



Overlay Engine

VNS3 Plugins Guide 2018

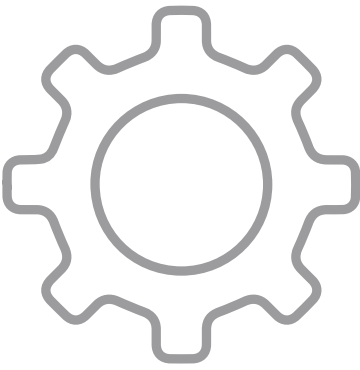
Table of Contents

Introduction	<u>3</u>
Overlay Engine Detail	<u>7</u>
Running the Overlay Engine Plugin	<u>12</u>
Overlay Engine Best Practices	<u>20</u>
Restrictions/Limitations	<u>22</u>
Resources	24

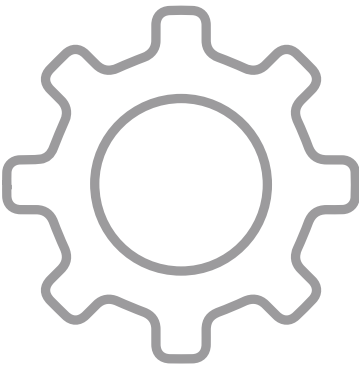
Introduction

VNS3:turret provides container based network services

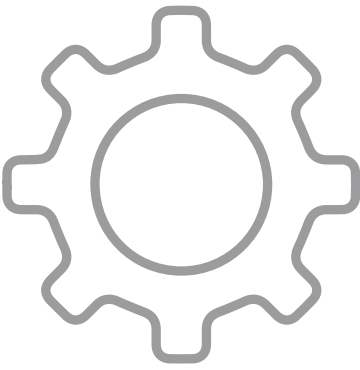
Isolated Linux containers within VNS3 allow partners and customers to embed features and functions safely and securely into their cloud network.



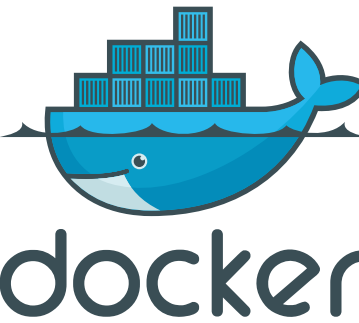
waf



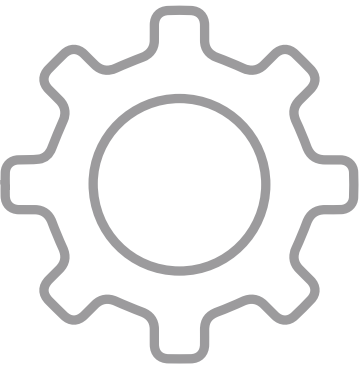
content caching



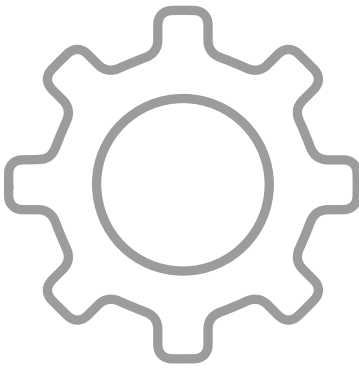
nids



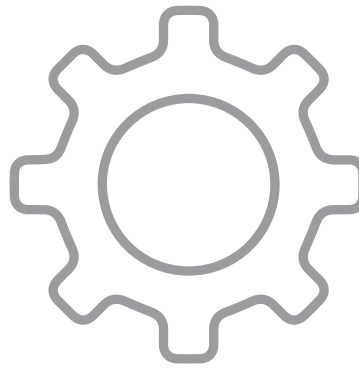
docker



proxy

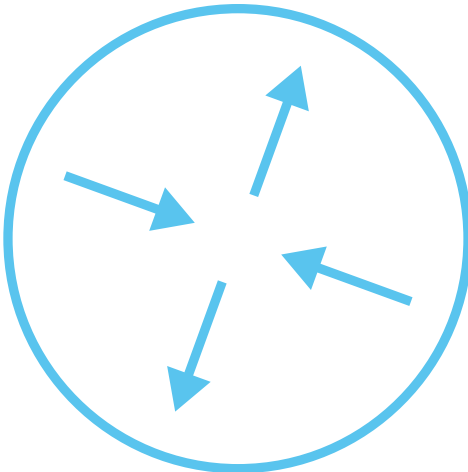


load balancing

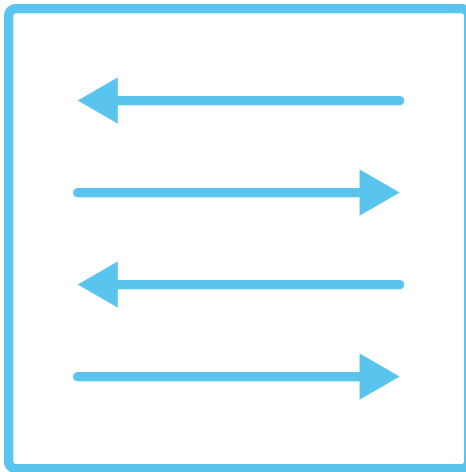


custom

VNS3 Core Components



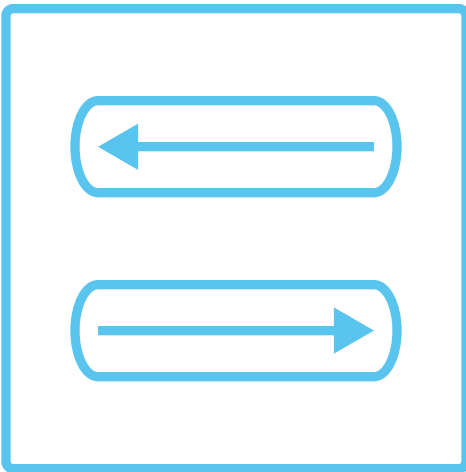
router



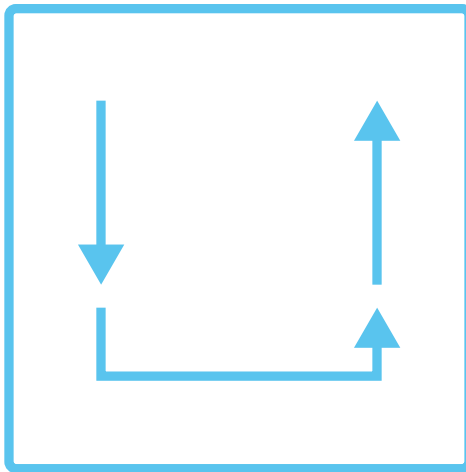
switch



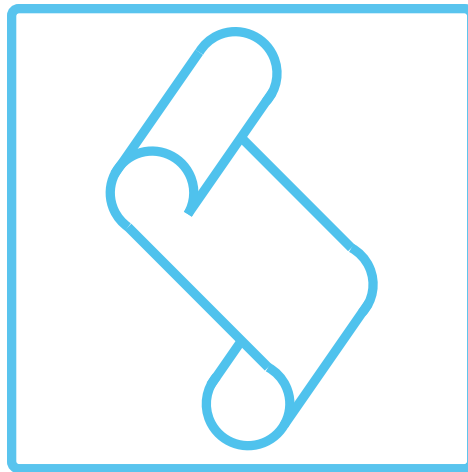
firewall



vpn concentrator



protocol redistributor



extensible nfv

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.

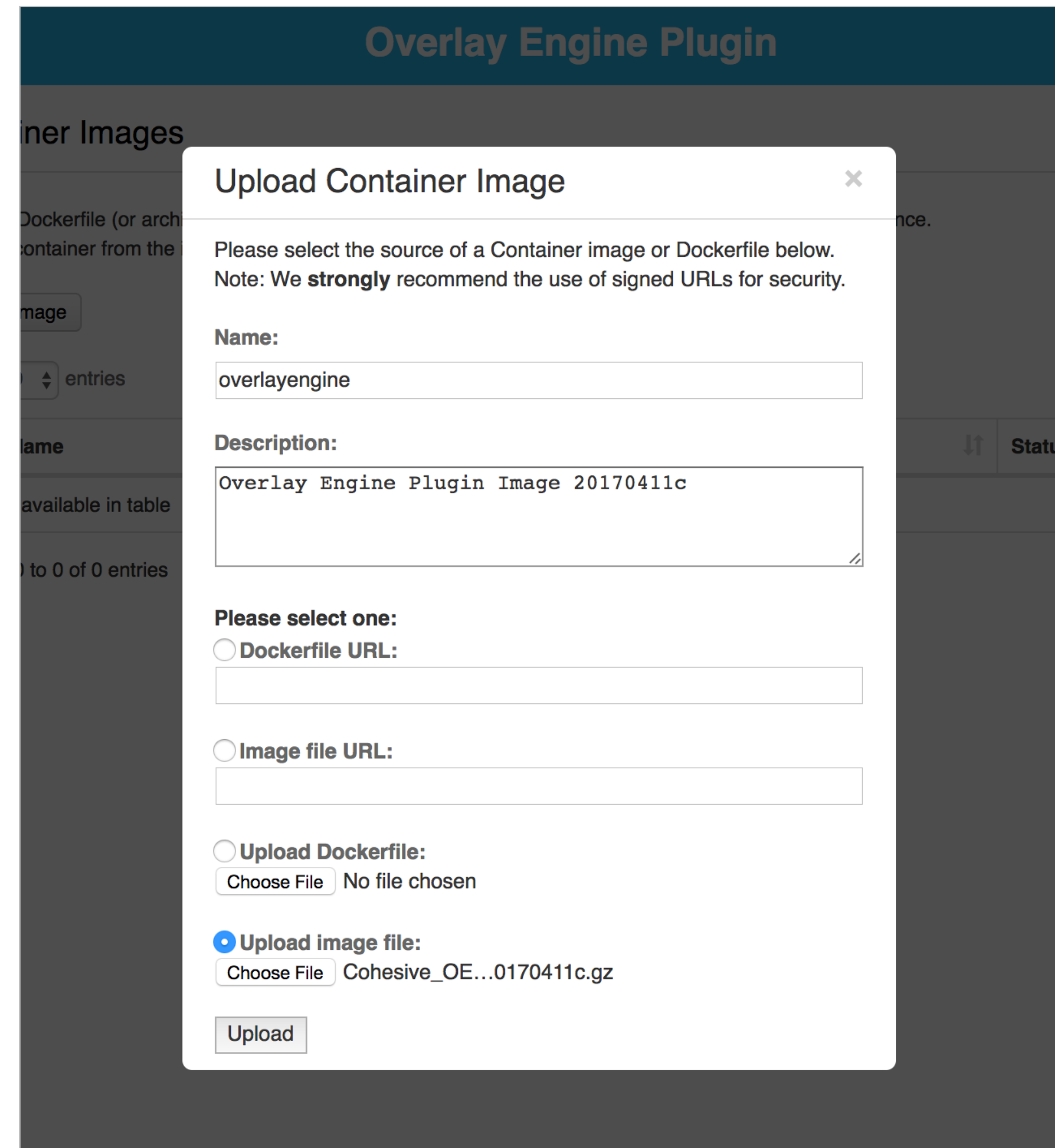
This URL can be used directly in a VNS3 Controller via the Web UI or API to import the container for use into that controller. (General screenshot walkthrough and help available in the plug-in configuration document.)

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.



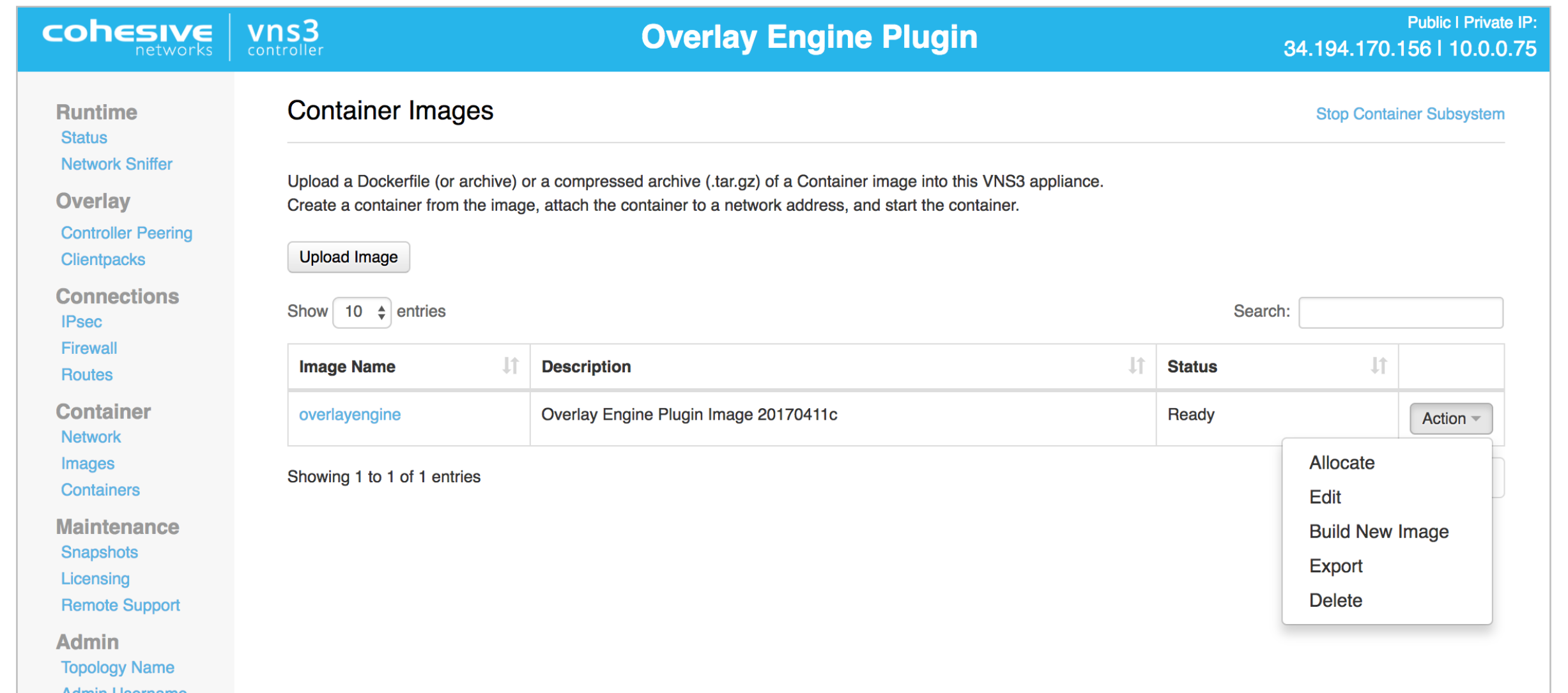
The screenshot shows the 'Overlay Engine Plugin' interface with a modal dialog titled 'Upload Container Image'. The dialog contains the following fields and options:

- Name:** A text input field containing 'overlayengine'.
- Description:** A text area containing 'Overlay Engine Plugin Image 20170411c'.
- Please select one:** A group of radio buttons with the following options:
 - Dockerfile URL: (with an empty text input field below it)
 - Image file URL: (with an empty text input field below it)
 - Upload Dockerfile: (with a 'Choose File' button and the text 'No file chosen')
 - Upload image file: (with a 'Choose File' button and the filename 'Cohesive_OE...0170411c.gz')
- Upload:** A button at the bottom of the dialog.

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.



The screenshot shows the VNS3 controller interface for the Overlay Engine Plugin. The page title is "Overlay Engine Plugin" and the status is "Ready". The interface includes a sidebar with navigation options: Runtime (Status, Network Sniffer), Overlay (Controller Peering, Clientpacks), Connections (IPsec, Firewall, Routes), Container (Network, Images, Containers), Maintenance (Snapshots, Licensing, Remote Support), and Admin (Topology Name, Admin Learnings). The main content area is titled "Container Images" and contains an "Upload Image" button, a "Show 10 entries" dropdown, and a search box. A table lists the container images:

Image Name	Description	Status	Action
overlayengine	Overlay Engine Plugin Image 20170411c	Ready	Allocate Edit Build New Image Export Delete

Showing 1 to 1 of 1 entries

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`

from the image, attach the container to a network address, and start the container.

Allocate Container

Name:
oe_1

Command:
/usr/bin/supervisord

Description:

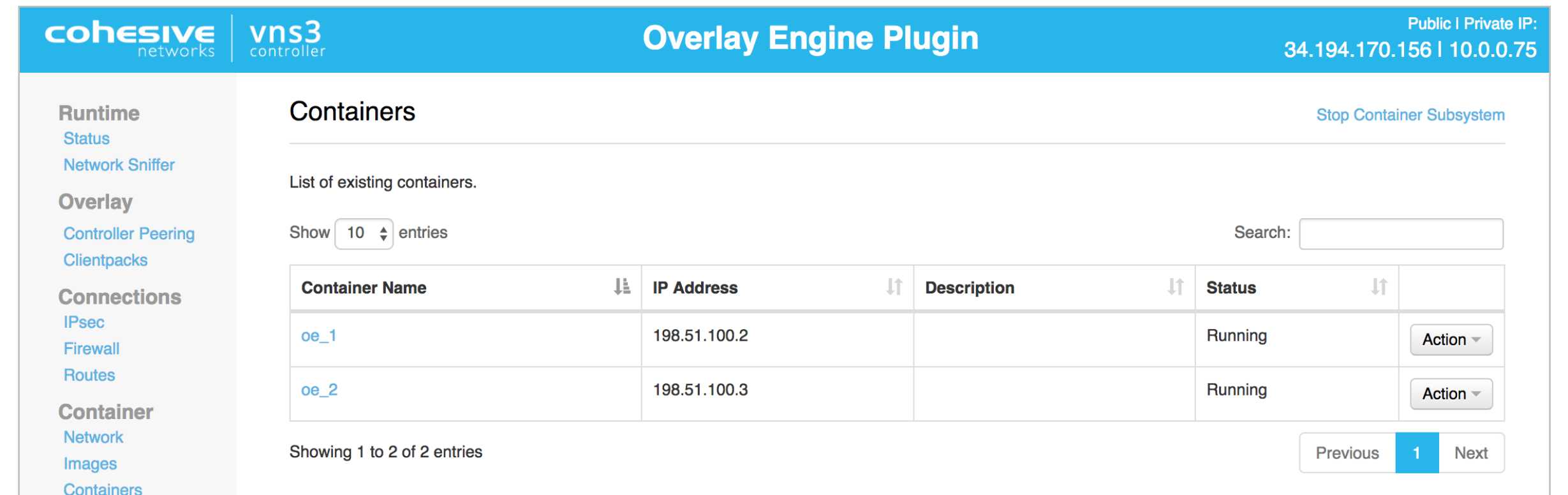
Available IP Addresses:
198.51.100.2

Allocate

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).



The screenshot shows the vns3 controller interface for the Overlay Engine Plugin. The page title is "Overlay Engine Plugin" and it displays the public and private IP addresses: 34.194.170.156 and 10.0.0.75. The left sidebar contains navigation options: Runtime (Status, Network Sniffer), Overlay (Controller Peering, Clientpacks), Connections (IPsec, Firewall, Routes), and Container (Network, Images, Containers). The main content area is titled "Containers" and includes a "Stop Container Subsystem" link. Below this, it says "List of existing containers." and "Show 10 entries" with a search box. A table lists two containers:

Container Name	IP Address	Description	Status	Action
oe_1	198.51.100.2		Running	Action
oe_2	198.51.100.3		Running	Action

At the bottom, it indicates "Showing 1 to 2 of 2 entries" and has "Previous", "1", and "Next" navigation buttons.

Overlay Engine Firewall Rules

Overlay Engines will need these 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 its 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
```

The screenshot shows the VNS3 controller interface for the Overlay Engine Plugin. The top navigation bar includes the Cohesive Networks logo, 'vns3 controller', and the title 'Overlay Engine Plugin'. On the right, it displays 'Public | Private IP: 34.194.170.156 | 10.0.0.75'. A left sidebar contains navigation menus for Runtime, Overlay, Connections, Container, Maintenance, and Admin. The main content area is titled 'Firewall' and indicates that a custom firewall is activated. It shows a table of current firewall rules and a text area for editing rules.

pkts	bytes	target	prot	opt	in	out	source	destination	
2583	334K	ACCEPT	all	--	plugin0	*	0.0.0.0/0	0.0.0.0/0	
1392	232K	ACCEPT	all	--	*	plugin0	0.0.0.0/0	0.0.0.0/0	
1	82	DNAT	udp	--	*	*	0.0.0.0/0	0.0.0.0/0	udp dpt:1193
1	82	DNAT	udp	--	eth0	*	0.0.0.0/0	0.0.0.0/0	udp dpt:1192
0	0	MASQUERADE	all	--	*	eth0	198.51.100.0/28	0.0.0.0/0	

Edit rules:

```
#oe_1 DNAT rule for inbound UDP 1193  
PREROUTING_CUST -p udp --dport 1193 -j DNAT --to 198.51.100.2:1194  
  
#oe_2 DNAT rule for inbound UDP 1192  
PREROUTING_CUST -i eth0 -p udp --dport 1192 -j DNAT --to 198.51.100.3:1194  
  
#Allow Plugin Network to use VNS3 as NAT to Internet  
POSTROUTING_CUST -o eth0 -s 198.51.100.0/28 -j MASQUERADE  
  
#Allow packets to/from Plugin Network to move through VNS3  
FORWARD_CUST -i plugin0 -j ACCEPT  
FORWARD_CUST -o plugin0 -j ACCEPT
```

Save and activate

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.

```
187 INSKAp7ZmWKBdQC+IWSA10qzLd8H57LZXS8bWnZQLX5SdrwyIvXdsqT0w8Hd5w
188 NbZyaw+c06hPV2QnyIpHQTb408DYrioxoEMk16DMAes6zpD6NMd3iFcptQK5dbid
189 BFTAsnZVpk/aoQ2M7vWYmJfcaAnLxe615oq8nflrItwLseZgTItVD6tVwQKBqQC2
190 4BuzDJq8AjY+e4x//1Zqpe5IwxUbQcho0KuN44GrLbJgiqN6pvib4Qt223peP0Fo
191 79vn0VB6i0srUXGDLI1tgR1e+TMlCVGtbvJvpQISq75pbAu9xFKWDUJ7Y3AiQHr3
192 EZxrKimmL6vdYTRzuSwW3jt+pN+Q/HX/0Gwwu5YBzwKBqFToIHJxAG5Y23qJ4ovs
193 dNnSee+14cdACngGM0LPRYKXE LUjsWiR7ahDHCLTrE8tS+hVUHhyyLzFbYphc/k0
194 4Ei4f7PrGK37iHLY5kv0iJnTe+7xMf22iG+jcrbxjEaFZmMv0l44V3HuU0LGDmTE
195 fYQ50btqsbSCRYHEdNk1YVr
196 -----END PRIVATE KEY-----
197 </key>
198
199 <tls-auth>
200 #
201 # 2048 bit OpenVPN static key
202 #
203 -----BEGIN OpenVPN Static key V1-----
204 1920a4cca6decaf647cb3dfef58992e8
205 790a70687fe28deef3f2c1b8f080de44
206 581e2b915fe2397b599ab34d7bb865d0
207 0e02a2b4b48096c653f59bb5be7fd834
208 37d032f59141fd15f3244122f161927d
209 d27e15d4169ce273d35e65fa75b50cda
210 a6d4602f47e83dd49554bbfb81147dd2
211 576a7b604553fbb6dd4baf4578823ef
212 12e95a5edb85c8ec0d0410395d1ab3ac
213 66c41b229b9cb2123675923c8c0a0a70
214 5cb13d734c36a8ad0f5231f6717a6592
215 8d5cba7171d74405b505ed45b0dc731d
216 b3e907afc2aa5dd380fc39a2c0cb1c
217 a47568c25bcf57a772d51378996c0431
218 7dab2b20273d0bf264d67d25b9699374
219 80bb7c4098bc5591ae64ef291f878fcc
220 -----END OpenVPN Static key V1-----
221 </tls-auth>
222
223 ### BEGIN POLICY INSERT ###
224
225
226 ### END POLICY INSERT ###
227
228 # add remote commands here
229
230 remote 10.0.0.75 1193
231
```

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.