `tx_queue_size=1024`.

From within the VM, at a shell prompt, confirm the ring queue sizes.
ethtool -g eth0
Ring parameters for eth0:
Pre-set maximums:
RX: 1024
TX: 1024
Current hardware settings:
RX: 1024
TX: 1024

If the number of queues was changed, confirm that as well:

ethtool -l eth0
Channel parameters for eth0:
Pre-set maximums:
Combined: 5

See also:
For more details, see:

- The libvirt-users mailing list, including this post describing the process.
- The libvirt XML format documentation

1.4 Recommended Components

TNSR runs on Linux for x86 processors. For best results in data center environments, the following CPUs are recommended:

- Intel Xeon Processor E3 Families (v2, v3, v4, v5, v6, and v7)
- Intel Xeon Processor E5 Families (v2, v3, v4, v5, v6, and v7)
- Intel Xeon Processor E7 Families (v2, v3, v4, v5, v6, and v7)
- Intel Xeon Scalable Processors
- Intel Xeon D Processors
- Intel Atom C2000 & C3000 Processors

Note: TNSR can utilize accelerator technology, such as Intel QuickAssist Technology (QAT) featured in the Intel Atom C2000 and C3000 processors, and the Netgate CPIC-8955 Cryptographic Accelerator Card.

Network Interface Card (NIC) support in TNSR is provided by drivers in the underlying Linux operating system. The following NIC drivers have been tested by Netgate:

**igb** NICs based on the following Intel Gigabit Ethernet controllers:
- 82575/6, 82580, I350, I354, I210/211

**ixgbe** NICs based on the following Intel 10 Gigabit Ethernet controllers:
- 82598/9, X520, X540, X550, X553, X552/X557-A
i40e NICs based on the following Intel 10/40 Gigabit Ethernet controllers:
  • X710, XL710, XXV710

mlx5 NICs based on the following Mellanox 10/25/40/50/100 Gigabit Ethernet controllers:
  • ConnectX-4, ConnectX-4 Lx, ConnectX-5
The best practice is to use a platform tested by Netgate. That said, the underlying components may support additional hardware that has not been tested by Netgate nor confirmed to run TNSR software successfully.

Netgate customers are often on the cutting edge of technology, so they may encounter systems and components that work with TNSR before Netgate has an opportunity to test them.

**Warning:** Netgate cannot offer any assurance that untested platforms will be compatible with TNSR software or perform at acceptable levels.

Hardware support is primarily determined by The Data Plane Development Kit (DPDK) and Vector Packet Processing (VPP) open source projects which are core parts of TNSR. Additionally, the underlying operating system drivers must support the hardware (i.e. boot, load the OS, etc.) and network interfaces used for host management. TNSR may run on a system if these components support the hardware in question, but experiences will vary.

For the latest information on hardware supported by DPDK and VPP, check here:

- [https://core.dpdk.org/supported/](https://core.dpdk.org/supported/)
- [http://doc.dpdk.org/guides/nics/overview.html#id1](http://doc.dpdk.org/guides/nics/overview.html#id1)