Secure Router Manual

Amazon AWS

Feb 06, 2026

CONTENTS

The Netgate® TNSR Router for Amazon AWS is a powerful routing and VPN appliance. TNSR leverages vector packet processing and acceleration techniques for extremely high speed routing and VPN performance. A TNSR instance running in AWS can securely connect between AWS and remote offices, data centers, or even make high speed links between AWS regions.

TNSR for AWS is available in the AWS Marketplace.

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

CHAPTER

ONE

GETTING STARTED

1.1 Prerequisites and Requirements

Using a Netgate® appliance instance to protect VPC subnets requires the following:

- Setup can take 30 minutes to two hours, depending on the user's familiarity with the tools.
- An AWS Account.
- Familiarity with AWS networking.
- A VPC.
- One internet-facing subnet, to which the Netgate appliance instance will have its internet-facing WAN interface connected.
- Two or more private subnets, to which the Netgate appliance instance will have its host management interface, client-facing LAN interface, and possibly additional optional interfaces connected.
- Separate routing tables for the internet-facing subnet and the private subnets.
- Separate security groups for the internet-facing subnet and the private subnets.
- An elastic IP address or public IP address for the WAN interface of the appliance.

For the purposes of this guide, the VPC will contain three subnets (one public and two private) as well as an Internet Gateway. The end result should look like the following diagram:

If all of these are already in place with an existing VPC, feel free to skip ahead to Launching an Instance.

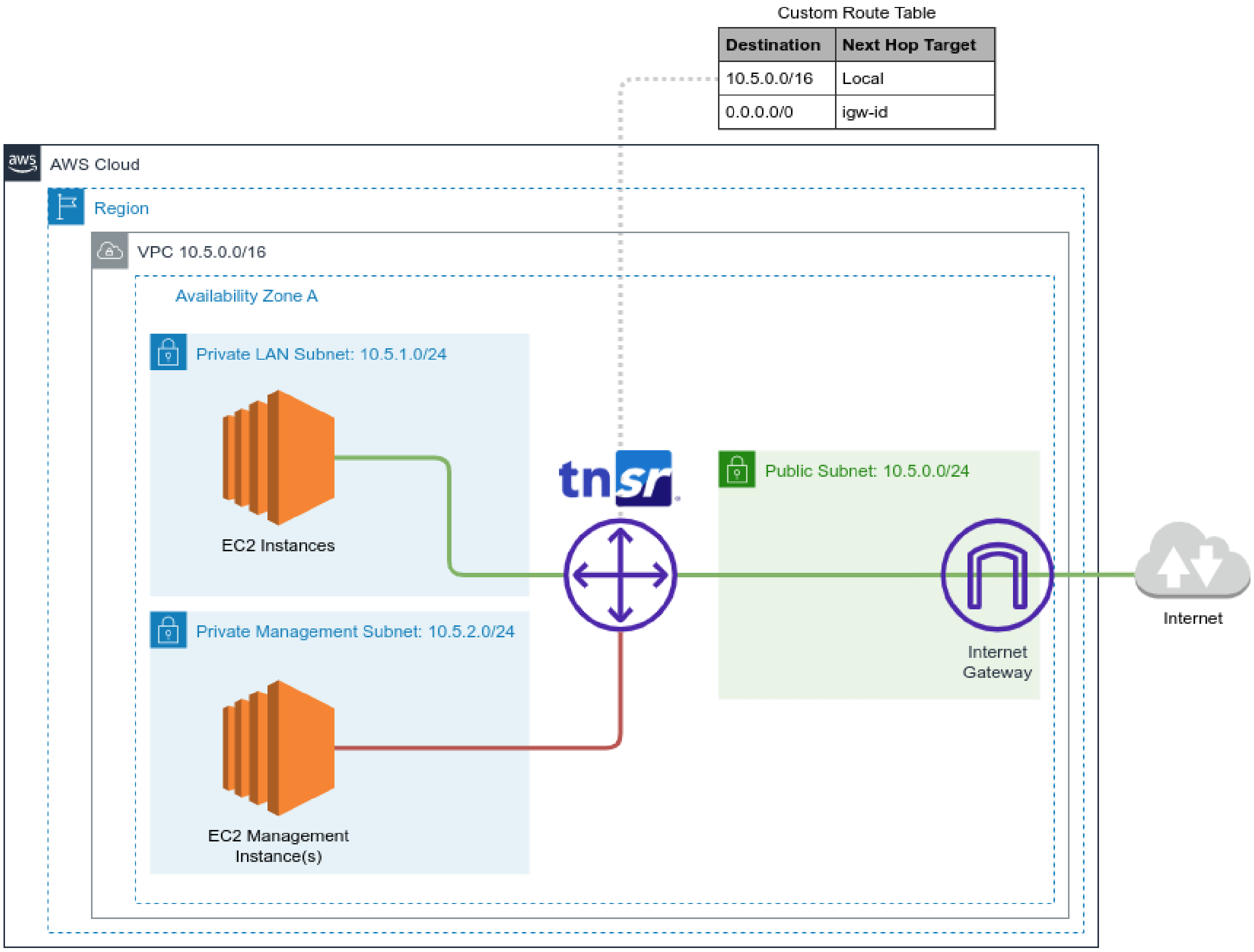
1.2 Choosing Instance Type and Sizing

There are a range of specifications to choose from, and this document guides through those choices.

1.2.1 Supported EC2 Instance Types

An instance type will depend on the expected network throughput as well as the types of services the Netgate® appliance will provide.

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 many subnets in the VPC or for networks that expect high throughput, one of the larger instance types is likely to be more appropriate.



Custom Route Table

Destination	Next Hop Target
10.5.0.0/16	Local
0.0.0.0/0	igw-id

AWS Cloud
Region
VPC 10.5.0.0/16
Availability Zone A
Private LAN Subnet: 10.5.1.0/24
EC2 Instances
tnsr
Public Subnet: 10.5.0.0/24
Internet Gateway
Internet
Private Management Subnet: 10.5.2.0/24
EC2 Management Instance(s)

Fig. 1: Architecture Diagram

Warning: Avoid using TNSR with spot instances as they are not persistent and can lead to instability.

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://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-eni.html#AvailableIpPerENI

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

1.2.2 Sizing the Storage

For general purpose routers, storage requirements will typically be small and the default **10GB** volume should be sufficient.

In situations where the appliance may be customized with additional local software, or other advanced features outside TNSR, consider increasing the volume size to something more appropriate, for example **64GB**.

1.3 AWS Service Limits

New services provisioned in a VPC may be assigned IP addresses or other resources, but Amazon puts limits on VPC resources per Region. Before provisioning a new resource, make sure to check these limits.

The following tables list the limits for Amazon VPC resources per Region. Unless indicated otherwise, requests can be made to increase these limits using the Amazon VPC limits form. For some of these limits, the current limit applied can be viewed using the **Limits** page of the Amazon EC2 console.

Note: If a limit increase is requested that applies per resource, AWS increases the limit for all resources in the Region. For example, the limit for security groups per VPC applies to all VPCs in the Region.

1.3.1 VPC and Subnets

Resource	Default limit	Comments
VPCs per Region	5	The limit for Internet gateways per Region is directly correlated to this one. Increasing this limit increases the limit on internet gateways per Region by the same amount.
Subnets per VPC	200	–
IPv4 CIDR blocks per VPC	5	This limit is made up of the primary CIDR block plus 4 secondary CIDR blocks.
IPv6 CIDR blocks per VPC	1	This limit cannot be increased.

1.3.2 DNS

For more information, see DNS Limits.

1.3.3 Elastic IP Addresses (IPv4)

Resource	Default limit	Comments
Elastic IP addresses per Region	5	This is the limit for the number of Elastic IP addresses for use in EC2-VPC. For Elastic IP addresses for use in EC2-Classic, see Amazon EC2 Limits in the Amazon Web Services General Reference.

1.3.4 Flow Logs

Resource	Default limit	Comments
Flow logs per single network interface, single subnet, or single VPC in a Region	2	This limit cannot be increased. There can effectively be 6 flow logs per network interface by creating 2 flow logs for the subnet, and 2 flow logs for the VPC in which the network interface resides.

1.3.5 Gateways

Resource	Default limit	Comments
Customer gateways per Region	50	–
Egress-only internet gateways per Region	5	This limit is directly correlated with the limit on VPCs per Region. To increase this limit, increase the limit on VPCs per Region. Only one egress-only internet gateway can attach to a VPC at a time.
Internet gateways per Region	5	This limit is directly correlated with the limit on VPCs per Region. To increase this limit, increase the limit on VPCs per Region. Only one internet gateway can be attached to a VPC at a time.
NAT gateways per Availability Zone	5	A NAT gateway in the pending, active, or deleting state counts against the limit.

1.4 Creating an IAM User in an AWS Account

A TNSR® AMI uses AWS Identity and Access Management (IAM) accounts for administration. Every AWS account includes at least one user. For security reasons, the root account should not be used for day-to-day administration. This section describes the process of creating and using an IAM user account for administering the TNSR® AMI.

See also:

To find out more about AWS security and credentials read Understanding and Getting Your Security Credentials.

There are multiple methods for creating users in IAM. The recommended method is to use the AWS Management Console. The process of creating a user and enabling that user to perform work tasks consists of the following steps:

1. Create the user.
2. Create credentials for the user.
3. As a best practice, create only the credentials that the user needs. For example, for a user who requires access only through the AWS Management Console, do not create access keys.

   Note: For cloud security the best practice is to limit access for the `root` account, so the `root` account is locked by default.

4. Grant the appropriate permissions to the user to administer the TNSR® AMI.
5. Provide the user with the necessary sign-in information.
6. (Optional) Configure multi-factor authentication (MFA) for the user.

1.4.1 Creating IAM Users (Console)

The AWS Management Console can create IAM users.

To create one or more IAM users (console):

1. Sign in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, choose **Users** and then choose **Add user**.
3. Type the username for the new user. This is the name they will use to sign in to AWS. To add up to 10 users at once, choose **Add another user** for each additional user and type their usernames.
4. Usernames can be a combination of up to 64 letters, digits, and these characters: `+=,.@-`
5. Names must be unique within an account and are *not* case-sensitive.
6. Select **AWS Management Console access**. This creates a password for each new user.

   Choose one of the following options for **Console password**:

   Autogenerated password
   Each user gets a randomly generated password that meets the account password policy in effect (if any).

   Note: The **Final** page allows viewing or downloading the passwords.

   Custom password
   Each user is assigned a given password.

Tip: The best practice is to select **Require password reset** to ensure that users are forced to change their password the first time they sign in.

7. Click **Next**. On the **Set permissions** page, specify how to assign permissions to this new user(s). Choose one of the following three options:

   Add user to group

   Choose this option to assign the user(s) to one or more groups that already have permissions policies. IAM displays a list of the groups in the account, along with their attached policies.

   Select one or more existing groups or choose **Create group** to create a new group.

   Copy permissions from existing user

   Choose this option to copy all access rights from an existing user to the new user(s).

   Attach existing policies to user directly

   Choose this option to see a list of the managed policies in the account. Select the policies to attach to the new users or choose **Create policy** to open a new browser tab and create a new policy.

8. Choose **Next: Review** to see all the choices made up to this point. Choose **Create user** to proceed.

9. To view user access keys (access key IDs and secret access keys), choose **Show** next to each password and access key to display. To save the access keys, choose **Download .csv** and then save the file to a secure location.

Danger: This is the **only** opportunity to view or download the secret access keys, and users **must** have this information before they can use the AWS API. Save the user new access key ID and secret access key in a safe and secure place.

There is no way to access the secret keys again after this step.

10. Choose **Send email** next to each user to send a message with account information. This opens a local mail client with a draft that to customize and send. The email template includes the following details to each user:

    - Username
    - URL to the account sign-in page. Use the following example, substituting the correct account ID number or account alias:
    - https://AWS-account-ID or alias.signin.aws.amazon.com/console

Important: The user's password is **not** included in the generated email as email is not a secure communications channel. Provide passwords to the user securely using a method which complies with security policies set by the organization.

1.5 Using IAM Roles

AWS IAM Roles are used to delegate access to users, applications, or services that require controlled access to AWS resources. IAM Roles should be used to manage all Netgate® TNSR® software instances. This unique role can be specified when launching a new instance, or attached to an existing instance.

The AWS Management Console is the recommended method for creating roles for use with TNSR® software. The best practice is to create these roles based on the *principle of least privilege*, also known as the *principle of least authority*, which is the assignment of lowest needed privileges based on necessity. These instructions attempt to follow this principle.

1.5.1 Create Policy for TNSR Software Management IAM Role

Create a custom policy that will be associated with an IAM role allowing access to the TNSR® Management GUI running on an EC2 Instance.

1. Sign in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane of the console, select **Policies** then choose **Create Policy**.
3. Click the **Service** dropdown and select **EC2**.
4. In the **Actions** dropdown check the box next to **All EC2 actions (ec2:)**

   Note: If stricter policies are required for the actions that can be performed on the TNSR® EC2 Instance, these can be set here.

5. Select the **Resources** dropdown arrow and review resulting warnings.
6. Click the **All resources** bubble
7. Select **Review policy**.
8. Populate the **Name** field (e.g. `TNSR_EC2_Access`) and **Description**, if desired.

   Note: Policy names must be unique within the AWS account, and the name of the policy cannot be changed once created.

9. Select **Create Policy**.

1.5.2 Create IAM Role for TNSR Software Management

Create a role that an IAM user, or users within an IAM Group, can assume and use to connect to and manage TNSR® running on an EC2 Instance.

Note: The administrator of the specified account can grant permission to assume this role to any IAM user in that account. To do this, the administrator attaches a policy to the user or a group that grants permission for the **sts:AssumeRole** action. That policy must specify the role's ARN as the Resource.

1. Sign in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane of the console, select **Roles** then choose **Create Role**.

3. Select the **Another AWS account** role type.
4. In the **Account ID** field, type the AWS account ID that will be allowed to access the destination resource.
5. The **Require external ID** checkbox should remain cleared unless granting permissions to users from an account not under the control of this organization. Reference AWS Documentation for External ID Roles in the event this is required.
6. The best practice is to restrict the role to users who sign in with multifactor authentication (MFA). Select **Require MFA** to add a condition to the role's trust policy to require MFA sign-in.
7. Select **Next: Permissions**.
8. Type the name of the previously created Custom policy in the search field. Check the box next to the correct Policy name.
9. Select **Next:Tags**

   Note: IAM tags are key-value pairs that can be used to organize, track, or control access for this role. This is an optional step. More information can be found within AWS Documentation for Tagging IAM Entities.

10. Select **Next: Review**.
11. Populate the **Role name** field (e.g. `TNSR_Admin`) and Role description if desired.

    Note: Role names must be unique within the AWS account, and the name of the role cannot be changed once created.

12. Review remaining configured settings then select **Create role**.

This role can now be assigned to an IAM User or all users in an IAM group allowing secure administrative access to the EC2 Instance(s) containing TNSR®.

CHAPTER

TWO

INSTANCE USAGE

2.1 Launching an Instance

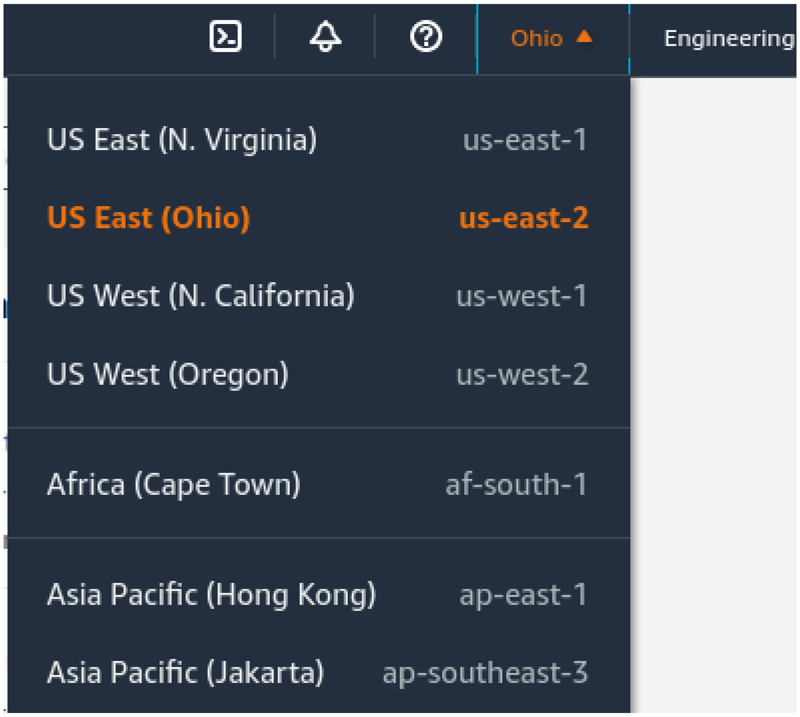
These instructions cover how to launch a new instance of the Netgate® TNSR® appliance from the Amazon EC2 Management Console.

1. Login to AWS, for example by navigating to https://console.aws.amazon.com/

   Note: This URL may be different if using other login functions, such as an IAM role or SSO authentication.

2. Select the region for the instance to run in:

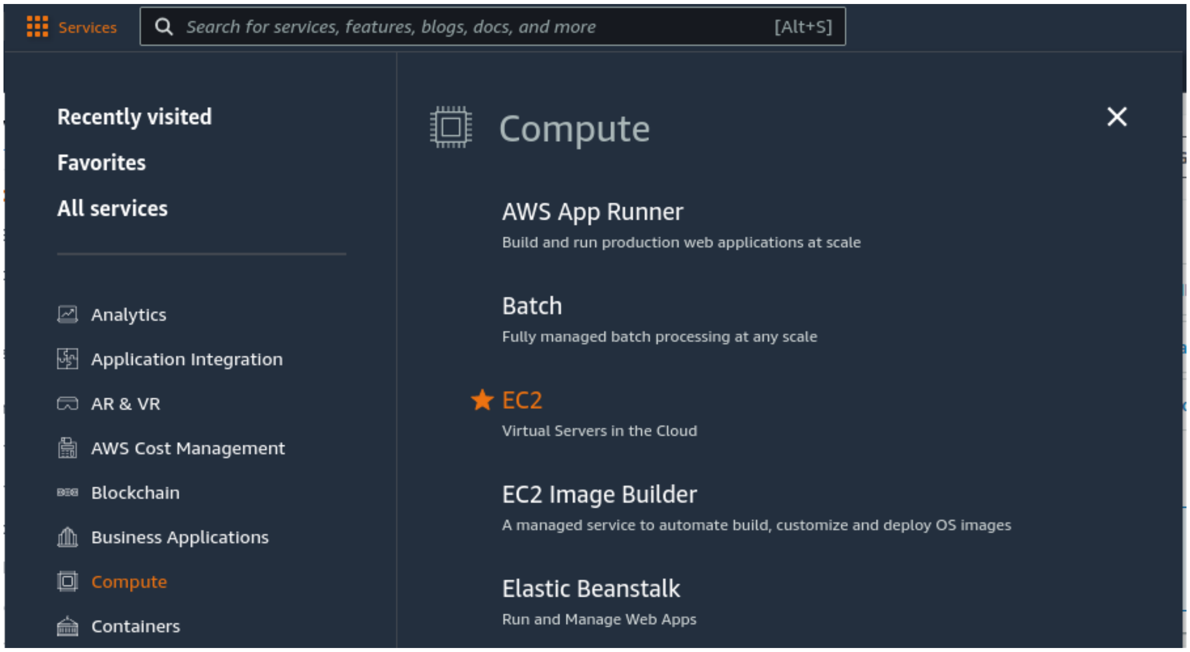
   - Click the current **Region** name near the upper right corner of the page
   - Select a new region if necessary


Ohio
Engineering
US East (N. Virginia) us-east-1
US East (Ohio) us-east-2
US West (N. California) us-west-1
US West (Oregon) us-west-2
Africa (Cape Town) af-south-1
Asia Pacific (Hong Kong) ap-east-1
Asia Pacific (Jakarta) ap-southeast-3

3. Navigate to the EC2 console

   - Click **Services** near the top left corner of the page

- Click **Compute** on the left navigation menu
- Click **EC2** on the main section of the menu

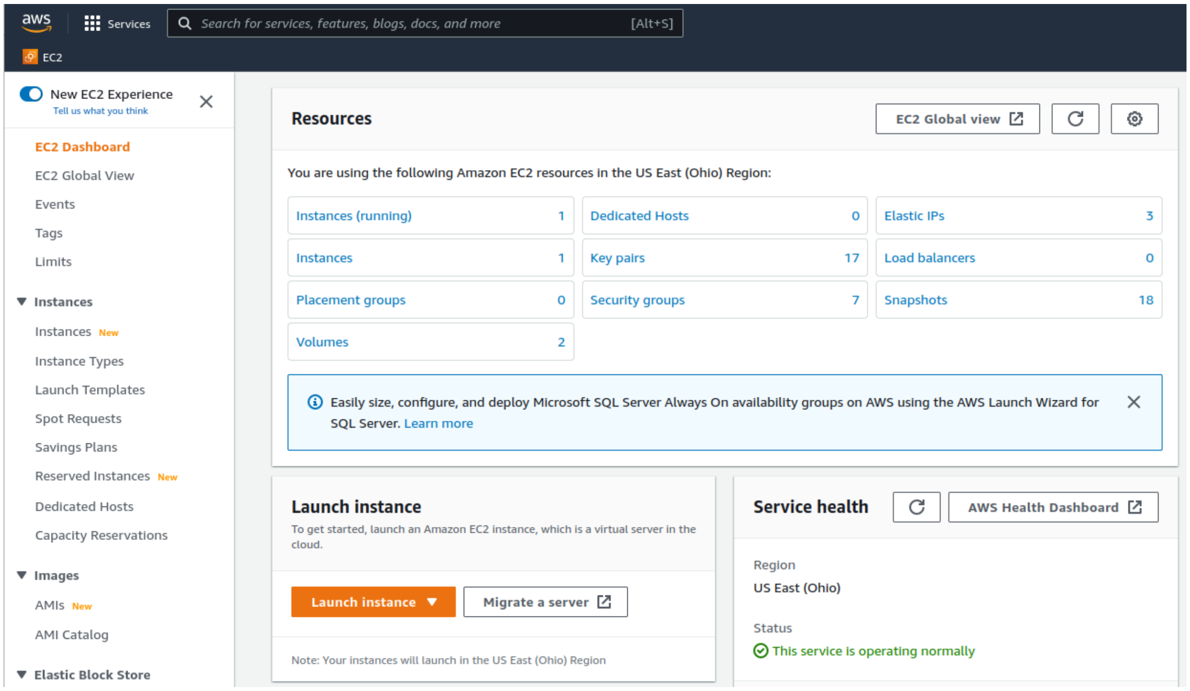

Services
Search for services, features, blogs, docs, and more [Alt+S]
Recently visited
Favorites
All services
Analytics
Application Integration
AR & VR
AWS Cost Management
Blockchain
Business Applications
Compute
Containers
Compute
AWS App Runner
Build and run production web applications at scale
Batch
Fully managed batch processing at any scale
EC2
Virtual Servers in the Cloud
EC2 Image Builder
A managed service to automate build, customize and deploy OS images
Elastic Beanstalk
Run and Manage Web Apps

4. Enter the **Launch Instance Wizard**

   - Click the **Launch Instance** button to open the **Launch Instance** menu

     This button is in the **Launch Instance** section which is located under the **Resources** section of the EC2 dashboard.

   - Click **Launch Instance** from the menu



You are using the following Amazon EC2 resources in the US East (Ohio) Region:

Instances (running)	1	Dedicated Hosts	0	Elastic IPs	3
Instances	1	Key pairs	17	Load balancers	0
Placement groups	0	Security groups	7	Snapshots	18
Volumes	2				

Easily size, configure, and deploy Microsoft SQL Server Always On availability groups on AWS using the AWS Launch Wizard for SQL Server. Learn more

To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

Note: Your instances will launch in the US East (Ohio) Region

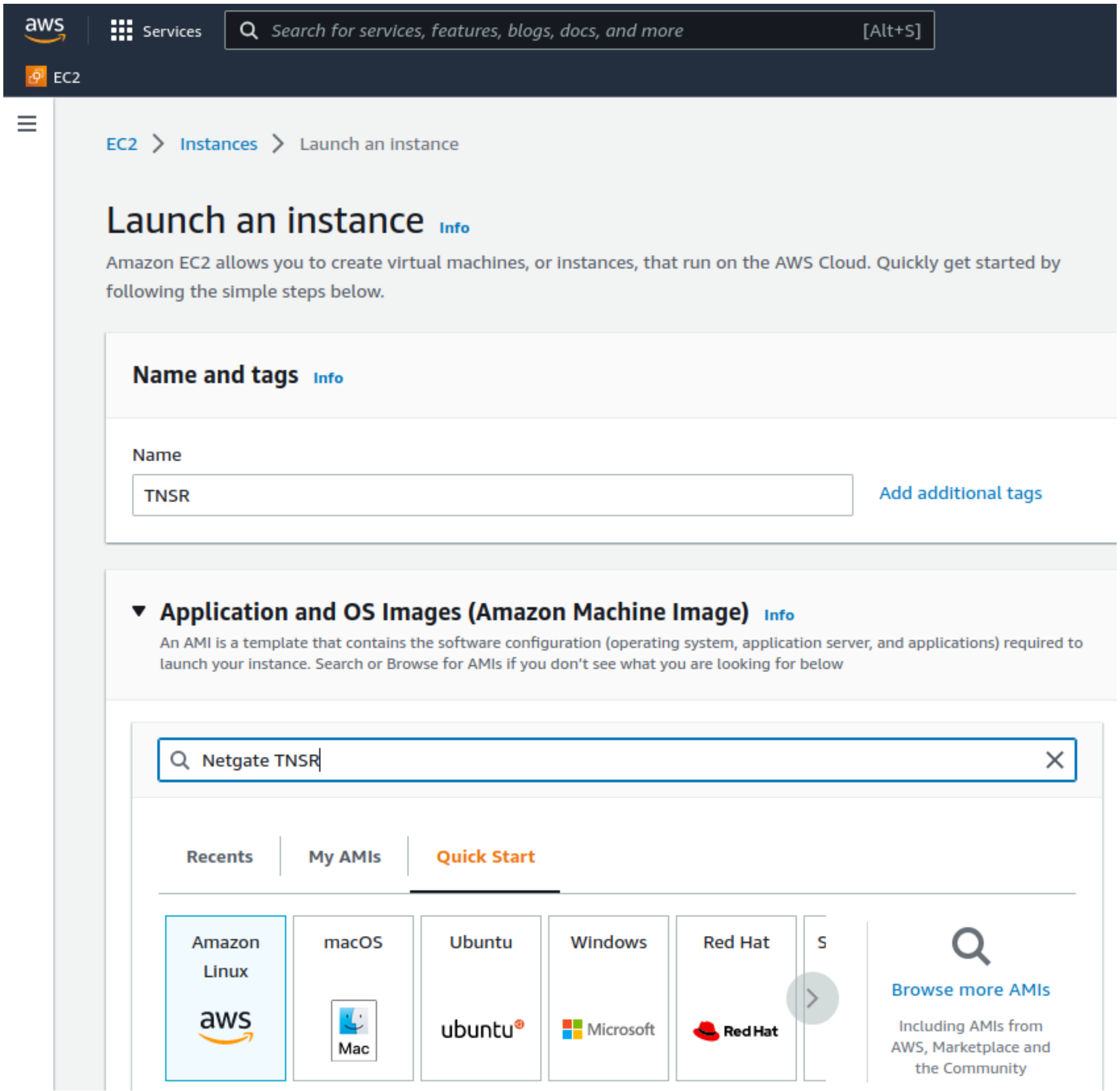
Region
US East (Ohio)

Status
This service is operating normally

5. Give the new instance a **Name**, such as `TNSR`

Optionally, click **Add Additional Tags** to create more tags which can be used to identify and locate this instance.

6. Type `Netgate TNSR` in the search box and press `Enter`.



aws
Services
Search for services, features, blogs, docs, and more
[Alt+S]
EC2
EC2 > Instances > Launch an instance

Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

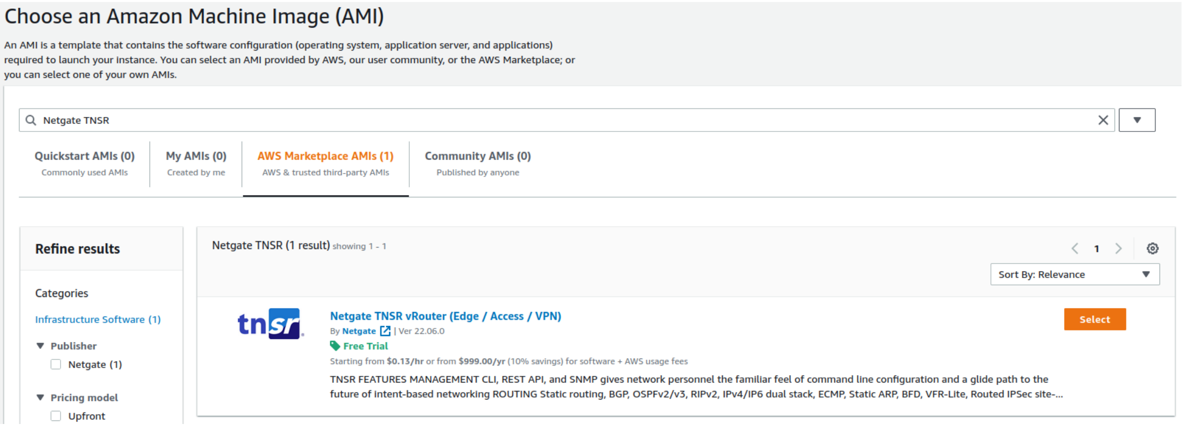
Name: TNSR

Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Netgate TNSR

7. Select **AWS Marketplace AMIs** if it is not automatically highlighted
8. Click the **Select** button for the **Netgate TNSR vRouter** entry in the search results.



Choose an Amazon Machine Image (AMI)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

Netgate TNSR

Netgate TNSR (1 result) showing 1 - 1

Netgate TNSR vRouter (Edge / Access / VPN)

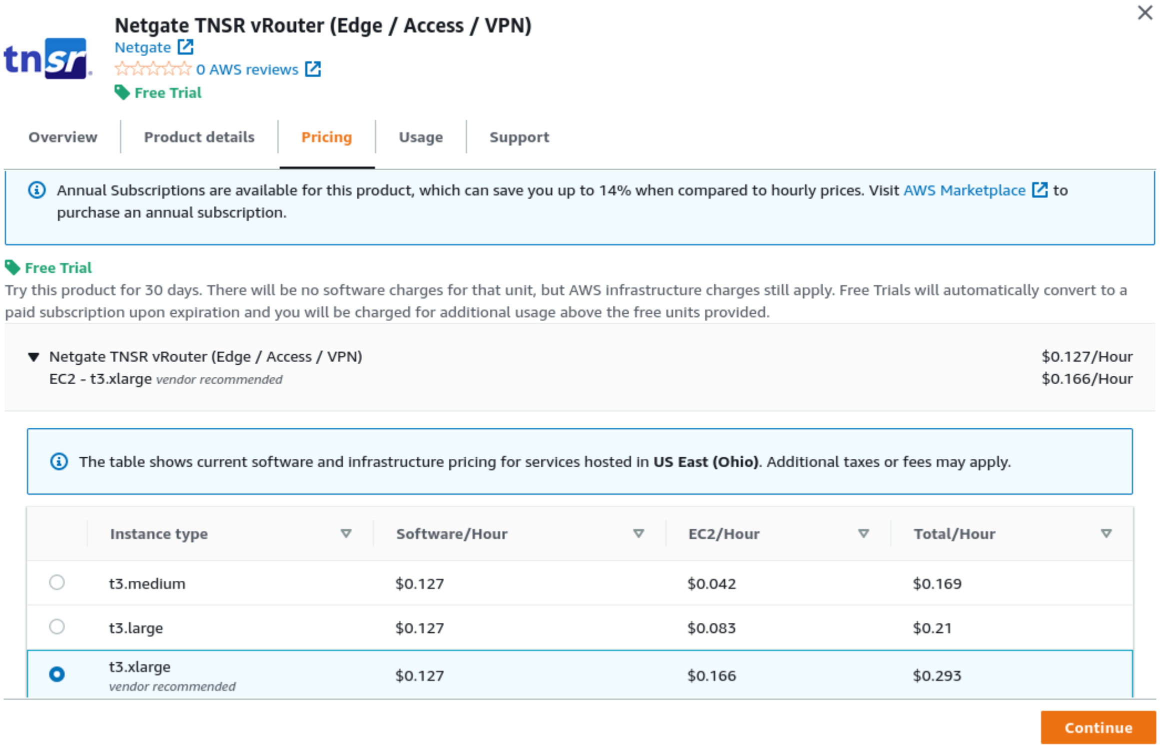
By Netgate | Ver 22.06.0

Free Trial

Starting from **$0.13/hr** or from **$999.00/yr** (10% savings) for software + AWS usage fees

TNSR FEATURES MANAGEMENT CLI, REST API, and SNMP gives network personnel the familiar feel of command line configuration and a glide path to the future of intent-based networking ROUTING Static routing, BGP, OSPFv2/v3, RIPv2, IPv4/IP6 dual stack, ECMP, Static ARP, BFD, VFR-Lite, Routed IPSec site-...

9. Review pricing and other helpful information, then click **Continue**.



Netgate TNSR vRouter (Edge / Access / VPN)

Netgate

0 AWS reviews

Free Trial

Annual Subscriptions are available for this product, which can save you up to 14% when compared to hourly prices. Visit AWS Marketplace to purchase an annual subscription.

Free Trial

Try this product for 30 days. There will be no software charges for that unit, but AWS infrastructure charges still apply. Free Trials will automatically convert to a paid subscription upon expiration and you will be charged for additional usage above the free units provided.

Netgate TNSR vRouter (Edge / Access / VPN) $0.127/Hour

EC2 - t3.xlarge *vendor recommended* $0.166/Hour

The table shows current software and infrastructure pricing for services hosted in **US East (Ohio)**. Additional taxes or fees may apply.

	Instance type	Software/Hour	EC2/Hour	Total/Hour
○	t3.medium	$0.127	$0.042	$0.169
○	t3.large	$0.127	$0.083	$0.21
●	t3.xlarge *vendor recommended*	$0.127	$0.166	$0.293

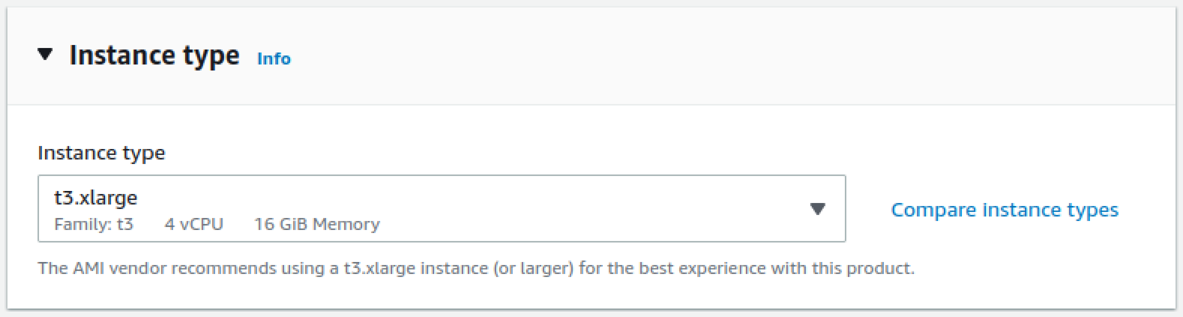
Note: TNSR software is also available with an annual subscription instead of hourly. The annual subscription may be purchased from the AWS Marketplace.

Information about support can be found on the *Support Resources* page.

10. Choose an **Instance Type** from the drop-down, then click **Next**

See also:

For guidance on which instance type to choose, see *Supported EC2 Instance Types*.



▼ Instance type

Instance type

t3.xlarge
Family: t3 4 vCPU 16 GiB Memory

The AMI vendor recommends using a t3.xlarge instance (or larger) for the best experience with this product.

11. Configure an SSH **Key Pair**

    The **Key Pair** section of the form sets the SSH key pair used by an SSH client when it connects to the TNSR instance for management.

    For an existing key pair:

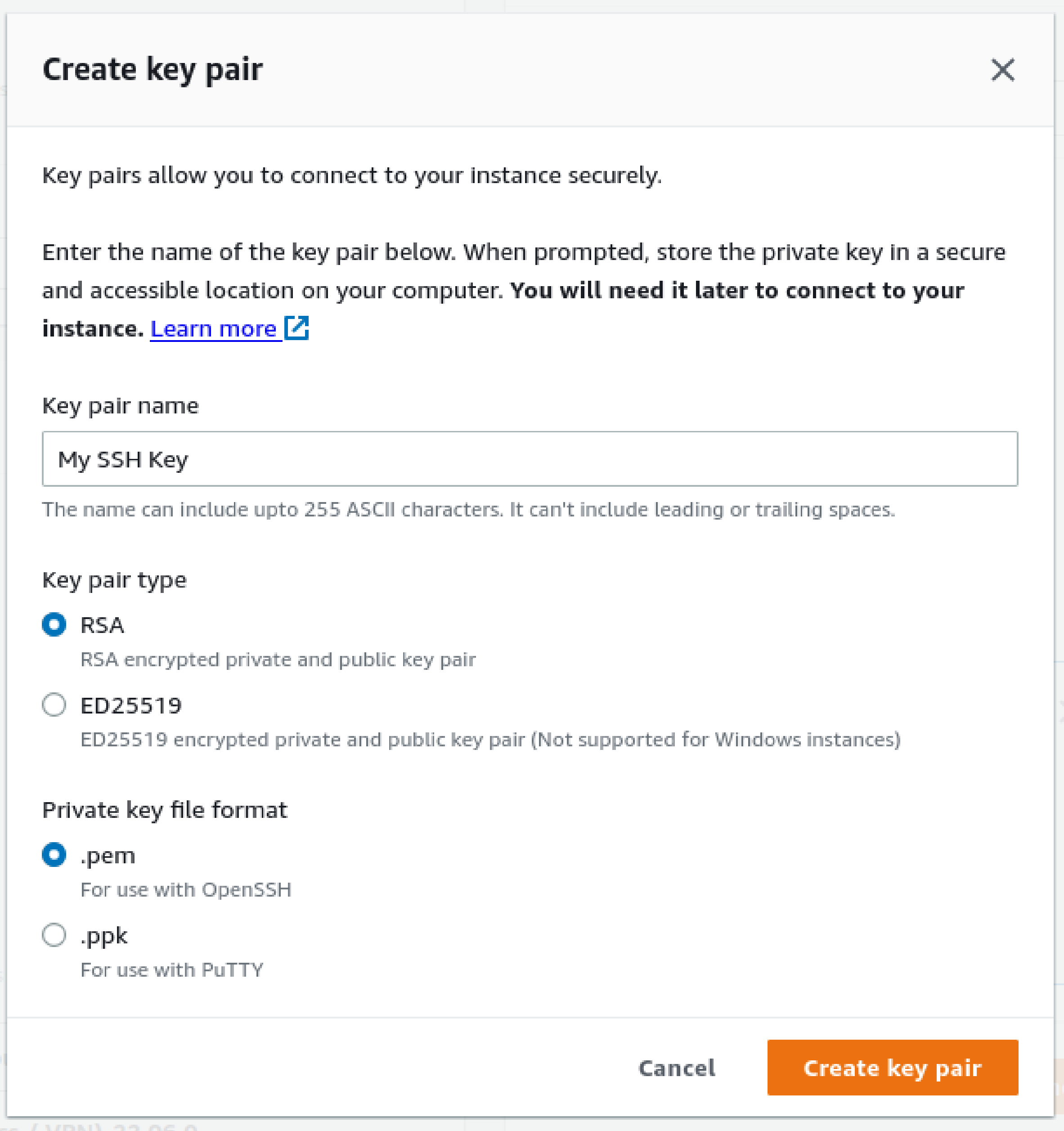
    - Click **Key pair name**
    - Search for and select an existing key pair entry

    To create a new key pair:

    - Click **Create new Key Pair**
    - Enter a **Key pair name**, such as **TNSR SSH Key**
    - Select a **Key Pair Type** and **Private Key Format**

      The chosen type and format must be compatible with whichever local SSH client will be used by TNSR administrators

    - Click the **Create key pair**
    - Select a location to save the key pair locally



Create key pair

Key pairs allow you to connect to your instance securely.

Enter the name of the key pair below. When prompted, store the private key in a secure and accessible location on your computer. **You will need it later to connect to your instance.** Learn more

Key pair name

My SSH Key

The name can include upto 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type

● RSA
RSA encrypted private and public key pair

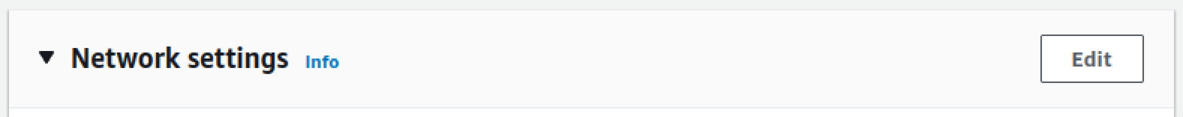
○ ED25519
ED25519 encrypted private and public key pair (Not supported for Windows instances)

Private key file format

● .pem
For use with OpenSSH

○ .ppk
For use with PuTTY

12. Click **Edit** under **Network Settings** to allow making changes for the next few steps.



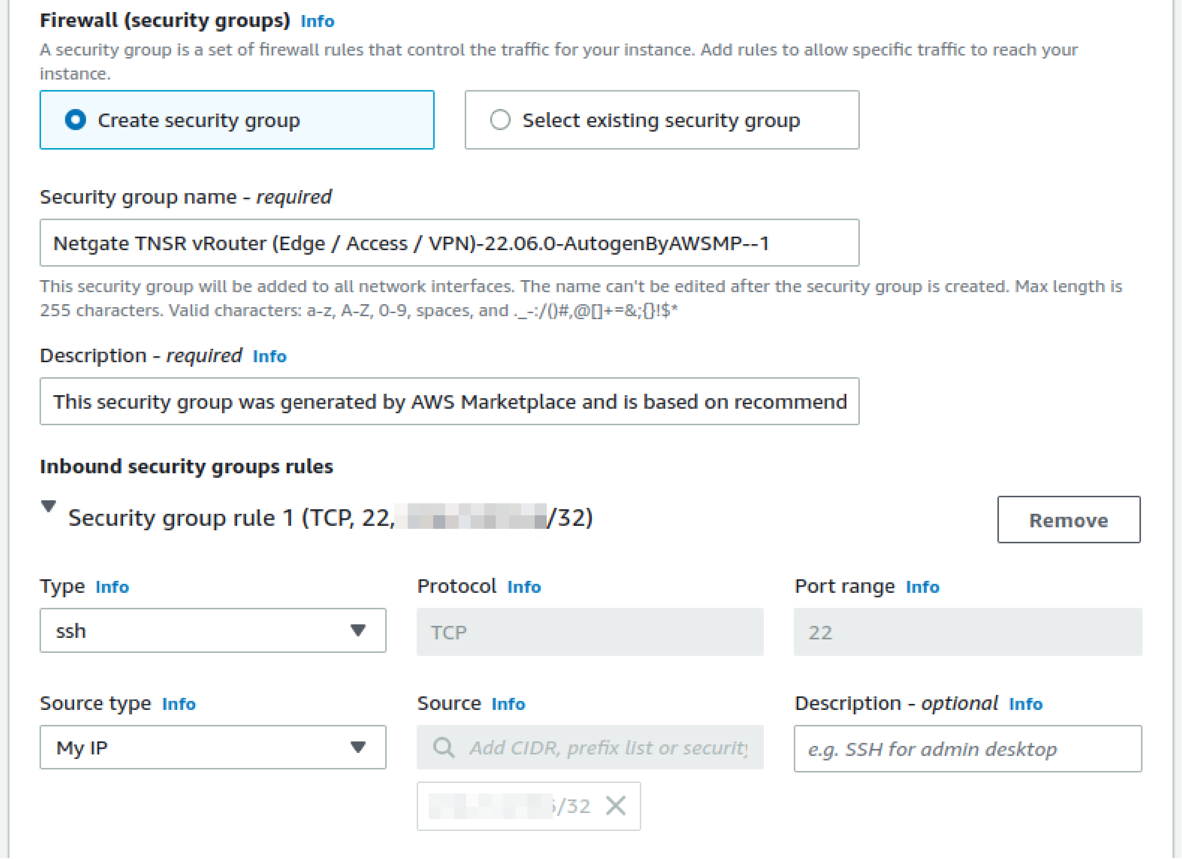
13. Configure **Security Groups**

The default security group only includes a rule to allow SSH. Since this group is for the management interface, allowing additional traffic is unlikely to be necessary, but there are still a few changes to make:

- Click **Create security group** under **Firewall (security groups)**
- Enter a **Security group name**, such as `TNSR Management` or leave it at the default automatic value.
- Enter a **Description** for the group, or leave it at the default value.

- Set the **Source type** on the default rule for SSH to **My IP** so it will restrict SSH access to the public address used by the person creating the AMI.

  This is optional, but more secure. If the address is not static, then it may not be viable to restrict this. Setting the value to **Anywhere** will allow SSH clients to connect from any source (`0.0.0.0/0`). While not ideal, allowing SSH connection from anywhere is OK because the TNSR for AWS default setup only allows key-based SSH authentication which is resistant to brute force attacks.



Firewall (security groups) Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

● Create security group ○ Select existing security group

Security group name - *required*

Netgate TNSR vRouter (Edge / Access / VPN)-22.06.0-AutogenByAWSMP--1

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and ._-:/()#,@[]+=&;{}!$*

Description - *required* Info

This security group was generated by AWS Marketplace and is based on recommend

Inbound security groups rules

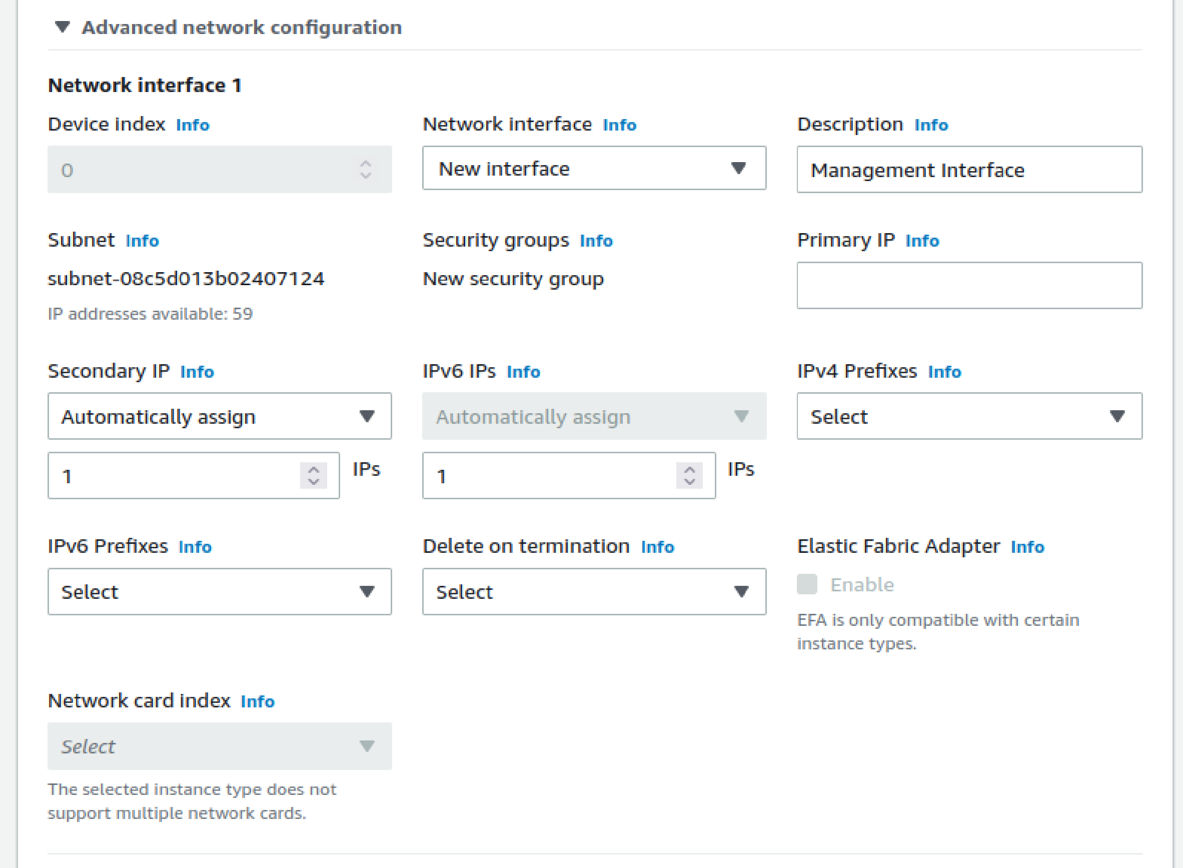
▼ Security group rule 1 (TCP, 22, [redacted]/32)

Type Info	Protocol Info	Port range Info
ssh ▼	TCP	22

Source type Info	Source Info	Description - *optional* Info
My IP ▼	*Add CIDR, prefix list or securit* [redacted]/32 ✕	*e.g. SSH for admin desktop*

14. Configure Network and Interfaces

    - Select the VPC in which to launch the instance
    - Click **Advanced Network Configuration** to expand the network interface list
    - Select the **Management subnet** as the subnet for **Network Interface 1**
    - Click the **Add Network Interface** button
    - Select the **WAN subnet** as the subnet for **Network Interface 2**
    - Click the **Add Network Interface** button
    - Select the **LAN subnet** as the subnet for **Network Interface 3**



▼ Advanced network configuration

Network interface 1

Device index: 0

Network interface: New interface

Description: Management Interface

Subnet: subnet-08c5d013b02407124
IP addresses available: 59

Security groups: New security group

Primary IP:

Secondary IP: Automatically assign, 1 IPs

IPv6 IPs: Automatically assign, 1 IPs

IPv4 Prefixes: Select

IPv6 Prefixes: Select

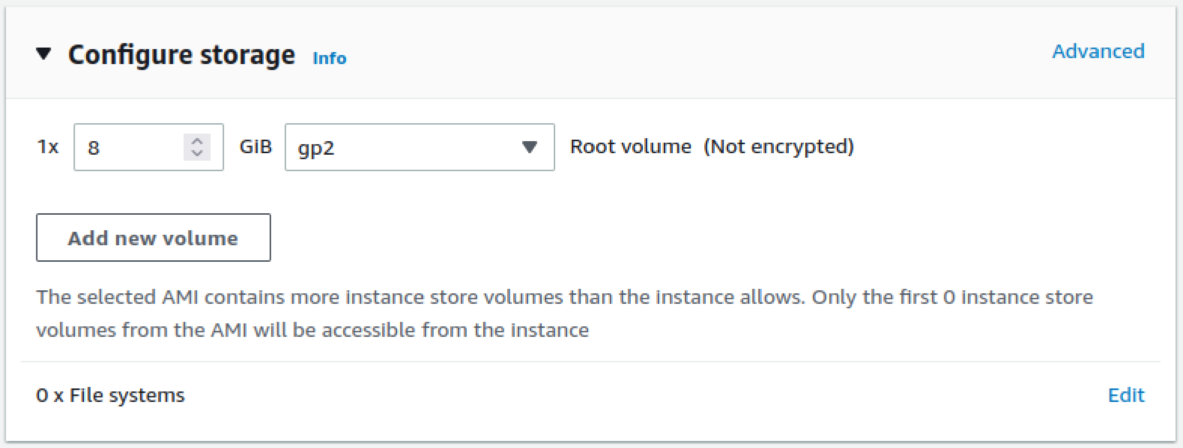
Delete on termination: Select

Elastic Fabric Adapter: □ Enable
EFA is only compatible with certain instance types.

Network card index: Select
The selected instance type does not support multiple network cards.

15. Configure storage

    If this instance will require more than the default 8 GiB disk, increase the value in the **Configure Storage** section



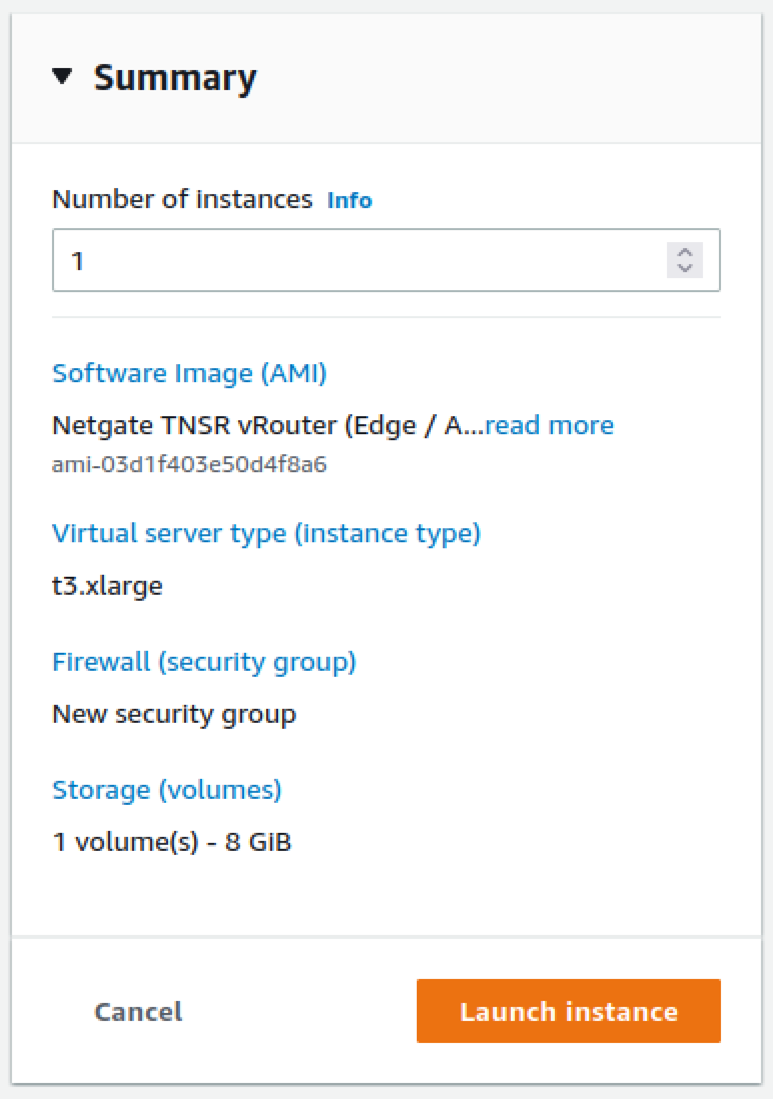
▼ Configure storage

1x 8 GIB gp2 Root volume (Not encrypted)

The selected AMI contains more instance store volumes than the instance allows. Only the first 0 instance store volumes from the AMI will be accessible from the instance

0 x File systems

16. Verify the settings selected in earlier steps and review any errors or recommendations displayed by AWS
17. Click **Launch instance** in the **Summary** box on the right side



Number of instances Info

1

Software Image (AMI)

Netgate TNSR vRouter (Edge / A...read more

ami-03d1f403e50d4f8a6

Virtual server type (instance type)

t3.xlarge

Firewall (security group)

New security group

Storage (volumes)

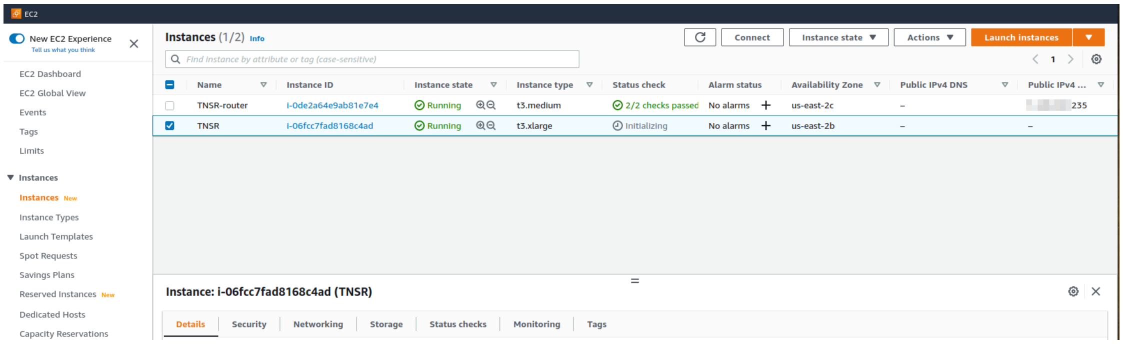
1 volume(s) - 8 GiB

2.2 Managing the Configuration of the Instance

Once the instance is launched, monitor its status using the **Instances** page of the EC2 Management Console.

The EC2 Management Console will display a list of instances. In this list it also indicates whether an instance is up and reachable and will also display its current public IP address and other related basic information.

Click the checkbox at the start of the row for an instance to display more information.

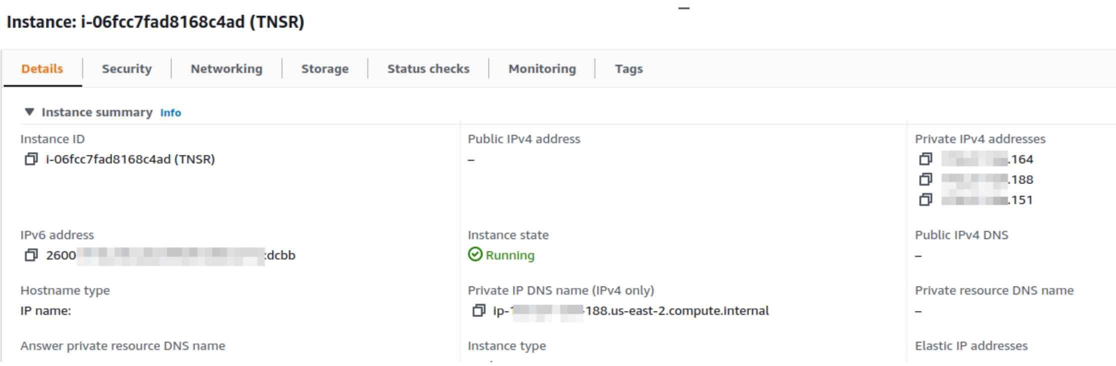


Instances (1/2) Info

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...
□	TNSR-router	i-0de2a64e9ab81e7e4	Running	t3.medium	2/2 checks passed	No alarms	us-east-2c	–	[redacted]235
☑	TNSR	i-06fcc7fad8168c4ad	Running	t3.xlarge	Initializing	No alarms	us-east-2b	–	–

Instance: i-06fcc7fad8168c4ad (TNSR)

With the instance selected, the bottom pane of the page displays detailed information about the instance. This includes the hostname and all IP address on the instance.



Instance: i-06fcc7fad8168c4ad (TNSR)

Instance summary Info

Instance ID: i-06fcc7fad8168c4ad (TNSR)

Public IPv4 address: –

Private IPv4 addresses: [redacted].164, [redacted].188, [redacted].151

IPv6 address: 2600[redacted]:dcbb

Instance state: Running

Public IPv4 DNS: –

Hostname type: IP name:

Private IP DNS name (IPv4 only): ip-[redacted]-188.us-east-2.compute.internal

Private resource DNS name: –

Answer private resource DNS name

Instance type

Elastic IP addresses

To manage the configuration of the instance, connect to it via SSH as described in *Connect to the instance*.

2.3 Advanced Usage

2.3.1 Protecting a private network in VPC

An instance of the Netgate® TNSR® appliance can be used as a firewall for a VPC subnet. This will generally require more manual configuration than using an instance to host a remote access VPN does. See the *VPC User Guide* for a more detailed explanation of how to configure a VPC and a Netgate TNSR® appliance instance to support this.

2.3.2 Connecting local devices using IPsec

A TNSR instance in AWS can act as an IPsec hub for one or more remote endpoints capable of using IPsec, such as local devices running pfSense® software. It can interconnect all the sites or even act as an Internet gateway.

For a complete example of using TNSR as an IPsec hub for multiple sites running pfSense software, see the recipe TNSR IPsec Hub for pfSense software nodes in the TNSR documentation.

2.3.3 Accessing the TNSR API

Accessing the API requires configuring the RESTCONF service securely as well as setting up a means of user authentication and NACM rules for authorization. There is a complete recipe in the TNSR documentation: RESTCONF Service Setup with Certificate-Based Authentication and NACM

> **Warning:** Ideally, the API should only be accessed from the management interface or at least over an encrypted channel such as a VPN.

For more information on the API in general, see the API Documentation.

2.3.4 Detect and Recover EC2 Instance Failure

It is also possible to create an Amazon CloudWatch alarm that monitors an Amazon EC2 instance and automatically recovers the instance if it becomes impaired due to an underlying issue.

For more information about instance recovery, see Recover Your Instance.

2.4 Frequently Asked Questions

2.4.1 How can an instance be accessed?

In order to manage the configuration of the instance, connect to it via **SSH** as described in *Connect to the instance*.

2.4.2 What are the default credentials for the tnsr user on the AWS instance?

The `tnsr` user on the TNSR for AWS instance does not have a default password. SSH connections to a TNSR for AWS instance require key-based authentication using an SSH key selected when launching the instance, which is much more secure than using password-based authentication.

The process of using key-based authentication to connect to an instance is covered in *Connect to the instance*.

2.4.3 How does NAT behave on AWS?

NAT behavior on AWS can be tricky, as in certain places NAT can be applied by TNSR and in other places by AWS. Determining where and how to perform NAT can be potentially problematic in that it is possible to unintentionally create asymmetric routing situations with an incorrect configuration.

See NAT Examples for multiple examples of NAT behavior with TNSR on AWS and how to avoid these pitfalls.

2.4.4 How can an instance be backed up and recovered?

The procedure to back up and restore the configuration databases and other key files is covered in the Configuration Backups section of the TNSR documentation.

2.4.5 How can an instance be monitored?

TNSR can be monitored in several ways compatible with standard utilities, such as SNMP, IPFIX, and Prometheus as well as customized monitoring by polling the TNSR API. The most common methods are covered in the Monitoring section of the TNSR documentation.

2.4.6 How can an instance be upgraded?

Upgrading a TNSR instance in-place requires a valid upgrade certificate issued by Netgate. The process for obtaining the certificate as well as for performing the upgrade is covered in the Updates and Packages section of the TNSR documentation.

Note: Requesting a TNSR upgrade certificate from Netgate TAC requires the current **AWS Customer ID** and **AWS Instance ID**. For more details, see the documentation on upgrading TNSR in AWS

Even without the TNSR upgrade configuration in place, the operating system can be upgraded to obtain security fixes for issues in the base OS.

2.4.7 How can credentials and keys be changed?

Credentials and keys, such as user account keys, certificates, VPN tunnel keys, and so on, should be changed periodically for security. Procedures to change these are located in Changing Credentials and Keys.

2.4.8 Further troubleshooting

More information on troubleshooting a variety of common TNSR issues can be found in the Troubleshooting section of the TNSR documentation.

CHAPTER

THREE

VIRTUAL PRIVATE CLOUD (VPC)

3.1 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.

The steps involved are:

3.1.1 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 1: Example AWS Network Configuration

Item	Value
VPC 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 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.

3.1.2 Launch an Instance

To launch this instance, follow the procedure in *Launching an Instance*.

3.1.3 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.

To allocate a new TNSR LAN Network Interface, create a new **Elastic Network Interface** on the LAN subnet following the instructions here https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-eni.html#create_eni

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

3.1.4 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.

To disable **Source/Destination Check** for a Network Interface, follow the instructions at https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-eni.html#change_source_dest_check

3.1.5 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.

For each interface that needs a public Elastic IP Address, allocate one by following the instructions at https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/elastic-ip-addresses-eip.html#using-instance-addressing-eips-allocating

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/
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

3.1.6 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`.

Note: The `root` account is present on the appliance but disabled by default. It is not necessary to enable or use the `root` account. Any administrative actions can be taken using the `tnsr` account from within the TNSR CLI or by using `sudo` from a shell prompt.

See Default Accounts and Passwords for additional information.

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/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: 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#
```

Tip: Additional host users can be added to TNSR so that each administrator can use their own account. See the User Management section of the TNSR documentation for details.

3.1.7 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 the TNSR instances, other than the Management Interface, are not displayed by `show interface`, restart the data plane and the missing interfaces will appear:

```
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
```

(continues on next page)

(continued from previous page)

```
    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 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
   tnsr(config-interface)# description eth1 eni-beaa7c21 WAN
   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. 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
    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

```
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 from 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 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 it 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

3.1.8 Configure Default Route in TNSR

In order for the TNSR data plane to forward packets outside 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-rttbl4-next-hop)# next-hop 1 via 10.5.0.1 VirtualFunctionEthernet0/6/0
tnsr(config-rttbl4-next-hop)# exit
tnsr(config-route-table-v4)# exit
tnsr(config)# exit
tnsr#
```

3.1.9 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/
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
64 bytes from 52.7.26.219: icmp_seq=4 ttl=45 time=48.433 ms

--- 52.7.26.219 ping statistics ---
5 packets transmitted, 5 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 48.433/48.863/49.238/0.323 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.

CHAPTER

FOUR

REFERENCES

4.1 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 1: AWS Available Regions

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

4.2 Support Resources

4.2.1 Commercial Support

TNSR® software is bundled with Netgate TAC Pro support, with TAC Enterprise available for upgrade.

Netgate TAC support options:

	TAC Pro	TAC Enterprise
TAC Support Hours	24/7	24/7
Target Initial Response SLA	24 Hour	4 Hour
Email / Support Portal	Yes	Yes
Telephone Support	No	Yes

For more information and purchasing, including the most up-to-date information on available TAC offerings, see: https://www.netgate.com/support.

4.2.2 Community Support

Community support is available through the Netgate Forum.

4.3 Additional Resources

4.3.1 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

4.3.2 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/

4.3.3 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/