cohesive networks

VNS3 4.0

Administration Guide

Table of Contents

Introduction

VNS3 is an Appliance as a Service that provides network security and connectivity - Security Appliance, Application Delivery Controller and Unified Thread Management all rolled into one - to your cloud-based applications.

SA
security appliance

+

ADC
application
delivery controller

+

UTM
unified threat
management

=

VNS3

This guide describes the basic concepts to initialize and configure a VNS3 controller or Mesh of controllers to provide Site-to-Site IPsec connectivity and VNS3 Overlay Network end-to-end encryption. Not all use-cases are covered in this document.

Getting Started with VNS3

Set up a cloud account at a public cloud provider. VNS3 is available in most public and private cloud formats including:

- **Public Clouds:** Amazon Web Services EC2, Amazon Web Services VPC, Microsoft Azure, CenturyLink Cloud, Google Compute Engine (GCE), Rackspace, IBM SoftLayer, ElasticHosts, Verizon Terremark vCloud Express, InterRoute, Abiquo
- **Private Clouds**: Openstack, Flexiant, Eucalyptus, Abiquo, HPE Helion, and more
- **Virtual Infrastructure**: VMware (all formats), Citrix, Xen, KVM, and more

Familiarize yourself with OpenVPN TLS client if you plan on using the encrypted VNS3 Overlay Network.

Familiarize yourself with your IPsec firewall/router network device if you plan on creating a site-to-site IPsec connection to your cloud application deployment via VNS3. VNS3 supports most IPsec data center solutions including:

- **Preferred**: Most models from Cisco Systems*, Juniper, Watchguard, Dell SONICWALL, Netgear, Fortinet, Barracuda Networks, Check Point*, Zyxel USA, McAfee Retail, Citrix Systems, Hewlett Packard, D-Link, WatchGuard, Palo Alto Networks, OpenSwan, pfSense, and Vyatta.
- **Best Effort**: Any IPsec device that supports: IKE1 or IKE2, AES256 or AES128 or 3DES, SHA1 or MD5, and most importantly NAT-Traversal standards.
- **Known Exclusions**: Checkpoint R65+ requires native IPSec connections as Checkpoint does not conform to NAT-Traversal Standards and Cisco ASA 8.4(2)-8.4(any) bugs prevent a stable connection from being maintained.

Getting Help with VNS3

Support for VNS3 is provided through the Cohesive Networks Support Site according to our Support Plans.

We recommend reviewing the Support Site FAQs and this document before opening a support ticket.

If you need more information on how to setup a specific cloud environment or prefer video instructions, please see our Product Resources page for additional links.

If you need specific help with project planning, POCs, or audits, contact our professional services team via sales@cohesive.net for details.

Firewall Considerations

The VNS3 network appliance uses the following portsVNS3 Controller instances use the following TCP and UDP ports.

VNS3 Web UI/API - TCP port 8000
HTTPS admin interface; must be accessible from hosts where you will want to obtain runtime status or configure peering, also needs to be open to and from the Controllers at least for the peering process, and needs to be accessible when downloading credentials for installation on overlay network clients.

VNS3 encrypted Overlay Network - UDP port 1194
For client VPN connections; must be accessible from all servers that will join VNS3 topology as clients.

VNS3 Controller Mesh Peering - UDP 1195-1203*
For tunnels between Controller peers; must be accessible from all peers in a given topology.

IPsec Phase1/ISAKMP - UDP port 500
UDP port 500 is used the phase 1 or IKE (Internet Key Exchange) component of an IPsec VPN connection.

IPsec Phase2/ESP or NAT-Traversal - UDP port 4500 or Protocol 50 (ESP)
Protocol 50 is used for phase 2 or ESP (Encapsulated Security Payload) component of an IPsec VPN connection only when negotiating with native IPsec. UDP port 4500 is used for the phase 2 or ESP (Encapsulated Security Payload) component of an IPsec VPN connection when using NAT-Traversal Encapsulation.

*VNS3:vpn and VNS3:net Lite Edition will not require UDP ports 1195-1197 access as it is not licensed for Controller Peering.

** Some public cloud providers require IPsec connections to use NAT-Traversal encapsulation on UDP port 4500

Topology Name

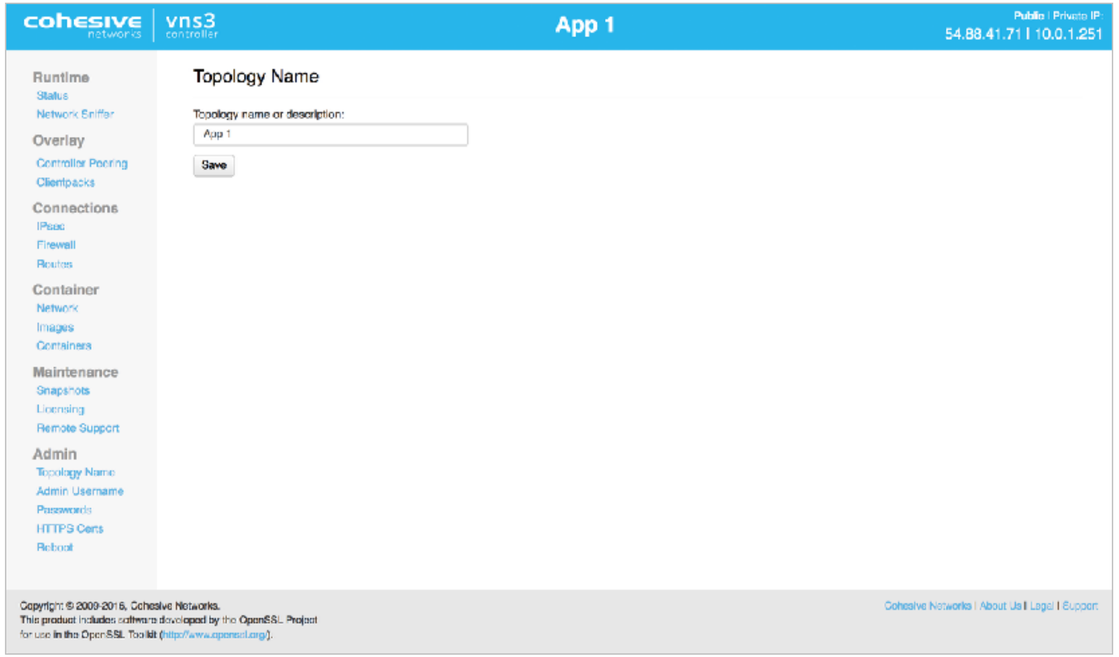
Changing the Topology Name

Topology Name is setup during keyset generation or when a keyset/configuration is fetched from a remote VNS3 Controller.

The Topology Name appears in the blue title header of the VNS3 Controller UI as well as in the meta title.

You can change the Topology Name at any time via the **Admin** left column menu heading and clicking **Topology Name**.

Enter in a new Topology Name and click **Save**.

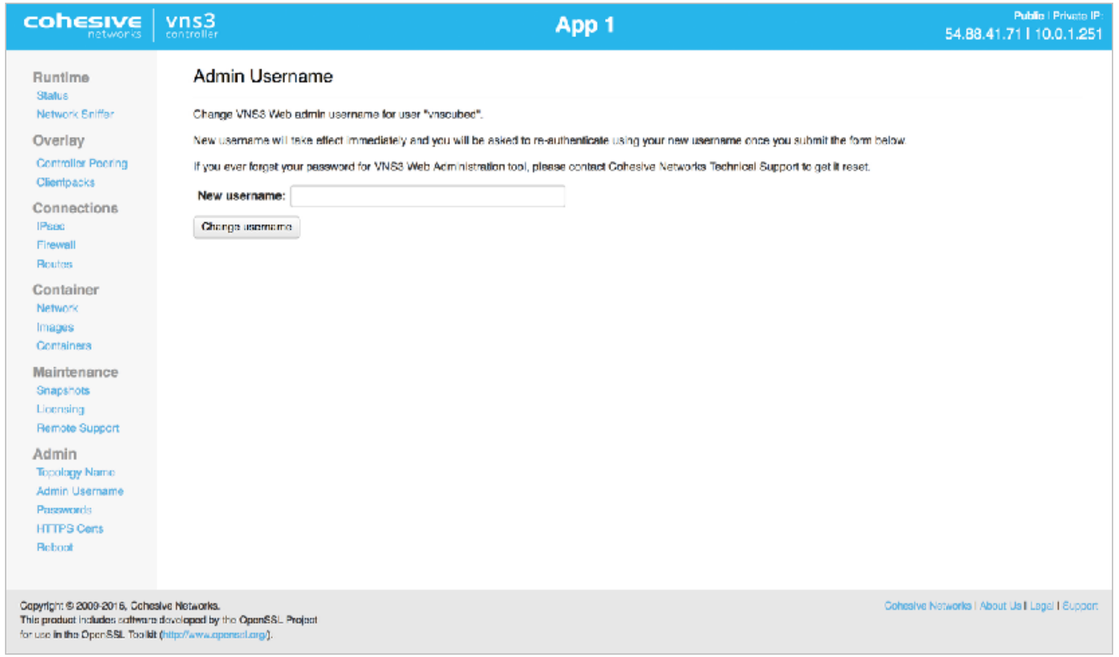


Change Admin Username

Change Admin Username

At the bottom of the left column menu, under **Admin** heading, click **Admin Username**.

Enter the new Admin Username and click **Change username**.

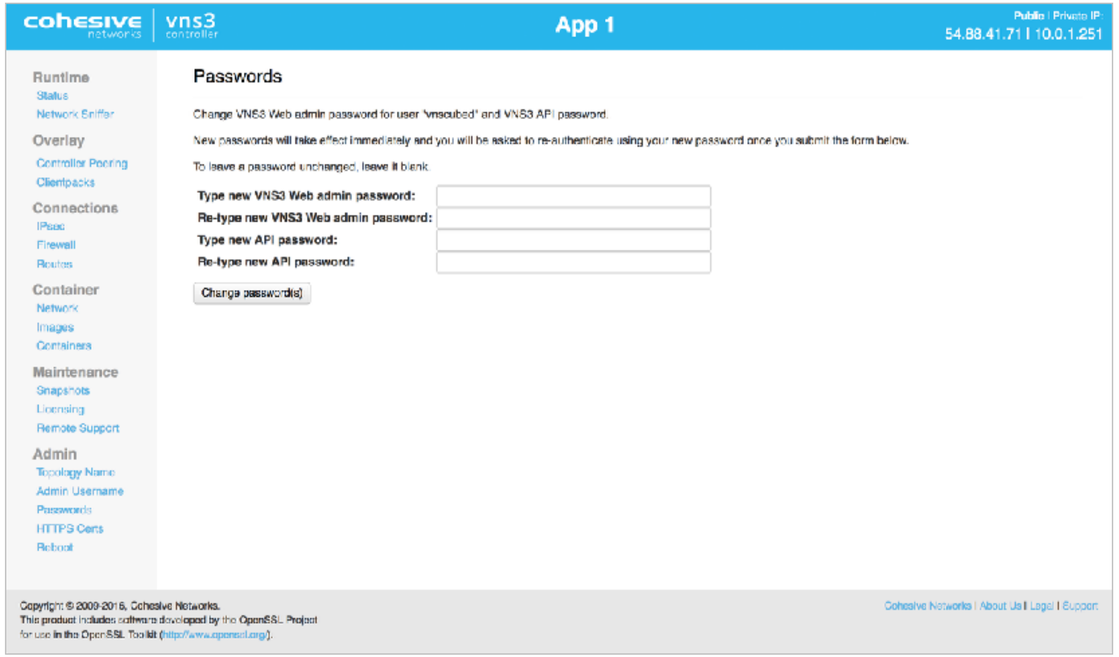


Change Password

Change Web UI and API Passwords

At the bottom of the left column menu, under **Admin** heading, click **Passwords**.

Enter the new Web UI password in the first two fields and/or the new API password in the last two fields and click **Change password(s)**.



HTTPS Certificates

HTTPS Certificate Upload

Before adding a custom SSL certificate to a VNS3 Controller, Cohesive Networks strongly recommends creating and downloading a VNS3 Snapshot from the Snapshots page. This VNS3 Snapshot backup can be used to re-instantiate the VNS3 Controller in the event the certificate/key pair creates an error (usually due to a mismatch or wrong files specified).

If you are unsure about the SSL Certificate files to upload, contact Cohesive Networks support staff to review.

Very simply stated, SSL Certificates for HTTPS interactions provide web users website ownership verification (so users can make sure they are interacting with an organization they intend) and encryption of communication to that website.

In order to be granted an SSL Certificate, a site owner needs to create a private key file (use later in the interactions between the web browser and the actual web server). This private key file is then used to generate a Certificate Signing Request (CSR) that is sent to a Certificate Authority (CA) like Verisign or Geotrust.

The Certificate Authority then sends back the SSL Certificate which includes information about the owner of the certificate, period of validity, URL that is certified and the ID of the CA, the public key used for encrypting communications and a hash to ensure the certificate is valid and not compromised.

The SSL Certificate along with the Private key used to generate the CSR are the two files required to add the Certificate to a VNS3 instance.

The next page reviews the relevant items and how to upload the SSL Certificate to VNS3.

HTTPS Certificate Upload

To order an SSL Certificate from a Certificate Authority you need to validate you are the owner of the specific URL you are certifying (typically via email validation or similar depending on the CA) and generate a CSR to send to the CA. The CA then uses the CSR to create the SSL Certificate.

To generate a CSR you first must create a Private Key. This document's example uses openssl. NOTE: VNS3 requires the private key to be an RSA key.

openssl genrsa -out vns3-example-com.key 2048

Once the private key is created, use it to generate the CSR with the following:

openssl req -new -sha256 -key vns3-example-com.key -out vns3-example-com.csr

The CA will send back one or multiple Certificates:

Root Certificate - typically not needed for VNS3

Intermediate Certificate - included if the CA is not a Root CA

End User Certificate - the certificate for the actual URL you plan on secure

Upload the SSL Cert or SSL Certificate Chain in the event your CA provided an intermediate Certificate (see example to the right) under the SSL Certificate file selection. General begins with "-----BEGIN CERTIFICATE-----".

Then upload the Private RSA key file used to generate the CSR under the SSL Key file section. Generally begins with "-----BEGIN RSA PRIVATE KEY-----".

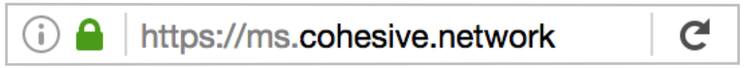
Click Upload and Install.

certificate-chain.crt

```
-----BEGIN CERTIFICATE-----
...
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
...
-----END CERTIFICATE-----
```

end-user certificate

intermediate certificate



Reset Factory Defaults

Reset Factory Defaults

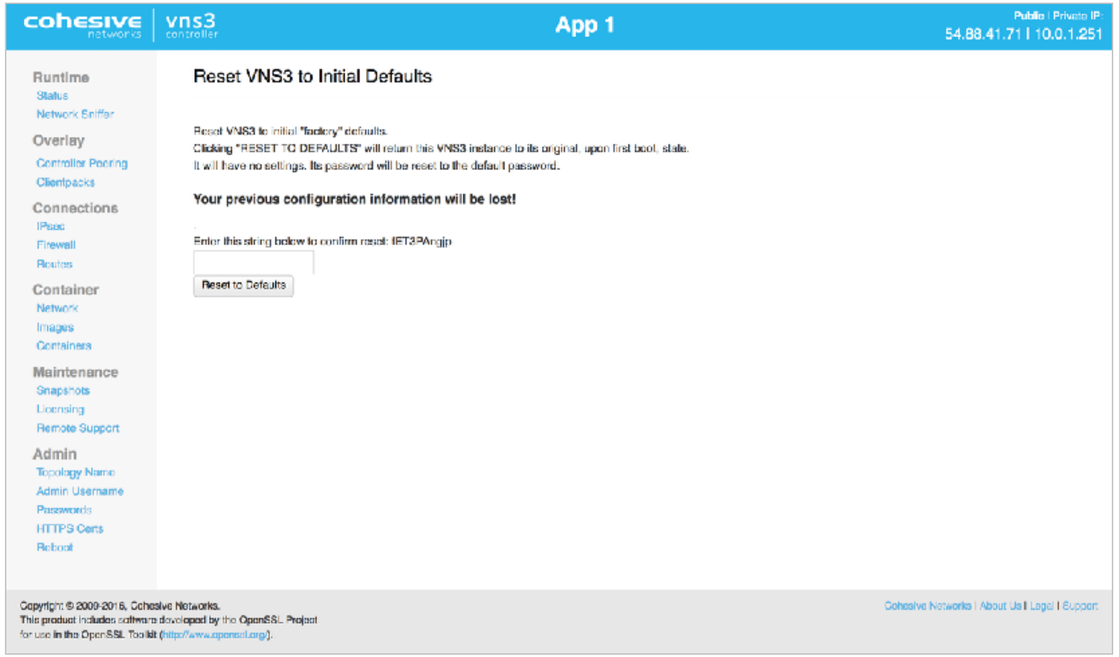
Increasingly there is a separation of duties between staff that can start/stop/reboot/terminate cloud instances and staff that configures and administers the VNS3 Controller device.

Reset Factory Defaults removes all configurations, licensing and settings on a particular VNS3 Controller instance. The only configuration parameter that will remain is the username and password (both UI and API) set on the Controller instance at the time of the reset operation.

To Reset Factory Defaults navigate to https://<managerIP>:8000/reset_defaults. This URL is not linked anywhere in the UI to eliminate the possibility of accidentally resetting a production server.

On the resulting page enter the code displayed to validate the reset and click Reset.

After Reboot, the Controller is reset and you can choose how to configure starting with Initialization.



VNS3 Snapshots

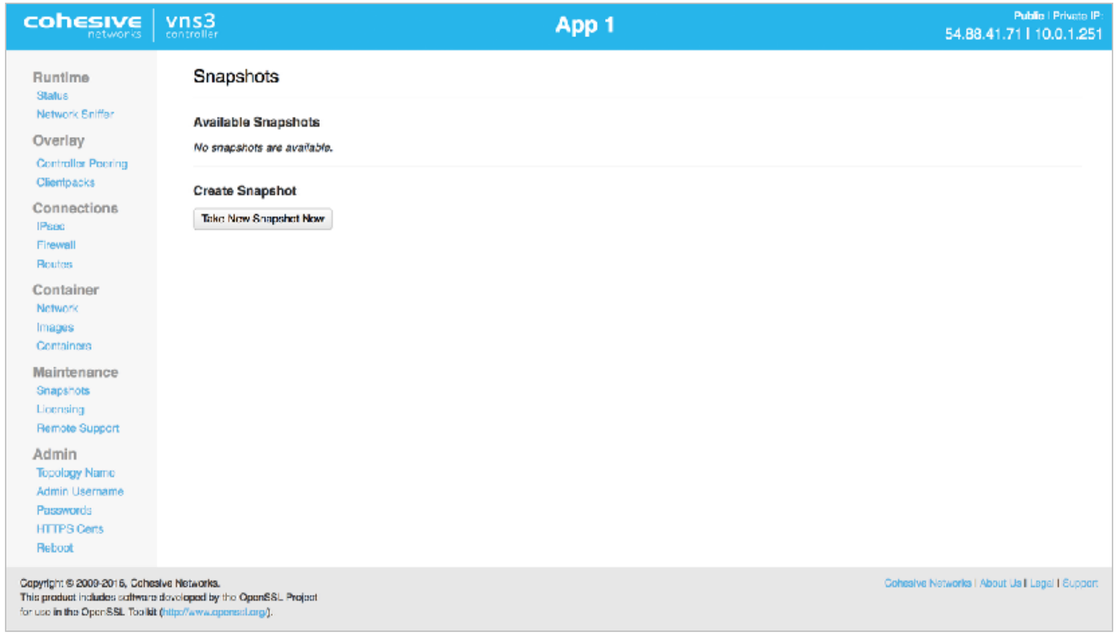
Runtime Snapshots save the Controller Configuration

Once your VNS3 Controllers and Clients are configured and running, save the configuration with Runtime Snapshots. Snapshots can be used to reconfigure a new Controller with the same SSL Certificates and Keyset with just one file upload.

Click **Snapshots** under Maintenance to take a new snapshot or view/download available snapshots.

Download the snapshot to your local network. In the event of a Controller failure or re-provisioning event, you can upload the snapshot file to a new VNS3 Controller. The new Controller will retain all the configuration settings as your saved snapshot.

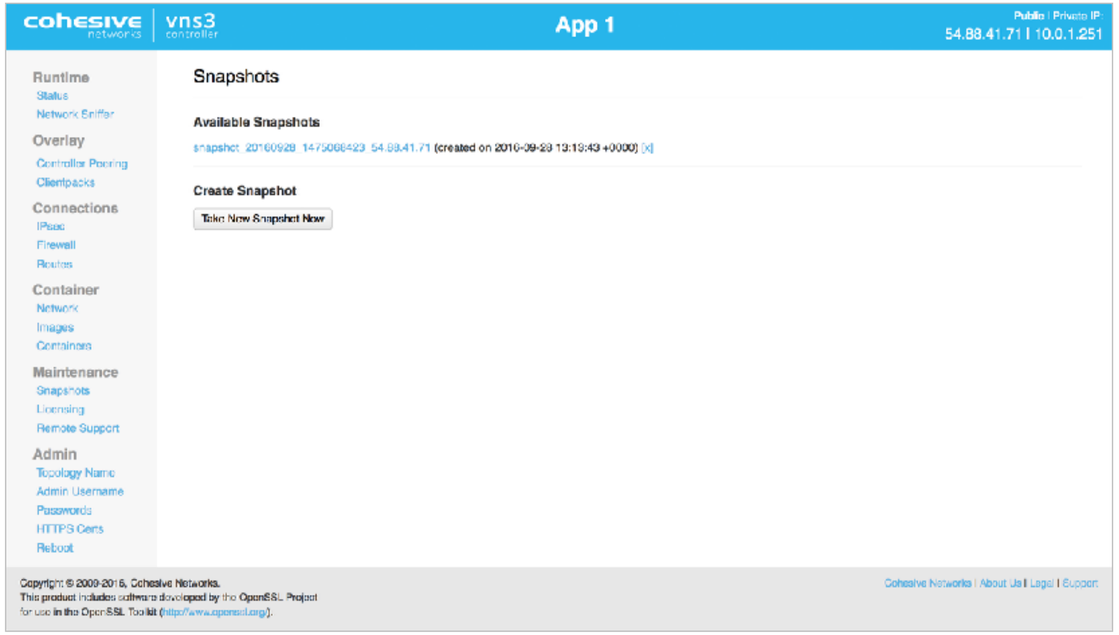
If you are using Elastic IPs, once the Elastic IP is transferred to the new Controller, your overlay network devices will automatically connect back with the Controllers. Save time on both Controller and client configuration.

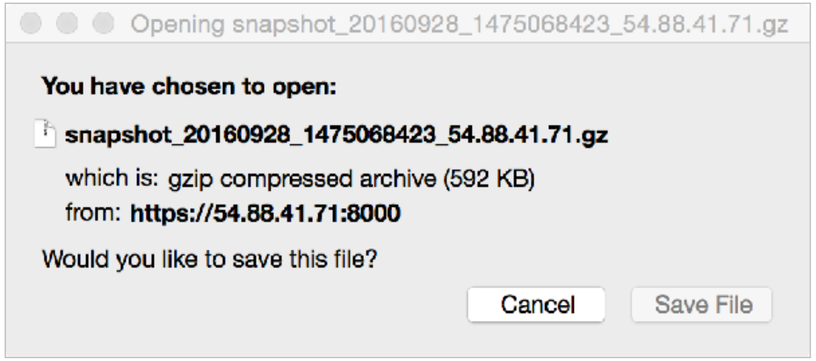


Save and Download a Snapshot

Click the “Take New Snapshot Now” button to generate a new Snapshot.

The resulting screen will have the snapshot download link. Download the Snapshot and save locally.





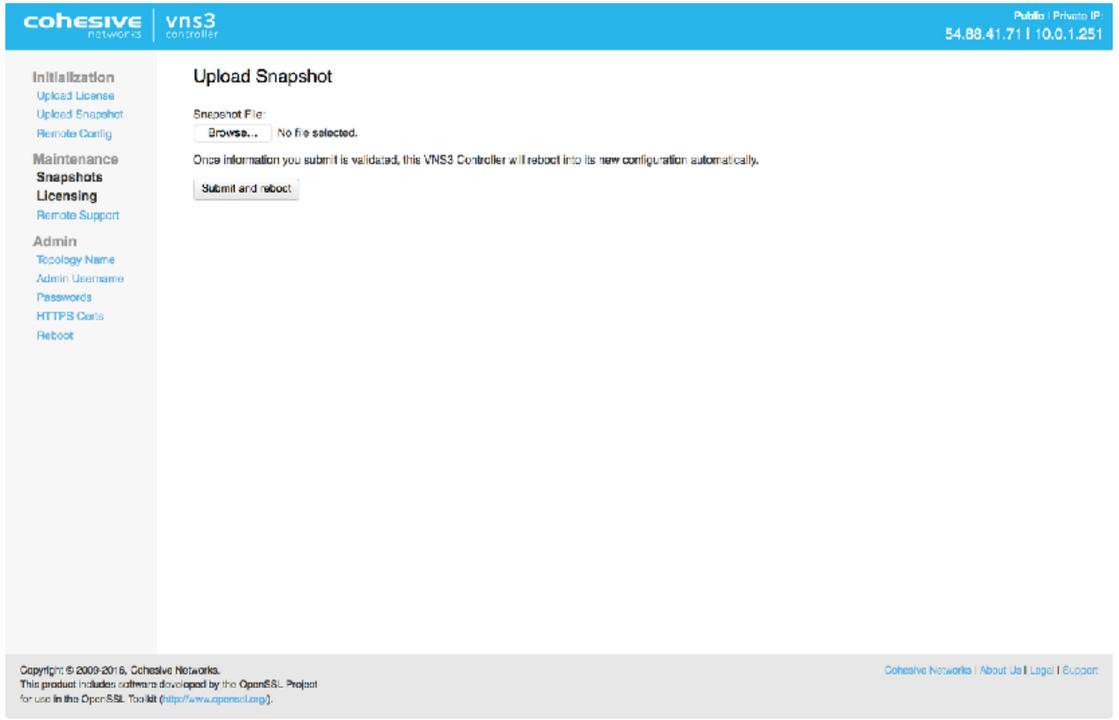
Save and Download a Snapshot

To use a Snapshot to configure a Controller click the **Import Runtime Snapshot** link.

Browse for your saved Snapshot and upload. The Controller will reboot with the updated configuration. The same Clientpacks will be available in the Controller, so redistribution to each server on the virtual network is not necessary.

A slight configuration change on each server on the virtual network is necessary if you have not assigned Elastic IPs to your Controller. The OpenVPN configuration file (vnscubed.<conf ovpn>) on each server needs the new IP of the new Controller referenced in the remote commands section.

To automate this step, you can assign an Elastic IP (see AWS billing for rates) to the Controller and reference the Elastic IP in each server's OpenVPN configuration file.



Upgrade License

VNS3 Upgrade License

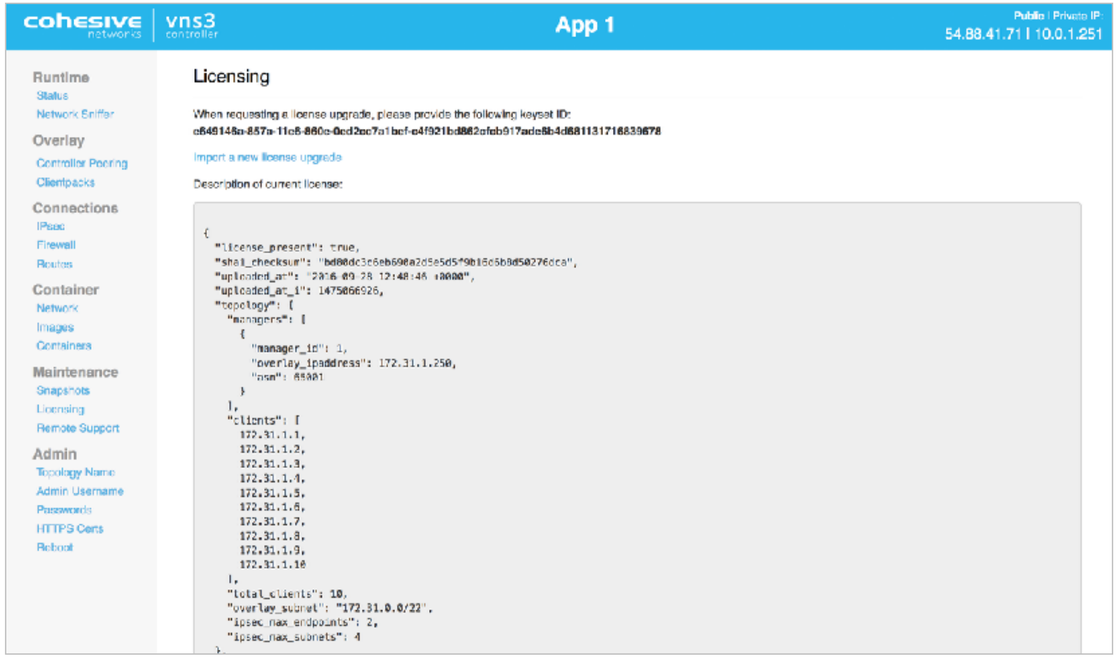
VNS3 controllers can be upgraded live without the need for an operational window.

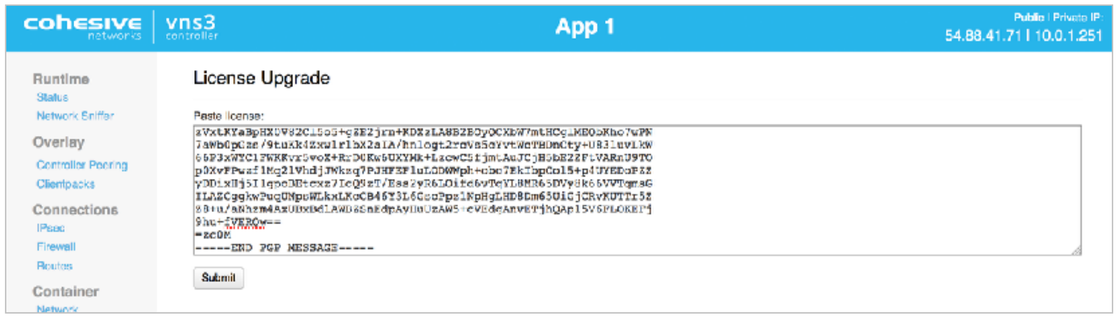
To upgrade a license click on the “License Upgrade” link in the left column of the Web User Interface.

A license upgrade needs to be deployed to all of the Controllers of a peered VNS3 topology.

In order to upgrade you will need the upgrade keyset ID which is shown in bold in the image displayed here. Provide that license keyset to Cohesive Support and they will use it to generate your upgrade license.

In order to apply the new license click “Import a new license upgrade”, and paste the contents of the license you received, and click “Submit”.





Confirming the successful license upgrade

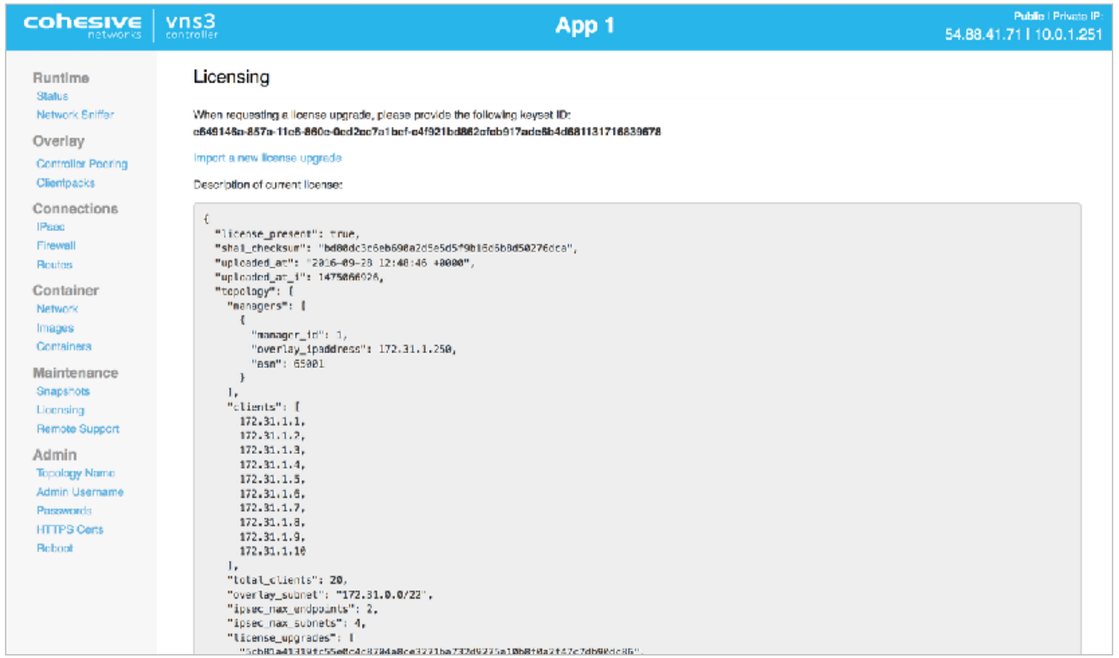
Along with the license upgrade key you will have received with the key a “License Upgrade ID”.

After clicking “Submit” your License Upgrade ID is displayed on the user interface. It should be the same as the one you received with your license upgrade.

If your license upgrade requires any new data such as IP Addresses to use for new Clientpacks, there will be an opportunity to enter it on the screen, and then you “finalize” the upgrade by hitting “Submit again.

You will then see the contents of the license displayed, and should confirm that your new license contents has the parameters you expected.

In this example case the license upgrade added the ability to have 2 more remote endpoint definitions for use with IPsec tunnels. Looking at the previous picture you will see the total of “ipsec_max_endpoints” was 50, and after the upgrade is increased to 52.



Remote Support

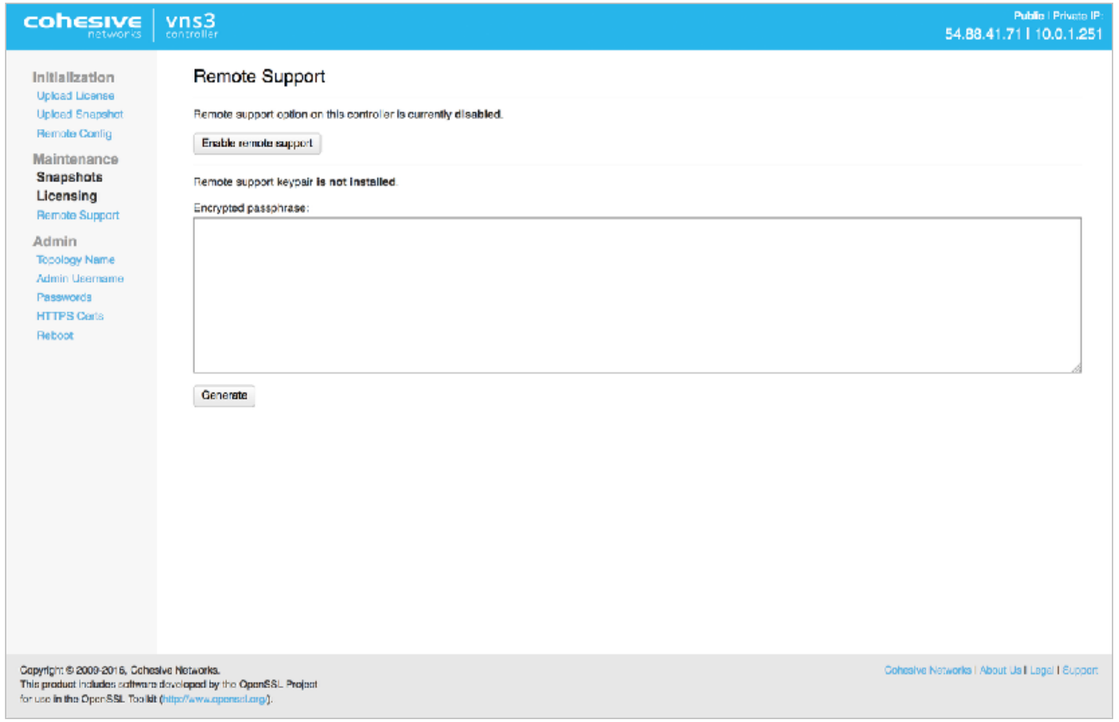
Remote Support

In the event Cohesive needs to observe runtime state of a VNS3 Controller in response to a tech support request, we will ask you to open Security Group access to TCP port 22 (SSH) from our support IP, 54.236.197.84, and Enable Remote Support via the Web UI.

Note that TCP 22 (ssh) is not required for normal operations.

Each VNS3 Controller is running a restricted SSH daemon, with access limited only to Cohesive for debugging purposes controlled by the user via the Remote Support toggle and key exchange generation.

Cohesive will send you an encrypted passphrase to generate a private key used by Cohesive Support staff to access your Controller. Access to the restricted SSH daemon is completely controlled by the user. Once the support ticket has been closed you can disable remote support access and invalidate the access key.



VNS3 Firewall

VNS3 Firewall Overview

The VNS3 Firewall adds a layer of security and control for cloud-based deployments.

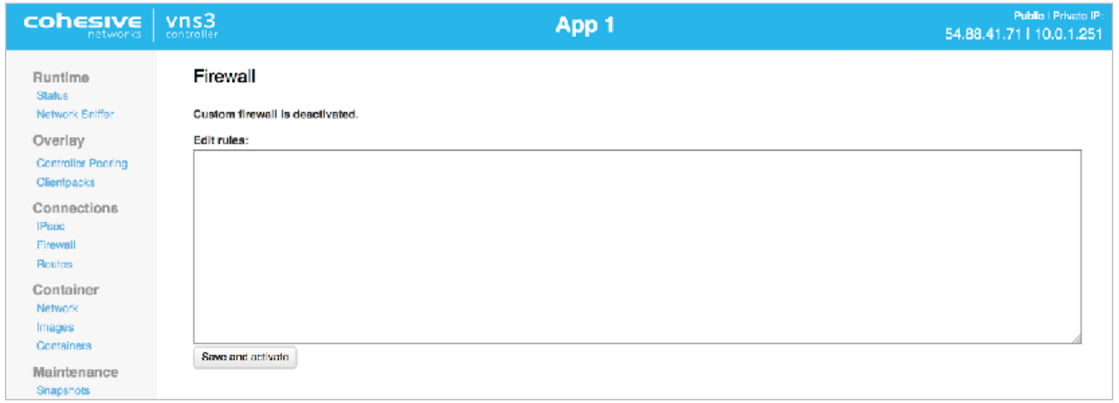
VNS3 Firewall features are controlled using IPTables syntax. For more information see - http://linux.die.net/man/8/iptables and look for the PARAMETERS section. Another useful guide is available here: http://www.thegeekstuff.com/2011/06/iptables-rules-examples/

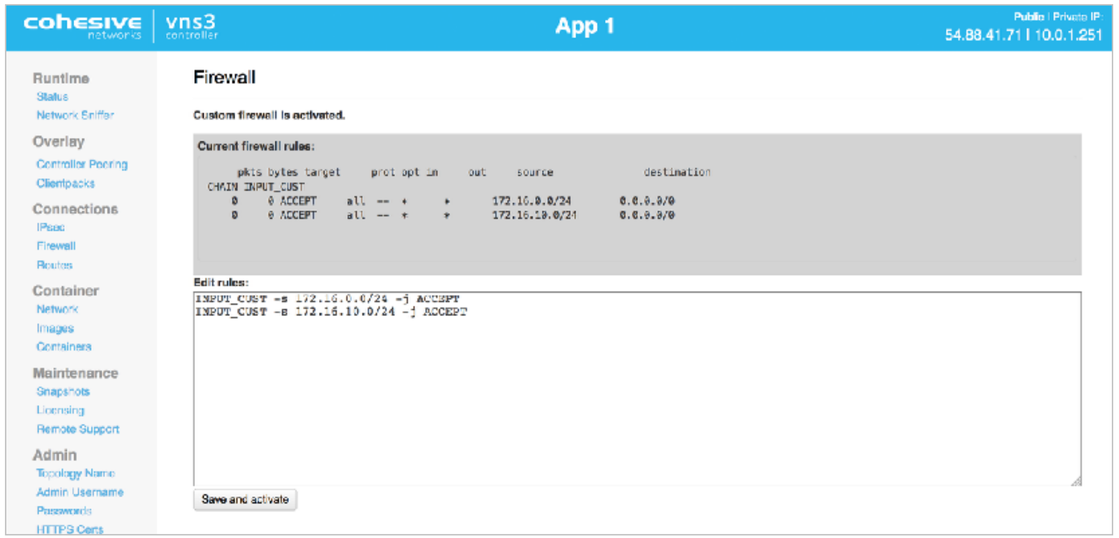
In general, you write a specification of a packet to match and then specify what to do with this packet. These are referred to as "customer" rules and are applied as appropriate in the overall firewall rule structure on the Controller. This means in addition to the standard security and firewall features of VNS3, you can create your own rules to restrict traffic to and through the VNS3 Controller.

The order of rules matter - rules are applied from top to bottom until the first match. If no match is found, the packet is allowed to continue on. Important note: If your customer rules don't reject a packet, it will be allowed by default.

However, this "default" is fairly restrictive. Traffic is allowed from "known" VLANS. Known VLANs are VLANS that are listed in IPSec tunnel rules, and the VNS3 virtual VLAN. Allowing traffic from other sources requires adding firewall rules to accept that traffic

NOTE: Firewall rules added to a specific controller are not automatically synced to other controllers in the peered mesh.





VNS3 Firewall - Basic Syntax

VNS3 creates a firewall table for your rules which is implicitly used by any firewall commands you enter.

As a result, the firewall syntax varies from standard iptables in that you don't specify an "append" or "-A". The chain names for your commands use restricted chain names of INPUT_CUST, OUTPUT_CUST, FORWARD_CUST, PREROUTING_CUST, and POSTROUTING_CUST.

These tables go into the top of the corresponding VNS3 internal firewall chains. The INPUT_CUST, OUTPUT_CUST, and FORWARD_CUST go into the iptables Filter table, and the PREROUTING_CUST and POSTROUTING_CUST go into the Nat table.

Cohesive also offers a chain MACRO_CUST which is a type of rule that combines some more complicated rule combinations spanning multiple chains and tables into a single rule.

For example the full IPTable syntax would be something like:

iptables -A INPUT_CUST -p tcp -s 172.31.1.1/32 --dport 8000 -m state --state NEW,ESTABLISHED -j DROP

In VNS3 enter the same elements without the "iptables" command and the "-A":

INPUT_CUST -p tcp -s 172.31.1.1/32 --dport 8000 -m state --state NEW,ESTABLISHED -j DROP

VNS3 Firewall Warning

The VNS3 firewall allows customers complete control of the INPUT, OUTPUT, FORWARDING, PREROUTING and POSTROUTING behavior of traffic as it first enters the VNS3 Controller and as it exits the VNS3 Controller.

The VNS3 internal firewall is still there to “protect” the internal mechanisms of VNS3, however, customer rules can be created that have undesirable effects. Essentially rules that ACCEPT or REJECT/DROP all traffic are likely to create a device that is un-reachable or one that is too permissive in accepting traffic.

Customer rules are evaluated and if there is not a match in the _CUST chains, then they flow through into the interior VNS3 chains which are quite restrictive. Accepting all traffic prevents most of the interior rules from being evaluated which might block unsafe traffic. Blocking all traffic prevents most of the interior rules from being evaluated which accept necessary traffic such as the API and WebUI management utilities. (Blocking port 8000 from all traffic will make the VNS3 instance un-manageable.)

Do not have rules of either of the following forms:

INPUT_CUST --dport 8000 -j REJECT

INPUT_CUST -j REJECT

INPUT_CUST -j ACCEPT

VNS3 Firewall - Examples

"-j ACCEPT" allows a packet. "-j DROP" drops a packet. "-j REJECT" sends an appropriate notification to sender saying such and such packet was rejected (depends on protocol).

Some Basic examples:

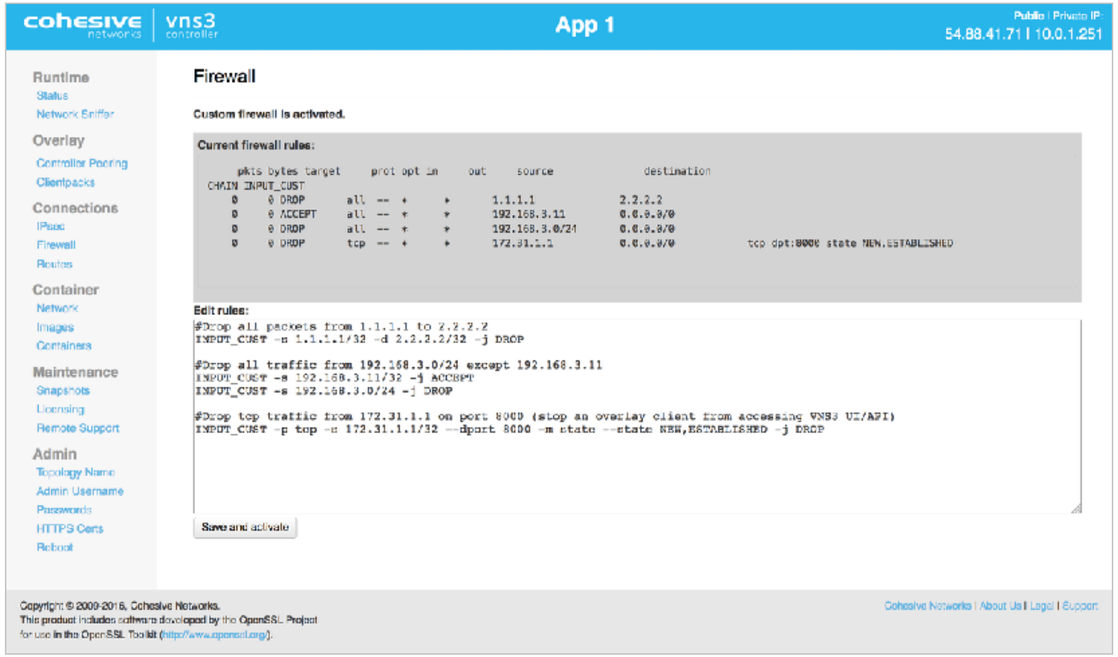
* Drop all packets from 1.1.1.1 to 2.2.2.2
INPUT_CUST -s 1.1.1.1 -d 2.2.2.2 -j DROP

* Drop all traffic from 192.168.3.0/24 (entire subnet) except 192.168.3.11:
INPUT_CUST -s 192.168.3.11/32 -j ACCEPT
INPUT_CUST -s 192.168.3.0/24 -j DROP

* Drop tcp traffic from 172.31.1.1 on port 8000 (Stop overlay clients from using the overlay IP of 172.31.1.1 with port 8000).

INPUT_CUST -p tcp -s 172.31.1.1/32 --dport 8000 -m state --state NEW,ESTABLISHED -j DROP

INPUT_CUST -p tcp --sport 8000 -m state --state ESTABLISHED -j DROP



VNS3 Firewall - NATing (network address translation)

It is now common for clouds to provide VLAN isolation. This provides a critical element of your overall security approach but creates the need for additional capabilities not needed in “plain old EC2”.

One of these is “NATing” which allows the machines in the VLAN to use the VNS3 Controller as a gateway to services on the Internet, with all VLAN machines sharing the Controller’s public IP address.

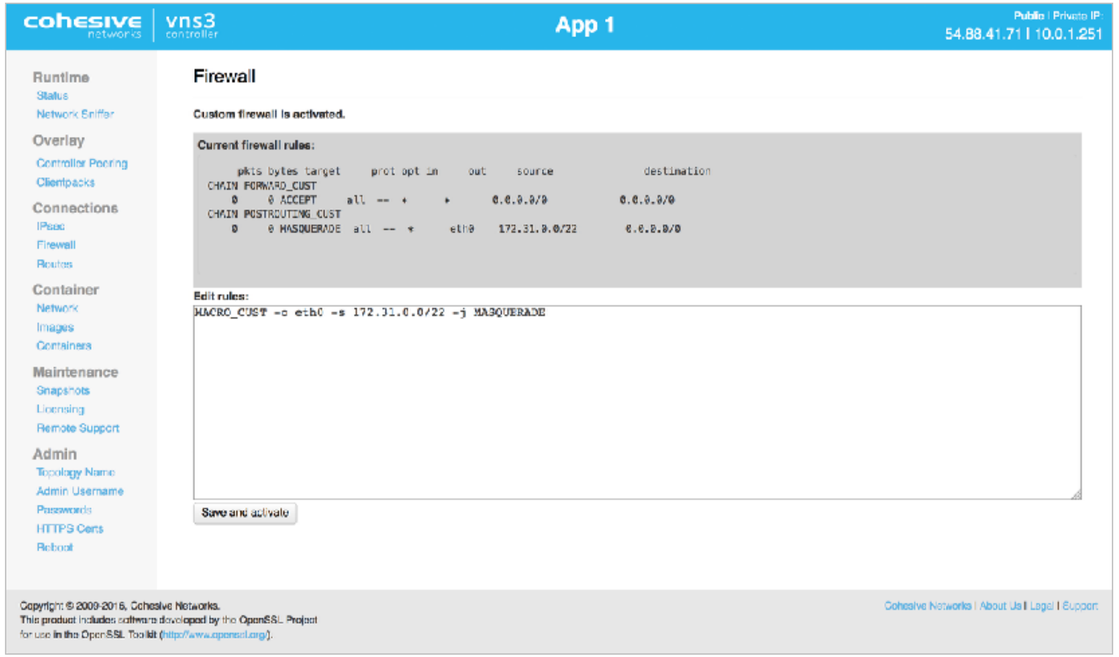
This is the same behavior used in your home or office, where many devices can access the Internet via one shared public ip address. When a VLAN device accesses the Internet, its return traffic is routed to it.

Basically, VNS3 lets you use your cloud VLAN just like you treat your home or office network, isolated from inbound requests for service, but allowing most outbound service requests.

Simple Syntax:
MACRO_CUST -o eth0 -s 172.31.1.0/24 -j MASQUERADE

In this example - your VNS3 Controller is in a VLAN subnet with a network from 172.31.1.0-172.31.1.255. Many clouds with VLAN capabilities map a public IP to the private IP on eth0 via DNS.

Here we are telling the VNS3 Controller to “masquerade” for traffic coming from that subnet out to the Internet and then return the response packets to the requesting machine.



VNS3 Firewall - Port Forwarding

With VPC, your cloud servers are not visible or accessible from the Internet unless you assign an Elastic IP, “putting” the server on the Internet.

What if you want to be able to access one of the machines (for example like you might do on your home network) from the Internet? This is where port forwarding comes in.

A common use case would be using a Windows Remote Desktop on one of your cloud servers, as the “jump” box for then remoting to all the other cloud servers in your VPC. VNS3 lets you do this with your VPC, just like you could for your home or office network, allowing specific traffic, from a specific source, on a specific port to be “forwarded” on to another machine.

Simple Syntax:
MACRO_CUST -o eth0 -s 10.199.1.0/24 -j MASQUERADE
PREROUTING_CUST -i eth0 -p tcp -s 61.61.70.70/32 --dport 3389 -j DNAT --to 10.199.130:3389

Using the same example network, assuming a source network public IP of 69.69.70.70 from which the RDP client is running, do the following:

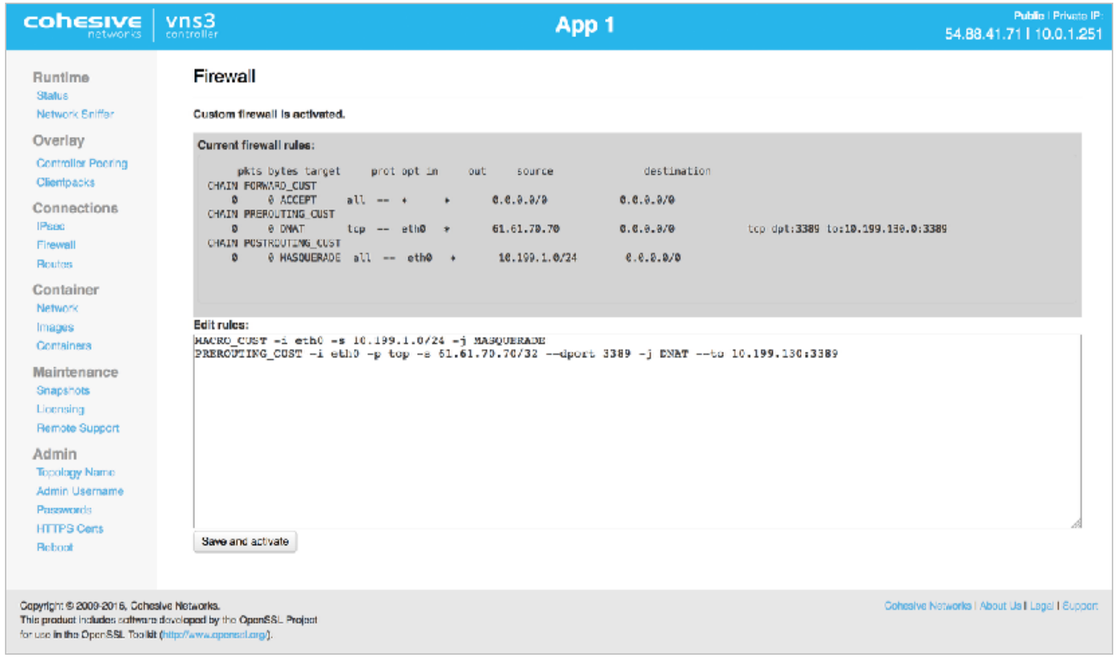
NATing needs to be enabled for port forwarding to work

Specify the port to be forwarded, in this case “RDP” or 3389

Specify the source network address, here 69.69.70.70/32

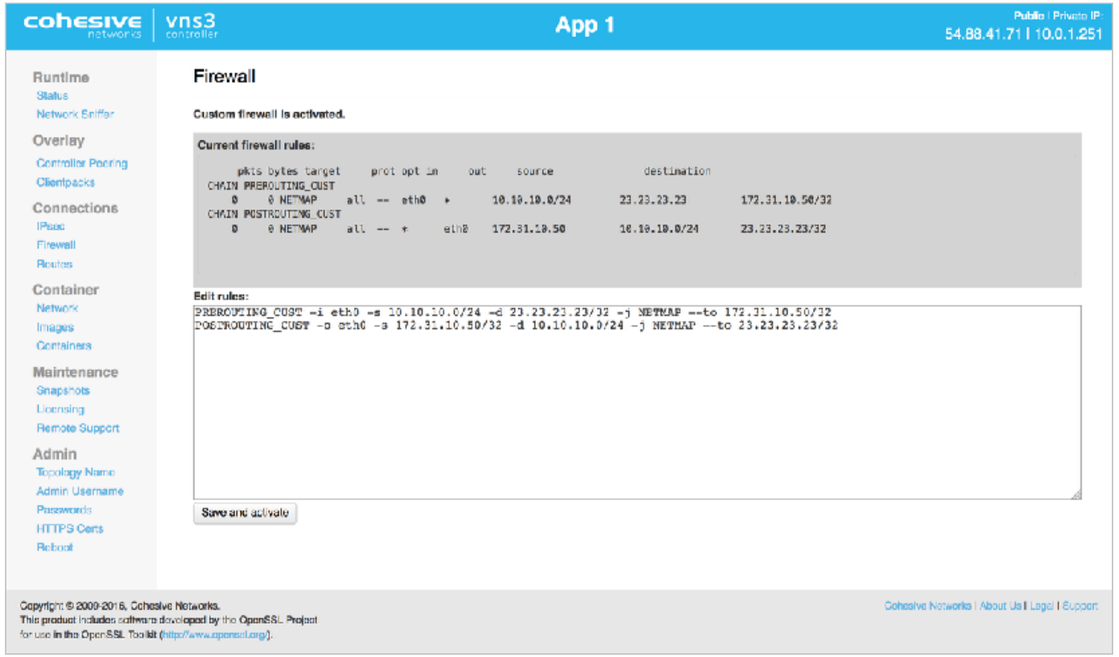
Specify the machine for port 3389 traffic, here 10.199.1.130 using the “--to” syntax

Use the “-j DNAT” syntax to specify destination network address translation.



VNS3 Firewall - Netmapping

Netmapping allows you to create IPsec tunnels to imaginary IPs on the VNS3 side of the connection and use the VNS3 firewall to map all traffic to/from the imaginary IP, to the actual host on your cloud side. This is extremely useful in situations where a connecting party has an address overlap with your Overlay or VLAN subnet.



Example

Remote subnet 10.10.10.0/24
VNS3 Overlay (clientpack network): 172.31.10.0/24
Local Server the Remote wants to access: 172.31.10.50
Customer will not connect their LAN (10.10.10.0/24) to a private network
Allocate an EIP to your account but DON'T associate: 23.23.23.23.
Build an IPsec the tunnel from 23.23.23.23/32 to 10.10.10.0/24

Simple Syntax:

PREROUTING_CUST -i eth0 -s 10.10.10.0/24 -d 23.23.23.23/32 -j NETMAP --to 172.31.10.50/32
POSTROUTING_CUST -o eth0 -s 172.31.10.50/32 -d 10.10.10.0/24 -j NETMAP --to 23.23.23.23/32

If the Local Subnet is a VLAN and not the Overlay Subnet add the following forward rule:
FORWARD_CUST -s 172.31.10.0/24 -d 10.10.10.0/24 -j ACCEPT
FORWARD_CUST -s 10.10.10.0/24 -d 172.31.10.0/24 -j ACCEPT

VNS3 Firewall - Copy Traffic to a Device

With the addition of the Docker-powered application container system there are scenarios where you might want to push a copy of the traffic from the eth0 or tun0 VNS3 interface to a particular IP. The obvious use-case is copying traffic to the Cohesive Utilities Container where you can do things like run tcpdump or iftop.

Example:
I want to copy all tun0 (Overlay Network) traffic to my Network Utils Container running on the VNS3 Controller on the Docker network at 172.0.10.2.

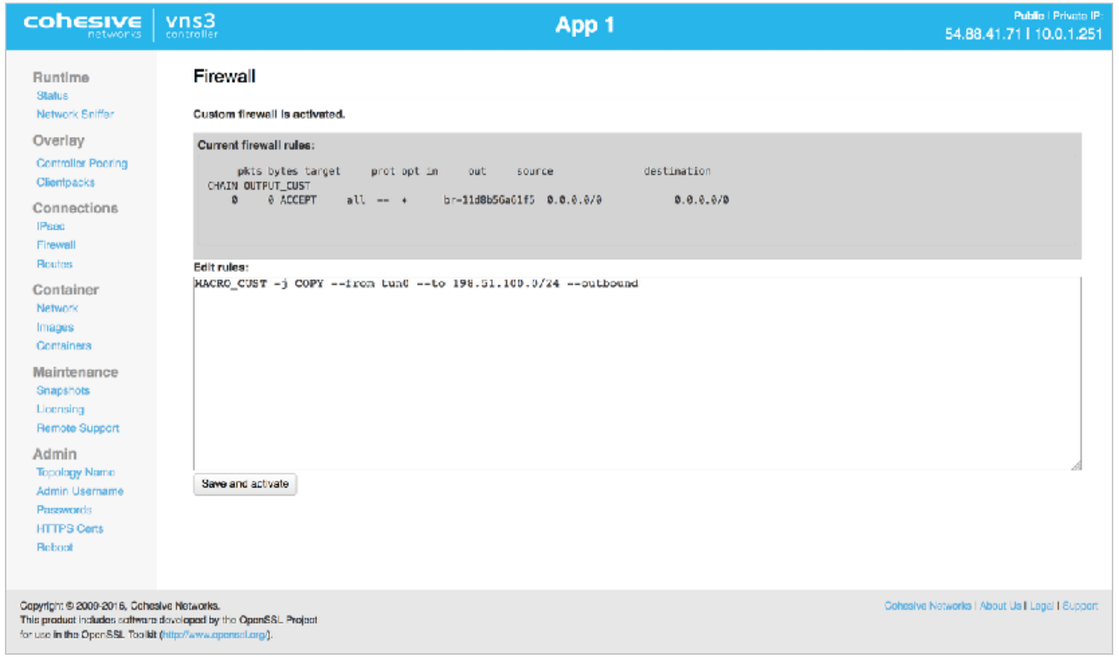
Simple Syntax:
#EXAMPLE: Copy all incoming tun0 (Overlay Network) traffic to the TCP Tools Container:

MACRO_CUST -j COPY --from tun0 --to <Container Network IP> --inbound

#EXAMPLE: Copy all outgoing tun0 (Overlay Network) traffic to the TCP Tools Container

MACRO_CUST -j COPY --from tun0 --to <Container Network IP> --outbound

NOTE: At this time analyze inbound OR outbound at any given time in order to prevent accidental traffic loops. It IS POSSIBLE to create a traffic cycle which could “brick” your controller if you create simultaneous inbound AND outbound rules with improper parameters.



VNS3 Routes

VNS3 Routes

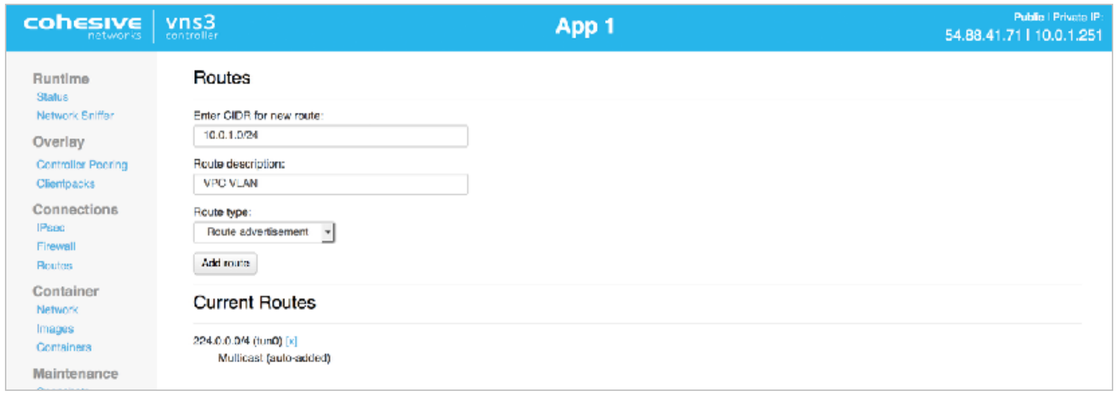
VNS3 allows you to configure Routes to allow a Controller to point to subnets not explicitly included in the IPsec or Overlay configurations. These routes can be automatically shared between other Controllers included in the topology via the Peering mechanism, and with Overlay Network connected devices. There are two types of routes:

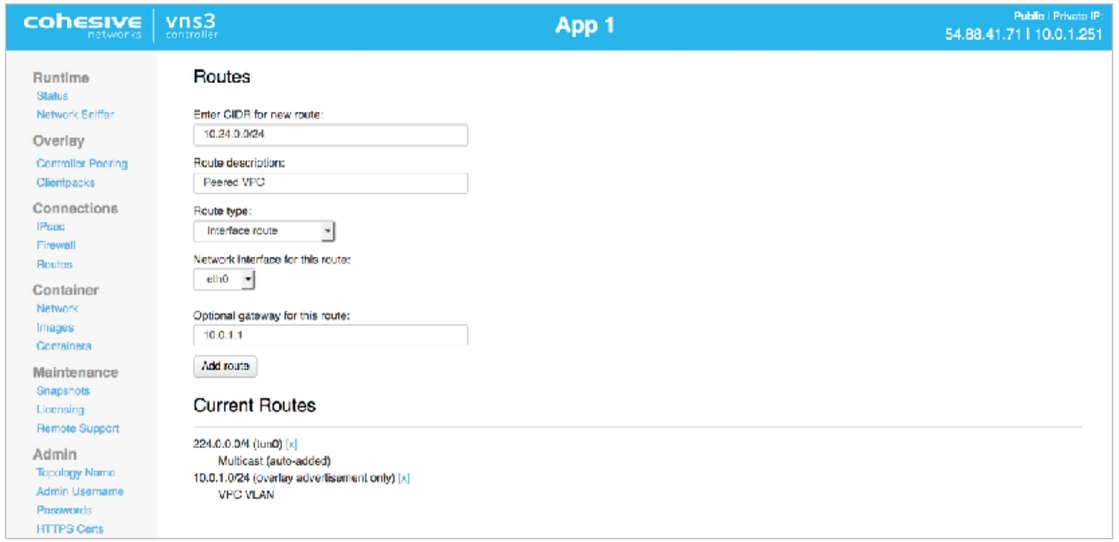
Route Advertisement

Simple route that tells all Overlay Network participants a certain CIDR destination is available through the VNS3 controller.

Interface Route

More complex route that also tells all Overlay Network participants a certain CIDR destination is available through the VNS3 controller but also allow configuration of the interface on the VNS3 controller where this is available and an optional Gateway IP.





VNS3 Routes

GRE Tunnel Route

GRE routes are a specialized subset of an "Interface Route". When there are route-based, site-to-site connections, established with GRE, those will be available for routes. Since the actual interface address of a GRE tunnel is not significant, the name of the tunnel which is routing encapsulated GRE traffic is shown. (In the picture on the left this is shown as "200net".)

Like "Interface Routes", the route can be advertised throughout the overlay by selecting the "Advertise route to Overlay" checkbox.

Routes to 0.0.0.0/0 (WARNING)

Making a route to 0.0.0.0/0

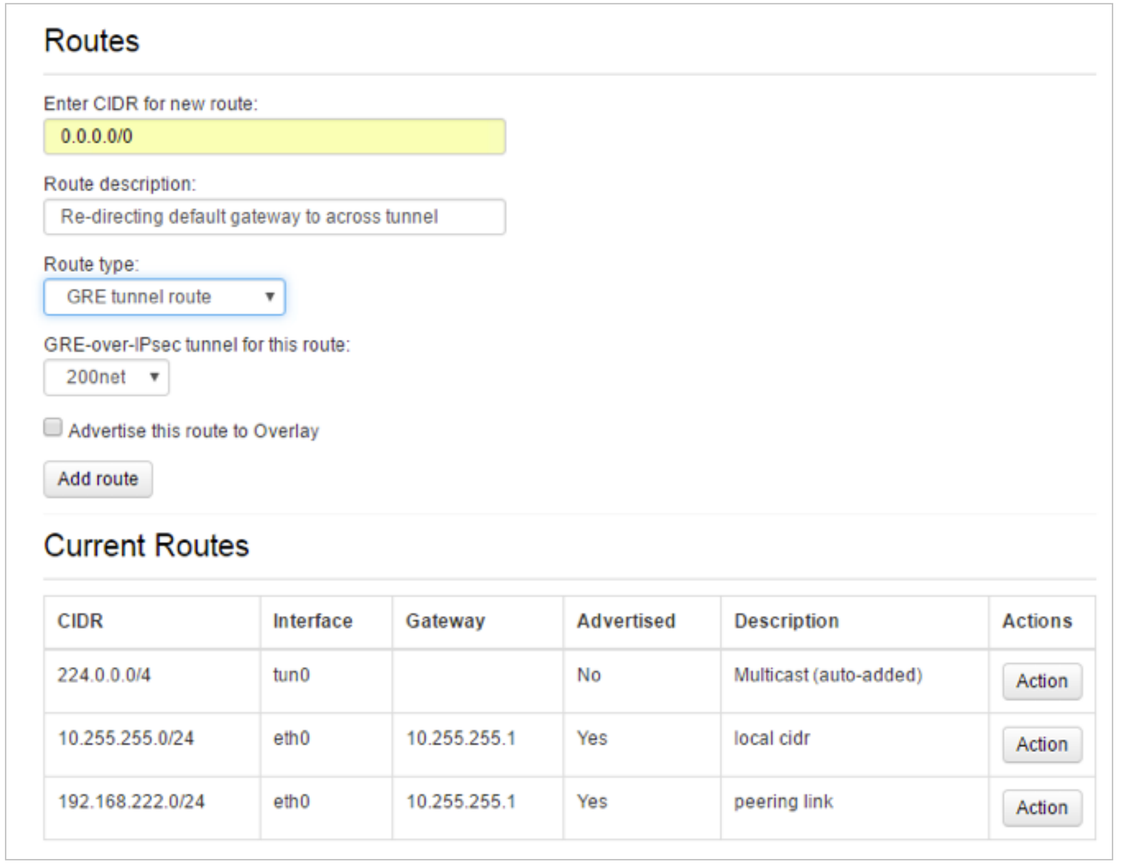
Sometimes it is necessary to make a route to 0.0.0.0/0.

If done improperly this can make your VNS3 Controller INACCESSIBLE and IRRECOVERABLE.

There are very solid and specific use-cases for this, but proper route sequencing must be followed.

In summary - you are replacing the VNS3 default gateway and telling it all traffic should go to that interface/gateway pair EXCEPT traffic to more specific routes.

To prevent losing access to your controller it is advised that you also have UI or API desktop to the controller from another host in the same specific subnet as the VNS3 controller.



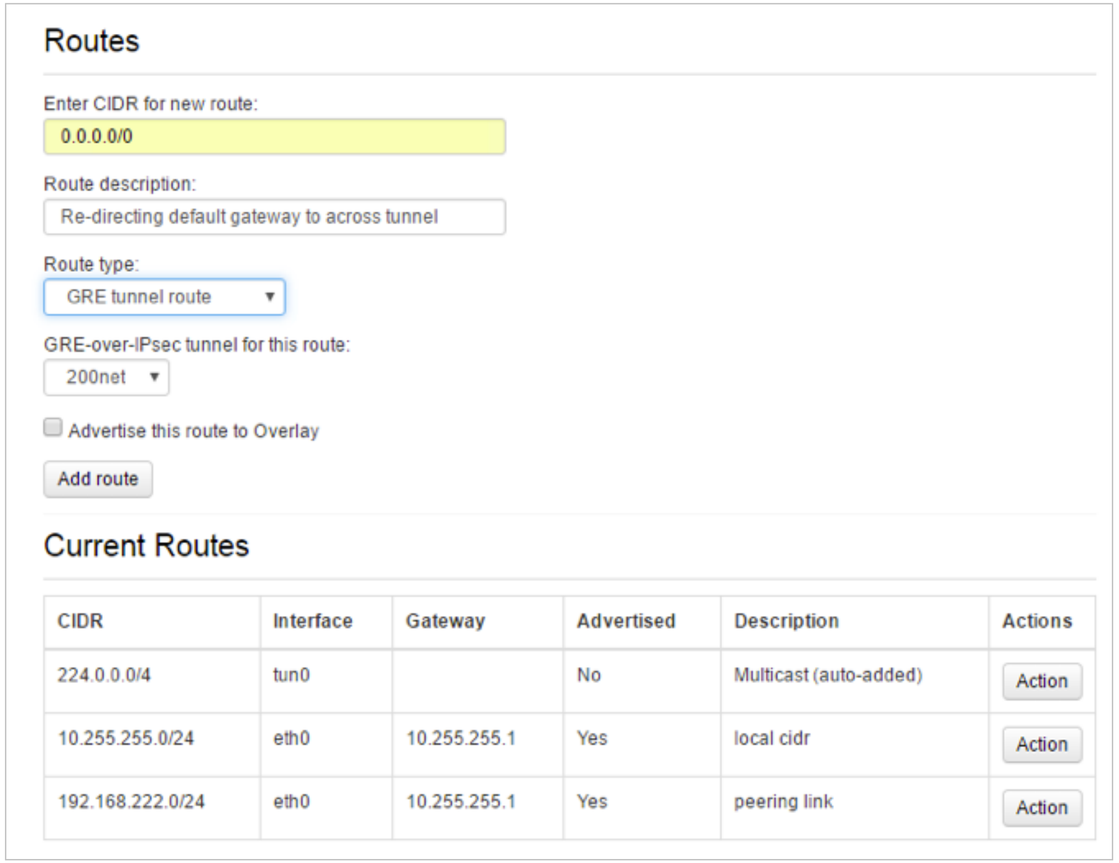
Adding route to 0.0.0.0/0 (WARNING)

In this example (from AWS, but basic approach required at Azure, Google, etc..), a 0.0.0.0/0 route is about to be entered.

BEFORE HITTING "Add Route"

- Ensure you have made an Interface Route to your VPC/ VNET/Subnet gateway. In the example this is 10.255.255.0/24 with a gateway of 10.255.255.1. (There are too many variables and distinctions for VNS3 to do this for you. You MUST provide the information!)

- If you have "peered" VPC/VNET/Subnets when you redirect the default gateway your peering via the cloud infrastructure will drop UNLESS you have a route(s) to the peered subnet(s) explicitly specified. In the example this is the Interface Route to 192.168.222.0/24 with the gateway of 10.255.255.1 (my VPC gateway).



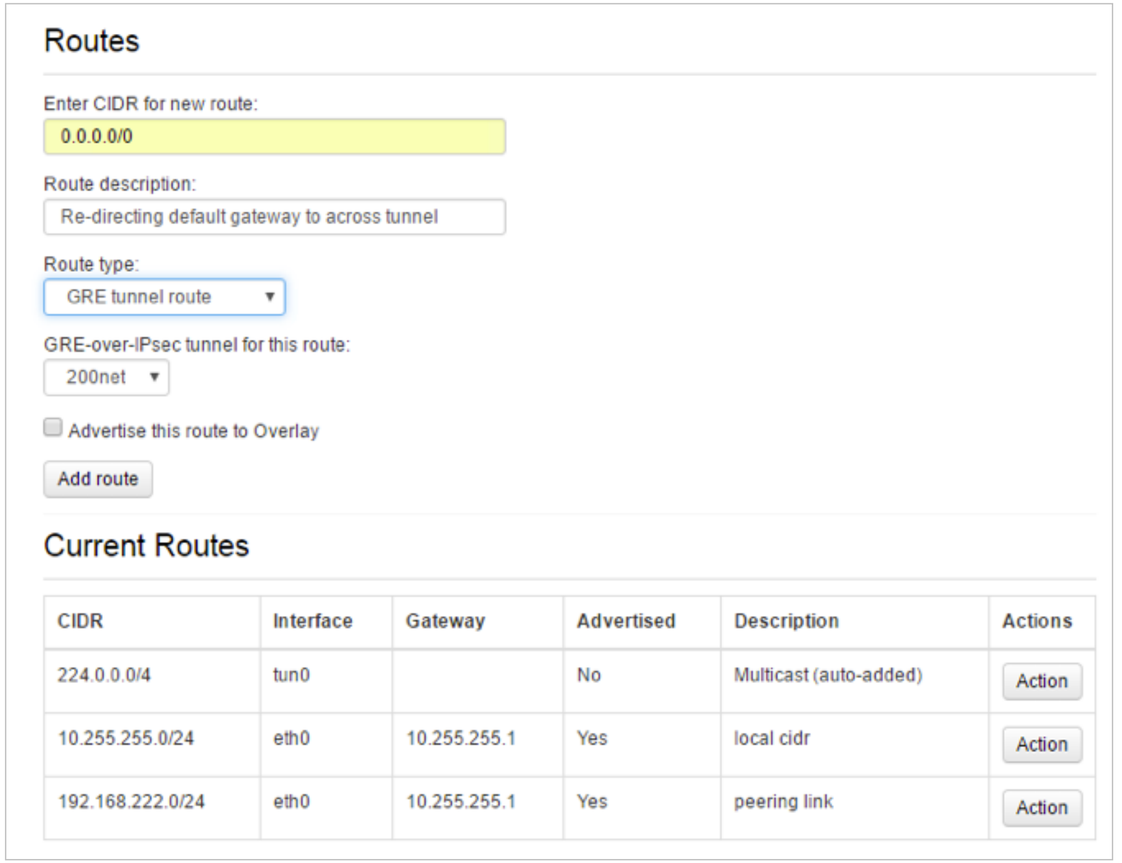
Adding route to 0.0.0.0/0 (WARNING)

In this example the route to "0.0.0.0/0" is seen as across a GRE tunnel route.

If the topology changes such that a re-directed default route is no longer needed, you can delete it via the "Action" menu on the "Current Routes" display table.

When the new default route was added, VNS3 saved the original gateway setting, and this setting will be used for the default gateway.

You DO NOT need to delete the routes you used to inform VNS3 of the local subnet CIDR, nor any peering routes. However, since the default route is now directed to your underlying subnet gateway, you do not need to keep them.



SNMP Support

SNMP Support

VNS3 now supports a number of industry standard MIBs for use from a monitoring system doing SNMP polling. We do not currently support any SNMP traps.

VNS3 SNMP support is enabled through the firewall. In the future we will provide API calls and user interface to provide more control of the SNMP experience.

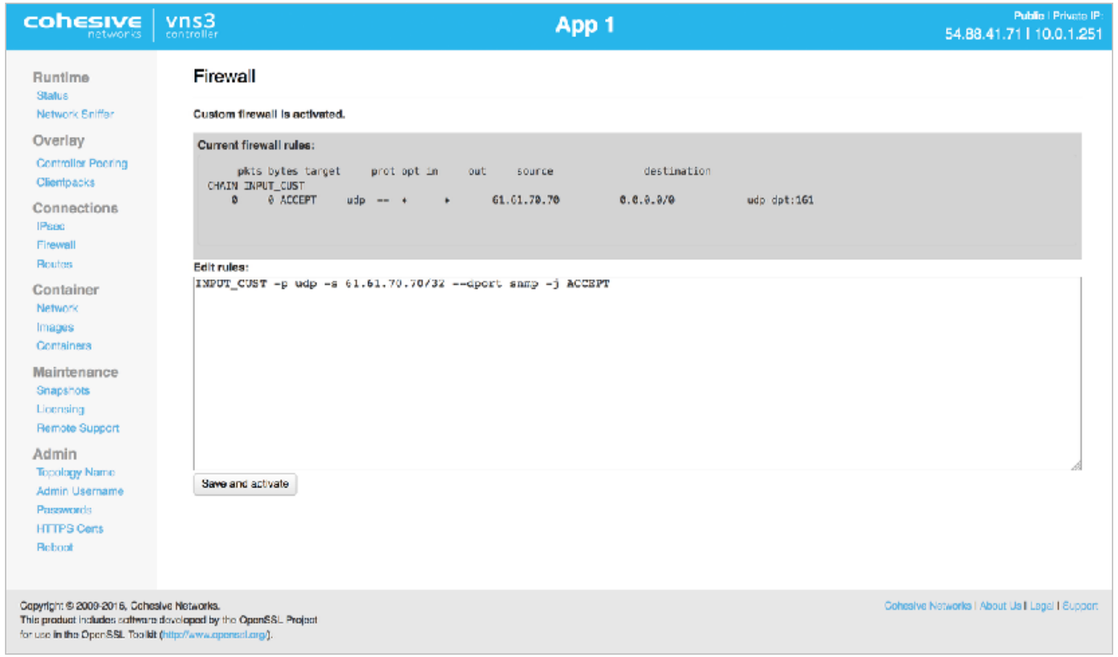
To enable access to the SNMP information add the following rule to your firewall using a source address from your network (either your public IP, or an internal IP available to the Controller via IPsec or Clientpack). There is no SNMP authentication in this beta. An example rule would be “INPUT_CUST -p udp -s 69.69.70.70/32 --dport snmp -j ACCEPT” (where 69.69.70.70 is your network’s public IP address).

On your SNMP monitoring system:

- Use SNMP v1c or v2
- Community string of “vns3public”
- The access to the SNMP information is “read only”

You should then be able to use a utility like “snmpwalk” to test:
snmpwalk -v 1 -c vns3public -O e <vns3_manager_public_ip>

In order to discuss additional MIBs needed please contact your Cohesive account representative or support (at) cohesive.net



Example of response from “snmpwalk”

```
snmpwalk -v 1 -c vns3public -O e <vns3_manager_public_ip>
SNMPv2-MIB::sysDescr.0 = STRING: Linux vpncubed 2.6.32-344-ec2 #46-
Ubuntu SMP Wed Mar 7 13:47:05 UTC 2012 x86_64
SNMPv2-MIB::sysObjectID.0 = OID: NET-SNMP-MIB::netSnmpAgentOIDs.10
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (1651090) 4:35:10.90
SNMPv2-MIB::sysContact.0 = STRING: support@cohesive.net
SNMPv2-MIB::sysName.0 = STRING: VNS3 version 3.0100.7-20130322173305
SNMPv2-MIB::sysLocation.0 = STRING: VNS3 Cloud Container
```

Network Sniffer

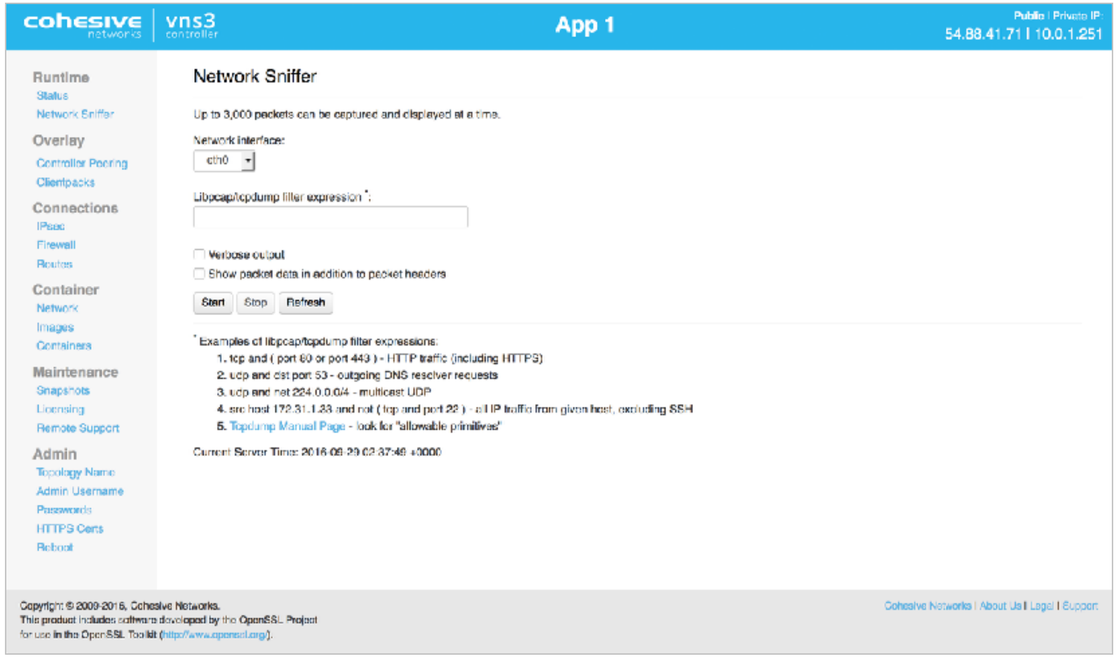
Network Sniffer

The Network Sniffer is a great troubleshooting tool. You can monitor both the public IP network interface and the Overlay Network interface of the VNS3 controller.

The Network Sniffer page includes some basic guidance for basic syntax. In the event your filter expression is malformed, the page will result in an "expression syntax error".

When using the Network Sniffer, there are two things to remember:

1. The page does not auto refresh. Once you enter in your expression, be sure to click the refresh button to update your view of the traffic.
2. The interface toggle is important.
   - eth0 - public IP network interface. This is where all encapsulated IPsec traffic and all unencrypted VLAN traffic is visible.
   - tun0 - Overlay Network interface.



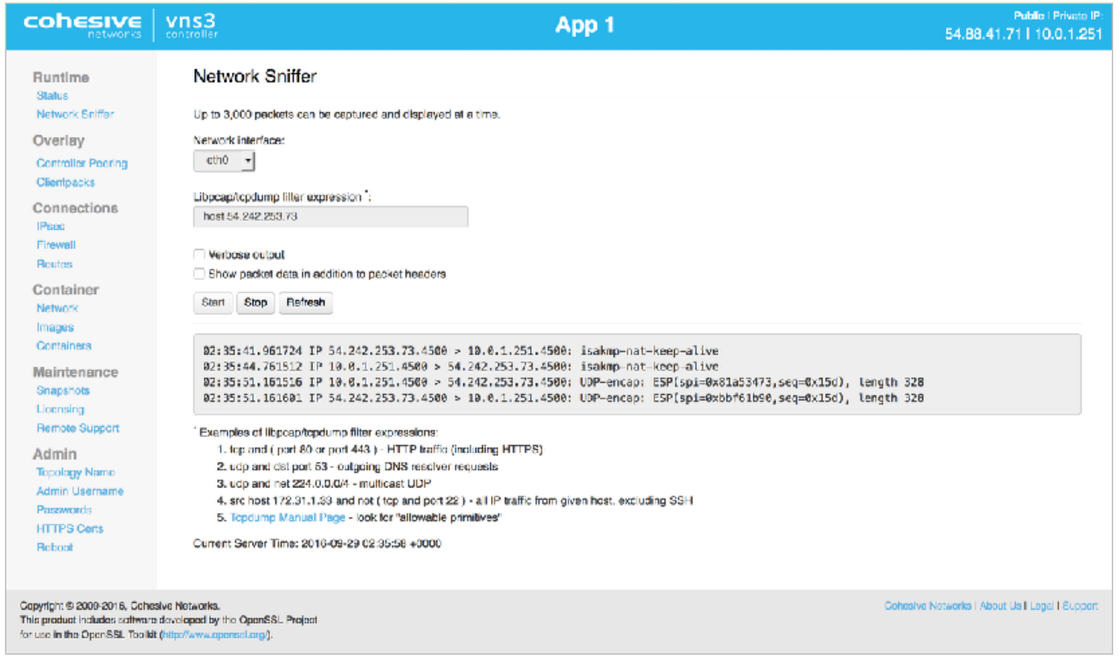
Network Sniffer - Troubleshooting an IPsec Connection

It is useful in troubleshooting an encrypted tunnel to first see if there is normal negotiation and keepalive traffic moving between the two IPsec Peers. This will help you to understand if there is a network connectivity or FW issue that prevents the negotiation.

Once the tunnel is negotiated, use the Network Sniffer to monitor the tunnel traffic, making sure encrypted/encapsulated packets are moving in both directions.

Use the following **eth0** filter to do both:

```
host <remote IPsec device IP>
```



Use the following **eth0** filter to do both:

```
host <remote IPsec device IP>
```

The result should be some UDP 500 traffic for maintenance traffic and encrypted traffic on UDP 4500 or ESP Protocol 50 (NAT-Traversal or Native IPsec respectively).

When troubleshooting, we recommend setting up a continuous ping down the tunnel with a larger-than-default size specified so that you can be sure the packets you are watching are your pings. To do this you can use the -l argument on Windows and the -s argument on Linux.

VNS3 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 Container Document

Configure and customize VNS3 and VNS3:turret products' Docker-based container plugin system.

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