cohesive networks

Overlay Engine

VNS3 Plugins Guide 2018

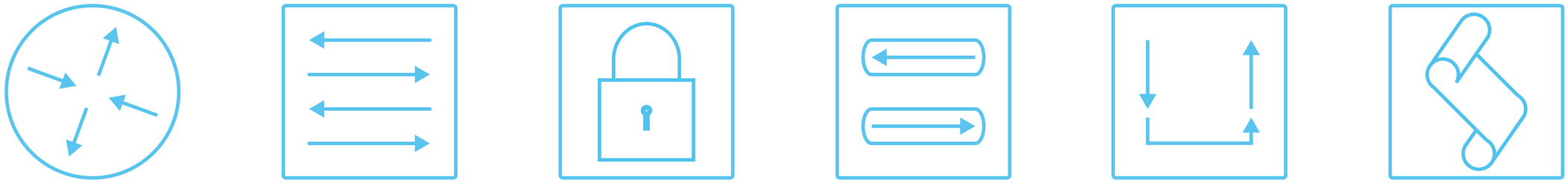
Table of Contents

Introduction

VNS3:turret provides container based network services

Isolated Linux containers within VNS3 allow partners and customers to embed features and functions safely and securely into their cloud network.

VNS3 Core Components



Requirements

You have a cloud or virtual infrastructure account that Cohesive Networks can use for enabling your access to the VNS3 Controller Images.

You have the ability to create an AWS IAM instance role.

You have agreed to the VNS3 Terms and Conditions.

Basic knowledge of Linux software installation and use of command line tools.

Getting Help with VNS3

This document assumes you have a VNS3 Controller instance launched and running in a security group, network or similar that has the appropriate access rules included for normal VNS3 operations. For any support issues, email us at support@cohesive.net

Please review the VNS3 Support Plans and Contacts before sending support inquiries. If you need specific help with project planning, POCs, or audits, contact our professional services team via sales@cohesive.net for details.

Overlay Engine Detail

Getting Started with VNS3 Plugin System

The following Overlay Engine functionality is deployed to VNS3 as a plugin using the container system. These instructions cover customization of the container image that will be used so that customer keys and rule sets can be employed.

Please be familiar with the VNS3 plug-in configuration guide: https://cohesive.net/dnld/Cohesive-Networks_VNS3-3.5-Container-System.pdf

Overlay Engine Plugin - What does it do?

Simply stated Overlay Engines make your Overlay Network faster.

With Overlay Engines running on VNS3 controller in a jumbo frame enabled cloud environment like AWS VPC, total throughput on c4.2xl of 12-20 Gbps is achievable depending on a tradeoff for the use-case between latency and throughput.

Overlay Engine Plugins add another Overlay Network process to a VNS3 Controller allowing more connections at higher speeds. The number of Overlay Engines allocated to a particular VNS3 controller depends on the number of plugin slots licensed.

Overlay Engine Plugin - How does it work?

Overlay Engines are run as a sealed plugin (no user access via SSH) that get access to the relevant Overlay Network keys and mechanisms in the host VNS3 controller. The Overlay Engine then runs all the processes necessary to handle Overlay Network client connections. This allows the Overlay Network to get multithreaded access to the VNS3 instance resources.

Overlay Engine Plugin - What does it need?

Overlay Engine plugins need two things:

- The source Container Image uploaded to the VNS3 controller needs *overlayengine* in the name.
- VNS3 firewall rules need to be added to DNAT ports to the specific Overlay Engine plugins

Running the Overlay Engine Plugin

Getting the Overlay Engine Plugin

The Linux-based (Ubuntu 14.04) Overlay Engine Plugin is accessible at the following URL:

https://vns3-containers-read-all.s3.amazonaws.com/Overlayengine/Cohesive_OE_20170908a.gz

This is a read-only Amazon S3 storage location. Only Cohesive Networks can update or modify files stored in this location.

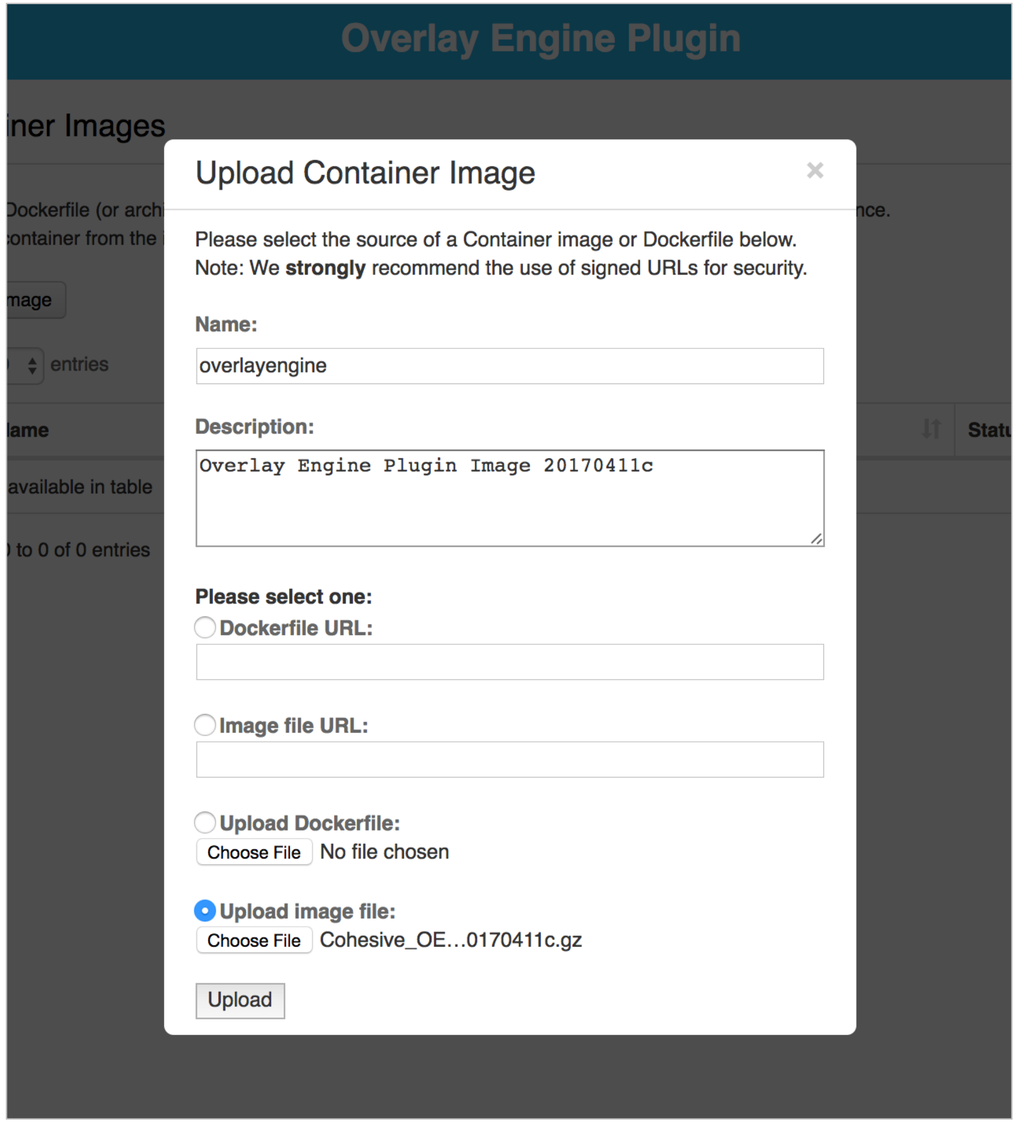
This URL can be used directly in a VNS3 Controller via the Web UI or API to import the container for use into that controller. (General screenshot walkthrough and help available in the plug-in configuration document.)

Uploading the Container Image to the VNS3 Plugin System

From the *Container —> Images* menu item, choose **Upload Image**.

Name the Container Image overlayengine. This is a requirement to get the allocated plugins to receive access to the required directories on the VNS3 host.

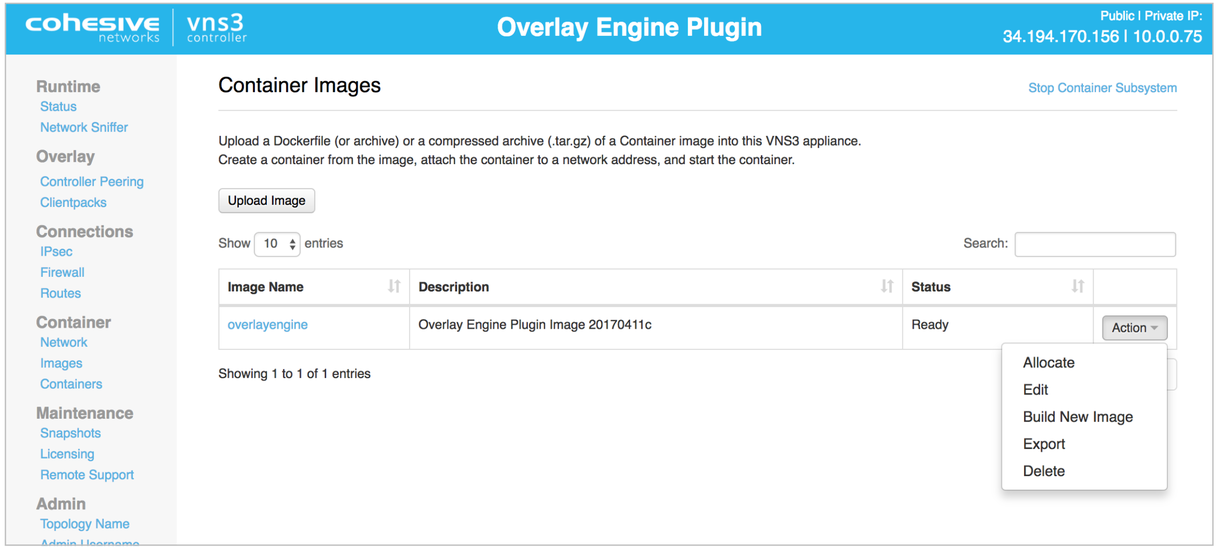
To use the pre-configured plugin paste the URL into the *Image File URL* box.



Allocating a Container from the Overlay Engine Image

When the Image has imported it will say **Ready** in the Status Column.

To then launch a running Overlay Engine plugin, choose **Allocate** from the *Action* menu.



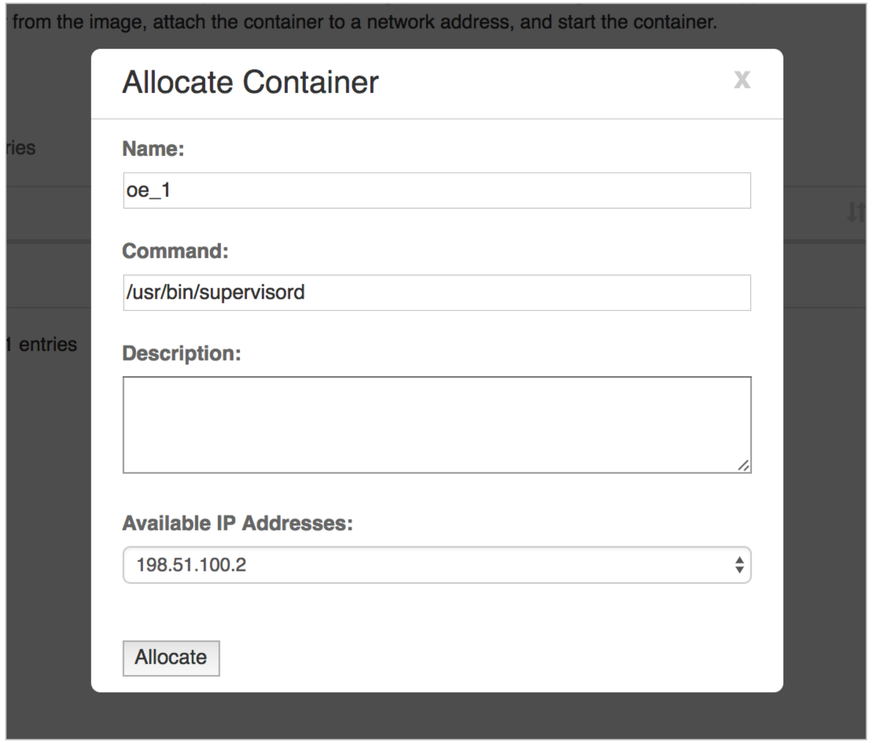
Launching the BGP HA Plugin

After selecting **Allocate** from the *Actions* menu you then name your container, provide a description and the command used to execute the container.

The **name** should be UNIQUE among all Overlay Engine containers allocated to a VNS3 controller.

For the Overlay Engine Plugin the command used is:

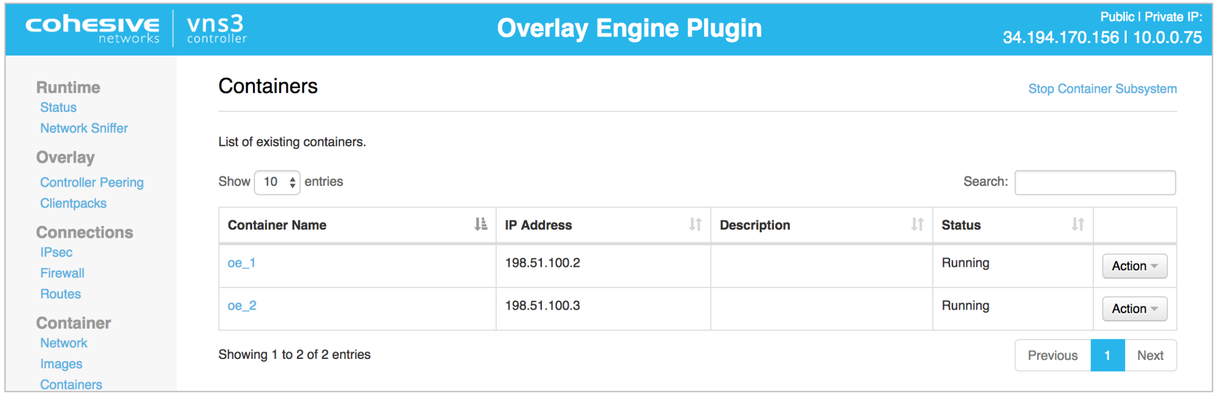
/usr/bin/supervisord



Confirming the BGP HA Plugin is running

After executing the **Allocate** operation you will be taken to the *Container Display page*.

You should see your BGP HA Plugin with the name you specified. The Status should be *Running* and it should have been given an **IP address** on your internal plug-in subnet (in this case 192.51.100.2).



Overlay Engine Firewall Rules

Overlay Engines will need there types of firewall rules added to the VNS3 controller.

1. DNAT rule to allow a clientpack with a specific UDP port other than 1194 to reach the Overlay Engine. Cohesive Networks recommends using 1193 and lower for each Overlay Engine:

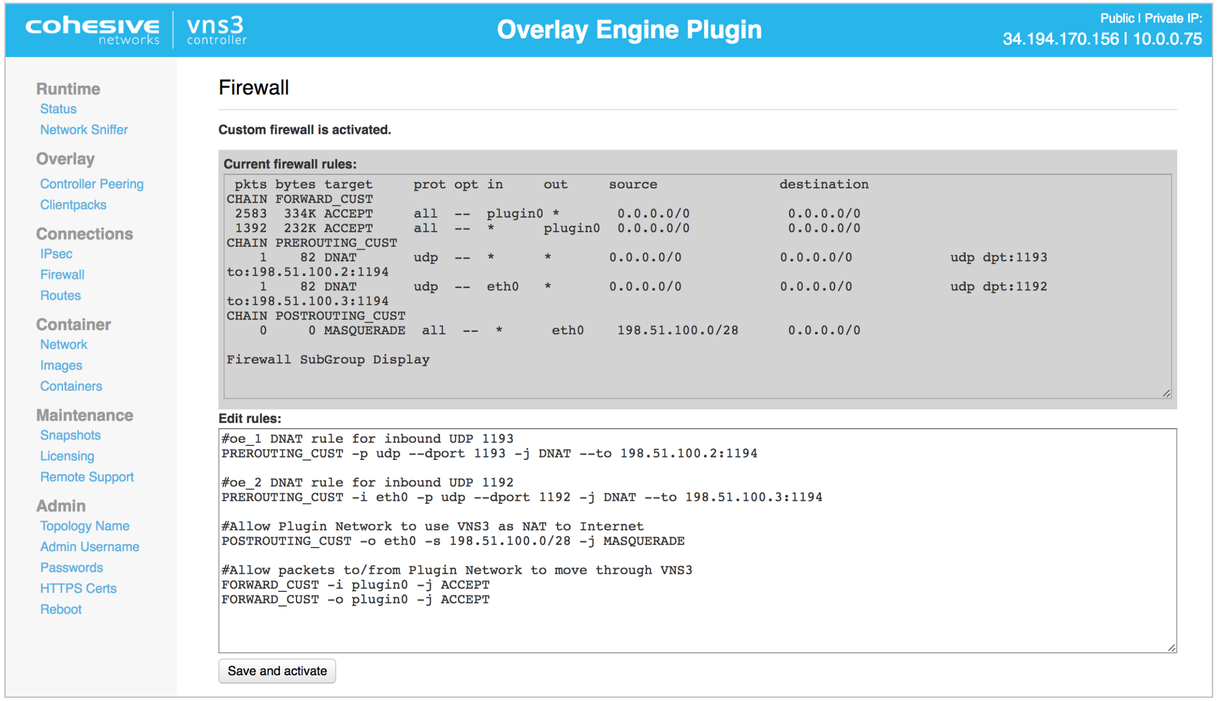
```
PREROUTING_CUST -p udp --dport 1193 -j DNAT --
to 198.51.100.2:1194
```

2. MASQUERADE rule to allow the Overlay Engine to communicate with the Overlay clients through it's host VNS3 controller:

```
POSTROUTING_CUST -o eth0 -s 198.51.100.0/28 -j
MASQUERADE
```

3. FORWARD rules to allow packets to move across the VNS3 Host that are to/from the Overlay Engine Plugin

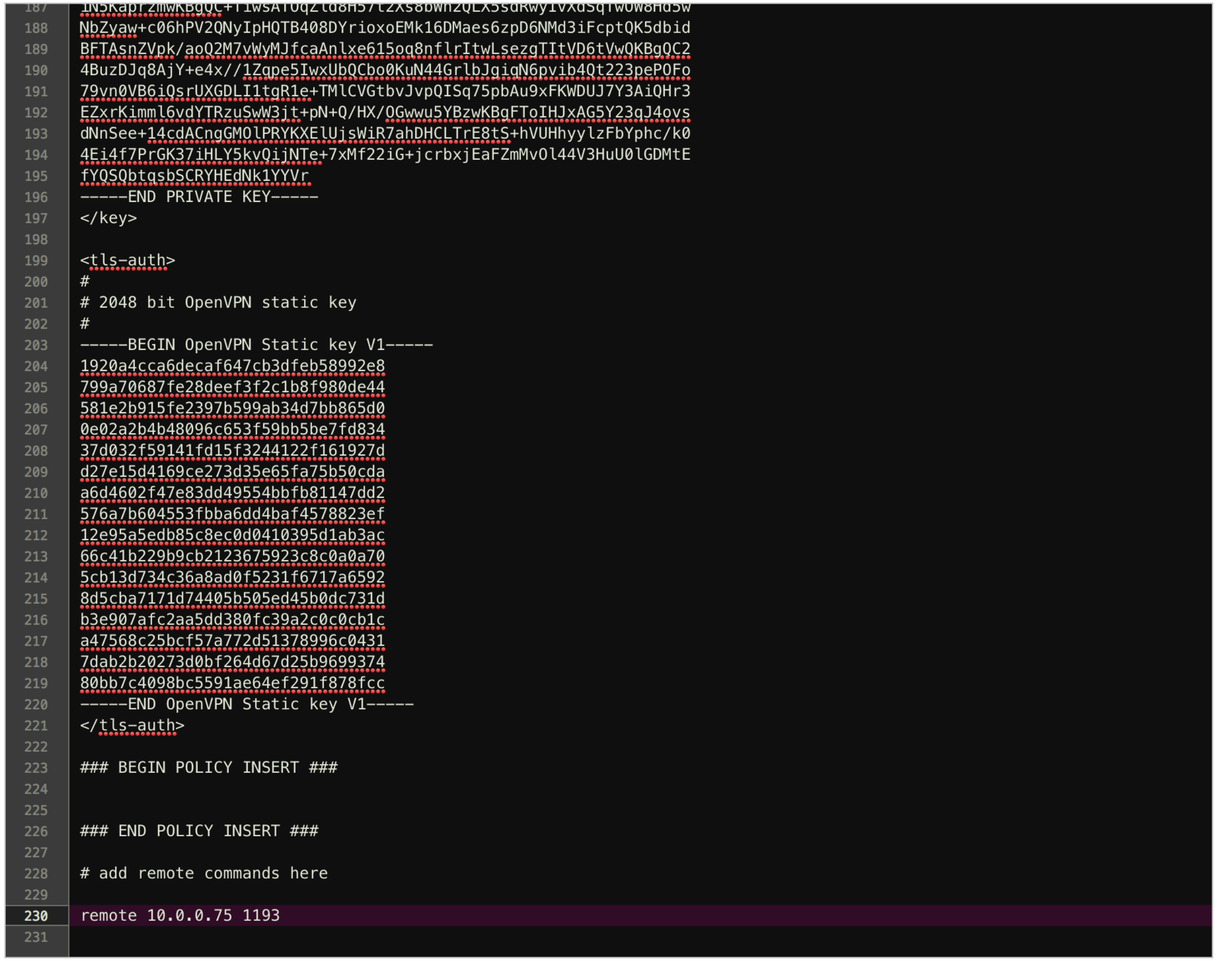
```
FORWARD_CUST -i plugin0 -j ACCEPT
FORWARD_CUST -o plugin0 -j ACCEPT
```



VNS3 Clientpack Configuration

The clientpack for clients you want to join a specific Overlay Engine will need a slight modification to the remote command section to use the appropriate UDP port used in the DNAT firewall rules.

This is an example of a clientpack file with the default remote command line port modified to use 1193.



Overlay Engine Best Practices

Best Practices for using the Overlay Engine Plugin

- **Overlay Engine Names need to be unique** - the Overlay Engine container name space is used to update status information on the host VNS3 controller. Overlap in naming will result in incorrect reporting with respect to connected client servers.
- **Not Rename Overlay Engine allocated Containers** - this reduces the chance of overlapping name space in allocated containers.
- **Run Logging Container** - Logging Container has access to all the Overlay Engine logs.
- **Use non-overlapping Container Network Subnets if running Overlay Engines on multiple Peered VNS3 Controller** - This provides maximum flexibility and ensures no collision of routing when running Overlay Engines in a peered mesh.
- **Use the client configuration gateway directive.** To ensure that all routes known to the network are reachable, overlay engine clients should use the "redirect-gateway def1" configuration line. This makes the connected client use the overlay engine, and the VNS3 controller as its default gateway for everything, including the cloud subnet. More information on this directive can be found here: https://cohesivenet.freshdesk.com/solution/articles/31000035384-routing-all-client-traffic-through-the-vns3-overlay

Restrictions/Limitations

Restrictions/Limitations

- **UDP Multicast not supported** - Normally client devices connected to VNS3 controllers can utilize the udp multicast protocol, despite it not being supported by the underlying cloud vendor. This capability runs over the encrypted overlay network. Client devices connected to overlay engines CANNOT send/receive multicast packets. Cohesive is working to remove this limitation.
- **Cohesive VNS3 Routing Agent not supported** - Normally overlay client devices receive their route specifications ONLY when they connect to a VNS3 Controller. This is equally true for client devices connected to an overlay engine running on a VNS3 Controller. This creates the need for client devices to have their TLS Tunneling Agent (openvpn client) stopped/started to get new routes for the network. To prevent this administrative burden, Cohesive provides a separate piece of software to run on overlay clients, the VNS3 Routing Agent. The VNS3 Routing Agent CANNOT receive route updates when running on a client device connected to an overlay engine. Cohesive is working to remove this limitation.

VNS3 Configuration Document Links

VNS3 Product Resources - Documentation | Add-ons

VNS3 Configuration Instructions (Free & Lite Editions | BYOL)
Instructions and screenshots for configuring a VNS3 Controller in a single or multiple Controller topology. Specific steps include, initializing a new Controller, generating clientpack keys, setting up peering, building IPsec tunnels, and connecting client servers to the Overlay Network.

VNS3 Administration Document
Covers the administration and operation of a configured VNS3 Controller. Additional detail is provided around the VNS3 Firewall, all administration menu items, upgrade licenses, other routes and SNMP traps.

VNS3 Troubleshooting
Troubleshooting document that provides explanation issues that are more commonly experienced with VNS3.