Delivering on the Promise of NFV

Virtual Broadband Network Gateway (vBNG)

Service Configuration Guide

v1.11

Published: June, 2021

The information in this document and any document referenced herein is provided for informational purposes only, is provided AS IS AND WITH ALL FAULTS and cannot be understood as substituting for customized service and information that might be developed by netElastic for a particular user based upon that user's particular environment. RELIANCE UPON THIS DOCUMENT AND ANY DOCUMENT REFERENCED HEREIN IS AT THE USER'S OWN RISK.

netElastic PROVIDES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION CONTAINED IN THIS DOCUMENT AND ANY DOCUMENT REFERENCED HEREIN. netElastic provides no warranty and makes no representation that the information provided in this document or any document referenced herein is suitable or appropriate for any situation, and netElastic cannot be held liable for any claim or damage of any kind that users of this document or any document referenced herein may suffer. Your retention of and/or use of this document and/or any document referenced herein constitutes your acceptance of these terms and conditions. If you do not accept these terms and conditions, netElastic does not provide you with any right to use any part of this document or any document referenced herein.

Information in this document, including URL and other Internet Web site references, is subject to change without notice. Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, e-mail address, logo, person, place or event is intended or should be inferred.

The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

Table of Contents

1 About this Document

This document is written to enable customers, who already have netElastic's vBNG installed and are ready to configure the vBNG, to run typical vBNG use cases such as PPPoE access, IPoE access, BGP routing, etc. By following the use cases in the document, the reader should be able to set up all the related components in the vBNG so that the use case can be run and verified.

1.1 Objective

The objective of the document is to enable customers, who already have netElastic's vBNG installed and are ready to configure the vBNG, to run BNG use cases for the first time. Configuring the vBNG to run use cases can be complicated and often involves configurations of multiple different components of the vBNG. This guide is written around typical vBNG use cases and provides step-by-step configuration instructions in all related components. Keep in mind, this guide is not a replacement of the user guide or the command line reference guide, but rather a supplement of those documents aimed at enabling users to quickly get started on running use cases on the vBNG. For detailed feature and command details, please refer to the netElastic vBNG user guide and the command line reference guide.

1.2 Audience

The primary audience for this Guide includes network operation personnel who are responsible for monitoring a network, configuring the network elements, and topology and provisioning services. This guide assumes that the reader is familiar with the following topics and products:

- Oracle Solaris
- Microsoft Windows
- Linux
- MacIntosh
- Supported web browsers
- Basic internetworking terminology and concepts
- Network topology and protocols

1.3 Document Organization

As stated above, the document is organized around use cases. The use cases we documented here are picked so that they represent typical vBNG access and network use cases.

1.4 Conventions

The table below lists the conventions used in this guide.

Convention	Item	Example
bold default font	Menu command paths	
	Button names	
	User interface labels	

	Window/Dialog box titles	
Courier font	User-entered text	
Default font, italic	Document titles	
Consolas Font	Terminal text	vBNG# config terminal
NOTE:	Helpful suggestions	

1.5 Technical Assistance

Customer Support for netElastic products is available, 24 hours a day, 7 days a week. For information or assistance with netElastic products, please contact netElastic using any of the methods listed below:

- Hours: 9:00 AM to 5:00 PM PST (Monday-Friday, except Holidays)
- Phone: 1.866.448.7198
- Email: support@netelastic.com

2 vBNG Router Access and Management

2.1 Login to the vBNG Router (confd)

How the router can be accessed depends on how it is deployed. Please refer to the following installation guides on how to get to the router confd command line interface and netconf interface.

- netElastic vBNG Application Host Mode Installation Guide.
- netElastic vBNG Appliance VM Mode Installation Guide

You can access the router via ssh connection either through out of band management interfaces or inband router interfaces. The **routerIP** is the IP address by which you access the router via ssh connection.

- **Out of Band Connections**: For host deployment, **routerIP** is the IP of any of the management interfaces on the host. For VM deployment, **routerIP** is the IP of any of the management interfaces on the VM.
- **Inband Connections**: **routerIP** is the IP of any of the forwarding interfaces on the router by which you can access the router. Keep in mind that inband access is disabled by default for security reasons. To enable inband management access, please refer to section 2.3 on how to enable inband management access on the router.

To login to the router as the "admin", use "**ssh admin@routerIP –p 2024**". The default admin login credential is admin/admin. Once logged in to confd as "admin", you can type "**aaa authentication users user admin change-password**" to change admin confd access password.

NOTE: you have to login to confd as the "admin" role to be able to change the admin password. If you enter confd through the command "**confd_cli**", you are entering as the "operator" role and you won't be able to change the admin password.

2.2 vBNG Router Configuration Management

vBNG router can be configured and managed either through the vBNG manager GUI or through confd CLI. To use vBNG manager GUI, please follow the vBNG Manager Installation Guide to install vBNG manager.

The following confd CLI essentials provide the basics for managing the router through confd CLI.

Navigate through confd

- Hit "tab" key to bring up command completion prompts. Type "**?**" to get command help.
- Type "**show running-config [configure path]**" to show the current running configuration at the [configure path]. For example, "**show running-config bras domain**" will display all domain configurations.
- Type "**config**" to enter confd configuration mode. The vBNG router commands are hierarchically organized. Type "**?**" to list all available commands at the current level. Use subcommand to navigate to the next level. Type "**exit**" to go back to the previous level. Type "**end**" to exit config mode directly from whichever configure level you are currently at. Type "**show full**" to show the current configuration at this level. Type "**commit**" to commit all pending changes.
- The symbol "!" is a special symbol to tell confd what follows is a comment. If a string contains "!", either escape it with \ or put the whole string in "". For example, if you need to use "J!mRock!" as the radius server key phrase. You can either configure as **server 1 ipv4-address 64.251.173.19 port 1812 key J\!mRock\!** or **server 1 ipv4-address 64.251.173.19 port 1812 key "J!mRock!"**

Save and restore configurations

The following screen capture shows how to save an existing configuration and restore/commit it.

```
[root@all-1-1 ~]# confd_cli
domain# config
Entering configuration mode terminal
domain(config)# save /tmp/current_bng_config.dat
domain(config)# load replace /tmp/current_bng_config.dat
Loading.
3.00 KiB parsed in 0.75 sec (3.97 KiB/sec)
domain(config)# commit
Commit complete.
domain(config)#
```

2.3 Enable Router Inband Management Access

In addition to the typical out of band management access, you can also enable inband router management access. This feature is especially needed when the router is deployed in an environment where there is no router out of band management access. Through inband access, you could access:

- confd command line ssh access (default port 2024)
- confd netconf ssh access (default port 2022)
- router host ssh access (default port 2222)

Enable Confd Commmand Line SSH Access

```
ssh-server in-band enable true
```

```
ssh-server in-band port 2024
```

Enable Confd Netconf SSH Access

```
netconf-server in-band enable true
netconf-server in-band port 2022
```

Enable Router Host SSH Access

```
host-server in-band enable true
host-server in-band port 2222
```

With any of the inband access method shown above, you can add interface and access list control to enhance security. Here is an example on how to configure inband netconf access with interface and access list controls

```
netelastic(config)# show full-configuration netconf-server
netconf-server in-band enable true
netconf-server in-band port 2022
netconf-server in-band bind interface gei-1/1/3
netconf-server in-band bind acl flexlink_access_list

netelastic(config)# show full-configuration flexlink
flexlink access-list
 rule flexlink_access_list
  global deny all
  ip-prefix 10.101.1.0/24 permit
 exit
exit
```

2.4 Enable TACACS Access

To enable TACACS access to the vBNG router, configure the vBNG with configuration that is similar to the following:

```
domain# show running-config system   #system level configuration
system hostname domain
system login authentication-order tacplus local
domain# show running-config tacplus  # TACPLUS configuration
tacplus enabled
tacplus group default
 source-ip 3.3.3.3
 timer response-timeout 5
 timer quiet      5
 authentication server 1 ipv4-address 2.2.2.2 port 49 shared-key test
 authorization server 1 ipv4-address 2.2.2.2 port 49 shared-key test
 accounting server 1 ipv4-address 2.2.2.2 port 49 shared-key test
exit
```

The TACACS configuration has two parts: system level configuration and TACACS module level configuration.

System Level Configuration

There is only one system level configuration that relates to TACACS. It is "**system login authentication-order**". It value can have the following four options

- **local tacplus**: vBNG will try local authentication first. It will try TACACS authentication only after local authentication fails.
- **tacplus local**: vBNG will try TACACS authentication first. It will try local authentication only after TACACS authentication fails.
- **local**: vBNG will only try local authentication.
- **tacplus**: vBNG will only try TACACS authentication.

TACACS Module Configuration

These fields need to be configured under tacplus

- **tacplus enabled/disabled:** This switch will globally enable or disable TACACS user authentication.
- **tacplus group default:** This will create TACACS AAA group default, under which TACACS rules and servers will be configured. Please note that the tacsplus group name has to be "default". This is because confd ssh access cannot carry TACACS group name, vBNG can only have one group called "default".
    - **source-ip [NAS IP]:** This is the vBNG NAS IP. This can be the same IP as the Radius NAS IP.
    - **timer response-timeout [timerInSeconds]:** This sets the timer by which vBNG will mark the TACACS server inactive after **timerInSeconds** passed without getting reply from TACACS server.
    - **timer quiet [timerInMinutes]:** This sets the timer by which the vBNG will reactivate the TACACS server after marking it inactive for **timerInMinutes** minutes.
    - **authentication server [serverID] ipv4-address [serverIPv4Address] port [portNumber] shared-key [serverKey]:** This sets the TACACS authentication server.
        - **serverID:** TACACS server ID (1-8). This serves as the TACACS authentication server index ID. vBNG supports up to 8 TACACS servers.
        - **serverIPv4Address:** TACACS server IPv4 address.
        - **portNumber:** TACACS server port number. The default port. for TACACS is 49.
        - **serverKey:** TACACS server secret key.
    - **authorization server [serverID] ipv4-address [serverIPv4Address] port [portNumber] shared-key [serverKey]:** This sets the TACACS authorization server. The fields in the authorization server setting are the same as those for the authentication. They can be set identically as the authentication server setting.
    - **accounting server [serverID] ipv4-address [serverIPv4Address] port [portNumber] shared-key [serverKey]:** This sets the TACACS accounting server. The fields in the accounting server setting are the same as those for the authentication. They can be set identically as the authentication server setting.

Check TACACS Access Status

TACACS access state information can be obtained with these commands:

- **show tacplus authentication**
  show TACACS authentication access status
- **show tacplus authorization**
  show TACACS authorization access status
- **show tacplus accounting**
  show TACACS accounting access status

3 Radius Server and Integration with vBNG

Remote Authentication Dial-In User Service (**RADIUS**) is a networking protocol that provides centralized Authentication, Authorization, and Accounting management for users who connect and use a network service. Since vBNG works very closely with Radius sever for user access management. We will start with installation of FreeRadius server and integration with

vBNG. If you already have working Radius server, you can skip the installation section and jump to section 3.2 for information on Radius integration with vBNG.

3.1 Deploying a FreeRadius Server.

To deploy a FreeRadius sever, it involves installing MariaDB, FreeRadius, and DaloRadius (optional). DaloRadius provides web interface backend for user to user web GUI to manage the Radius server.

There are two ways to deploy a FreeRadius server: install from scratch or instantiate a Radius Server VM from a pre-packaged Radius server VM image.

3.1.1 Install Radius Server from Scratch.

If you prefer to install a Radius server from scratch either on a bare metal host or on a VM, you can follow this installation guide.

Install FreeRADIUS and Daloradius on CentOS/RHL 7

3.1.2 Create Radius Server VM from Pre-Installed Image

netElastic provides a qemu/kmv compatible VM image file that has MariaDB, FreeRadius, and DaloRadius already installed and configured. The radius database within the VM image is also pre-populated with a test user so you can functionally validate the Radius server upon instantiation. To instantiate your own Radius server VM from this image, please follow these steps.

- Download the radius server image from this link.
- Use virt-install (or your own orchestrator) to create a VM from the downloaded Radius server image. You can specify interfaces with the VM creation. If you don't specify any interface (as shown in the example below), one default interface will be created.

```
virt-install \
--connect qemu:///system \
-n freeRadius-clone \
--description "Free Radius Server" \
--ram=2048 \
--vcpus=2 \
--disk path=freeRadiusSvr-gold.qcow2,format=qcow2,bus=virtio,size=8 \
--graphics none \
--import \
--debug
```

- After the server starts, you can log in with root/netElastic (server's default login). At this point, you can do a local loopback user access check by issuing the following command. The server is already preload with a test user credential test_user/test_user_pw
  #radtest test_user test_user_pw 127.0.0.1 1812 netElastic

  The test should produce the following success message.

```
[root@localhost ~]# radtest test_user test_user_pw 127.0.0.1 1812 netElastic
Sent Access-Request Id 170 from 0.0.0.0:51772 to 127.0.0.1:1812 length 79
        User-Name = "test_user"
        User-Password = "test_user_pw"
        NAS-IP-Address = 127.0.0.1
        NAS-Port = 1812
        Message-Authenticator = 0x00
```

```
        Cleartext-Password = "test_user_pw"
Received Access-Accept Id 170 from 127.0.0.1:1812 to 0.0.0.0:0 length 20
```

- Since the server already has DaloRadius installed and configured, you should also be able to access the Radius server through a web browser with the following URL
  Error! Hyperlink reference not valid.
  where <server-ip-address> is the IP address of the Radius server you just instantiated. The default login for the web interface is administrator/radius

3.2 Load netElastic Vendor Specific Attributes (VSA) Radius Dictionary

netElastic's vBNG comes with vendor specific attributes (VSA) Radius dictionary that allows user to customize netElastic vBNG specific operations through Radius. To be able to use these features, it is necessary to load netElastic VSA dictionary to the Radius server. How to load VSA dictionary depends on the Radius software distribution that you are using. For FreeRadius, please follow the following steps to load netElastic's VSA dictionary:

1. Download netElstic's VSA Radius dictionary from this link. netElastic VSA Radius Dictionary
2. Copy the dictionary file such as "dictionary.netElastic" to the **/usr/share/freeradius/** directory. You should see dictionaries from other vendors in there as well. Keep in mind that dictionaries copied here manually will be lost upon a FreeRadius software upgrade.
3. Modify the **/usr/share/freeradius/dictionary** file to include the VSA dictionary such as "dictionary.netElastic". Keep in mind that modifications to this file will be lost upon a FreeRadius software upgrade. To add site local VSAs, please modify the **/etc/raddb/dictionary** file to include a line like the following.
   $INCLUDE dictionary.netElastic
4. Restart the **rediusd** process (**systemctl restart radiusd**) or reboot the radius server for FreeRadius to pick up the updated dictionaries.

Note: With the installation of DaloRadius (Refer to the FreeRadius Installation Guide), a dictionary table in the radius database (radius.dictionary) will be created and pre-populated with some vendor VSA dictionary entries. This table is used by DaloRadius. Instead of using this dictionary table in the radius database, FreeRadius uses dictionaries configured in **/usr/share/freeradius/dictionary** and **/etc/raddb/dictionary** (site local attributes) for VSAs.

3.3 Verifications of RADIUS connection on vBNG

The configuration shown in this section is based on the assumption that the vBNG and Radius server are L3 connected. In this example, they are in the same network with the following IP assignments.

RADIUS Server	192.168.7.157
vBNG	192.168.7.158

At this point, we assume the Radius server has been assigned with the IP 192.168.7.157 and is ping-able from the vBNG confd.

```
domain# ping 192.168.7.157
Sending 5, 64-byte ICMP Echos to 192.168.7.157, timeout is 2s:
!!!!!
--- ping statistics ---
5 packets transmitted, 5 received, 0.00% packet loss
round-trip min/avg/max = 11/15/16 ms
```

Now we need to configure IP 192.168.7.158 on a forwarding interface or a sub interface off a forwarding interface on the vBNG. Here is the interface configuration on the vBNG for this example.

```
domain# show running-config interface gei-1/1/2
interface gei-1/1/2
 ipv4 address 192.168.7.158 24
exit
```

Now we need to configure the radius section on the vBNG to add NAS IP and radius server IP. Here is the radius configuration on the vBNG for this example.

```
domain# show running-config radius
radius vendor-id     54268
radius accounting-on enable
radius attribute-usermac-as mac
radius authentication group my-radius-auth-grp
 server-type    ipv4-server
 timeout        3
 retry-times    3
 nas-ip-address 192.168.7.158
 algorithm      master
 dead-time      5
 dead-count     10
 class-as-car   disable
 filter-id-type user-acl
 server 1 ipv4-address 192.168.7.157 port 1812 key netElastic
exit
radius dmcoa group
 server-type ipv4-server
exit
```

At this point, we can do a radius ping test with a user's name and password from confd. Assuming the user already exists on the Radius server with credentials matching what are used in the ping test, we should receive an authentication "access accept" message from Radius as shown below.

```
domain# radius-ping authentication group my-radius-auth-grp user-name test_user
password test_user_pw pap
Ping radius authentication-group my-radius-auth-grp with test_user at 2020-01-08
07:58:31!
Ping server 192.168.7.157 at 2020-01-08 07:58:31!
Reply from server 192.168.7.157 access accept at 2020-01-08 07:58:31!
domain#
```

Now we have completed the Radius authentication test from vBNG.

4 User Access Management

If you are reading this guide, you probably have already installed vBNG and it is running without errors. You can check the running state of all vBNG processes by typing command "**flexbng**" on the CP VM Linux prompt. If you have applied valid license to the vBNG, you should also be able see all your active data interfaces by typing "**show running-config interface**" in confd CLI. If you don't see you data interfaces listed, it might be that your license is not valid or expired. Here is a sample list of interfaces.

```
all-1-1# show running-config interface
interface gei-1/1/2
exit
interface gei-1/1/3
exit
interface null0
exit
```

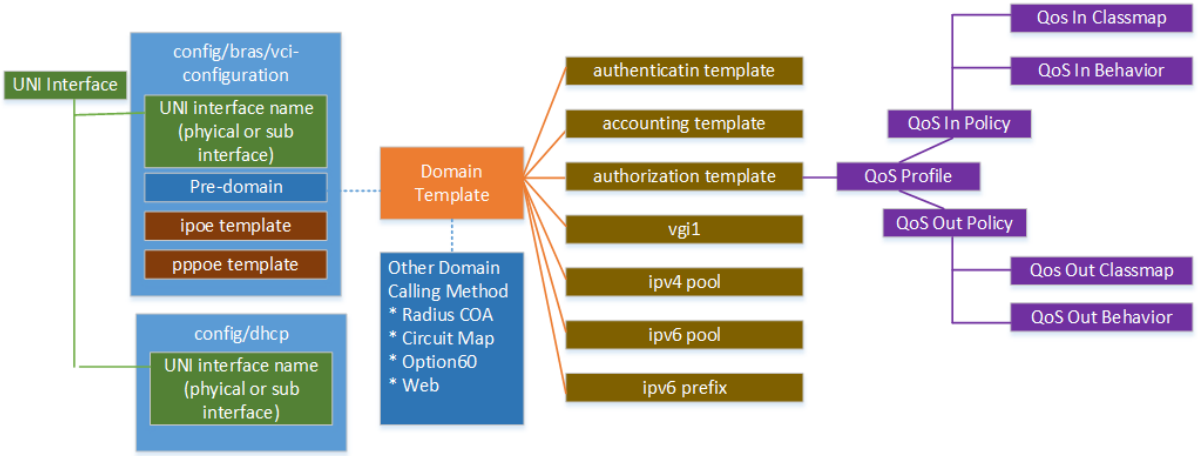
Note: to enter confd CLI configuration mode, type "**confd_cli**" on the CP VM Linux prompt.

Now it is time to map the interfaces to your network topology to identify which interfaces will be configured as access interfaces and which interfaces will be configured as network interfaces.

In section 4.1, the access configuration hierarchy will be explained. In section 7, detailed access configuration for some common access use cases will be provided as configuration examples.

4.1 Access Configuration Topology.

The access configuration is hierarchical. The following diagram shows a typical statically mapped (domain statically mapped to user) configuration flow for IPoE access.



When user access initiation packets come in to the vBNG, they usually carry the following information.

- User credentials, i.e. user name/password and access domain. For PPPoE access, users come in with user name and password in PPPoE discovery packets. For IPoE access user credential string usually come in as whole or part of option60 and option 82 strings. vBNG provides a very flexible ways to extract user name, password, and domain from these strings.
- Device ID, this is usually the access device's MAC address.
- Access circuit information. This is usually the VLAN ID carrier inserted in the packets. These VLAN IDs can either be originated from the access CPE device and/or inserted by the switch or gpon device between the CPE device and the vBNG.

The vBNG user authentication process can be summarized as in the following steps:

- Get or derive user name, password from the subscriber access request information (e.g. pppoe initiation or dhcp discovery packets etc). We will discuss this in Section 4.2

- Get or derive domain name from the subscriber access request information (e.g. pppoe initiation or dhcp discovery packets etc) or locally configured values. We will discuss this in Section 4.3
- Use the user name and password combo to compare against either locally created records or records in a Radius database and decide if the subscribers should be authenticated or not. The manner how this compassion should be performed is defined in the domain definition. See section 4.3 for more information on access domain definition.

4.2 About User Access Credentials

User credentials (user name, password) are what needed for user authentication. With Radius authentication, the vBNG will always send user name and password to Radius server for authentication. What are sent as user name and password to Radius comes from subscriber's access initiation or discovery packets. How the user credentials are extracted and formatted to send to Radius depends on the access method and how formatting is configured.

4.2.1 User Access Credentials for PPPoE

For PPPoE, the user name and password always comes from the subscriber's PPPoE initiation packets. vBNG will extract the user name and password from the PPPoE initiation packet and use them as the credential to check against stored record either locally or from Radius to determine if the subscriber should be authenticated or not.

NOTE: The user name field in the PPPoE discovery packet inherently can carry both user name and domain in the general format of "[user name][domain delimiter][domain]" (e.g. JohnSmith@officeNetwork). What part of the string is actually used for authentication is configurable in the authentication template under the key value "**user-name-format**" as described in section 5.1.2

4.2.2 User Access Credentials for IPoE

For IPoE, the user name and password are not explicitly carried in DHCP discovery packets. The vBNG will derive user name, password, and domain information from various mappings, including mapping from the various strings that the DHCP discovery packet carries. How the user name, password, and domain are derived depends on how the ipoe template is configured on the vBNG. A typical ipoe template configuration looks like the following. The switches "**authentication-type ipv4**" and "**authentication-type ipv6**" control how vBNG maps user name, password, and domain information from IPoE access requests. Other switches such as "**dhcp-v4 auth-on-up username-type**" and "**dhcp-v4 auth-on-up password-type**" determine the format by which the vBNG derives the user name and password from DHCP discovery packets.

```
bras
 ipoe template my_ipoe_temp
  authentication-type ipv4 dhcpv4 none
  authentication-type ipv6 dhcpv6 option
  dhcp-v4 auth-on-up password-type mac
  dhcp-v4 auth-on-up username-type mac
  dhcp-v4 auth-on-up domain-type pre-domain
  dhcp-v6 auth-on-up password-type mac
  dhcp-v6 auth-on-up username-type mac
  dhcp-v6 auth-on-up domain-type option
 exit
exit
```

User Name/Password/Domain Formation for IPoE

The mode by which User Name/Password/Domain information are formed for IPoE authentication is controlled by the switch "**authentication-type ipv4 dhcpv4** " and "**authentication-type ipv6 dhcpv6**" for IPv4 and IPv6 respectively. Use the "**authentication-type ipv4 dhcpv4**" switch in the ipoe template to configure how the user's access user name/password/domain information should be obtained for IPv4 IPoE. "**authentication-type ipv4 dhcpv4**" can take the following values.

- **none**: Applicable only to no authentication. IPoE module will not attempt to from user name and password for users.
- **cir-map:** User Name/Password/Domain are obtained by the mapping defined under circuit map definition.
- **option**: User Name/Password/Domain are obtained from DHCP discovery packet and whose format are defined by the switches "**dhcp-v4 auth-on-up username-type**" and "**dhcp-v4 auth-on-up password-type**"
- **option-web:** option first, then web if option failed.
- **web:** User Name/Password/Domain are obtained from web portal.

User Name/Password/Domain from Circuit Map

User access circuit (interface/vlan) can be directly mapped to user name/password/domain information by using the circuit map local definitions on the BNG. The vBNG can support up to 65535 circuit map entries. Here is an example with two circuit map definitions:

```
bras
 circuit-map 1
  interface gei-1/1/0 qinq external 23 to 23 internal 100 to 100
  username johnSmith domain myDomain password JohnSmithPassword
 exit
 circuit-map 2
  interface gei-1/1/0 qinq external 25 to 25 internal 100 to 100
  username paulRyan domain myDomain password paulRyanPassword
 exit
exit
```

To use circuit map, the switch "**authentication-type ipv4 dhcpv4**" needs to be set to **cir-map** in IPoP template.

User Name Formats for IPoE

When "**authentication-type ipv4 dhcpv4**" is set to **option**, use the "**dhcp-v4 auth-on-up username-type**" switch in the ipoe template to configure how user's access user name should be formed:

- **mac** : use subscriber's mac address as username
- **option60** : user Option60 string as username
- **option82-circuit-id** : use Option82 circuit-id string as username
- **option82-remote-id** : use Option82 remote-id string as username
- **default-user-name** : use customized user name (see section below)

Password Formats for IPoE

When "**authentication-type ipv4 dhcpv4**" is set to **option**, use the "**dhcp-v4 auth-on-up password-type**" switch in the ipoe template to configure how user's access password should be formed:

- **config** : use the passowrd that is configured (hardcoded) here
- **mac** : use subscriber's mac address as the password

- **option82-circuit-id** : use Option82 circuit-id string as password
- **option82-remote-id** : use Option82 remote-id string as password
- **optionstring** : user Option60 string as password
- **default-password** : use customized password (see section below)

Domain Formats for IPoE

Use the "**dhcp-v4 auth-on-up domain-type**" switch in the ipoe template to configure how user's access domain should be formed:

- **option** : use option60 string as domain
- **optionparse**: parse option60 to extract domain by the domain delimiter defined under bras-> domain-name-delimiter
- **optionstring** : use domain named "option"
- **pre-domain** : use the domain defined under the pre-domain key in the vci-configuration

Use Customized Username and Password for IPoE

In addition to the user name and password formats shown above, when user name and password formats are setup to use default-user-name and default-password, the access request user name and password can be generated from the following fields in customized formats:

- **ip-address** user IPv4 address
- **mac** user MAC address, xx:xx:xx:xx:xx:xx
- **option18** DHCPv6 option18
- **option37** DHCPv6 option37
- **port** port number
- **second-vlan** internal vlan for QINQ or middle vlan for QINQINQ
- **slot** Slot number
- **sysname** System name
- **third-vlan** innermost vlan for QINQINQ
- **vlan** vlan for DOT1Q, or external vlan for QINQ, or outmostvlan for QINQINQ

To configure the vBNG to use customized username and password, follow these steps:

1. Define customized username and password templates
2. Reference the customized username and password template in vci-configuration
3. Specify to use customized username and password in IPoE template

Below is a configuration example where we defined a customized username template "userName-port-vlans" and password template "userPasswd-port-vlans". Please note that when constructing a list of elements, you must use square brackets to enclose the list of elements. These templates were referenced under the vci-configuration. Finally in the user's IPoE template, we configure the "dhcp-v4 auth-on-up username-type" and "dhcp-v4 auth-on-up username-type" to use customized username and password.

```
bras
 default-user-name template userName-port-vlans
  type [ port second-vlan vlan ] format %s.%s.%s
 exit
 default-user-name template userPasswd-port-vlans
  type [ port second-vlan vlan ] format %s.%s.%s
 exit
exit

bras
 vci-configuration
```

```
  interface 10gei-1/1/0.33
   ipoe template ipoe
   max-ipox-session      32000
   max-pppox-session     32000
   encapsulation         multi
   default-user-name     userName-port-vlans
   default-user-password userPasswd-port-vlans
   pre-domain            ipoe-domain
   ip-access-type        ipv4
  exit
 exit
exit

bras
 ipoe template ipoe
  authentication-type ipv4 dhcpv4 option
  authentication-type ipv6 dhcpv6 cir-map
  dhcp-v4 auth-on-up password-type default-password
  dhcp-v4 auth-on-up username-type default-user-name
  dhcp-v4 auth-on-up domain-type optionparse
  dhcp-v6 auth-on-up password-type mac
  dhcp-v6 auth-on-up username-type mac
  dhcp-v6 auth-on-up domain-type optionparse
 exit
exit
```

4.3 About Access Domain

User access behavior is largely determined by access domain defined for the user or access interface. Access domain will define properties such as

- Authentication template: this will define how user is going to authenticated such as none, local, or radius, etc.
- Authorization template: this will define how user is going to authorized with properties such as QoS, ACL, NAT etc.
- Accounting template: this will define how user's online records are going to accounted.
- vgi definition: this defines user's gateway.
- IP Pools: this will define IP Pools for subscribers.
- etc.

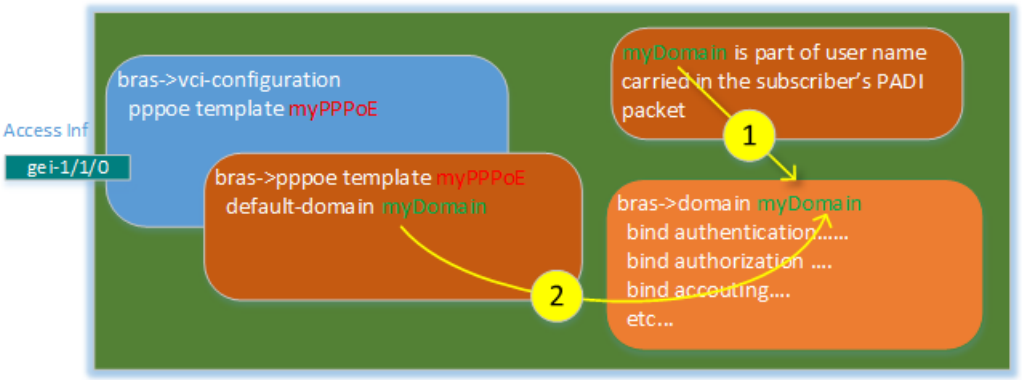
User online process has two stages: authentication and authorization. A user has to be able to find its associated domain during each of these two stages. These two stages can use the same domain or they can use different domains. During the authentication stage, user has to be able to find how it is going to be authenticated. Therefore, the associated domain during the authentication stage must have authentication template defined. After successful authentication, the online process enters the authorization stage, the same domain used during authentication can be used for authorization, or another domain for authorization can be associated with the user via Radius authentication reply message or local-subscriber domain specification. It is obvious that the only situations where authorization domain could be different from the authentication domain are when Radius authentication or local-subscriber management is used.

Although there is only one way that the authorization domain can be determined when different for authentication domain (i.e. by Radius reply message), there are many ways an authentication domain can be specified to accommodate the various ways a user's online authentication process can be handled by ISPs. How an authentication domain is specified depends on the access method such as PPPoE, IPoE, and others.

4.3.1 Access domain specification for PPPoE

For pppoe, subscriber's authentication domain specification can come from two places:

1. From subscriber's PPPoE PADI packets: pppoe initiation (PADI) packet coming to vBNG already carries user name and password. vBNG will use the user name and password for authentication. If the user name coming in is of the format username@domain, vBNG will treat the string after the "@" sign as the access domain associated with the user.
2. From the **default-domain** specification in the PPPoE template definition: See section 7.5 for PPPoE template definition.

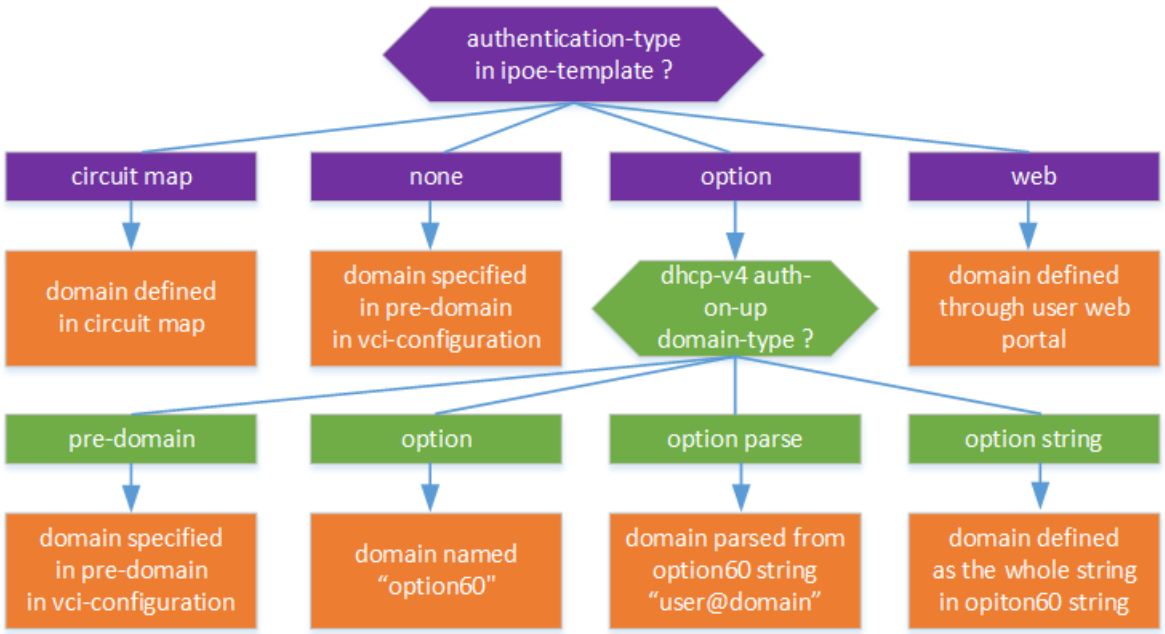


The above diagram shows these two methods for a PPPoE subscriber to find its access domain.

NOTE: Method 1 has higher precedence than method 2. If domain name is carried as part of the user name in PADI packet, the domain matching that domain name must be configured on the vBNG. Otherwise, the subscriber won't be able to be authenticated. The domain specified as the default-domain in PPPoE template takes effect only if the user name in PADI packet does not carry domain name.

4.3.2 Access domain specification for IPoE

As mentioned above, unlike PPPoE, IPoE connection initiation (DHCP discovery packet) does not come in with user name and password. Instead, DHCP discovery packets come in with user ID (MAC) and possibly other DHCP option strings. netElastic's vBNG offers flexible ways to determine which access domain the IPoE user belongs to from the information carried in the DHCP discovery packets. This diagram shows how the access domain is determined for IPoE access through the ipoe-template configuration.



4.3.3 Domain specification summary

User online process has two stages, authentication and authorization. A domain has to be associated with each stage. The same domain can be used for both authentication and authorization. Domains have to be predefined in configuration and one can define as many domains as needed. Domains can be specified dynamically or statically in configuration file.

- Statically configured:
  - For PPPoE, specify as **default-domain** in PPPoE template.
  - For IPoE, speify as **pre-domain** in vci-configuration.
- Dynamically specified:
  - For PPPoE, as part of user name (the string after the domain delimiter)
  - Various ways for IPoE as illustrated under section 4.3.2
  - Radius authentication reply message for Both PPPoE and IPoE

4.4 Check User Access Status

4.4.1 Display User Session Summary and Detail

The following commands display user session information:

- **show smgr-session summary**
  display user session summary by access types, interfaces, domains
- **show smgr-session all**
  list all user sessions in a tabular format.
- **show smgr-session detail**
  shows all active sessions in detailed format record by record. When significant amount of user are online, the list can be too long to display. You can selectively display user records by applying filters as shown in the next command.
- **show smgr-session detail by-xxx**
  displays user sessions in detailed format for users meeting certain filtering criteria. The “xxx” in the command represents a myriad of options such as access domain, accounting id, authorization domain, dot1q, ipv4 address, qinq, qinqinq, user name, etc.

The user session details lists the following fields for the users displayed.

```
  access-interface          framed-route              policy-name
  access-mode               gateway-address           session-id
  accounting-info           igmp-profile              subcar-input
  auth-status               ip-access-type            subcar-output
  auth-type                 ippool-name               tcp-adjust-mss
  author-domain             ippoolv6-name             timeout
  classmap-input            ipv4-address              unicast-traffic
  classmap-output           ipv6-address              unicast-traffic-web
  create-time               l2tp-info                 user-access-type
  dns-v4                    lawful-intercept          user-index
  dns-v6                    mac-address               user-name
  domain-name               mru                       vgi-interface
  dropped-traffic           multicast-traffic         vlan
  duid                      multicast-traffic-web     vrf-name
  family-info               nat-info                  webforce-info
  framed-ipv6-route         online-time
```

You can pick and choose any one or more of the fields to display. To select more than one fields, use the | operator with select. Here is an example:

```
domain# show smgr-session detail user info user-name | select info auth-status |
select info mac-address
USER                        AUTH
TYPE  MAC ADDRESS           STATUS  USER NAME
-------------------------------------------------------
ipoe  e4:b9:7a:88:f1:d5     accept  e4-b9-7a-88-f1-d5
      84:2b:2b:aa:86:4f     accept  84-2b-2b-aa-86-4f
```

4.4.2 Display User Connection Rate

To display a user's connection rate, use the following command: **show subscriber-traffic-rate by [ipv4-address, ipv6-address, mac-address, user-name]**. Here is an example

```
netelastic# show subscriber-traffic-rate by user-name e4-b9-7a-88-f1-d5
Collecting data, that may take a few seconds.
up-Bps:6498 up-pps:19 down-Bps:3909 down-pps:18 upv6-Bps:0 upv6-pps:0 downv6-Bps:0
downv6-pps:0
```

In the above example, the measured upstream rate is 6498 bytes/s and downstream rate is 3909 bytes/s during the measurement interval.

4.5 User Access Troubleshooting

Here are some of the commonly used commands and practices for troubleshooting user access related issues.

4.5.1 Check Online Fail Record Log

vBNG keeps track user access fail records and attempts with diagnosed failed reasons. Here is an example.

```
netelastic# show vbras online-fail-record
                                        ACCESS  USER  DOMAIN  ACCESS
USER MAC          USER LOGIN STAMP      TYPE    NAME  NAME    INTERFACE  VLAN  USER IP  FAIL REASON
-------------------------------------------------------------------------------------------------------
84:2b:2b:aa:86:4f  2021-05-20 18:30:06.377  ipoe                gei-1/1/2  0     0.0.0.0  ip access type error
e4:b9:7a:88:f1:d5  2021-05-20 18:29:55.637  ipoe                gei-1/1/4  0     0.0.0.0  ip access type error
```

To clear the online fail records, use the command "**clear-vbras-online-fail-record**"

4.5.2 Check Abnormal Offline Record Log

vBNG keeps track user abnormal offline records and attempts with diagnosed failed reasons. Here is an example.

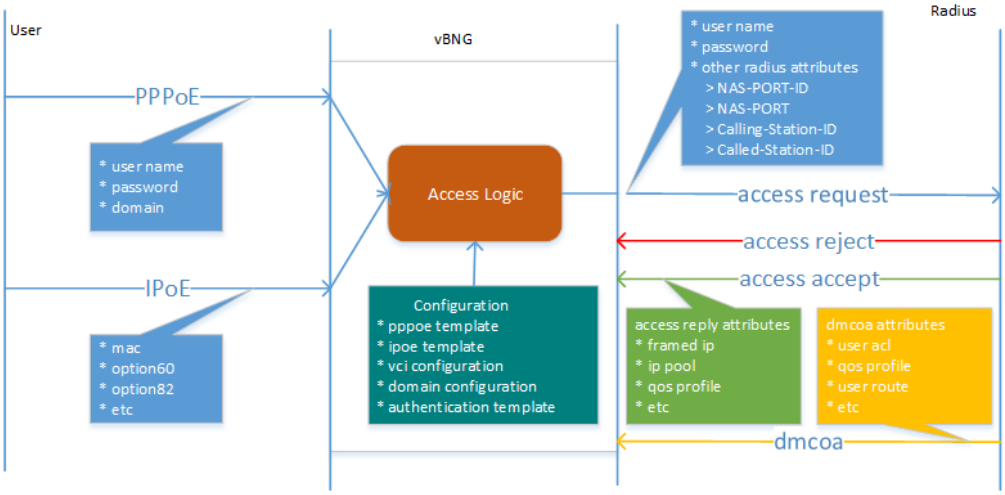
```
netelastic# show vbras abnormal-offline-record
% No entries found.
```

To clear the abnormal offline records, use the command "**clear-vbras-abnormal-offline-record**"

5 Radius AAA

Authentication here means comparing subscriber credentials (user name/password as described in section 4.2) received in the wire against what is stored either locally or on Radius, and then made a decision as to whether or not the subscriber should be allowed on the vBNG. The manner in which subscribers are authenticated is described in the authentication template. Subscribers coming in from an access interfaces need to find their authentication template definitions to be validly authenticated even if the authentication method is specified as "none" in the authentication template. The authentication template for a subscriber is specified in its access domain of which we described in section 4.3.1 and section 4.3.2 for PPPoE and IPoE respectively.

In the case of radius authentication, vBNG will send user's user name and password together with other optional and configurable attributes to the radius server for authentication. The radius server will reply with either "authentication accept" or "authentication deny" message. In the case of authentication accept, radius can also send optional authorization attributes such as user IP, framed route, qos plans etc. The following diagram shows this flow.



From the above diagram, we can see there are three possible interactions between the vBNG and the radius

- vBNG sends authentication access request with user credential and other attributes.
- Radius sends vBNG access reject or access accept with certain attributes.
- Raidus sends DMCOA messages to vBNG to change user online behaviours on the fly with certain attributes.

We will talk about each of these interaction and associated configurations in the following subsections.

5.1 Radius Access Request and User Authentication

The manner in which the subscriber's Radius access request is performed is configured through authentication template, which in turn will refer to a radius authentication group definition.

5.1.1 Radius Authentication Group Definition

First we have to define a radius authentication group where we specify parameters such as timeouts and radius server IPs and access secrets. Here is an example:

```
radius authentication group radius_netElastic
 server-type    ipv4-server
 timeout        3
 retry-times    3
 nas-ip-address 10.10.0.169
 algorithm      master
 dead-time      5
 dead-count     10
 class-as-car   disable
 filter-id-type user-acl
 server 1 ipv4-address 192.168.7.149 port 1812 key netElastic
exit
```

5.1.2 Authentication Template

A typical configuration of the authentication template with radius authentication is shown below.

```
bras
 authentication radius_auth
  authentication-type         radius
  radius-authentication-group radius_netElastic
  user-name-format            strip-domain
  nas-port-format             class1
  called-station-id-format    class1
  nas-port-id-format class1
  calling-station-id-format class1
  invalid-vlan-tag            0
 exit
exit
```

The authentication template configures:

1. How authentication should be done (Radius, Local, etc)
2. How the vBNG will form and send auxiliary Radius attributes with authentication request packet.

Some of the typical fields in the template are:

- **authentication-type** : this field dictates how the subscriber should be authenticated. Here are the options:
  - ***local*** : local authentication, user must be configured locally on the vBNG. See section 7.2 and 7.6 for local user configuration examples.
  - ***local-radius*** : vBNG will try local authentication first. If user cannot be found locally, vBNG will send authenticate request to Radius for Radius authentication. User access attempt will be rejected if both fail.
  - ***none*** : no authentication. As noted earlier, you still need to create an authentication template and set the authentication-type to this value even if you don't need authentication.

  - ***radius*** : Radius only. If user cannot be authenticated through radius, user access attempt will be rejected.
  - ***radius-local*** : vBNG will try Radius authentication first. If user cannot be authenticated through Radius, vBNG will try local authentication. User access attempt will be rejected if both fail.
  - ***radius-none*** : vBNG will try Radius authentication first. If Radius does not reply to the authentication request, vBNG will authenticate user as if the user's authentication-type is set to none.
- **radius-authentication-group** : specify the Radius authentication group you would have defined under "radius authentication group" where you specify radius server access information (IP, ports, and secret)
- **user-name-format** : This specifies how vBNG extract the user name from the user name string sent from the subscriber. The user name string from the subscriber is in the general form of "*stringA@stringB*", where @ denotes the delimiter character set under "bras-> domain-name-delimiter". Here are the options:
  - ***strip-domain*** : "stringA" will be used as user name
  - ***include-domain*** : "stringA@stringB" will be used as user name
  - ***only-domain*** : "stringB" will be used as user name
- **nas-port-id-format** : This specifies how vBNG forms the NAS-PORT-ID (RADIUS Attribute 87)string to be sent to radius.
  - ***class1*** : The NAS-PORT-ID string will be formulated as "slot=xx;subslot=xx;port=xx;vlanid=xx;vlanid2=xx". For single vlan (dot1Q), vlanid carries and vlan ID and vlanid2 will be 0. For double vlan (QinQ), vlanid carries inner vlan ID (cVlan) and vlanid2 will carry outer vlan ID (sVlan)
  - ***class2*** : The NAS-PORT-ID string will be formulated as "slot=xx;subslot=xx;port=xx;vlanid=xx;vlanid2=xx". For single vlan (dot1Q), vlanid carries and vlan ID and vlanid2 will be 0. For double vlan (QinQ), vlanid carries outer vlan ID (sVlan) and vlanid2 will carry inner vlan ID (cVlan). Note that for QinQ, how vlan IDs are carried for class2 is exactly the opposite of that for class1.
  - ***class3*** : The NAS-PORT-ID string will be formulated as "certusnet eth 0/slot/subslot/port:{vlan|evlan.ivlan}".
  - ***class4*** : The NAS-PORT-ID string will be formulated as "{atm|eth|trunk} NAS_slot/NAS_subslot/NAS_port:XPI.XCI AccessNodeIdentifier/ANI_rack/ANI_frame/ANI_slot/ANI_subslot/ANI_port[:ANI_XPI.ANI_XCI]", where

    atm|eth|trunk: BRAS/SR interface type。"atm' for atm interface; "eth" for Ethernet interface; and "trunk" for Trunk type Ethernet interface. Currently this field is set to "eth" by vBNG.

    NAS_slot: BRAS slot ID.

    NAS_subslot: BRAS sub slot ID.

    NAS_Port: BRAS port number.

    XPI: If the interface type is ATM, XPI corresponds to VPI, XPI is an integer value within range 0~255; If the interface type is eth or trunk, XPI corresponds to PVLAN, XPI is an integer value within range 0~4095.

    XCI: If the interface type is ATM, XCI corresponds to VCI, XCI is an integer value within range 0~65535; If the interface type is eth or trunk, XCI corresponds to CVLAN, XCI will be an integer value within range 0~4095

    AccessNodeIdentifier/ANI_rack/ANI_frame/ANI_slot/ANI_subslot/ANI_port[:ANI_XPI.ANI_XCI]: This is L2 access information. Currently vBNG sets these values all to 0

  - ***class5*** : The NAS-PORT-ID string will be the Circuit-ID in PPPoE or IPoE packets.
  - ***KEEP_AGENT_CIRCUIT_ID***: The NAS-PORT-ID string will be formulated as "eth slot/subslot/port:vlan.vlan".
  - ***USER_DEFINED***: The NAS-PORT-ID string will be formulated as user defined string. Currently only "slot", "port", and "vlan" are supported.
- **nas-port-format**: This specifies how vBNG forms the NAS-PORT (RADIUS Attribute 5) value (32 bit) to be sent to radius.
  - ***class1***: slot ID (8 bit) + sub slot ID (4 bit) + port number (8 bit) + VLAN(12 bit). For QinQ, VLAN is inner VLAN ID.
  - ***class2***: slot ID (12 bit) + sub slot ID (8 bit) + VLAN(12 bit). For QinQ, VLAN is inner VLAN ID.
  - ***class3***: slot ID (3 bit) + sub slot ID (1 bit) + port number (4 bit) + QinQVLAN(12 bit) + VLAN(12 bit).
  - ***class4***: slot ID (8 bit) + sub slot ID (4 bit) + port number (8 bit) + VLAN (12 bit). For QinQ, VLAN is outer VLAN ID.
  - ***class5***: Only inner VLAN ID.
- **calling-station-id-format**: This specifies how vBNG forms the Calling-Station-ID (RADIUS Attribute 31) to be sent to radius
  - **class1**: user MAC address in the form of "xx:xx:xx:xx:xx:xx"
  - **class2**: "certusnet#0/slot/subslot/port #{vlan|exVlan:inVlan}"
  - **class3**: "xx-xx-xx-xx-xx-xx@vlan", where "@" represents the delimiter defined in the domain-name-delimiter under bras.
  - **Class4**: The value of "PPPoE remote-id"
- **called-station-id-format**: This specifies how vBNG forms the Called-Station-ID (RADIUS Attribute 30) to be sent to radius
  - **class1**: configured "webserver-ssid"
  - **class2**: The value of "PPPoE service-name"

NOTE: We only talked about a few of the Radius attributes that are sent over to Radius with authentication request packets. The complete list can be found from the following link.

List of vBNG Radius Attributes Sent With Access Request.

5.1.3 Check Authentication Request Status

Passing authentication is the first step for a subscriber to get online. When an authentication requests fails], it is important to be able to check access details such as what were the user access credentials received by vBNG, what were actually sent to Radius for authentication, what were the radius reply messages for a particular access request, etc. vBNG provides multiple ways to display access information at this level of detail.

Check Radius Access Status with radius-ping Test

For radius authentication, the first thing you want to try is to ensure radius can accept your authentication request. This can be tested by using command radius-ping. Radius-ping can test both radius authentication and radius accounting. It exercises the whole radius authentication or accounting protocols with a test user as shown in the following example.

```
domain# radius-ping authentication group my-radius-auth-grp user-name test_user
password test_user_pw pap
Ping radius authentication-group my-radius-auth-grp with test_user at 2020-01-08
07:58:31!
Ping server 192.168.7.157 at 2020-01-08 07:58:31!
```

```
Reply from server 192.168.7.157 access accept at 2020-01-08 07:58:31!
domain#
```

Check Access Status with Access Request Log

vBNG uses 342 log files to keep track of almost all user online and networking activities in corresponding log files. Here is the list of some of the access related log files that you may want to examine to check or debug access related issues.

- /var/log/certus/pppoe
  pppoe related activity log.
- /var/log/certus/ipoe
  ipoe related activity log.
- /var/log/certus/l2tp
  lt2p related activity log.
- /var/log/certus/radius
  radius access related activity log.
- /var/log/certus/ippool
  ip pool allocation activity log.
- /var/log/certus/dhcp
  dhcp request activity log.
- /var/log/certus/smgr
  smgr including accounting record update activity log.
- /var/log/certus/nat
  nat sessions activity log.

Check Access Status with Console Debug Message

You can interactively debug user online status by turning on debug message print out to the console. To enable displaying debug messages.

1. Turn on global debug message print out with command **debug monitor on**
2. Turn on printing out debug messages for various modules. For example:
   a. **debug pppoe all** – display pppoe online messages.
   b. **debug ipoe all** – display ipoe online messages.
   c. **debug radius all** – display radius interaction messages.
   d. **debug dhcp all** – display dhcp interaction messages.
   e. **debug ippool all** – display ipv4 pool allocation messages.

NOTE: The debug display settings are section specific. It is active only for the currently logged in session. All settings will be lost once you log out.

Check Access Status by Examining On-line Fail Records

vBNG also keeps track of failed access records with the reason why the access was failed. This provides very helpful information as to why access attempts failed and thus provide hints on how to correct them. To display these records, type the command **show vbras online-fail-record**. Here is a sample of the records.

```
                                  ACCESS  USER  DOMAIN  ACCESS
USER MAC          USER LOGIN STAMP        TYPE    NAME  NAME    INTERFACE  VLAN  USER IP  FAIL REASON
-------------------------------------------------------------------------------------------------------------------
e4:b9:7a:88:f1:d5  2021-01-08 19:22:36.879  ipoe                  gei-1/1/4  0     0.0.0.0  ip access type error
84:2b:2b:aa:86:4f  2021-01-08 19:22:20.789  ipoe                  gei-1/1/2  0     0.0.0.0  ip access type error
e4:b9:7a:88:f1:d5  2021-01-08 19:22:04.879  ipoe                  gei-1/1/4  0     0.0.0.0  ip access type error
e4:b9:7a:88:f1:d5  2021-01-08 19:21:48.925  ipoe                  gei-1/1/4  0     0.0.0.0  ip access type error
84:2b:2b:aa:86:4f  2021-01-08 19:21:48.789  ipoe                  gei-1/1/2  0     0.0.0.0  ip access type error
e4:b9:7a:88:f1:d5  2021-01-08 19:21:40.869  ipoe                  gei-1/1/4  0     0.0.0.0  ip access type error
00:21:70:d7:d3:ca  2021-01-08 19:21:40.552  pppoe                 gei-1/1/5  0     0.0.0.0  PPPoE discover timeout
e4:b9:7a:88:f1:d5  2021-01-08 19:21:36.879  ipoe                  gei-1/1/4  0     0.0.0.0  ip access type error
e4:b9:7a:88:f1:d5  2021-01-08 19:21:34.869  ipoe                  gei-1/1/4  0     0.0.0.0  ip access type error
```

To clear this record, use the command **clear-vbras-online-fail-record**.

5.2 Radius Access Reply and User Authorization.

If authentication is not successful, radius will reply with access reject message that carries not attributes. If authentication is successful, radius will reply with access accept message that can carries relevant attributes to direct the vBNG how to handle certain properties of the user. This process is called radius authorization and its behavior is defined in the user authorization template.

5.2.1 Authorization Template

The authorization template server two purposes:

1. Control how subscriber's authorization attributes such as IP address, qos plan, acl rule etc. should be obtained
2. Specify locally defined authorization attributes such as IP address, qos plan, acl rule etc.

A typical user authorization template is shown below.

```
bras
 authorization radiusAuthorization
  authorization-type mix-radius
  user-qos-profile   user_qos_1000kbpsUp_5000kbpsDown
  user-acl-profile   user_acl_list
  sub-car-input cir 2000 pir 2000 cbs 250000 pbs 250000
  sub-car-output cir 4000 pir 4000 cbs 500000 pbs 500000
  bind nat-domain-name myNatRule
  nat-type           inside
  radius-nat-switch  disable
 exit
exit
```

Some of the typical fields in the template are:

- **authorization-type**: this field dictates how the subscriber's authorization attributes should be obtained. Here are the options:
  local: user's authorization attributes strictly come from locally configured values on the vBNG.
  radius: user's authorization attributes strictly come from radius.
  mix-radius: vBNG will use authorization attributes coming from the radius first. It will use locally configured values only if these attributes are not sent from radius.
  NOTE: we normally set this to mix-radius when using radius AAA.
- **user-qos-profile**: this field specifies the user qos profile. See section 6.7 for user qos profile definition.
  NOTE: the value specified here will only be used when either authorization-type is local OR authorization-type is mix-radius but radius does not send user-qos-profile.
- **user-acl-profile**: this field specifies the user acl rules. See section 6.2 for user ACL configuration.
  NOTE: the value specified here will only be used when either authorization-type is local OR authorization-type is mix-radius but radius does not send user-acl-profile.
- **sub-car-input**: this field specifies the user input (upload) rate limiting CAR parameters. See section 6.7.1 for user CAR rate control configuration.
  NOTE: Do not configure this parameter is user qos profile is used for user rate control instead.
- **sub-car-output**: this field specifies the user output (download) rate limiting CAR parameters. See section 6.7.1 for user CAR rate control configuration.
  NOTE: Do not configure this parameter is user qos profile is used for user rate control instead.

- **bind nat-domain-name:** this field specifies the NAT rule defined in the NAT section. See section 6.6 for NAT configuration.

5.2.2 Commonly used Radius reply attributes.

When user's radius authentication is successful, Radius server will send Access-Accept packets to the vBNG. The Radius Access-Accept packets can carry a lot of very useful public and private Radius attributes that are very useful and can provide a lot of access control flexibility for subscribers. These attributes will overwrite locally configured authorization attribute values when **authorization-type** is set to mix-radius in the authorization template. For complete list of supported Radius reply attributes by the vBNG, please refer the table by following the following link.

List of Radius Reply Attributes Supported by vBNG

Here we list out a few very commonly used attributes from the list and more detailed explantion.

Framed-IP-Address Attribute (public 8).

When a subscriber dials in to vBNG configured for RADIUS authentication, the vBNG begins the process of contacting the RADIUS server in preparation for user authentication. Typically, the IP address of the dial-in host is not communicated to the RADIUS server until after successful user authentication. However if the subscriber already has an IP assigned to it by the vBNG, the vBNG can send the IP address of the dial-in host to the RADIUS server as attribute 8 together with other user information, such as the user name, password to the RADIUS server.

After the RADIUS server receives the user information from the vBNG, it has two options:

- Regardless Radius authentication request contains attribute 8 or not, if the user profile on the RADIUS server already includes attribute 8, the RADIUS server will fill or override the IP address sent by the vBNG with the IP address defined as attribute 8 in the user profile. The address defined in the user profile is returned to the vBNG. The vBNG will then assign this IP to the user.
- If the user profile does not include attribute 8, the RADIUS server can accept attribute 8 from the NAS, and the same address is returned to the vBNG.

The format to assign Framed IP address on FreeRadius is the following:

Framed-IP-Address=192.168.3.8

Please note that for vBNG to accept Framed-IP-Address attribute from Radius, a valid IP address pool that contains the assigned IP needs to be defined and associated with the user's access domain. At authorization time for the user, vBNG will try to assign the user an IP from the defined pool unless "**Framed-IP-Address**" is found in the Radius accept reply message. To instruct vBNG to exclusively use the IP from Radius to assign to the user, you need to reserve the IP section that you intend to assign from the Radius in the IP pool definition to prevent vBNG from assigning them by default. Here is an example of IP pool definition with IP reservation (highlighted in red).

```
ippool group private_radius_pool
 gateway-ip 172.16.10.1 gateway-mask 255.255.255.0
 lease-time         3600
 dns-primary 8.8.8.8 secondary 8.8.4.4
 ippool-status      unlock
 warning-threshold 80
```

```
 warning-exhaust   disable
 frame-ip lease manage disable
 section start-ip 172.16.10.2 end-ip 172.16.10.255
  reserved-section reserved-start-ip 172.16.10.2 reserved-end-ip 172.16.10.255
 exit
exit
```

Framed-Route Attribute (public 22).

This is a public attribute (22). With this attribute, Radius can specify the routing information to be configured for the user on the vBNG. vBNG supports multiple framed routes per user. When forming the attribute string on the Radius, multiple routes have to be separated by an agreed-upon route delimiter. The delimiter is configured on the vBNG under "radius" as shown below. The default delimiter is ";"

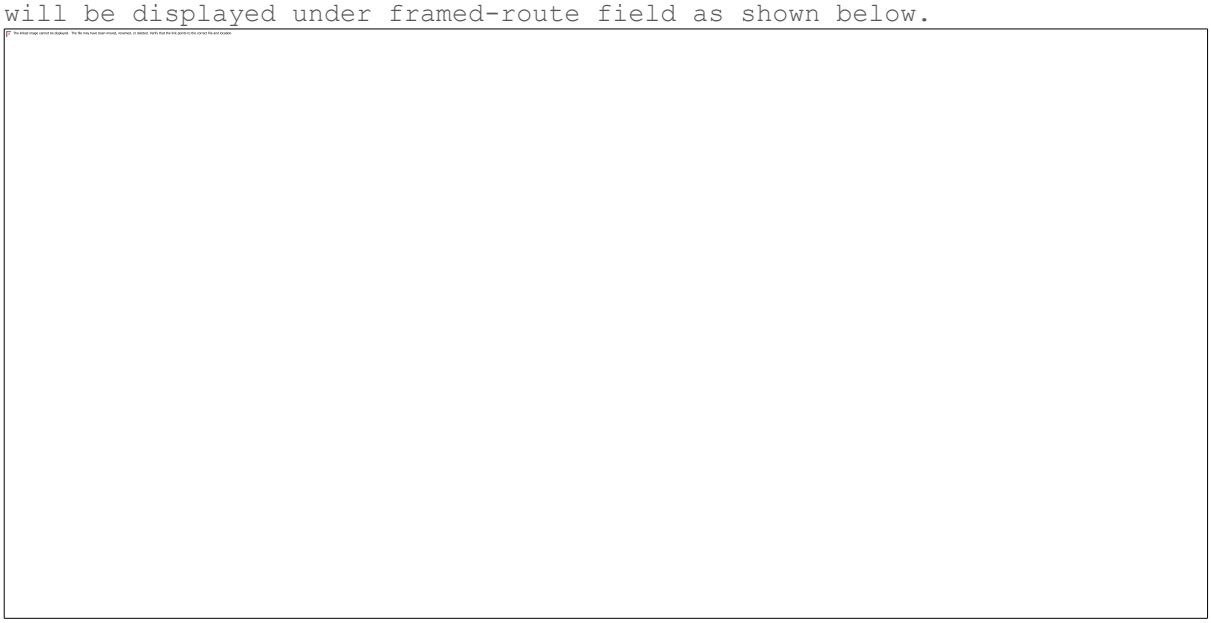
```
domain# show running-config radius
radius vendor-id       54268
radius accounting-on   enable
radius attribute-usermac-as mac
radius framed-route-delimiter ";"
radius username-override disable
radius dmcoa group
 server-type ipv4-server
exit
```

On the Radius, form the attribute string for framed route with the delimiter defined on the vBNG. Here is an example of Framed-Route attribute with three routes using the delimiter ";"

Framed-Route= 99.0.0.0/24 0.0.0.0 1;1.0.0.0/24 0.0.0.0 1;2.0.0.0/24 0.0.0.0 1;

In the above example, we are assigning three routes associated with the user. They are 99.0.0.0/24, 1.0.0.0/24, and 2.0.0.0/24. The Framed-Route string format is [route] [next hop] [metric], where the next hop should always be configured as 0.0.0.0. vBNG will tie the framed routes to the user whose own these routes.

To check the framed routes configured for a particular user on the vBNG, use the **show smgr-session detail user** command. The assigned frame routes will be displayed under framed-route field as shown below.



NetElastic-Data-Filter Attribute (private 82)

This is netElastic's private attribute. If the user profile on the RADIUS server already includes this attribute, Radius can send an ACL list name to the vBNG and the ACL list with that name will be applied to the user. With this attribute, you can apply subscriber's white list or black list on the fly. The format to assign Framed IP address on Radius is like the following:

NetElastic-Data-Filter=user_acl_list

Note: the access list "user_acl_list" needs to be preconfigured on the vBNG.

NetElastic-Qos-Profile-Name Attribute (private 31)

This is netElastic's private attribute. If the user profile on the RADIUS server already includes this attribute, Radius can send the qos profile name to the vBNG and the user_qos_profile with that name will be applied to the user. With this attribute, you can change subscriber's qos plan on the fly. The format to assign qos profile on Radius is like the following:

NetElastic-Qos-Profile-Name =user_qos_profile

Note: The qos profile "user_qos_profile" needs to be preconfigured on the vBNG

NetElastic-Domain-Name Attribute (private 138)

This is netElastic's private attribute. If the user profile on the RADIUS server already includes this attribute, Radius will send the domain name to the vBNG and the domain with that name will be applied to the user. With this attribute, you can change subscriber's access domain on the fly. The format to assign domain on Radius is like the following:

NetElastic-Domain-Name =user_domain

Note: The domain "user_domain" needs to be preconfigured on the vBNG

5.3 Radius DMCOA

netElastic's vBNG supports dynamically change of user's online behavior through Radius DMCOA. The following table lists the DMCOA actionable attributes.

Category	Description	Attribute Number	Attribute String
DM	Disconnet sub by accounting ID	44	Acct-Session-Id
COA (Dynamically Update User Properties)	change user's accounting update interval	85	Acct-Interim-Interval
	update user's framed route	22	Framed-Route
	update user's idle timeout	28	Idle-Timeout
	update user's session timeout	27	Session-Timeout
	update user's acl rule	82	NetElastic-Data-Filter
	update user's subcar rate	1	NetElastic-Input-Burst-Size
		2	NetElastic-Input-Average-Rate
		3	NetElastic-Input-Peak-Rate
		4	NetElastic-Output-Burst-Size

		5	NetElastic-Output-Average-Rate
		6	NetElastic-Output-Peak-Rate
		77	NetElastic-Input-Peak-Burst-Size
		78	NetElastic-Output-Peak-Burst-Size
	update user's qos profile or policy	31	NetElastic-Qos-Profile-Name
		95	NetElastic-Policy-Name
	update group user's QoS profile. User group are configured under config->bras->subscriber-manage->user-group-identify	17	NetElastic-ISP-ID
	update user's prepaid plan type and setting	251	NetElastic-Remanent-Volume-Type
		15	NetElastic-Remanent-Volume
	update user's primary DNS	135	NetElastic-Primary-DNS
	update user's secondary DNS	136	NetElastic-Secondary-DNS
	update user's portal related properties	138	NetElastic-Domain-Name
		252	NetElastic-Web-Coa
		85	NetElastic-HW-Portal-Mode
		140	NetElastic-HTTP-Redirect-URL
		11	Filter-Id

Some extra details on some of these attributes:

- **Update User's Subcar rate:**
  There are 8 attributes associated with this function. You can use these attributes to change subscriber's CAR on the fly.

  NOTE: Keep in mind that if you are using user qos profile (attribute 31 or 95) to control the subscriber rate, you should not use these subcar rates. They are not supposed to be used at the same time.

  To configure subscriber's subcar rate on the fly, the following attributes need to be assigned to proper values.
  - **NetElastic-Input-Average-Rate(2):** Specifies the input committed information rate (CIR), which is the average rate of traffic that can pass through an interface. The value is an integer that ranges from 64000 to 10000000000, in bit/s.
  - **NetElastic-Input-Peak-Rate(3):** Specifies the input peak information rate (PIR), which is the maximum rate of traffic that can pass through an interface. The value is an integer that ranges from 64000 to 10000000000, in bit/s.
  - **NetElastic-Input-Burst-Size(1):** Specifies the input committed burst size (CBS), which is the average volume of burst traffic that can pass through an interface. The value is an integer that ranges from 1 to 4294967295, in bytes. If the PIR is not set, the default CBS is 188 times the CIR in kbps. If the PIR is set, the default CBS is 125 times the CIR in kbps.
  - **NetElastic-Input-Peak-Burst-Size(77):** Specifies the input peak burst size (PBS), which is the maximum volume of burst traffic that can pass through an interface. The value is an integer that ranges from 10000 to 4294967295, in bytes. If the PIR is not set, the default PBS is 313 times the CIR in kbps. If the PIR is set, the default PBS is 125 times the PIR in kbps.

- **NetElastic-Output-Average-Rate(5):** Specifies the output committed information rate (CIR). Refer to "NetElastic-Input-Average-Rate" for attribute explanation.
- **NetElastic-Output-Peak-Rate(6):** Specifies the output peak information rate (PIR). Refer to "NetElastic-Input-Peak-Rate" for attribute explanation.
- **NetElastic-Output-Burst-Size(4):** Specifies the output committed burst size (CBS). Refer to "NetElastic-Input-Burst-Size" for attribute explanation.
- **NetElastic-Output-Peak-Burst-Size(78):** Specifies the output peak burst size (PBS). Refer to "NetElastic-Input-Peak-Burst-Size" for attribute explanation.

To enable DMCOA on the vBNG, all you need to configure is to create the DMCOA group as shown in the following example. After the dmcoa group is configured, the vBNG is ready to accept DMCOA requests.

```
radius dmcoa group
 server-type ipv4-server
 server 1 ipv4-address 128.201.138.55 key radiusKey
 server 2 ipv4-address 128.201.138.56 key radiusKey
 server 3 ipv4-address 128.201.138.57 key radiusKey
exit
```

You can configure up to 16 radius servers in the DMCOA group.

Here are a couple of examples of DMCOA requests from FreeRadius.

5.3.1 Disconnect Subscribers

By accounting id

```
# echo "Acct-Session-Id=D91FE8E51802097" > coa_message.txt
# echo "User-Name=somebody" >> coa_message.txt  #user name is optinoal
# echo "NAS-IP-Address=10.0.0.1" >> coa_message.txt
# cat coa_message.txt | radclient -x 10.0.0.1:3799 disconnect "radius_secret"

Sending Disconnect-Request of id 214 to 10.0.0.1 port 3799
      Acct-Session-Id = "D91FE8E51802097"
      User-Name = "somebody"
      NAS-IP-Address = 10.0.0.1
rad_recv: Disconnect-ACK packet from host 10.0.0.1 port 3799, id=214, length=20
```

By user name

```
# echo "User-Name=somebody" > coa_message.txt
# echo "NAS-IP-Address=10.0.0.1" >> coa_message.txt
# cat coa_message.txt | radclient -x 10.0.0.1:3799 disconnect "radius_secret"

Sending Disconnect-Request of id 214 to 10.0.0.1 port 3799
      Acct-Session-Id = "D91FE8E51802097"
      User-Name = "somebody"
      NAS-IP-Address = 10.0.0.1
rad_recv: Disconnect-ACK packet from host 10.0.0.1 port 3799, id=214, length=20
```

5.3.2 Switch Subscriber's QoS Plan

```
# echo "Acct-Session-Id = 1565289198482099000 2094000062" > coa_message.txt
# echo "Session-Timeout = 56000" >>  coa_message.txt
# echo "Idle-Timeout = 9990" >> coa_message.txt
# echo "Acct-Interim-Interval = 8" >> coa_message.txt
# echo "netElastic-Qos-Profile-Name = user_20M_updown" >> coa_message.txt
# cat coa_message.txt | radclient -x 192.168.25.117:3799 coa "radius_secret"
```

Here is the same example by user name

```
# echo “User-Name = 84-2b-2b-aa-86-4f > coa_message.txt
# echo “Session-Timeout = 56000” >>  coa_message.txt
# echo “Idle-Timeout = 9990” >> coa_message.txt
# echo “Acct-Interim-Interval = 8” >> coa_message.txt
# echo “netElastic-Qos-Profile-Name = user_20M_updown” >> coa_message.txt
# cat coa_message.txt | radclient -x 192.168.25.117:3799 coa “radius_secret”
```

5.3.3 Put Subscriber to Walled Garden

Sometimes it is needed to control subscriber’s access to certain website and services to push certain notification and special services to the subscriber. The walled garden feature in the vBNG allows the ISP to direct subscribers’ http traffic to certain website and to restrict their access to the internet. This feature can be dynamically turned on and off via Radius COA while the subscriber is online. To configure the walled garden service and make the service executable via Radius COA, please follow these steps:

Configure an ACL profile to restrict subscriber’s access

First we need to create an ACL rules and associated profile to restrict the subscriber’s traffic only to the desired website or other special services. All other internet access will be denied. Here is a sample configuration.

```
domain# show running-config access-list
access-list walled_garden_in
 rule 10 permit ip source 192.168.7.0/24 destination any
 rule 20 permit ip source 10.10.0.0/24 destination any
 rule 30 deny ip source any destination any
exit
access-list walled_garden_out
 rule 10 permit ip source any destination 192.168.7.0/24
 rule 20 permit ip source any destination 10.10.0.0/24
 rule 30 deny ip source any destination any
exit
domain#
```

```
domain# show running-config bras user-acl-profile
bras
 user-acl-profile walled_garden_acl_profile
  input-acl-profile  walled_garden_in
  output-acl-profile walled_garden_out
 exit
exit
domain#
```

Enable special-acl in Radius configuration

The ACL defined above to restrict subscriber’s access for walled garden control is considered a special ACL rule as opposed to user ACL rules. It is important to configure the “filter-id-type” to be “special-acl” in Radius configure (see the following example) so that the walled garden ACL can be properly turned off when walled garden restriction is turned off via Radius COA.

```
domain# show running-config radius
radius vendor-id       54268
radius accounting-on   enable
radius attribute-usermac-as mac
radius framed-route-delimiter ";"
radius username-override disable
radius authentication group my_radius
 server-type    ipv4-server
```

```
 timeout        3
 retry-times    3
 nas-ip-address 192.169.7.199
 algorithm      master
 dead-time      5
 dead-count     10
 class-as-car   disable
 filter-id-type special-acl    ! to restore user-acl when walled garden disabled
 server 1 ipv4-address 192.169.7.232 port 1812 key myRadiusKeyString
exit
radius dmcoa group
 server-type ipv4-server
exit
domain#
```

Put Subscriber to Walled Garden via Radius COA

To initiate putting subscriber to walled garden, send the following four attributes through Radius COA call.

- **Acct-Session-Id** //user radius accounting ID
- **NetElastic-Portal-Mode** // turn on http redirect, 1-on, 0-off
- **NetElastic-HTTP-Redirect-URL** //walled garden http url address
- **Filter-Id** //acl rule profile when walled garden enable

Here is an example by accounting ID

```
# cat “Acct-Session-Id = 15687879783069791aabbcdddecd2” > walled_garden.txt
# cat “NetElastic-Portal-Mode  = 1” >>  walled_garden.txt
# cat “NetElastic-HTTP-Redirect-URL = http://172.19.100.215” >>   walled_garden.txt
# cat “Filter-Id = walled_garden_acl_profile” >> walled_garden.txt
# cat walled_garden.txt | radclient -x 192.168.25.117:3799 coa “radius_secret”
```

Here is the same example by user name

```
# cat “User-Name = 84-2b-2b-aa-86-4f” > walled_garden.txt
# cat “NetElastic-Portal-Mode  = 1” >>  walled_garden.txt
# cat “NetElastic-HTTP-Redirect-URL = http://172.19.100.215” >>   walled_garden.txt
# cat “Filter-Id = walled_garden_acl_profile” >> walled_garden.txt
# cat walled_garden.txt | radclient -x 192.168.25.117:3799 coa “radius_secret”
```

Remove Subscriber from Walled Garden via Radius COA

To remove subscriber to walled garden, you only need to send the following two attributes through Radius COA call.

- **Acct-Session-Id** //user radius accounting ID
- **NetElastic-Portal-Mode** // turn on http redirect, 1-on, 0-off

Here is an example

```
# cat “Acct-Session-Id = 15687879783069791aabbcdddecd2” > walled_garden_disable.txt
# cat “NetElastic-Portal-Mode  = 1” >>  walled_garden_disable.txt
# cat walled_garden.txt | radclient -x 192.168.25.117:3799 coa “radius_secret”
```

5.4 Enable Radius Accounting

netElastic’s vBNG supports Radius accounting. After enabled and setup, the vBNG will periodically send subscriber’s accounting records to the designated Radius accounting server. The following link is a table that lists the radius accounting attributes sent from the vBNG to radius accounting servers.

List of vBNG Radius Accounting Attributes.

To enable Radius accounting, follow the following steps:

Turn on Radius accounting

```
radius accounting-on   enable
```

Note: This is a global switch under config->radius

Create a Radius accounting group

```
radius accounting group radius_acct_grp
 server-type    ipv4-server
 timeout        3
 retry-times    3
 nas-ip-address 128.201.138.15
 algorithm      master
 dead-time      5
 dead-count     10
 flow-unit      byte
 server 1 ipv4-address 128.201.138.55 port 1813 key radiusKey
exit
```

Note: multiple Radius accounting servers can be specified in one accounting group.

Create a Radius accounting template

Under bras, create an accounting template, in which the Radius accounting group created in the last step is bound as shown below highlighted in red.

```
bras
 accounting radius_acct_tmpl
  accounting-type               radius
  accounting-update             600
  first-radius-accounting-group radius_acct_grp
  accounting-start-fail         online
  accounting-update-fail online
  accounting-update-immediately disable
  l2tp-accounting               vpdn-model
  user-name-format              strip-domain
  nas-port-format               class1
  called-station-id-format      class1
  nas-port-id-format class1
  calling-station-id-format class1
  invalid-vlan-tag              0
 exit
exit
```

In the accounting template, you can specify to let what accounting attributes to carry what user access attributes to the radius as accounting records. Here is the list of some of the most popular accounting attributes and their formats based on configured format specifications

- **User-Name**
  Assuming the general user name format is "userName@domain", where @ is the delimiter configured under the key domain-name-delimiter under bras configuration, the content of the User-Name attribute in the radius accounting records depends on how the key user-name-format is configured, which can take any of the following values:
  - **strip-domain**: "username"
  - **include-domain**: "userName@domain"
  - **only-domain**: "domain"

- **Nas-Port-Id**
  Nas-Port_Id can take the following values depending on how the key nas-port-id-format is defined. nas-port-id-format can take any of the following values:

  - **class1:** "slot=xx;subslot=xx;port=xx;vlanid=xx;vlanid2=xx". For Dot1q, vanid is vlan and vlanid2 will be 0. For QinQ, vanid is inner vlan and vlanid2 will be outer vlan.
  - **class2:** "slot=xx;subslot=xx;port=xx;vlanid=xx;vlanid2=xx". For Dot1q, vanid is vlan and vlanid2 will be 0. For QinQ, vanid is outer vlan and vlanid2 will be inner vlan.
  - **class3:** the attribute sting will be "netElastic eth 0/slot/subslot/port:{vlan|evlan.ivlan}"
  - **class4:** the attribute sting will be "{atm|eth|trunk} NAS_slot/NAS_subslot/NAS_port:XPI.XCI AccessNodeIdentifier/ANI_rack/ANI_frame/ANI_slot/ANI_subslot/ANI_port[:ANI_XPI.ANI_XCI]", where:
    - {atm|eth|trunk}: interface type with this mapping atm->ATM, eth->Ethernet, trunk->Ethernet trunk.
    - NAS_slot, NAS_subslot, NAS_port: NAS slot, subslot, and port number.
    - XPI: For ATM, XPI is the VPI value (0-255). For Ethernet, XPI is the PVLAN value (0-4095).
    - XCI: For ATM, XCI is the VCI value (0-65535). For Ethernet, XCI is that CVLAN value (0-4095).
  - **class5:** "Circuit-Id" for both IPoE and PPPoE
  - **keep-agent-circuit-id:** "eth slot/subslot/port:vlan.vlan"
  - **user-defined:** Define your own string format. The available elements are [slot, port, vlan, second-vlan, third-vlan]. Here is a sample user defined configuration "user-defined [slot port vlan second-vlan] format %d%d%d%d"

- **Nas-Port**
  Nas-Port is a 32-bit integer and its value changes depending on how the key nas-port-format is defined. nas-port-format can take any of the following values:
  - **class1: [**slotID(8bit)**][**subSlotID(4bit)**][**portID(8bit)**][**vlan (12bit)**].** For QinQ, only vlan field will be inner vlan ID. For example, Nas-Port 16785408 or 0x01002000 in hex can be interpreted as slotID=1, subSlotID=0, portID=2, and vlan=0.
  - **class2: [**slotID(12bit)**][**portID(8bit)**][**vlan(12bit)**].** For QinQ, only vlan field will be inner vlan ID.
  - **class3: [**slotID(3bit)**][**subSlotID(1bit)**][**portID(4bit)**][[**innerVlan(12bit)**] [**OuterVlan(12bit)**].**
  - **class4: [**slotID(8bit)**][**subSlotID(4bit)**][**portID(8bit)**][[**innerVlan(12bit)**].**
  - **class5: [**0(20bit)**][**innerVlan(12bit)**].**

- **Calling-Station-Id**
  The content of the Calling-Station-Id attribute in the radius accounting records depends on how the key calling-station-id-format is configured, which can take any of the following values:
  - **class1:** user MAC address in the format of "xx:xx:xx:xx:xx:xx"
  - **class2:** "certusnet#0/slot/subslot/port #{vlan|exVlan:inVlan}"
  - **class3 delimiter [delimiter char]:** "xx-xx-xx-xx-xx-xx@vlan", where "xx-xx-xx-xx-xx-xx" is the user mac separated by char "-", "@" is the configured delimiter char. "vlan" is the user's access vlan tag.
  - **class4:** PPPoE remote-id.

- **Called-Station-Id**
  The content of the Called-Station-Id attribute in the radius accounting records depends on how the key called-station-id-format is configured, which can take any of the following values:

- **class1**: webserver-redirect-ssid configured under domain
- **class2**: PPPoE service-name

Bind the accounting template in the domain template

In the subscriber's domain template, bind the Radius accounting template created in the last step as seen in the following example.

```
bras
 accounting radius_acct_tmpl
  accounting-type                radius
  accounting-update              600
  first-radius-accounting-group radius_acct_grp
  accounting-start-fail          online
  accounting-update-fail online
  accounting-update-immediately disable
  l2tp-accounting                vpdn-model
  user-name-format               strip-domain
  nas-port-format                class1
  nas-port-id-format class1
  calling-station-id-format class1
  invalid-vlan-tag               0
 exit
exit
```

Check Radius Accounting Access with radius-ping Test

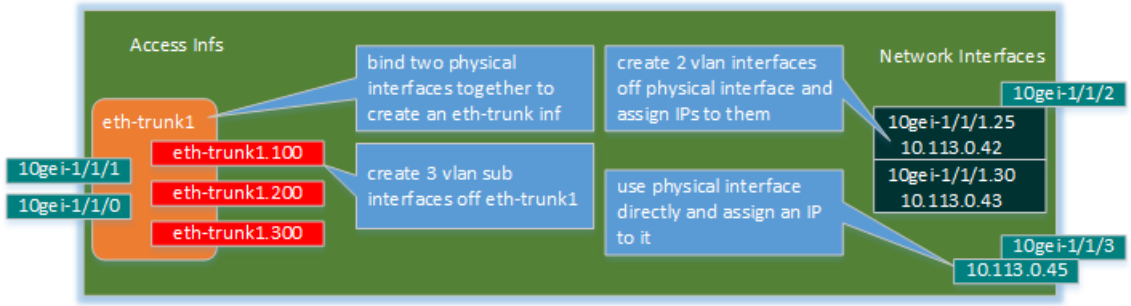
After setting up Radius accounting, the first thing you want to try is to ensure radius accounting can go through to the radius server. This can be tested by using command radius-ping as shown in the following example. The test exercises the whole radius accounting protocols with a test user as shown in the following example.

```
netelastic# radius-ping accounting group radius_acct_grp user-name 84-2b-2b-aa-86-
4f
Ping radius accounting-group radius_netElastic_accounting with 84-2b-2b-aa-86-4f at
2021-02-19 18:09:54!
Ping server 192.168.7.149 at 2021-02-19 18:09:54!
Reply from server 192.168.7.149 accept at 2021-02-19 18:09:54!
netelastic#
```

6 vBNG Configuration by Components.

6.1 Interface configuration

After installation and applying valid license, all physical interfaces specified as the forwarding interfaces during the installation will show up in the vBNG router. All user traffic enters or exits the vBNG through its forwarding interfaces. You configuration starts with configuration the interfaces first. The interfaces can be used directly or they can be lagged together to form eth-trunk interfaces. VLAN sub interfaces can be created off either physical interfaces or eth-trunk interfaces. The following diagram depicts these possibilities.



6.1.1 Trunk LAG interface configuration

To create eth-trunk interface, go to confd, enter configuration mode, then type "**interface eth-trunk[*integer*]**", where ***integer*** needs to be an integer larger or equal to 1.

Here is an example on how to create eth-trunk1

```
domain# config
Entering configuration mode terminal
domain(config)# interface eth-trunk1
domain(config-interface-eth-trunk1)#
```

Here is an example on an eth-trunk interface configuration.

```
domain# show running-config interface eth-trunk1
interface eth-trunk1
 description **Trunk-To-SW218**
 trunk-mode lacp-static
 balance    ip-simple
 trunk-port gei-1/1/0
  lacp port-priority 32768
  lacp timeout  long
 exit
 trunk-port gei-1/1/1
  lacp port-priority 32768
  lacp timeout  long
 exit
exit
```

After configuring trunk LAG interface, you can use the command "**show trunk-info**" to check trunk interface information including LAG status.

6.1.2 VLAN sub interface configuration.

VLAN sub interface can be of Dot1Q, dot1q range, QinQ, QinQ range, qinqinq (triple vlans), qinqiq-range. Any of these can be configured on a physical interface or on a trunk interface.

To get into sub interface configuration, you have to first create a sub interface off by appending .xxx to a parent physical or trunk interface, where xxx represents a valid vlan integer value. The same format applies to single, double, or triple VLAN sub interfaces. Here is an example that shows how to create a sub interface off a physical interface.

```
netelastic(config)# interface gei-1/1/1.100
netelastic(config-interface-gei-1/1/1.100)# sh fu
interface gei-1/1/1.100
exit
```

Single VLAN Interface Configuration.

The following example shows the dot1q vlan sub interface configuration off physical interface 10gei-1/1/1 with vlan 1902.

```
interface 10gei-1/1/1.1902
 ip tcp adjust-mss 1436
 ipv4 address 192.168.56.6 30
 dot1q 1902
exit
```

The following example shows the dot1q vlan sub interface configuration off physical interface gei-1/1/5 with vlan range from 1000 to 1005.

```
interface gei-1/1/5.1000
 dot1q-range 1000 to 1005
exit
```

If you have discrete single vlan values that need to be tied to one sub interface, you can enumerate the list such shown in the following example.

```
interface gei-1/1/1.700
 dot1q 701
 dot1q 705
 dot1q 710
 dot1q-range 712 to 750
exit
```

QinQ VLAN Interface Configuration.

QinQ involves use multiple VLAN tags in an Ethernet header so that one VLAN ID can carry another 4096 VLAN IDs in a second tag. This makes a simple and useful tunneling strategy.

The first/inner tag is the one set by the customer, and the second/outer (next to source MAC) tag would be set by the network. It's common in the Service Provider industry to refer the first/inner tag as Customer VLAN (C-VLAN) and second/outer tag as Service VLAN (S-VLAN).

The following example shows a qinq vlan sub interface configuration off a physical interface with C-VLAN 100 and S-VLAN 2000.

```
interface 10gei-1/1/1.100
 description "vlan inf with c-tag 100 and s-tag 2000"
 qinq internal 100 external 2000
exit
```

QinQ sub interface can also be configured as vlan range on any of the tags. The following example shows the qinq vlan sub interface configuration off a trunk interface with inner vlan range from 1 to 4094 and outer vlan 2668.

```
interface eth-trunk4.2668
 description **Hyperloop VIC**
 qinq-range internal 1 to 4094 external 2668 to 2668
exit
```

NOTE: For QinQ, we support both protocol 0x8100 and 0x88a8. If the QinQ VLANs are encapsulated in 802.1ad format, you should set the qinq-protocol to 88a8 as shown below.

```
interface eth-trunk4.2668
 description **Hyperloop VIC**
 qinq-range internal 1 to 4094 external 2668 to 2668
 qinq-protocol 88a8
exit
```

Triple VLAN Interface Configuration.

In rare situations, another vlan tag is added beyond the outer tag. This makes the Ethernet header has three vlan tags.

The following example shows a vlan sub interface configuration off a physical interface with three vlan tags. They are 100, 1000, 2000 from inner to the outmost (next to source MAC) position.

```
interface gei-1/1/1.100
 description "vlan inf with three tags "
 qinqinq outmost 2000 preoutmost 1000 inmost 100
exit
```

QinQinQ sub interface can also be configured as vlan range on any of the tags. The following example shows a vlan sub interface configuration off a physical interface with three vlan tags. They are 100, 1000-1500, 2000 from inner to the outmost position.

```
interface gei-1/1/1.100
 description "vlan inf with three tags "
 qinqinq-range outmost 2000 to 2000 preoutmost 1000 to 1500 inmost 100 to 100
exit
```

6.1.3 VGI interface and loopback interfaces.

The rule to create VGI Interface is the "vgi" string following by an integer. VGI is subscriber's gateway. It is a very important concept. In section 6.3, we will focus on VGI and its implications with other parts of the vBNG configurations.

Loopback interface can be created with the interface command. Loopback interface name is the "loopback" string followed by an integer. Loopback interfaces represent the "self" of the vBNG router. Typical uses of loopback interfaces are:

- Configure an IP to it and use it as the NAS port IP.
- Configure as the vBNG router ID with BGP peering.

6.1.4 Access interface v.s. network interface

By default, all interfaces (physical, trunk, or sub interfaces) are network interface upon creation. You can assign IP to them and they are all routable interfaces.

An interface becomes an access interface the moment it bound to a vci-configuration where it is tied to a PPPoE, or IPoE template that defines user's access behavior. As shown in the following example, eth-trunk3.1 and eth-trunk3.10 become access interface because they are bound under the vci-configuration.

```
bras
 vci-configuration
  interface eth-trunk3.1
   pppoe template my_pppoe_template
   max-ipox-session           32000
   max-pppox-session          32000
   encapsulation              multi
   access-delay 2000
   ip-access-type             ipv4
  exit
  interface eth-trunk3.10
   pppoe template my_pppoe_template
   max-ipox-session           32000
   max-pppox-session          32000
   encapsulation              multi
   access-delay 2000
   ip-access-type             ipv4
  exit
```

```
exit
```

6.2 ACL Configuration

vBNG supports flexible ACL rules that are based on matching a comprehensive set of L2 and L3 packet header fields. You can create as many as 1000 ACL group with each group can have as many as 4000 rules. The maximum of rules across all groups are limited to 32000.

ACL rules can be applied to

- Class maps
- Interfaces

The following shows a configuration example that controls the flows (white/black lists) on the network interface 10gei-1/1/0.

```
access-list in_from_network
 rule 10 permit ip source 173.243.64.7/32 destination any
 rule 20 deny tcp source any gt 0 destination any eq 111
 rule 30 deny tcp source any gt 0 destination any range 135 139
 rule 40 deny tcp source any gt 0 destination any range 161 162
 rule 50 deny tcp source any gt 0 destination any eq 445
 rule 60 deny tcp source any gt 0 destination any eq 520
 rule 70 deny tcp source any gt 0 destination any eq 1020
 rule 80 deny tcp source any gt 0 destination any range 1433 1434
 rule 90 deny tcp source any gt 0 destination any eq 2433
 rule 100 deny tcp source any gt 0 destination any eq 3306
 rule 110 deny tcp source any gt 0 destination any eq 3389
 rule 120 deny tcp source any gt 0 destination any eq 2179
 rule 130 deny tcp source any gt 0 destination any eq 593
 rule 200 deny udp source any gt 0 destination any eq 111
 rule 210 deny udp source any gt 0 destination any eq 135
 rule 220 deny udp source any gt 0 destination any range 137 139
 rule 230 deny udp source any gt 0 destination any range 161 162
 rule 240 deny udp source any gt 0 destination any eq 389
 rule 250 deny udp source any gt 0 destination any eq 445
 rule 260 deny udp source any gt 0 destination any eq 1434
 rule 300 permit tcp source any gt 0 destination 173.243.79.128/25 eq 25
 rule 310 permit tcp source any gt 0 destination 8.18.76.0/24 eq 25
 rule 320 permit tcp source any gt 0 destination 137.83.103.0/24 eq 25
 rule 4096 permit ip source any destination any
exit

access-list out_to_network
 rule 10 permit ip source 173.243.64.7/32 destination any
 rule 20 deny tcp source any gt 0 destination any eq 111
 rule 30 deny tcp source any gt 0 destination any range 135 139
 rule 40 deny tcp source any gt 0 destination any range 161 162
 rule 50 deny tcp source any gt 0 destination any eq 445
 rule 60 deny tcp source any gt 0 destination any eq 520
 rule 70 deny tcp source any gt 0 destination any eq 1020
 rule 80 deny tcp source any gt 0 destination any range 1433 1434
 rule 90 deny tcp source any gt 0 destination any eq 2433
 rule 100 deny tcp source any gt 0 destination any eq 3306
 rule 110 deny tcp source any gt 0 destination any eq 3389
 rule 120 deny tcp source any gt 0 destination any eq 2179
 rule 130 deny tcp source any gt 0 destination any eq 593
 rule 200 deny udp source any gt 0 destination any eq 111
 rule 210 deny udp source any gt 0 destination any eq 135
 rule 220 deny udp source any gt 0 destination any range 137 139
 rule 230 deny udp source any gt 0 destination any range 161 162
 rule 240 deny udp source any gt 0 destination any eq 389
 rule 250 deny udp source any gt 0 destination any eq 445
 rule 260 deny udp source any gt 0 destination any eq 1434
 rule 300 permit tcp source 173.243.79.128/25 gt 0 destination any eq 25
 rule 310 permit tcp source 8.18.76.0/24 gt 0 destination any eq 25
 rule 320 permit tcp source 137.83.103.0/24 gt 0 destination any eq 25
exit

interface 10gei-1/1/0
 description "External Interface"
 bind acl in ipv4 in_from_network
 bind acl out ipv4 out_to_network
```

```
 ipv4 address 173.243.64.130 30
exit
```

6.3 IPv4 Pool Configuration

IP Pools have to be configured on the vBNG in order or the vBNG to allocate IP addresses to subscribers. This is true even the subscriber's IP is allocated from Radius. In that case, you still need to configure corresponding IP pool, IP section blocks, and reserve sections that whose IPs are going to be allocated from Radius.

Since vBNG is subscriber's default gateway, which is represented by VGI interfaces on the vBNG, the IP Pool configuration is often related to VGI configuration described in section 6.4. The following is an example of typical IP pool configuration with its corresponding vgi configuration

```
ippool group localPool
 gateway-ip 10.10.10.1 gateway-mask 255.255.255.0
 lease-time        60
 dns-primary 8.8.8.8 secondary 8.8.4.4
 ippool-status     unlock
 warning-threshold 80
 warning-exhaust   disable
 frame-ip lease manage disable
 section start-ip 10.10.10.2 end-ip 10.10.10.200
  reserved-section reserved-start-ip 10.10.10.2 reserved-end-ip 10.10.10.20
 exit
exit

interface vgi1
 ipv4 address 10.10.10.1 24
exit
```

In the above example, we configure the gateway-ip from the pool to be the ipv4 address of the corresponding vgi interface.

Once IP pool and its corresponding vgi interface are defined, we need to bind them to the subscriber's access domain as a pair as illustrated in the following example.

```
bras
 domain myDomain
  bind authentication-template localAuthentication
  vgi                       vgi1
  domain-status             unlock
  user-routing-distribute enable
  tunnel-domain             disable
  flow-statistic            enable
  radius-attribute qos-acl-profile no-exist-policy offline
  quota-out offline
  bind-pool 1 localPool
 exit
exit
```

NOTE: vgi interface, once defined, has to be added to **bras->vgi-configuration** first before you can add it to a domain. If vgi interface is not in the **vgi-configuration** already, you will not be able to add it to user's access domain.

When dealing with IP pools that contain multiple subnets, the gateway configuration for PPPoE IP pool is slightly different from that for IPoE due to the point-to-point nature of PPPoE connections. We will use examples to discuss these two cases separately.

6.3.1 PPPoE IP Pool With Multiple Subnets

Here is an example for PPPoE IP Pool with two different subnet ranges. Since the two subnets do not overlap and cannot share a common gateway, we

have to user a 32-bit gateway access. The gateway address can be any valid IP from the IP range of any of the subnets as long as we use 32-bit network mask (highlighted in green text). We then create the corresponding vgi interface to use the same address as the gateway-ip set in the IP pool configuration. Finally we need to bind the IP pool and vgi in the user's access domain

```
ippool group local-PPPoE-Pool
 gateway-ip 10.10.10.1 gateway-mask 255.255.255.255
 lease-time         60
 dns-primary 8.8.8.8 secondary 8.8.4.4
 ippool-status      unlock
 warning-threshold 80
 warning-exhaust    disable
 frame-ip lease manage disable
 section start-ip 10.10.10.2 end-ip 10.10.10.200
 exit
 section start-ip 192.168.0.1 end-ip 192.168.10.255
 exit
exit

interface vgi1
 ipv4 address 10.10.10.1 32
exit

bras
 domain myDomain
  bind authentication-template localAuthentication
  vgi                       vgi1
  domain-status             unlock
  user-routing-distribute enable
  tunnel-domain             disable
  flow-statistic            enable
  radius-attribute qos-acl-profile no-exist-policy offline
  quota-out offline
  bind-pool 1 local-PPPoE-Pool
 exit
exit
```

6.3.2 IPoE IP Pool With Multiple Subnets

For IPoE with multiple subnets, we do need to configure multiple IP pools with corresponding gateways as the subscribers' gateways for the network they are in. However we still only need to configure one VGI. Here is an example for IPoE IP pools with two different subnet ranges. We create two IP pools and two sets of gateway IPs. We then create the corresponding vgi interface to use all the gateway IPs set in the IP pool configurations with one being the primary and the rest being secondary IPs. Finally we need to bind the IP pools and vgi in the user's access domain.

```
ippool group local-IPoE-Pool-1
 gateway-ip 10.10.10.1 gateway-mask 255.255.255.0
 lease-time         60
 dns-primary 8.8.8.8 secondary 8.8.4.4
 ippool-status      unlock
 warning-threshold 80
 warning-exhaust    disable
 frame-ip lease manage disable
 section start-ip 10.10.10.2 end-ip 10.10.10.254
 exit
exit

ippool group local-IPoE-Pool-2
 gateway-ip 172.20.10.1 gateway-mask 255.255.255.0
 lease-time         60
 dns-primary 8.8.8.8 secondary 8.8.4.4
 ippool-status      unlock
 warning-threshold 80
 warning-exhaust    disable
 frame-ip lease manage disable
 section start-ip 172.20.10.2 end-ip 172.20.10.254
 exit
exit
```

```
interface vgi1
 ipv4 address 10.10.10.1 24
 ipv4 address 172.20.10.1 24 secondary
exit

bras
 domain myDomain
  bind authentication-template localAuthentication
  vgi                          vgi1
  domain-status                unlock
  user-routing-distribute enable
  tunnel-domain                disable
  flow-statistic               enable
  radius-attribute qos-acl-profile no-exist-policy offline
  quota-out offline
  bind-pool 1 local-IPoE-Pool-1
  bind-pool 2 local-IPoE-Pool-2
 exit
exit
```

6.3.3 Check IP Pool Status

Use the following commands to display IP Pool status and usage information.

- **show ippool allocate-status** - display ip pool allocation status summary.
- **show ippool status** - display ip pool usage information by pools.
- **show ippool detail** - display ip pool individual IP allocation status.

6.4 DHCP Configuration

The vBNG router can be configured either as a DHCP server or as a relay agent at each interface level.

6.4.1 DHCP Configured as a Server

6.4.2 DHCP Configured as a Relay Agent

When the vBNG router is configured as a relay agent on an interface, the DHCP requests from that interface will be forwarded to external DHCP servers. vBNG will subsequently relay the DHCP responses from the external DHCP servers back to the clients with the option to insert certain DHCP options.

6.4.3 Configure DHCP Policies

6.4.4 Check DHCP Status

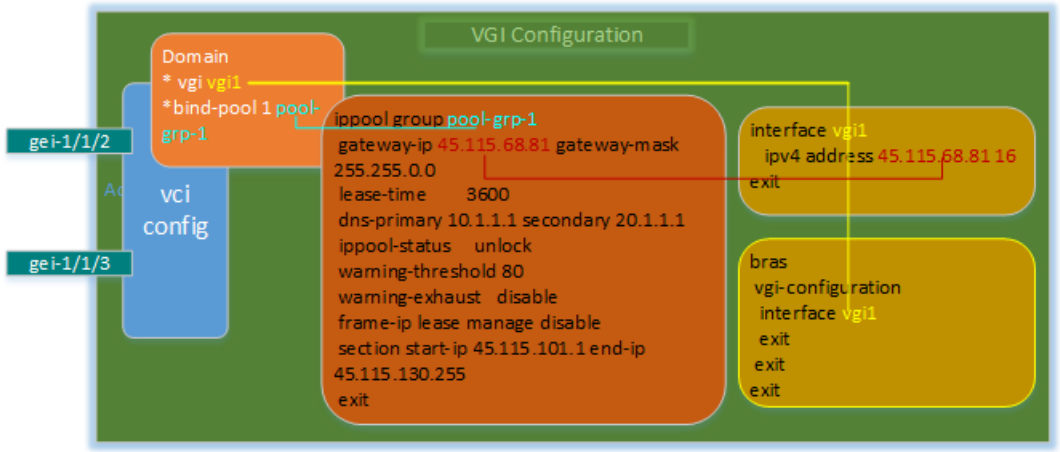
The command "**show dhcp detail | tab**" shows dhcp allocation status. You can customize what to display with the optional select command as shown in the example.

```
netelastic# show dhcp detail info interface | select ipv4-address | select option60 opt60-ascii | select option82 opt82-cid-
ascii | select option82 opt82-rid-ascii | select state | select expiration | select option12 opt12-ascii

                                                                                                    OPT82   OPT82
                                                                                  OPT60     OPT12   CID     RID
VPN                          IPV4                                                 ASCII     ASCII   ASCII   ASCII
ID   MAC ADDRESS              ADDRESS       STATE  EXPIRATION           INTERFACE  STRING    STRING  STRING  STRING
-------------------------------------------------------------------------------------------------------------------
0    00:21:70:d7:d3:ca  10.10.10.21  BOUND  2021-06-04 19:03:08  gei-1/1/5  MSFT 5.0  lab-PC
0    e4:b9:7a:88:f1:d5  10.10.10.22  BOUND  2021-06-04 19:02:53  gei-1/1/4  MSFT 5.0  Weixiao-PC
```

6.5 VGI Configuration

We have used VGI configuration earlier in the few configuration cases. Since VGI is such an important concept in user access networking and it needs to be configured or referenced in multiple places for access service to work properly, we devote this section to discuss VGI configuration on the vBNG.

The vgi configuration can be illustrated in the following diagram.



Here are the configuration steps.

1. Create vgi interface under interface configuration. The vgi interface name has to be in the format vgi followed by a numeric number, such as vgi1, vgi2, etc. When creating the vgi interface, you have to specify an IP address, which will serve as the gateway for subscribers whose access domain references this vgi.
2. Reference the created vgi interface under bras->vgi-configuration.
3. Reference the vgi ip address as the gateway ip in ippool configuration.
4. Finally tie the vgi interface and ippool to an access domain definition.

   NOTE: If you try to add vgi interface in a domain without adding it to vgi-configuration first as shown in step2, you won't be able to commit the configuration.

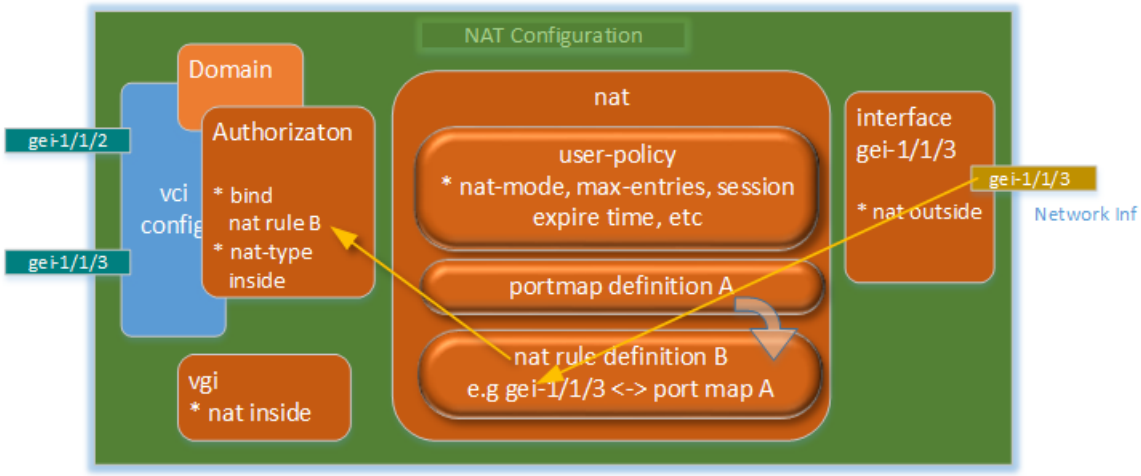
The following is a vgi configuration example

```
interface vgi55
 ipv4 address 45.115.68.81 16
exit

ippool group ipoe-mul
 gateway-ip 45.115.68.81 gateway-mask 255.255.0.0
 lease-time          3600
 dns-primary 10.1.1.1 secondary 20.1.1.1
 ippool-status       unlock
 warning-threshold 80
 warning-exhaust     disable
 frame-ip lease manage disable
 section start-ip 45.115.101.1 end-ip 45.115.130.255
 exit

bras
 vgi-configuration
  interface vgi55
  exit
 exit
exit

bras
 domain ipoe-mul
  bind authentication-template ipoe-mul
```

```
  bind authorization-template ipoe-mul
  vgi                         vgi55
  domain-status               unlock
  user-routing-distribute enable
  tunnel-domain               disable
  flow-statistic              enable
  radius-attribute qos-acl-profile no-exist-policy offline
  quota-out offline
  bind-pool 1 ipoe-mul
 exit
exit
```

6.6 CGNAT Configuration

netElastic's vBNG supports CGNAT, which can be configured with both dynamic and static rules. The configuration flow of CGNAT is illustrated in the following diagram.



The above diagram illustrates a NAT configuration example where users access through two access interfaces (gei-1/1/2 and gei-1/1/3) and users' traffic is NATed before routed out through the network interface (gei-1/1/3). The following lists the configuration flow:

Enable nat on the interfaces.

We need to enable nat on both the network interfaces (nat outside) and the access side user gateway (nat inside).

Here is the sample interface configuration for the case illustrated in the above diagram.

```
interface gei-1/1/3
 description "network inf"
 nat outside
 ipv4 address 10.10.0.169 24
exit
interface vgi1
 nat inside
 ipv4 address 172.20.0.1 16
exit
```

Enable nat in the authorization template.

Finally we need to enable nat in the authorization template. Here is a sample authorization template configuration for the case illustrated in the above diagram.

```
bras
 authorization myAuthorization
```

```
  authorization-type local
  bind nat-domain-name myNatRule
  nat-type           inside
  radius-nat-switch  disable
 exit
exit
```

nat configuration

6.6.1 Nat configuration with single public IP

nat configuration has four sub-sections.

1. user_policy: This is the session where nat mode, number of nat sessions, and session expiration times are configured.
2. log: here you can configure to enable or disable nat logging.
3. portmap: here you define the nat port starting value and port range size.
4. nat rules: here you define nat rules, which include both static and dynamic rules. The rules will bind the network interfaces or ip range to the portmap defined in step 3

Here is the sample nat configuration for the case illustrated in the above diagram

```
nat
 user-policy
  nat-mode                      full-cone
  working-form                  bras
  max-entries                   100000
  icmp-expire-time              20
  udp-expire-time               180
  tcp-expire-time               240
  tcp-fin-expire-time           30
  single-user-max-entries       1000
  alarm-enable                  disable
  alarm-total-entries-threshold 80
 exit
 log
           enable
  log-style type1
 exit
 portmap group myNat
  start-port 8000
  size       10000
  portrange-enable 1000
 exit
 rule group myNatRule
  type            dynamic
  radius-origin   disable
  ip-alloc-random disable
  if-name gei-1/1/3 portmap-name myNat
 exit
exit
```

6.6.2 Nat configuration with a pool of public IPs

In the example shown in Section 6.6.1, the outward facing NAT configuration is tied to the interface gei-1/1/3, which has a public IP configured. All NAT traffic leaving vBNG will have this IP as their source IP.

However, there are situations where all NAT users share a pool of public IP addresses. netElastic's vBNG allows you to configure the NAT so that this pool of public IP addresses are evenly distributed among all active NAT sessions. Here is a sample configuration for using public IP pool. Note

the public IP pool definition and its reference in the NAT rule definition as highlighted in red.

```
nat
 user-policy
  nat-mode                        full-cone
  working-form                    bras
  max-entries                     100000
  icmp-expire-time                20
  udp-expire-time                 180
  tcp-expire-time                 240
  tcp-fin-expire-time             30
  single-user-max-entries         1000
  alarm-enable                    disable
  alarm-total-entries-threshold 80
 exit
 log
           enable
  log-style type3
 exit
 ippool group cgnat_ipv4
  section start-ip 128.201.138.1 end-ip 128.201.138.127
 exit
 portmap group cgnat_ports
  start-port 3000
  size       60000
  portrange-enable 1000
 exit
 rule group myNatRule
  type            dynamic
  radius-origin   disable
  ip-alloc-random disable
  ippool-name cgnat_ipv4 portmap-name cgnat_ports
 exit
exit
```

When a pool of public IPs is used, there are different algorithms to map user sessions to available IPs. There are three algorithms we support, normal, pat, and, spr. Here is how to configure nat to enable the different mapping algorithms.

1. PAT (Port Address Translation) - NAT will use this algorithm when "portrange-enable" is not configured under portmap rules.
2. SPR- NAT will use this algorithm when "portrange-enable" is configured with non-zero values under portmap rules.
3. Normal - This is equivalent to static NAT rules.

Some applications such interactive gaming require NAT to support EIM (endpoint-independent mapping) and EIF (endpoint-independent filtering). For these use cases, you should configure NAT to use SPR algorithm by configuring "portrange-enable" under portmap group definition. There are two types of portrange-enable settings:

- **portrange-enable [N]**: Here **N** represents the maximum number of user sessions that NAT will reserve for the user. Each user will be allocated **N** number of sessions and within which User's public IP won't change. New sessions won't be able to be created once **N** sessions are exhausted.
- **portrange-enable [N] alarm-threshold [threshold] extend-port [M] extend-times [T]:** With this configuration, the subscriber will be allocated **N** initial sessions. When the usage percentage reaches **threshold**, **M** more sessions will be allocated. This process can repeat a maximum of **T** times. For example, the configuration "**portrange-enable 400 alarm-threshold 80 extend-port 800 extend-times 5**" means the subscriber will be allocated 400 initial sessions, when the usage reaches 80%, another 800 sessions will be allocated. This process can be repeated 5 times for a maximum total 5*800+400=4400 sessions.

6.6.3 Selectively NAT based on User IP Address

Often, you would encounter situations where subscribers can get either private IP or public IP (usually from Radius). In this case, you would only apply NAT to users with private IPs. To achieve this, we need to

1. Create an acl rule filter to match all users with private IPs.
2. Apply the acl rule filter to the nat rule so that only IPs that matches the acl rule filter will be subject to the nat rule.

Here is an example:

ACL configuration:

```
access-list private_ip_block
 rule 10 permit ip source 10.10.10.0/24 destination any
 rule 20 deny ip source any destination any
exit
```

Corresponding NAT configuration:

```
nat
 user-policy
  nat-mode                      full-cone
  working-form                  bras
  max-entries                   100000
  icmp-expire-time              20
  udp-expire-time               180
  tcp-expire-time               240
  tcp-fin-expire-time           30
  single-user-max-entries       1000
  alarm-enable                  disable
  alarm-total-entries-threshold 80
 exit
 log
  switch    off
  log-style type1
 exit
 ippool group cgnat_ipv4
  section start-ip 128.201.138.1 end-ip 128.201.138.127
 exit
 portmap group cgnat_ports
  start-port 3000
  size       60000
  portrange-enable 1000
 exit
 rule group my_nat_rule
  type            dynamic
  radius-origin   disable
  ip-alloc-random disable
  ippool-name cgnat_ipv4 portmap-name cgnat_ports acl-list-name private_ip_block
 exit
exit
```

6.6.4 Nat configuration with static NAT rules

For users who needs statically mapped NAT rules, we need to create rules that map the user's private IP and port to the desired public IP and port specification. The following example shows statically mapped NAT rue

```
rule group my_static_nat_rule
  type            static
  radius-origin   disable
  ip-alloc-random disable
  id 1 local-ip 172.15.1.10 global-ip 108.123.237.12 local-port 22 global-port 2222
  id 2 local-ip 172.15.1.20 global-ip 108.123.237.13
 exit
```

The above example shows two types of static NAT rule mapping:

Static Map with Port Specification:
Rule 1 specifies that private IP 172.15.1.10 with port 22 maps to public IP

108.123.237.12 on port 2222. This rule is equivalent to forward forwarding functionality found in traditional home routers. Only flows with ports explicitly specified in mapping rules will be mapped. Flows without mapping specifications will be ignored and not NATed.

Static Map without Port Specification:
Rule 2 specifies that all flows from private IP 172.15.1.20 are mapped to the public IP 108.123.237.13 on the same ports. This implies all traffic will be statically NATed.

When **working-form** of the NAT is configured as **bras**, a user can either be on a dynamic NAT rule or a static NAT rule. It cannot be on both. It then does not make a lot of senses to use static NAT rules for vBNG use cases. Since one static IP is always mapped to one public IP, you might as well assign those users public IPs and exclude them from going through NAT as described in section 6.6.3

Note: After the static NAT rule is created, it needs to be referenced in the authorization template for the user whose NAT behavior is subject to the static NAT rule defined.

6.6.5 Enable NAT Logging.

vBNG logs the session creation and deletion activities of all NAT sessions. Depending on configurations, NAT log can log these activities at different level of log details.

Enable NAT Logging and Log Locally

To enable NAT logging on the vBNG, create the following configurations under the **nat** configuration
For version 2019Q3 and prior, use the following format

```
nat
 log
           enable
  log-style type3
 exit
exit
```

For version 2020Q1 and later, use the following format

```
nat
 log
  switch    on
  log-style type3
 exit
exit
```

You also need to configure the following on the BNG to enable logging locally.

```
syslog facility local0
syslog severity all
syslog filesize 1024
syslog confd daemon false
syslog confd audit false
syslog confd netconf false
syslog confd snmp false
```

NOTE: If you make any changes to syslog configuration, you do have to restart syslog service for the changes to take effect beyond committing the changes. To restart syslog service, type command "**syslog restart**" in confd.

The NAT syslog records will be written to the file **/var/log/flexbng-syslog**. When its size reaches the specified file size limit under syslog configuration, the **/var/log/flexbng-syslog** log file will be backed up to **/var/log/flexbng-syslog.bak**. The size of **/var/log/flexbng-syslog** will be reset to 0 and begin to accept the log stream again.

To view local NAT log records, you can use the **journalctl** command. Here are some examples.

- **journalctl -f | grep NAT**
  display the newest NAT entries as they arrive in the journal.

- **journalctl --since today | grep NAT**
  show all NAT session logs since today

- **journalctl --since "2021-3-30" --until "2021-3-31" | grep NAT**
  show all NAT session logs within a particular day

- **journalctl --since "2021-3-30 22:53" --until "2021-3-31" | grep NAT**
  show all NAT session logs within a time frame on a particular day

Enable NAT Logging via Syslog

NAT logging messages can be sent to external syslog servers. To enable nat logging to syslog servers, create the following syslog configuration on the vBNG in addition to nat logging enablement configuration shown above.

To enable in-band (meaning syslog sent through forwarding interfaces) syslog, use the following reference configuration. Please note that:

1. You need to specify the interface from which syslog is sent out.
2. Make sure "syslog out-band" is not configured.

```
syslog facility local0
syslog source interface gei-1/1/4
syslog severity all
syslog filesize 1024
syslog server ip 10.155.20.24 port 514
```

To enable out-band (meaning syslog sent through system interfaces) syslog, use the following reference configuration.

```
syslog facility local0
syslog severity all
syslog filesize 1024
syslog out-band
syslog server ip 10.155.20.24 port 514
```

In the above reference configurations, 10.155.20.24 is the syslog server IP and 514 is the syslog server port number.

NOTE: If you make any changes to syslog configuration, you do have to restart syslog service for the changes to take effect beyond committing the changes. To restart syslog service, type command "**syslog restart**" in confd.

6.6.6 Check NAT Sessions and Status

Once NAT is configured, you want to check to make sure the NAT rules are applied to the intended users and NAT resources usages are normal. The following shows how to do display NAT related information on the vBNG.

Check User NAT Rule and Sessions

To check if a NAT rule has been applied to a subscriber, use the **show smgr-session detail user info mac-address [mac address]** or **show smgr-session detail user info ipv4-address [ipv4 address]**. Here is an example of user session detail printout with the nat rule applied to the user highlighted in red.

```
netelastic# show smgr-session detail user info ipv4-address 10.10.10.21
smgr-session detail user ipoe
 info
  mac-address       e4:b9:7a:88:f1:d5
  ip-access-type    ipv4
  auth-type         local
  auth-status       accept
  user-name         e4-b9-7a-88-f1-d5
  domain-name       myDomain
  author-domain     myDomain
  create-time       "2021-02-03 15:24:08"
  online-time       1268
  access-interface gei-1/1/4
  vlan              0
  vgi-interface     vgi1
  vrf-name          ""
  ippool-name       localPool
  ipv4-address      10.10.10.21
  gateway-address   10.10.10.1
  dns-v4            [ 8.8.8.8 8.8.4.4 ]
  accounting-info   acct-type:none
  nat-info          "nat-type:inside nat-domain:myNatRule public-ip:0.0.0.0 start-
port:0 end-port:0 nat-interval:0"
  family-info       "family-id:0 family-qos-profile:"
  policy-name       "acl: qos:user_qos_200000kbpsUp_200000kbpsDown user-group:"
  timeout           "session-timeout:0(second) prepay:-(second) -(kbyte) idle-
timeout: 0(second) 0(KB)"
  webforce-info     "webforce-flag:0 adforce-flag:0 special-acl: http-url:
advertisement-url:"
  subcar-input      "cbs:0(B) cir:0(kbps) pbs:0(B) pir:0(kbps)"
  subcar-output     "cbs:0(B) cir:0(kbps) pbs:0(B) pir:0(kbps)"
  unicast-traffic   "update-time:2021-02-03 15:45:14.615 up-stream:5971959(byte) up-
packets:13836 down-stream:4684894(byte) down-packets:12903"
  dropped-traffic   "update-time:2021-02-03 15:45:14.615 up-stream:0(byte) up-
packets:0 down-stream:0(byte) down-packets:0"
```

To list NAT sessions by user, user the command **show information data-plane nat-session rule user [user IP]**. Here is a sample output

```
netelastic# show information data-plane nat-session rule user 10.10.10.21
                          PRIVATE      PRIVATE                          PUBLIC   AGING
ID   RULE NAME   USER ADDR    ADDR         PORT     PROTO   PUBLIC ADDR  PORT     TIME
-------------------------------------------------------------------------------------
1    myNatRule   10.10.10.21  10.10.10.21  42270    TCP     10.10.0.169  14840    949 s
                              10.10.10.21  42282    TCP     10.10.0.169  14851    971 s
                              10.10.10.21  49161    UDP     10.10.0.169  16792    173 s
                              10.10.10.21  42216    TCP     10.10.0.169  14881    987 s
                              10.10.10.21  42328    TCP     10.10.0.169  14893    996 s
```

Check NAT Status and Statistics

Use **show nat status** to display the NAT session status. Here is a sample output with some important fields highlighted.

```
SLOT  TOTAL  TOTAL  TOTAL  TOTAL  TOTAL                                  USED
ID    USER   TCP    UDP    ICMP   SESS   INDEX  POOL NAME    MODE  CFG TBL   TBL     USAGE
SESS   SESS   SESS
-----------------------------------------------------------------------------------------
1     1445   62794  69380  98     132272 0      rule-nuron   SPR   13696000  613200  4.477%
```

- **TOTAL USER**: total number of users that are currently natted.
- **TOTAL SESS**: total number of active sessions.
- **POOL NAME**: the nat rule that is currently being applied.

- **MODE:** the NAT mode that is actively applied.
- **CFG TBL:** total number of possible NAT sessions. This equals [port size]x[number of public IPs].
- **USED TBL:** total number of session blocks that is currently allocated. Note this number can be different from the total number of active sessions. In SPR mode, each user is pre-allocated with a number of session blocks that is equal to the portrange size set under portrange statement. Even if a user is not using all allocated blocks, the allocated size will be counted as USED TBL.
- **USAGE:** this is [USED TBL]/[CFG TBL] in percentage.

Use **show nat statistic** or **show nat statistic overflow info** to display NAT session overflow users. Here is a sample output:

```
nuron-bng-snena# show nat statis
SLOT  TOTAL  OVERFLOW                                                    USED
ID    USER   USER      INDEX  PRIVATE ADDR    PUBLIC ADDR    PORTRANGE  PORTS  VRF
-------------------------------------------------------------------------------
1     1445   4         0      10.10.128.101   103.98.78.96   5200       3938
                       1      10.10.133.6     103.98.78.97   5200       3975
                       2      10.10.146.59    103.98.78.98   5200       3619
                       3      10.10.130.137   103.98.78.98   5200       3968

SLOT  TOTAL  EXHAUST         PRIVATE  PUBLIC            USED
ID    USER   USER     INDEX  ADDR     ADDR    PORTRANGE  PORTS  VRF
-------------------------------------------------------------------
1     1445   0
```

- **OVERFLOW USER:** This metric lists users who used up all preallocated initial sessions. If no session extensions for these users are enabled, these users will not be able to create new sessions. If these users are allowed to have extension, their currently allocated sessions will be listed.

  In the example shown above, there are four overflow users. Each of them is pre-allocated with 400 sessions with the possibility to extend. Their currently allocated sessions are 5200.
- **EXHAUST USER:** this metric lists user who used up all allocated sessions and could not create any new sessions. You want make sure the EXHAUST USER count is always 0

6.7 Setup QoS

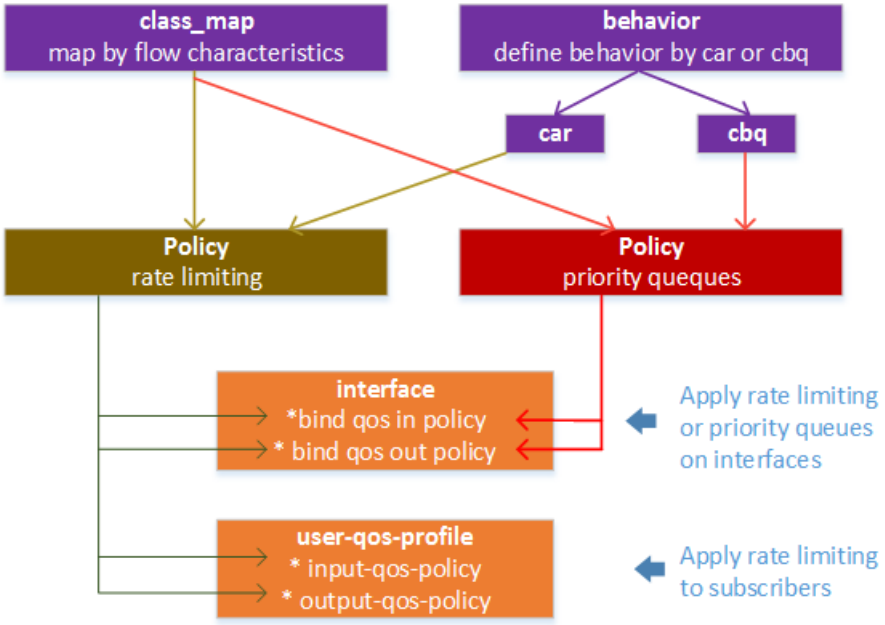
vBNG supports rate limiting and priority queues QoS for various traffic flows. Rate limiting QoS can be applied to subscribers, or interfaces, or both. Because of the high memory cost in implementing queues in software, vBNG is currently only supporting priority queues QoS on interfaces.

Setting up QoS on the vBNG involves the following steps.

1. Create class_map to define the flows for which QoS behaviours are intended to be applied on. class_map can be defined either directly by listing flow characteristics or by referencing defined acl lists.
2. Create intended behaviours for the class_map rules defined. The behaviours supported by vBNG are car, cbq, remark, etc.
3. Create policies to create class_map and behaviour pairs and setup the relative priority among them. Each policy can have up to 8 class_map/behaviour pairs.
4. QoS policies can be directly applied to interfaces.

5. If QoS policies need to be applied to subscribers, user qos profiles need to be created where both the upstream and downstream policies can be specified. The defined user qos profile is then referenced in the authorization template of the user's access domain. All users accessing through this domain are subject to the QoS policies defined in the user qos profile.

The following diagram depicts the relationship among these components.



6.7.1 QoS - Rate Limiting

The vBNG supports rate limiting QoS either through subcar or through definition of rate limiting QoS profiles. Each of these two methods has its own advantage and suitable use cases and they are meant to complement each other.

- **Subcar**
  - Easier to configure: minimal configuration is needed when used statically, or no configuration at all on the vBNG when used dynamically through Radius reply attributes or COA.
  - Limited flexibility: With subcar the rate limit applies to all traffic flows. Subcar cannot perform flow-based rate limiting.
- **QoS Profile**
  - More complicated to configure: needs to configure class map, behavior, policy, and then QoS profile.
  - Maximum flexibility: Using QoS profile, you can achieve maximum rate limiting flexibility based on many L2 and L3 flow characteristics.

Subcar Rate Limiting QoS

With subcar, you can independently control subscriber connection rate in both the upstream and downstream directions. Subcar can be statically configured as part of an authorization template or it can be dynamically assigned to subscriber with Radius reply or Radius COA.

- **Statically Configure Subcar**
  subcar can be configured as part of an authorization template as shown below. The authorization template will be referenced in the subscriber's access domain. See section 4.3 on how subscriber's access domain is determined.

```
bras
 authorization myAuthorization
  authorization-type local
  sub-car-input cir 2000 pir 2000 cbs 250000 pbs 250000
  sub-car-output cir 20000 pir 20000 cbs 2500000 pbs 2500000
  bind nat-domain-name myNatRule
  nat-type           inside
  radius-nat-switch  disable
 exit
exit
```

In the above example, the upstream rate is set to 2Mbps and downstream rate is set to 20Mbps. The parameters used are:

- cir – committed information rate in kbps
- pir – peak information rate in kbps
- cbs – committed block size. This is usually set to 125×cir
- pbs – committed block size. This is usually set to 125×pir

NOTE: In the above example, the "authorization-type" is set to "local". This means the subcar rate defined here will be honored by the vBNG. If the "authorization-type" is set to "radius" the subcar rate defined here won't be used at all. If the "authorization-type" to or "mix-radius", the subcar rate defined here will only be used if there are no subcar attributes coming from radius.

- **Dynamically Configure Subcar**

  Subcar can also be dynamically assigned from radius either as radius authentication reply attributes or via radius COA. The following VSA attributes are relevant for subcar configuration. See section 5.2.1 for complete list of COA attributes.

Attribute Name	Note
NetElastic-Input-Average-Rate	upstream cir (bps)
NetElastic-Input-Burst-Size	upstream cbs (125×cir(kbps))
NetElastic-Input-Peak-Rate	upstream pir (bps)
NetElastic-Input-Peak-Burst-Size	upstream pbs (125×pir(kbps))
NetElastic-Output-Average-Rate	downstream cir (bps)
NetElastic-Output-Burst-Size	downstream cbs (125×cir(kbps))
NetElastic-Output-Peak-Rate	downstream pir (bps)
NetElastic-Output-Peak-Burst-Size	downstream pbs (125×pir(kbps))

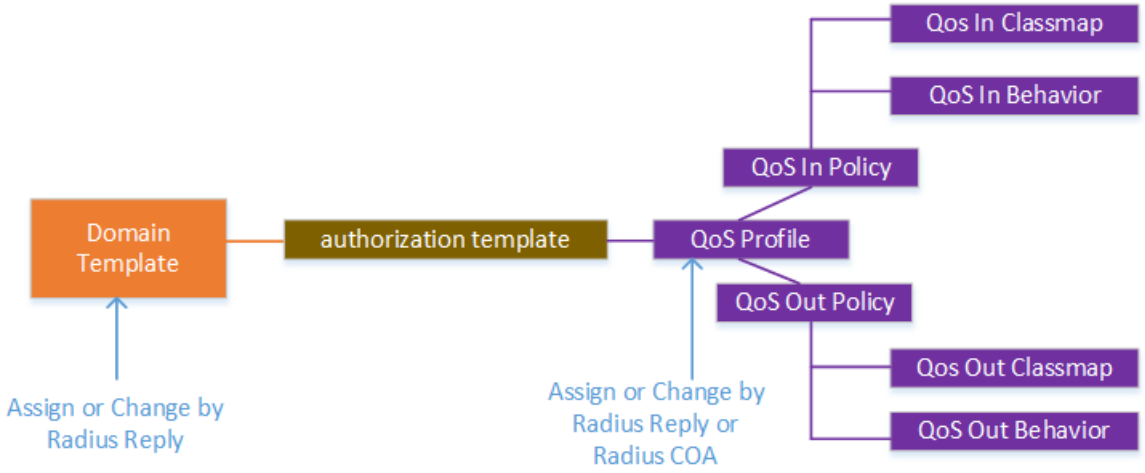
NOTE: Please note that unlike the rate units used in statically configured subcar parameters, the rate units in subcar parameters sent from radius is in bps instead of kbps.

For example, if you need to set 50 Mbps download and 20 Mbps upload for a user, Radius should reply with the following Radius attribute values to the user's access request or via radius COA.

Attribute Name	Operator	Value
NetElastic-Input-Average-Rate	:=	20000000
NetElastic-Input-Burst-Size	:=	2500000
NetElastic-Input-Peak-Rate	:=	20000000
NetElastic-Input-Peak-Burst-Size	:=	2500000
NetElastic-Output-Average-Rate	:=	50000000
NetElastic-Output-Burst-Size	:=	6250000
NetElastic-Output-Peak-Rate	:=	50000000
NetElastic-Output-Peak-Burst-Size	:=	6250000

QoS Profile Based Rate Limiting

A more flexible way to control subscribe connection rate is through creating user QoS profiles and then associate these profiles to subscribers either statically by domain/authorization template or by Radius reply message attributes. The QoS profile creation flow and related association to authorization and domain templates is illustrated below.



Next we will show some examples on how to create rate limiting QoS profiles. But before we do that, please note that the QoS profile can be dynamically associated to subscribers either by Radius reply attribute or Radius COA attribute, while domain can only be dynamically associated to subscribers by radius reply attribute.

Here is an example of creating QoS profile with the following requirements.

- Rate limit for Google cache at 3Mbps, up and down
- Rate limit for Facebook cache at 2Mbps, up and down
- Rate limit for rest of traffic at 6Mbps, up and down

1. We define an access lists for Facebook Cache (FB), Google Cache (GGC) to identify the flows associated with them. Here we are classifying the flows with destination IP ranges as shown below.

```
access-list FB
 rule 100 permit ip source any destination 185.125.148.64/26
 rule 101 permit ip source any destination 185.125.157.0/26
 rule 200 deny ip source any destination any
exit
access-list GGC
 rule 100 permit ip source any destination 185.4.253.192/27
 rule 200 deny ip source any destination any
exit
```

2. We define three class maps that match FB Cache, Google Cache, and all of the rest traffic flows.

```
class_map FB match-way match-all
 match ipv4-access-list FB
exit
class_map GGC match-way match-all
 match ipv4-access-list GGC
exit
class_map all match-way match-all
 match all
exit
```

3. We then create 3 CAR rate limiting behaviors.

```
behavior 2M
 item 1
```

```
  car cir 2000 pir 2000 cbs 250000 pbs 250000
 exit
exit
behavior 3M
 item 1
  car cir 3000 pir 3000 cbs 375000 pbs 375000
 exit
exit
behavior 6M
 item 1
  car cir 6000 pir 6000 cbs 750000 pbs 750000
 exit
exit
```

4. Next we define a QoS policy to tie the class maps together with the rate limiting behaviors defined. At this point, we can bind this policy to interfaces to apply rate limiting on the interfaces.

```
policy FB2M-GGC3M-ALL6M-Policy
 class_map FB behavior 2M priority 5
 class_map GGC behavior 3M priority 4
 class_map all behavior 6M priority 1
exit
```

5. Finally, if we want to apply these policies to subscribers, we need to create a rate limiting QoS profile by applying the above defined policy in both the upstream and downstream directions to create symmetric rate limiting in both directions (see below example). For asymmetrical rate limiting, you need to create two separate policies and tie them to both the input-qos-policy and the output-qos-policy in the user-qos-profile. Keep in mind that the direction connotation in "output-qos-policy" and "in-qos-policy" is derived from the subscriber's vantage point. "input-qos-policy" means "input" rate (download rate) for subscribers. "output-qos-policy" means "output" rate (upload rate) for subscribers.

```
bras
 user-qos-profile FB2M-GGC3M-ALL6M-Profile
  input-qos-policy  FB2M-GGC3M-ALL6M-Policy
  output-qos-policy FB2M-GGC3M-ALL6M-Policy
 exit
exit
```

Now that we have created the rate limiting QoS profile "FB2M-GGC3M-ALL6M-Profile", we can either statically assign it to an authorization/domain template or dynamically assign it to subscribers through Radius reply message (Attribute "NetElastic-Domain-Name") or by Radius COA (See section 5.2.1).

6.7.2 QoS - Priority Based Queues

vBNG has six priority queues that can be assigned each to a subscriber or an interface. The designator of these 6 queues are **ef**, **af1**, **af2**, **af3**, **af4**, **be**. Of these 6 queues, **ef** has the highest priority and **be** has the lowest priority. **af1, af2, af3, af4** are weighted fair queues whose priority is weighted proportionally to the bandwidth assigned.

Here we are showing a configuration example for the following use case:

- We have 4 flows that we want to assign to the 4 queues.
  - Voip traffic goes to the highest priority queue (ef)
  - Two video streams go to two weighted fair queues with one stream takes twice bandwidth than the other.
  - Everything else goes to the lowest priority queue(be)
- We need this policy to be applied to the network interface (gei-1/1/2) on the vBNG.

```
!create class maps based on flow characteristics
class_map all match-way match-all
 match all
exit
class_map video_cache1_map match-way match-any
 match ipv4-dest-address 122.1.1.55
exit
class_map video_cache2_map match-way match-any
 match ipv4-dest-address 122.1.1.59
exit
class_map voice_traffic match-way match-any
 match ipv4-dest-address 122.1.1.23
exit
!create priority queues as behaviors
behavior my_queue_af1
 item 1
  cbq queue af1 bandwidth 60000
 exit
exit
behavior my_queue_af2
 item 1
  cbq queue af1 bandwidth 30000
 exit
exit
behavior my_queue_be
 item 1
  cbq queue be
 exit
exit
behavior my_queue_ef
 item 1
  cbq queue ef
 exit
exit
!create policy to tie class maps to priority queues assignments
policy voip_and_video_policy
 class_map voice_traffic behavior my_queue_ef priority 8
 class_map video_cache1_map behavior my_queue_af1 priority 7
 class_map video_cache2_map behavior my_queue_af2 priority 6
 class_map all behavior my_queue_be priority 1
exit
!apply policy to interfaces to prioritize traffic by priority queuing
interface gei-1/1/2
 bind qos in voip_and_video_policy
 bind qos out voip_and_video_policy
exit
```

6.7.3 Time Based QoS

vBNG supports time frame based QoS switching. Once configured, QoS rules will switch automatically to desired rate or priority settings at the designated time frame. Time frame definitions repeat daily within a 24 hour period and are based on vBNG local time. Therefore, it is important to set the vBNG system clock to local time.

To configure, you define the time frames first and then use it as one of the matching criteria in ACL rules. The following is an example with the following time based rate control requirements:

- From 16:00-22:00 is prime peak with the bandwidth limited to 1M
- From 22:00-01:00 is secondary peak with bandwidth limited to 2M
- No limit on other time frames

Here is the configuration:

```
!define time range
time-range TR-16-22
 daily start 16:00:00 end 22:00:00
exit
time-range TR-22-01
```

```
 daily start 22:00:00 end 01:00:00
exit

! define classmap
class_map all_traffic match-way match-any
 match all
exit

!define CAR rate limiter for different time frames
behavior peak-limiter
 item 1
  car cir 1000 pir 1000 cbs 125000 pbs 125000
  tr-name TR-16-22
 exit
 item 2
  car cir 2000 pir 2000 cbs 250000 pbs 250000
  tr-name TR-22-01
 exit
exit

!define QoS policy
policy peak-policy
 class_map all_traffic behavior peak-limiter
exit

!define user qos profile
bras
 user-qos-profile peak-profile
  input-qos-policy  peak-policy
  output-qos-policy peak-policy
 exit
exit
```

NOTE 1: Not all behavior CAR or CBQ definitions need to be accompanied by a tr-name configuration. If you do not want a CAR or CBQ to be tied to any time frame limitation, simply don't configure tr-name under that item.
NOTE 2: item [id] determines match priority. If you have multiple CARs and CBQs under behavior definition with some of them tied to time frames and some of them not, always put the ones with tr-name specifications at the top. The smaller the item value, the higher is that item's priority.
NOTE 3: If all CARs or CBQs under behavior definition has tr-name specified and yet the union of tr-name does not cover the whole 24-hour period. The uncovered segment of the 24-hour period will not be subject to any CAR or CBQ rules.

6.8 L2TP Configuration.

In the case of handling L2TP connections, the vBNG router can be configured either as LAC client to initiate L2TP connections or as a LNS server to terminate L2TP connections.

6.8.1 L2TP LNS Configuration.

In the case of handling L2TP connections, vBNG can be configured as an LNS server to terminate L2TP connections from a L2TP LAC device so that subscribers can terminate on the vBNG across L2TP VPN links.

Another use case for LNS is private VPN connection for remote management of subscribers. Often it is desirable to have a VPN connection to the vBNG router so you can get an IP that is in the same subnet of the subscribers. By making your IP routable to subscribers, you can then remotely manage subscribers and perform customer CPE router management. This is especially true when the vBNG is deployed in an out-of-network environment such as a data center.

A sample LNS configuration on the vBNG is shown below:

```
bras
 pppox template l2tp-pppox-template
```

```
  check-magic-number disable
  ppp-authentication auto
  ac-name            ne-NLS
  mru                1492
  default-domain     <The domain name for your client connections>
  keepalive-time     60
  keepalive-count    3
  ppp-ncp-admit-any  disable
 exit
exit

l2tp group ne-LNS
 group-for               LNS
 access-address <L2TP LNS IP> interface <LNS Access Interface>
 retransmit-interval     3
 retransmit-max-times    5
 no-session-timeout      3
 check-session-id-in-zlb disable
 tunnel-authentication   disable
 tunnel-hello            60
 tunnel-hostname         netElasticLNS
 tcp adjust-mss          1460
 pppox template l2tp-pppox-template
exit
```

In addition to the above L2TP related configuration and pppox template, other related configurations include:

- **interface:** The L2TP connection interface needs to be configured with the LNS IP
- **authentication, authorization, and domain templates:** The configuration for these items are similar to how they are configured for normal PPPoE access.

Below is an example of a complete L2TP connection configurations with local authentication for incoming L2TP vpn connections. The associated ippool, vgi, authentication, authorization, domain, pppox template, local subscribers, and interface configurations are all included with the example. With these configurations, an L2TP VPN client should be able to dial in with login credentials "l2tp_conx_user1/l2tp_conx_user1". Once logged in, the client will get an IP from the pool range 10.10.10.2 to 10.10.10.200. The L2TP connection client should be able to ping vgi2 (10.10.10.1).

```
ippool group l2tp_conx_pool
 gateway-ip 10.10.10.1 gateway-mask 255.255.255.0
 lease-time        60
 dns-primary 8.8.8.8 secondary 8.8.4.4
 ippool-status     unlock
 warning-threshold 80
 warning-exhaust   disable
 frame-ip lease manage disable
 section start-ip 10.10.10.2 end-ip 10.10.10.200
 exit
exit

interface vgi2
 ipv4 address 10.10.10.1 24
exit

bras
 authentication l2tp_conx_authentication
  authentication-type      local
  user-name-format         strip-domain
  nas-port-format          class1
  called-station-id-format class1
  nas-port-id-format class1
  calling-station-id-format class1
  invalid-vlan-tag         0
 exit
exit

bras
```

```
 authorization l2tp_conx_authorization
  authorization-type local
 exit
exit

bras
 domain l2tp_conx_domain
  bind authentication-template l2tp_conx_authentication
  vgi                     vgi2
  domain-status           unlock
  user-routing-distribute enable
  tunnel-domain           disable
  flow-statistic          enable
  radius-attribute qos-acl-profile no-exist-policy offline
  quota-out offline
  bind-pool 1 l2tp_conx_pool
 exit
exit

bras
 local-subscriber l2tp_conx_user1 domain l2tp_conx_domain
  bind authorization-template l2tp_conx_authorization
  password l2tp_conx_user1
 exit
 local-subscriber additinalUser domain l2tp_conx_domain
  bind authorization-template l2tp_conx_authorization
  password additinalUserPassword
 exit
exit

bras
 pppox template l2tp-pppox-template
  check-magic-number disable
  ppp-authentication auto
  ac-name            ne-NLS
  mru                1492
  default-domain     l2tp_conx_domain
  keepalive-time     60
  keepalive-count    3
  ppp-ncp-admit-any  disable
 exit
exit

l2tp group ne-LNS
 group-for               LNS
 access-address 41.90.10.246 interface 10gei-1/1/2
 retransmit-interval     3
 retransmit-max-times    5
 no-session-timeout      3
 check-session-id-in-zlb disable
 tunnel-authentication   disable
 tunnel-hello            60
 tunnel-hostname         netElasticLNS
 pppox template l2tp-pppox-template
exit

interface 10gei-1/1/2
 description "l2tp lns interface"
 ipv4 address 41.90.10.246 28
exit
```

6.8.2 L2TP LAC Configuration.

When a vBNG router is configured as a LAC device, vBNG will concentrate PPPoE connections and forward PPPoE traffic to the LNS server where PPPoE connections will be eventually terminated. The LNS server will server IP addresses and gateways to the subscribers route their traffic to the next hop router.

To configure LAC, you need to create an access domain and its associated authentication, authorization, and accounting templates for PPPoE connections. With authentication template configuration, you have the

option to choose to authenticate the subscriber on the LAC device or passon the authentication to LNS. Here is the configuration sequence:

1. Create an authentication template. Set **authentication-type** to **none** if you want to pass authentication to LNS; Otherwise you can set authentication-type to **local** or **radius** if you choose to authenticate subscribers before sending their connections to LNS.
2. Create an authorization template. You can set the **authorization-type** to **none** for LAC since service level authorization will be actually done on the LNS device.
3. Create an accounting template and set the **accounting-type** to **none** as actual accounting will be done on the LNS device.
4. Create an access domain and bind the AAA templates defined in step 1, 2, and 3. Also the key **tunnel-domain** in the domain needs to be set to **enable**.
5. Create a **pppox template** for PPPoE subscribers to connect to this LAC device and set the key **default-domain** to the domain defined in step 4.
6. Add the access interface to **vci-configuration** and bind pppox template define in step 5
7. Configure an IP address on the interface by which l2tp tunnel will be established between this LAC device and the peering LNS device.
8. Finally create a LAC l2tp group where you will bind:
    a. The domain defined in step 4
    b. The IP address of peer LNS device.
    c. The LAC upstream interface and its address defined in step 7

The following is an example of LAC configuration with some of the important items highlighted.

```
bras
 authentication lac-authentication
  authentication-type         none
  user-name-format            strip-domain
  nas-port-format             class1
  nas-port-id-format class1
  calling-station-id-format   class1
  invalid-vlan-tag            0
 exit
exit

bras
 authorization lac-authorization
  authorization-type none
  nat-type           none
  radius-nat-switch  disable
 exit
exit

bras
 accounting lac-accounting
  accounting-type               none
  accounting-update             600
  accounting-start-fail         online
  accounting-update-fail online
  accounting-update-immediately disable
  l2tp-accounting               vpdn-model
  user-name-format              strip-domain
  nas-port-format               class1
  nas-port-id-format class1
  calling-station-id-format     class1
  invalid-vlan-tag              0
 exit
exit

bras
 domain lac-domain
  bind authentication-template lac-authentication
  bind accounting-template lac-accounting
  bind authorization-template lac-authorization
  domain-status        unlock
```

```
  tunnel-domain         enable
  flow-statistic        enable
  radius-attribute qos-acl-profile no-exist-policy offline
  quota-out offline
 exit
exit

bras
 pppox template lac-pppoe
  check-magic-number enable
  ppp-authentication auto
  ac-name            NETELASTIC-BRAS
  mru                1492
  default-domain     lac-domain
  keepalive-time     60
  keepalive-count    3
  ppp-ncp-admit-any  disable
 exit
exit

bras
 vci-configuration
  interface 10gei-1/1/3      ! pppoe access interface
   pppox template lac-pppoe  ! pppox template
   max-ipox-session  32000
   max-pppox-session 64000
   encapsulation     multi
   ip-access-type    ipv4
  exit
 exit
exit


interface 10gei-1/1/2
 ipv4 address 66.1.1.136 24  ! interface IP on which l2tp tunnel establish
exit

l2tp group lac
 group-for                LAC
 access-domain            lac-domain
 session-limit-per-tunnel 65535
 retransmit-interval      3
 retransmit-max-times     5
 no-session-timeout       3
 check-session-id-in-zlb  disable
 tunnel-authentication    disable
 tunnel-hello             60
 tunnel-hostname          netElasticLAC
 peer-ip 66.1.1.101 source-ip 66.1.1.136  bind-interface 10gei-1/1/2
 static-tunnel retry-timeout 120
exit
```

6.9 Router Configuration.

User traffic needs to be routed to the internet after authentication and authorization process. Multiple dynamic routing protocols can be configured on the router in addition to static route to exchange routes with neighbouring routers so user traffic can be routed to their desired destinations.

6.9.1 Enable User Routes

Although vBNG will create route entries for the networks associated with the IP pools configured on the vBNG, it, by default, does not create 32-bit route entries in the routing table for each individual subscriber. To instruct the vBNG to create 32-bit user routes, you have to enable it by setting the key "user-routing-distribute" value to "enable" in the subscriber's access domain configuration as shown in the following example.

```
bras
 domain myDomain
```

```
  bind authentication-template localAuthentication
  bind-addr-pool          ipoe_ipv6_addr_pool
  vgi                     vgi1
  domain-status           unlock
  user-routing-distribute enable
  tunnel-domain           disable
  flow-statistic          enable
  radius-attribute qos-acl-profile no-exist-policy offline
  quota-out offline
  bind-pool 1 localPool
 exit
exit
```

Then the user route entries will show up in the vBNG routing table as shown in the following example.

```
netelastic# show route database
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP, N - NAT, M - MAP-T
       O - OSPF, IA - OSPF inter area, U - User network route
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       > - selected route, * - FIB route, p - stale info
       IP Route Table for VRF default
S     *> 0.0.0.0/0 [1/0] via 10.10.0.1, gei-1/1/3
C     *> 10.10.0.0/24 is directly connected, gei-1/1/3
C     *> 10.10.0.169/32 is directly connected, gei-1/1/3
C     *> 10.10.10.0/24 is directly connected, vgi1
C     *> 10.10.10.1/32 is directly connected, vgi1
U     *> 10.10.10.21/32 [10/10] is directly connected, vgi1
U     *> 10.10.10.22/32 [10/10] is directly connected, vgi1
U     *> 10.10.10.24/32 [10/10] is directly connected, vgi1
S     *> 43.241.71.109/32 [1/0] via 108.217.237.214, gei-1/1/6
S        70.89.142.193/32 [1/0] via 108.217.237.214, vgi1 inactive
```

6.9.2 Set Up Static Routes

For static route configuration, the vBNG supports ifname (interface name), ifname-nexthop (interface name and next hop IP), and nexthop (next hop IP) as the next hop designation. Here are some sample configurations.

```
domain(config-router-static)# show full
router static
 ip route 0.0.0.0 0.0.0.0 ifname-nexthop gei-1/1/3 10.10.0.1
exit
```

```
router static
 ip route 0.0.0.0 0.0.0.0 nexthop 128.201.136.1
 ip route 10.180.1.0 255.255.255.0 nexthop 10.255.0.2
 ip route 128.201.137.16 255.255.255.255 nexthop 10.255.0.2
 ip route 128.201.137.81 255.255.255.255 nexthop 10.255.0.2
 ip route 128.201.137.96 255.255.255.224 nexthop 10.180.1.34
 ipv6 route :: 0 nexthop 2804:3ef0:c0ca:c0ca::1
exit
```

6.9.3 Set Up OSPF

Here is an example of OSFP configuration

```
router ospf instance 0
 router-id 10.255.0.13
 area 0.0.0.5
  authentication-mode null
  nssa
 exit
 interface 10gei-1/1/1
  network             point-to-point
  cost                10
  priority            1
  retransmit-interval 5
```

```
  authentication-mode null
  dead-interval       40
  hello-interval      10
 exit
 network 10.254.100.0/30 area 0.0.0.5
 redistribute connected metric-type 1
 redistribute nat metric-type 1
 redistribute static metric-type 1
 redistribute unr metric-type 1
exit
```

NOTE: Pay attention to the area type (nssa or stub) setting. They need to match the type setting on other routers to vBNG to successfully establish neighbors with other routers.

As shown in the above example,

- **network 10.254.100.0/30 area 0.0.0.5** distribute a network to an area.
- **redistribute connected** - distribute all connected routes to OSPF neighbors.
- **redistribute nat** - distribute nat routes to OSFP neighbors.
- **redistribute static** - distribute static routes to OSFP neighbors.
- **redistribute unr** - distribute user side routes to OSFP neighbors.

After you setup OSFP, you can use the following commands to reset and display OSPF related state and neighbor information.

- **clear-ospf-process** – reset ospf process so vBGN will re-establish neighbors with peering routers.
- **show ospf-state database all** - show ospf database
- **show ospf-state neighbor all detail** - show ospf neighbor information
- **show ospf-state route all** - show all ospf routes in the routing table

6.9.4 Set Up BGP

Setup BGP and Advertise Certain Internal Routes

Here is an example of BGP configuration with settings to advertise internal routes to neighbors through route-map configuration

```
prefix-list my-internal-IPV4 1 permit 128.201.136.0 22 le 24
route-map IX-OUT 1
 action permit
 match ip address prefix-list my-internal-IPV4
exit
router bgp 266630
 router-id 138.201.136.15
 address-family ipv6-unicast
  network-v6 2804:3ef0::/32
 exit
 address-family ipv4-unicast
  network 138.201.136.0/24
  network 138.201.137.0/24
  network 138.201.138.0/24
  network 138.201.139.0/24
 exit
 neighbor 188.16.198.252
  remote-as 20121
  address-family ipv4-unicast
   route-map IX-OUT out
  exit
 exit
 neighbor 188.16.198.253
  remote-as 266630
  shutdown
  address-family ipv4-unicast
   route-map IX-OUT out
  exit
 exit
```

```
exit
```

The above example shows how vBNG router is configured to exchange routes with neighbors through iBGP and eBGP.

Check BGP Status and BGP Routes

To check BGP status, use the command "**show bgp summary [*vrf-name*]**". Use ***default*** as the vfr-name for the default routing instance.

Setup BGP and Accept Certain External Routes

The router can also be configured to selectively accept certain incoming routes from it BGP neighbors. The following example shows the BGP configuration with settings to deny certain routes from neighbors through route-map configuration.

```
access-list bgp-route-acl    !create bgp route filter acl (black list routes)
 rule 100 deny ip source 192.0.1.0/24 destination any
 rule 200 deny ip source any destination 192.0.1.0/24
 rule 300 permit ip source any destination any
exit

route-map bgp-route-map 1    !create route map based on access list
 action permit
 match ip address access-list bgp-route-acl
exit

router bgp 1001              !bgp router configuration
 router-id 9.9.9.9
 aspath-access-list test deny 2001.*
 address-family ipv4-unicast
  network 5.5.5.0/24
 exit
 neighbor 8.8.8.8
  update-source loopback1
  remote-as     2001
  ebgp-multihop 2
  address-family ipv4-unicast
   route-map bgp-route-map in
  exit
 exit
exit
```

Setup BGP with AS filter

In the following example, BGP is setup to reject any AS path that starts with 2001 from certain neighbor.

```
router bgp 1001
 router-id 9.9.9.9
 aspath-access-list my_as_filter deny 2001.*
 address-family ipv4-unicast
  network 5.5.5.0/24
 exit
 neighbor 8.8.8.8
  update-source loopback1
  remote-as     2001
  ebgp-multihop 2
  address-family ipv4-unicast
   route-map bgp-route-map in   !bgp-route-map was defined in the last example
   filter-list my_as_filter in
  exit
 exit
exit
```

Clear Up BGP Routes

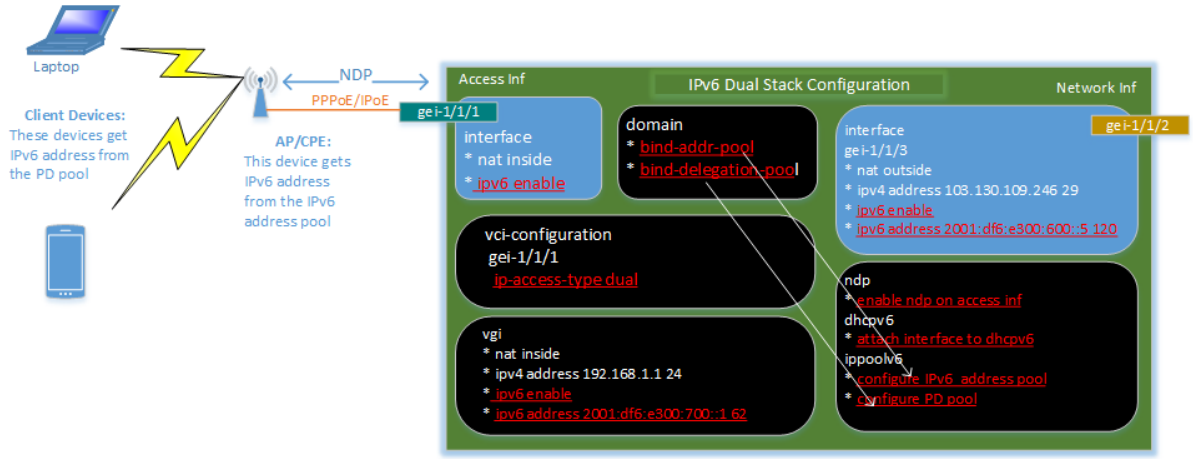
To clear all bgp routes without tearing down existing neighbors, use the "**clear-bgp all in**" command. Tis will trigger bgp to recalculate incoming routes. For this to work, the key "soft-reconfiguration-inbound" needs to be enabled for each neighbor that the incoming routes are coming from. The following example shows this configuration (highlighted in red).

```
router bgp 1001
 neighbor 1.1.1.1
  address-family ipv4-unicast
   soft-reconfiguration-inbound
  exit
 exit
exit
```

6.10 IPv6 Dual Stack Configuration

netElastic's vBNG router supports dual stack IPv6 address allocation for subscribers. It not only supports IPv6 address allocation to client devices, but also supports prefix delegation to routers and home gateways with prefix delegation capabilities. Here is an illustration of a typical use case that involves both IPv6 address and PD delegation allocation from the vBNG.

vBNG router supports both stateful and stateless IPv6 address allocation. With stateful IPv6 address allocation, the CPE device (e.g. the AP in the following diagram) gets its IPv6 address from the configured IPv6 address pool. If the CPE device (e.g. the AP in the following diagram) is configured for stateless IPv6 address allocation, it will get IPv6 prefix from the configured prefix pool and generate the remainder of the whole 128 bit IPv6 address locally on the CPE.



As illustrated above, the vBNG dual stack configuration has the following components:

Enable IPv6 on related network, access and vgi interfaces

```
interface gei-1/1/1          !access interface
 mtu-v6 1480
 nat inside
 ipv6 enable
exit
interface gei-1/1/2          !network interface
 nat outside
 ipv4 address 103.130.109.246 29
 ipv6 enable
 ipv6 address 2001:df6:e300:600::5 120
```

```
exit
interface vgi1              !vgi interface
 nat inside
 ipv4 address 192.168.1.1 24
 ipv6 enable
 ipv6 address 2001:df6:e300:700::1 62
exit
```

Enable NDP on the related access interface

```
ndp
 interface gei-1/1/1       !run ndp on this access interface
  nud reachable-time 30000
  dad attempts 1 interval 1000
  ra suppress     disable
  ra interval     300
  ra router-lifetime 1800
  ra hop-limit    64
  ra preference   medium
  ra auto-config managed-address enable  !enable stateful ip allocation
  ra auto-config other enable
 exit
exit
```

Some of the key elements of the ndp configuration:

ra auto-config managed-address: when "**ra auto-config managed-address**" is set to **enable** as show above, vBNG is set for stateful ip allocation. CPE client will get complete IP address from the IPv6 address pool. When this setting is set to **disable**, vBNG is set for stateless IPv6 address allocation. CPE client will get only get the prefix from the IPv6 prefix pool. The remainder of the complete IPv6 address will be generated locally on the CPE device.

ra suppres: when "**ra suppres**" is set to **enable**. vBNG will only respond to RS (Router Solicitation), and not proactively send RA (Router Advertisement). If you CPE device does not periodically send RS, you should sent this key to **disable** as shown in the example so vBNG will periodically send RA to keep the link connected.

ra auto-config other: Always set "**ra auto-config other**" **enable**. This is especially important when IPv6 addresses are allocated in DHCH v6 manner.

Enable dhcpv6 on the related access interface

```
dhcpv6
 dhcp enable              !globally enable dhcpv6
 pool my_dhcpv6_grp       !define a dhcpv6 property grp
  life-time valid-lifetime 3600 preferred-lifetime 3600
 exit
 interface gei-1/1/1      !enable dhcpv6 on this inf
  mode       server
  dhcp-pool my_dhcpv6_grp  !associate the inf to the above-defined property grp
 exit
exit
```

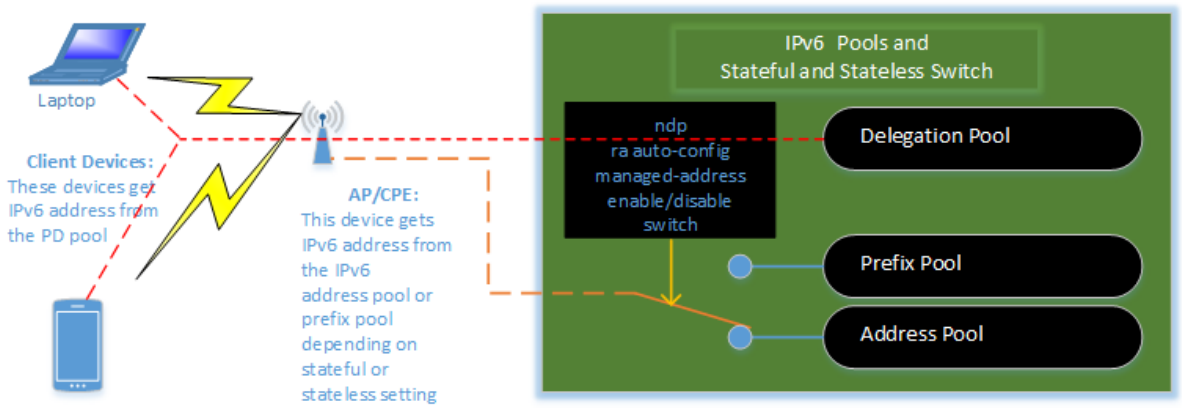
Create IPv6 prefix, address, and PD pools

```
ippoolv6 prefix-pool 1   !define dhcpv6 prefix pool
 prefix-address 2001:df6:e300:700::1 prefix-length 62
 dns-primary 2001:4860:4860::8888 secondary 2001:4860:4860::8844
 pool-status unlock
 assign-mode normal
 calc-mode   Byte
exit
ippoolv6 delegation-pool pppoe_ipv6_del_pool   !define PD pool
 prefix-address 2001:df6:e300:800::1 prefix-length 64 delegation-length 64
 pool-status       unlock   !the clients in above diagram get ip from this pool
 warning-threshold 80
```

```
 warning-exhaust   disable
exit
ippoolv6 addr-pool ipv6-pool  !define ipv6 address pool (the AP in the above
 pool-status       unlock     !diagram gets its ip from this pool)
 warning-threshold 80
 warning-exhaust   disable
 addr-range start-ipv6-ip 2001:df6:e300:700::2 end-ipv6-ip 2001:df6:e300:700::32
  gateway-address 2001:df6:e300:700::1
 exit
exit
```

As discussed earlier, three types of IPv6 pools can be configured on the vBNG. CPE and client devices can get IPv6 addresses from different pools depending on CPE device and vBNG configuration as shown in the following illustration.

- **Address Pool:** This is the address pool from which a device directly connected to vBNG (such as the AP in the diagram) gets its IPv6 address in stateful mode from the vBNG. The connected device will get a full 128-bit IPv6 address from the vBNG.
- **Prefix Pool:** This is the prefix address pool from which a device directly connected to vBNG (such as the AP in the diagram) gets its IPv6 prefix address in stateless mode from the vBNG. The rest of the full 128-bit IPv6 address of the device is generated by the device itself.
- **Delegation Pool:** This is the pool of IP pools vBNG will delegate to the connected device (such as the AP in the diagram). The connected device will use the obtained pool as the IPv6 pool for the devices that it will allocate IPv6 address to. For example, if you specify prefix-address in the delegation pool as "**prefix-address 2400:ca07:f037::1 prefix-length 52 delegation-length 64**", the vBNG can allocate 2^{12}=4096 delegation pools to possibly 4096 AP devices as shown in the following diagram with each pool having network prefix "2400ca07f03700". Each AP can have $2^{128-64}=2^{64}$ IPv6 addresses to allocate to it connected devices.



Enable dual stack on the related access interface under vci-configuration

```
bras
 vci-configuration
  interface gei-1/1/1
   ipoe template ipoe1
   pppoe template PPPoE1
   max-ipox-session  32000
   max-pppox-session 32000
   encapsulation     multi
   pre-domain        my_access_domain
   ip-access-type    dual   !enable dual stack on this interface
  exit
```

```
 exit
exit
```

Bind IPv6 pools in the related access domain

```
bras
 domain my_access_domain
  bind authentication-template my_auth_template
  bind authorization-template my_author_template
  bind-prefix-pool 1      1                   !bind prefix pool
  bind-delegation-pool    pppoe_ipv6_del_pool !bind PD pool
  bind-addr-pool          ipv6-pool           !bind address pool
  vgi                     vgi1
  domain-status           unlock
  user-routing-distribute disable
  tunnel-domain           disable
  flow-statistic          enable
  radius-attribute qos-acl-profile no-exist-policy offline
  quota-out offline
  bind-pool 1 public1
 exit
exit
```

6.11 Multicast Configuration

netElastic's vBNG supports multicast services so ISPs can provide services such as IPTV to their subscribers. To configure multicast on vBNG, follow these steps.

6.11.1 Enable Multicast on the BNG router

```
domain(config)# show full ip multicast-routing
ip multicast-routing
domain(config)#
```

6.11.2 Enable SM (Sparse Mode) PIM on the Network Interfaces

```
domain(config)# show full router pim
router pim interface gei-1/1/3
 address-family ipv4
  sm
 exit
exit
domain(config)#
```

6.11.3 Multicast Access Configuration

The multicast configuration on the access side involves the following steps:

Setup multicast template

```
domain(config-bras)# show full umgmd
bras
 umgmd profile ug
 exit
exit
domain(config-bras)#
```

Bind defined multicast template in authorization template

```
domain(config-bras-authorization-myAuthorization)# show full
bras
 authorization myAuthorization
  authorization-type local
  igmp               ug
  bind nat-domain-name myNatRule
  nat-type           inside
  radius-nat-switch  disable
 exit
exit
domain(config-bras-authorization-myAuthorization)#
```

Note: for multicast, the **authorization-type** has to be set to either **local** or **mix-radius** in the authorization template.

Enable SM (Sparse Mode) PIM on the user side VGI interface

```
domain(config)# show full-configuration router pim
router pim interface gei-1/1/3
 address-family ipv4
  sm
 exit
exit
router pim interface gei-1/1/2
 address-family ipv4
  sm
 exit
exit
router pim interface vgi1
 address-family ipv4
  sm
 exit
exit
domain(config)#
```

Note: This configuration is not done on the vgi interface directly. It is essentially binding vgi interface to the router pim configuration as we did early to the network interface in section 6.11.2. In the above example, router pim configuration lists all the interfaces where SM PIM is enabled, namely, the network interface (gei-1/1/3), the access interface (gei-1/1/2), and vgi interface (vgi1).

The above configurations covers basis igmp configuration on the vBNG. Subscribers shall be able to get online and join igmp groups. vBNG currently support igmp v1/v2/v3.

6.11.4 Advanced multicast configurations

Traffic duplication from one access port to another

Sometimes, it is required to duplicate all multicast traffic from one interface to another. In the following example, the vBNG is configuration to duplicate traffic from the access interface 10gei-1/1/1.10 to the access interface 10gei-1/1/1.200. Of course, mvlan interface 10gei-1/1/1.200 needs to be configured already under interface configuration.

```
bng1(config)# show fu bras vci-configuration interface 10gei-1/1/1.10
bras
 vci-configuration
  interface 10gei-1/1/1.10
   pppoe template HarbourISP
   max-ipox-session           32000
   max-pppox-session          32000
   encapsulation              multi
   access-delay 2000
   ip-access-type             ipv4
```

```
   authentication-method-ipv6 ppp
   mvlan-interface            10gei-1/1/1.200  /////need to cconfig mvlan interface
  exit
 exit
exit
bng1(config)#
```

Subscriber static join of a multicast group

The following example shows the configuration to allow a static user (user with IP225.1.1.10) to join a multicast group "*myIgmpGrp*"

```
domain(config-bras-umgmd-profile-myIgmpGrp)# show full
bras
 umgmd profile myIgmpGrp
  static-group 225.1.1.110
 exit
exit
domain(config-bras-umgmd-profile-myIgmpGrp)#
```

6.11.5 Check Multicast Status

vBNG has a few commands to check multicast status

Multicast User Status

```
domain# show umgmd
Possible completions:
  mroute               Route information
  packet-statistics    Userside igmp packet statistics
  statistics           Main statistics
  user                 Subscriber information
```

Multicast Related Table Information

```
domain# show ip pim
Possible completions:
  bsr-router   PIMv2 Bootstrap information
  interface    PIM interface information
  mroute       PIM mroute information
  neighbor     PIM neighbor information
  rp           RP information
```

6.12 SNMP Configuration

In this section, we will describe how to setup SNMP server on the vBNG, how to get and load netElastic's MIB files, and how to perform a test by snmpwalk.

6.12.1 Setup SNMP Server on the vBNG.

To enable SNMP server on netElastic's vBNG, login to confd and follow the following reference configuration. By default, the community string on netElastic's snmp server is set to "public".

```
domain# show running-config snmp-server
snmp-server agent enabled
snmp-server agent out-band enable true
snmp-server agent out-band port 161
snmp-server agent in-band enable true
snmp-server agent in-band port 161
snmp-server version v1 true
snmp-server version v2c true
snmp-server version v3 true
snmp-server packet-max-size 50000
```

```
snmp-server trap enable
snmp-server inform enable
snmp-server community public view-name public rw
snmp-server view public 1.3.6.1 included
snmp-server host 127.0.0.1 udp-port 162 trap-outband version v2c community public
domain#
```

6.12.2 SNMP In-band and Out-band Access

From the sample configuration shown above, you can see that vBNG support both in-band access (through vBNG router forwarding ports) and out-band access (through the host management ports). To enable one, or the other, or both, make sure the corresponding enable switches are set to true. The in-band or out-band ports configured mean that the vBNG SNMP agent will only accept connections whose source ports matches the ports configured.

6.12.3 Load netElastic's SNMP MIB Files

netElastic MIB files come with every vBNG version release package. Unpack the version package; there should be a **mib** folder under the unpacked root folder. The MIB files are located in the **mib/netelastic** folder.

The vBNG version package can be found either in the installer package under the folder **installer_root/resource/image** or in the version folder of the CP VM or host (**/usr/local/certus/version/**) after the installation.

How to load the MIB files depends on the SNMP client you use. Please refer to the user guide of the SNMP client you use on how to load third party MIB files. Here is an example on how to load MIBs if you are using **net-snmp** and **net-snmp-utils** packages on CentOS.

1. Copy the netElastic MIBs to **/usr/share/snmp/mibs** folder. If the folder does not already exit, create it.
2. Create snmp configuration file **snmp.conf** under the **/etc/snmp** directory. If the folder does not already exit, create it.
3. Add a line with "**mibs +ALL**" to the **snmp.conf** file. If you prefer to selectively adding individual MIBS instead of adding them all, you can add them one by one in **snmp.conf** as shown below.

   mibs +NETELASTIC-FLEXBNG-ALARM
   mibs + NETELASTIC-FLEXBNG-IPPOOL
   …

6.12.4 Test SNMP server by snmpwalk.

If you have installed **net-snmp-utils** packages, you can use snmpwalk to test the SNMP server access. Here is an example.

```
snmpwalk -v2c -Of -c public 127.0.0.1 1.3.6.1
```

Here I used local loopback IP as I installed snmpwalk on the same host where vBNG is installed. Here I used OID 1.3.6.1, all objects whose OID starts with 1.3.6.1 will be displayed. If you load netElastic's MIB correctly, you should see netElastic's OID objects printed out as shown below.

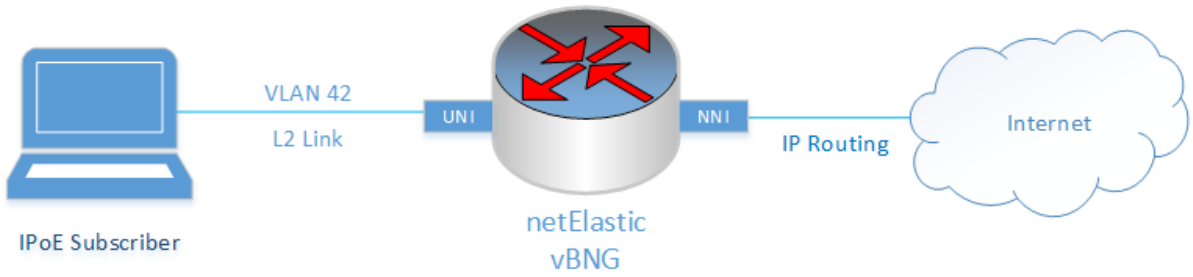
```
.iso.org.dod.internet.private.enterprises.netelastic.flexbng.bras.pppoeMib.connection-success.0 = Gauge32: 0
.iso.org.dod.internet.private.enterprises.netelastic.flexbng.bras.pppoeMib.connection.0 = Gauge32: 0
.iso.org.dod.internet.private.enterprises.netelastic.flexbng.bras.pppoeMib.discovery-timeout.0 = Gauge32: 0
.iso.org.dod.internet.private.enterprises.netelastic.flexbng.bras.pppoeMib.lcp-fail.0 = Gauge32: 0
.iso.org.dod.internet.private.enterprises.netelastic.flexbng.bras.pppoeMib.lcp-fail-other.0 = Gauge32: 0
.iso.org.dod.internet.private.enterprises.netelastic.flexbng.bras.pppoeMib.auth-fail.0 = Gauge32: 0
.iso.org.dod.internet.private.enterprises.netelastic.flexbng.bras.pppoeMib.auth-fail-other.0 = Gauge32: 0
```

7 vBNG Configuration Examples

Here we will provide step by step instructions on how to configure vBNG for various commonly used services.

7.1 IPoE Access without Authentication

Use Case Summary: In this use case, layer-2 connected IPoE subscribers are connected to the vBNG access interface with VLAN 42. The DHCP server on the vBNG assigns IP addresses to IPoE subscribers. The subscribers will be connected to the vBNG without authentication. The following diagram shows the network topology:



Configuration of the vBNG involves the following:

- Configure access interface and enable DHCP server on that interface
- Creating an IPoE template
- Creating a VGI
- Creating AAA (Authentication none only)
- Creating an IPPool
- Creating a domain
- Creating and configuring VCI

Create a sub-interface with VLAN 42

Typically, the data interfaces on the vBNG fall under two categories; user network interfaces (UNI) and network-to-network interfaces (NNI). UNI interfaces are typically L2 interfaces and NNI are L3 interfaces. When a vBNG is installed, all available data interfaces will be NNI interfaces by default. We have to put an interface under VCI configuration to make that interface a UNI interface.

For interfaces with VLANs, we need to create a sub-interface off the physical interface with the proper dot1Q or QinQ settings.

In this example, the vBNG has two interfaces; **gei-1/1/2** and **gei-1/1/3**. Let's use **gei-1/1/2** as the UNI and as **gei-1/1/3** NNI.

To configure **gei-1/1/2** as UNI with VLAN 42, we need to do the following.

1. Create a sub-interface off **gei-1/1/2** with VLAN 42
2. Put the sub-interface under vci-configuration to make it an UNI.

In this step, we will create the sub-interface first. We will configure the VCI later in section 0 as it also involves IPoE templates that we will configure next.

Here is how the VLAN 42 sub-interface gei-1/1/2.42 is created:

```
all-1-1# config
```

```
Entering configuration mode terminal
all-1-1(config)# interface gei-1/1/2.42
all-1-1(config-interface-gei-1/1/2.42)# dot1q 42
all-1-1(config-interface-gei-1/1/2.42)# commit
all-1-1(config-interface-gei-1/1/2.42)# exit
all-1-1(config)#
```

Note: The command "interface gei-1/1/2.42" will create sub-interface gei-1/1/2.42 if it does not exist. If the sub-interface already exists, command "interface gei-1/1/2.42" will enter edit mode for that sub-interface.

At this point, the interface configuration should look like this:

```
all-1-1(config)# show full-configuration interface
interface gei-1/1/2
exit
interface gei-1/1/2.42
 dot1q 42
exit
interface gei-1/1/3
exit
interface null0
exit
all-1-1(config)#
```

Enable DHCP service on the access sub interface

Now we need to enable DHCP service on the access sub interface. Follow the example below:

```
all-1-1(config)# dhcp
all-1-1(config-dhcp)# interface gei-1/1/2.42
all-1-1(config-dhcp-interface-gei-1/1/2.42)# commit
all-1-1(config-dhcp-interface-gei-1/1/2.42)# exit
all-1-1(config-dhcp)#
```

The DHCP configuration would look like this.

```
all-1-1(config-dhcp)# show full
dhcp
 dhcp enable
 relay max-user 128000
 relay option82 policy     keep
 relay option82 format     china-tel
 relay option82 user-configuration-policy interface
 interface gei-1/1/2.42
  mode       server
  user-quota 32000
 exit
exit
```

You can clearly see that dhcp is enabled on the interface gei-1/1/2.42 as shown in the configuration.

Create an IPoE Template

To enable IPoE access, we need to create an IPoE template, which will then be bound to an UNI interface through VCI configuration. In this IPoE Template, we need to specify that IPoE access should be granted without authentication. Below is how the IPoE template called "my_ipoe_template" is created with the specification of no authentication for IPoE access.

```
all-1-1# config
Entering configuration mode terminal
all-1-1(config)# bras
all-1-1(config-bras)# ipoe template my_ipoe_template
```

```
all-1-1(config-bras-ipoe-template-my_ipoe_template)# authentication-type ipv4
dhcpv4 none
all-1-1(config-bras-ipoe-template-my_ipoe_template)# commit
Commit complete.
all-1-1(config-bras-ipoe-template-my_ipoe_template)# end
all-1-1#
```

The complete IPoE template "my_ipoe_template" looks like this:

```
all-1-1(config-bras-ipoe-template-my_ipoe_template)# show ful
bras
 ipoe template my_ipoe_template
  authentication-type ipv4 dhcpv4 none
  authentication-type ipv6 dhcpv6 option
  dhcp-v4 auth-on-up password-type mac
  dhcp-v4 auth-on-up username-type mac
  dhcp-v4 auth-on-up domain-type optionparse
  dhcp-v6 auth-on-up password-type mac
  dhcp-v6 auth-on-up username-type mac
  dhcp-v6 auth-on-up domain-type optionparse
 exit
exit
```

Create Authentication Template

We need to create an authentication template. The authentication type "none" in the IPoE template does not mean "no authentication"; rather, it means the vBNG will be using the authentication template in the domain specified in the pre-domain (under vci_configuration). The various and flexible authentication methods offered in our vBNG such as Radius, local, or none can be enabled or disabled by specifying the appropriate parameters in the authentication template. In this example, we are specifying no authentication in the authentication template. Below is how the authentication template called "my_authentification_template" is created to specify no authentication. Keep in mind that the authentication template is not tied to a specific access method such as PPPoE or IPoE. An authentication template can be applied to both PPPoE and PPoE.

```
all-1-1# config
Entering configuration mode terminal
all-1-1(config)# bras
all-1-1(config-bras)# authentication my_authentication_template
all-1-1(config-bras-authentication-my_authentication_template)# authentication-type
none
all-1-1(config-bras-authentication-my_authentication_template)# commit
% No modifications to commit.
all-1-1(config-bras-authentication-my_authentication_template)# end
all-1-1#
```

The complete authentication template (my_authentication_template) looks like this:

```
all-1-1(config-bras-authentication-my_authentication_template)# show full
bras
 authentication my_authentication_template
  authentication-type none
  user-name-format    strip-domain
  nas-port-format     class1
  nas-port-id-format class1
  calling-station-id-format class1
  invalid-vlan-tag    0
 exit
exit
```

Create an IP Pool

Now we need to configure an IP pool from which IPoE access subscribers' IP address will be assigned via DHCP. netElastic's vBNG provides flexible IP pool configurations that can span multiple disjoint segments. In this example, we will configure one IP segment 172.16.1.1/24 with gateway IP 172.16.1.1.

```
all-1-1# config
Entering configuration mode terminal
all-1-1(config)# ippool group my_ippool
all-1-1(config-ippool-group-my_ipoe_ippool)# gateway-ip 172.16.1.1 gateway-mask
255.255.255.0
all-1-1(config-ippool-group-my_ipoe_ippool)# section start-ip 172.16.1.1 end-ip
172.16.1.254
all-1-1(config-ippool-group-my_ipoe_ippool-section-172.16.1.1/172.16.1.254)# commit
Commit complete.
all-1-1(config-ippool-group-my_ipoe_ippool-section-172.16.1.1/172.16.1.254)# end
all-1-1#
```

The ippool my_ipoe_ippool configuration looks like this:

```
all-1-1(config-ippool-group-my_ipoe_ippool)# show full
ippool group my_ippool
 gateway-ip 172.16.1.1 gateway-mask 255.255.255.0
 lease-time        3600
 ippool-status     unlock
 warning-threshold 80
 warning-exhaust   disable
 frame-ip lease manage disable
 section start-ip 172.16.1.1 end-ip 172.16.1.254
 exit
exit
```

Create a VGI interface

Subscribers need to have an access gateway configuration on the vBNG to have network access. netElastic's vBNG implements the concept of Virtual Gateway Interface(VGI) to configure subscriber's access gateway. To configure VGI, you need to:

1. Create a vgi interface and assign a gateway IP. This is done under **config->interface [vgi interface name]**. Note: the vgi interface name has be in the format of "vgi" followed by a numerical string such as vgi1, vgi2, etc. Other vgi interface names will not be accepted.
2. Specify the newly created vgi interface under **bras->vgi-configuration**

Note: the vgi interface IP address shall match the gateway address in the ippool configuration as described in section 0.

Here is an example of a vgi configuration with gateway IP 172.16.1.1/24:

```
all-1-1# config
Entering configuration mode terminal
all-1-1(config)# interface vgi1
all-1-1(config-interface-vgi1)# ipv4 address 172.16.1.1 24
all-1-1(config-interface-vgi1)# exit
all-1-1(config)# bras
all-1-1(config-bras)# vgi-configuration
all-1-1(config-bras-vgi-configuration)# interface vgi1
all-1-1(config-bras-vgi-configuration-interface-vgi1)# commit
Commit complete.
all-1-1(config-bras-vgi-configuration-interface-vgi1)# end
all-1-1#
```

The vgi-related configuration shall look like this.

```
all-1-1(config-interface-vgi1)# show full
interface vgi1
 ipv4 address 172.16.1.1 24
exit
all-1-1(config-bras-vgi-configuration)# show ful
```

```
bras
 vgi-configuration
  interface vgi1
  exit
 exit
exit
```

Create a domain

We have created an authentication template, an ippool, and a vgi interface. Now we need to create a domain to tie all these together and bind the domain to IPoE access to achieve the desired access behaviour. A user access domain defines user access behaviour. Multiple domains can be defined for the same access method to define different behaviours. User's access domains can be switched during operations (through Radius COA or command line) to alter access behaviours.

In the following example, an ipoe access domain called "my_ipoe_domain" is created to tie the defined authentication, ippoo, and vgi together.

```
all-1-1# config
Entering configuration mode terminal
all-1-1(config)# bras
all-1-1(config-bras)# domain my_domain
all-1-1(config-bras-domain-my_domain)# bind authentication-template
my_authentication_template
all-1-1(config-bras-domain-my_domain)# bind-pool 1 my_ippool
all-1-1(config-bras-domain-my_domain)# vgi vgi1
all-1-1(config-bras-domain-my_domain)# commit
Commit complete.
all-1-1(config-bras-domain-my_ipoe_domain)#
```

The domain my_ipoe_domain configuration should look like this.

```
all-1-1(config-bras-domain-my_domain)# show full
bras
 domain my_domain
  bind authentication-template my_authentication_template
  vgi                     vgi1
  domain-status           unlock
  user-routing-distribute disable
  tunnel-domain           disable
  flow-statistic          enable
  radius-attribute qos-acl-profile no-exist-policy offline
  quota-out offline
  bind-pool 1 my_ippool
 exit
exit
```

Create a VCI interface and bind with IPoE Template

Finally we need to create a VCI configuration to tie the IPoE template and the domain to the access interface so the access behaviour for traffic coming to the interface will be subject to what we have defined in the IPoE template and domain template.

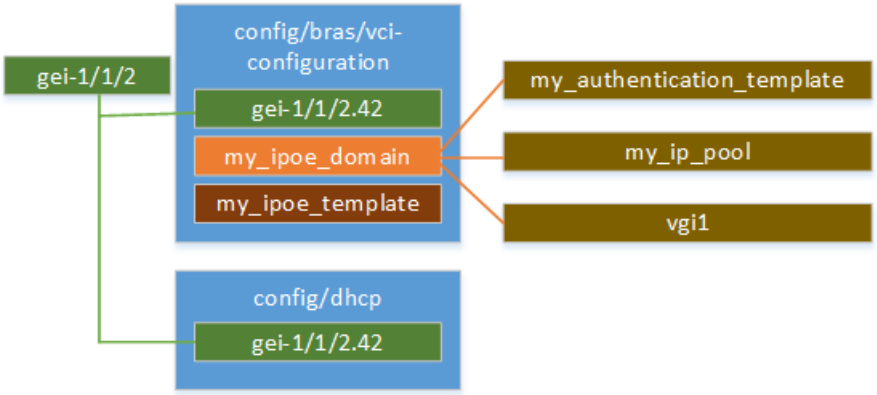
```
all-1-1# config
Entering configuration mode terminal
all-1-1(config)# bras
all-1-1(config-bras)# vci-configuration
all-1-1(config-bras-vci-configuration)# interface gei-1/1/2.42
all-1-1(config-bras-vci-configuration-interface-gei-1/1/2.42)# ipoe template
my_ipoe_template
all-1-1(config-bras-vci-configuration-interface-gei-1/1/2.42)# pre-domain
my_ipoe_domain
all-1-1(config-bras-vci-configuration-interface-gei-1/1/2.42)# commit
% No modifications to commit.
```

The VCI configuration should look like this:

```
all-1-1(config-bras-vci-configuration)# show full
bras
 vci-configuration
  interface gei-1/1/2.42
   ipoe template my_ipoe_template
   max-ipox-session            32000
   max-pppox-session           32000
   encapsulation               multi
   pre-domain                  my_ipoe_domain
   ip-access-type              ipv4
   authentication-method-ipv6 ppp
  exit
 exit
exit
```

Configuration Summary

The following graph shows the configuration logic between the various components and how they were tied together to provide the desired IPoE access service without authentication.



7.2 IPoE Access with Local Authentication and QoS Plan.

Based on the case "IPoE Access without Authentication" described in section 7.1, let's add to the configuration so that user can have a QoS rate plan. Since QoS rate plans always tie to users, we have to create some "user" identity so that QoS plans can be associated with it. As we know IPoE connection does not have the concept of user identity in the form of a user name, we create local user identity based on its MAC address and use it to represent the user.

Create Rate Limiting QoS Plan Profile

We create a 2M rate limit plan profile for both upstream and downstream traffic as shown below.

```
class_map all_traffic match-way match-any
 match all
exit
behavior rate_limit_2m
 car cir 2000 pir 2000 cbs 250000 pbs 250000
exit
policy policy_2m
 class_map all_traffic behavior rate_limit_2m
exit
bras
```

```
 user-qos-profile profile_2m
  input-qos-policy  policy_2m
  output-qos-policy policy_2m
 exit
exit
```

Create authentication template

Instead of using "none" for authentication as described in section 7.1, we will use local authentication.

```
bras
 authentication localAuthentication
  authentication-type local
  user-name-format    strip-domain
  nas-port-format     class1
  nas-port-id-format class1
  calling-station-id-format class1
  invalid-vlan-tag    0
 exit
exit
```

Create the access domain

Specify authentication template defined above in the domain definition.

```
bras
 domain myDomain
  bind authentication-template localAuthentication
  vgi                          vgi1
  domain-status                unlock
  user-routing-distribute disable
  tunnel-domain                disable
  flow-statistic               enable
  radius-attribute qos-acl-profile no-exist-policy offline
  quota-out offline
  bind-pool 1 localPool
 exit
exit
```

Create IPoE template

Here we want to use the domain specified by **pre-domain** under vci-configuration as the authentication domain. As discussed in section 4.3.2, there are only two ways to use that domain defined there: one is to set **authentication-type ipv4 dhcpv4** to "none", the other is to set **authentication-type ipv4 dhcpv4** to "option" and then set **dhcp-v4 auth-on-up domain-type** to "pre-domain". In the configuration shown below, we specify to use "option" to authenticate and then specify use mac for authentication. We also specify to use the domain bound to pre-domain in the vci-configuration as the authentication domain.

```
bras
 ipoe template my_ipoe
  authentication-type ipv4 dhcpv4 option
  authentication-type ipv6 dhcpv6 web
  dhcp-v4 auth-on-up password-type mac
  dhcp-v4 auth-on-up username-type mac
  dhcp-v4 auth-on-up domain-type pre-domain
  dhcp-v6 auth-on-up password-type mac
  dhcp-v6 auth-on-up username-type mac
  dhcp-v6 auth-on-up domain-type option
 exit
exit
```

Create vci-configuration

Under the vci-configuration, specify the ipoe template defined above and the authentication domain defined above as pre-domain

```
bras
 vci-configuration
  interface gei-1/1/2
   ipoe template my_ipoe
   max-ipox-session  32000
   max-pppox-session 32000
   encapsulation     multi
   pre-domain        myDomain
   ip-access-type    ipv4
  exit
 exit
exit
```

Create Local IPoE User

Create a local IPoE user and specify its associated domain and authorization template. Here we directly associate the authorization template to the local user. Alternatively we could also bind the authorization template in "myDomain" definition as well.

Note: here we created a user using its MAC address. Username and password are case sensitive. It is important to user all lower cases for letters in the MAC address.

```
bras
 local-subscriber 84-2b-2b-aa-86-4f domain myDomain
  bind authorization-template myAuthorization_2m
  password 84-2b-2b-aa-86-4f
 exit
exit
```

Verification

After user is online, check its access details in show smgr-session to verify its QoS policy.

```
domain# show smgr-session all user info
           SESSION                          AUTH   AUTH                  IPV6     TUNNEL                          DOMAIN
USER TYPE  ID       MAC ADDRESS             TYPE   STATUS  IPV4 ADDRESS  ADDRESS  SESSION  USER NAME              NAME      CIRCUIT    VLAN
-------------------------------------------------------------------------------------------------------------------------------------
   ipoe    -        84:2b:2b:aa:86:4f       local  accept     172.20.0.3     -        -        84-2b-2b-aa-86-4f      myDomain  gei-1/1/2  0/0
```

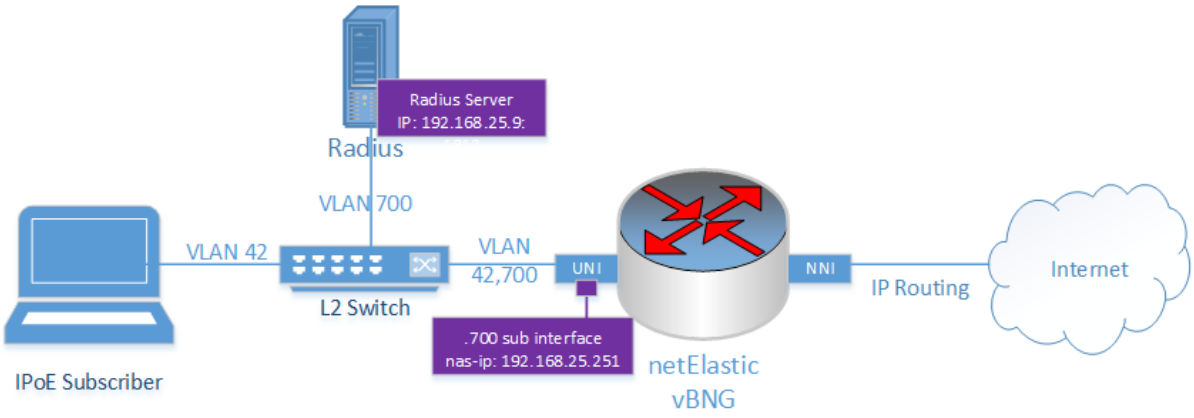
```
domain# show smgr-session detail user info
smgr-session detail user ipoe
 info
  mac-address      84:2b:2b:aa:86:4f
  ip-access-type   ipv4
  auth-type        local
  auth-status      accept
  user-name        84-2b-2b-aa-86-4f
  domain-name      myDomain
  author-domain    myDomain
  create-time      "2020-06-24 09:40:38"
  online-times     191300
  access-interface gei-1/1/2
  vlan             0/0
  vgi-interface    vgi1
  vrf-name         ""
  ippool-name      localPool
  ipv4-address     172.20.0.3
  gateway-address  172.20.0.1
  dns-v4           [ 8.8.8.8 8.8.4.4 ]
  accounting-info  acct-type:none
  nat-info         "nat-type:inside nat-domain:myNatRule public-ip:0.0.0.0 start-port:0 end-
port:0 nat-interval:0"
  family-info      "family-id:0 family-qos-profile:"
  policy-name      "acl: qos:profile_2m user-group:"
  timeout          "session-timeout:0(second) prepay:-(second) -(kbyte) idle-timeout:
0(second) 0(KB)"
  webforce-info    "webforce-flag:0 adforce-flag:0 special-acl: http-url: advertisement-url:"
  subcar-input     "cbs:0(B) cir:0(kbps) pbs:0(B) pir:0(kbps)"
```

```
  subcar-output    "cbs:0(B) cir:0(kbps) pbs:0(B) pir:0(kbps)"
  unicast-traffic  "update-time:2020-06-26 14:48:55.815 up-stream:57176170(byte) up-
packets:276520 down-stream:114954732(byte) down-packets:253482"
  mru              0
```

7.3 IPoE Access with Radius Authentication.

Refer to Section 2 for Radius server setup details if you do not have a radius server set up already.

Successful IPoE connections rely on the DHCP server running with vBNG to assign IP address. The network setup for this test is shown in the figure below:



The process of provisioning an IPoE session on vBNG involves:

- Configuring access sub interface with VLAN and enable DHCP server
- Creating an IPoE template
- Creating a radius authentication group
- Creating a VGI
- Creating AAA (Authentication none only)
- Creating an IPPool
- Creating a domain option60
- Creating and configuring VCI

Layer 2 Switch Configuration

For the setup to work properly, the switch shown in the figure above needs to be configured with the proper VLAN partitions. Please refer to the user guide of your layer 2 switch in your setup to set up the following configuration.

- Create VLAN 42
- Create VLAN 700
- Configure ports connected to subscribers to have access VLAN 42
- Configure port connected to Radius server to access VLAN 700
- Configure port connected to vBNG to trunk VLAN 42 and 700

Create an additional sub-interface and enable DHCP service on the access interface

The steps to create access sub interface with VLAN 42 and to enable dhcp are exactly the same as in the previous test case in section 7.1. Refer to

that section for the configuration of the sub-interface and the enablement of dhcp on that interface.

One additional sub interface we will need to create is the sub-interface for VLAN 700, through which CP needs to access the Radius server. We need to assign to this interface an IP address (nas port ip address) that is in the same network as the Radius server IP. In this case, 192.168.25.251 is assigned to this interface.

Here is how this is configured.

```
all-1-1# config
Entering configuration mode terminal
all-1-1(config)# interface gei-1/1/2.700
all-1-1(config-interface-gei-1/1/2.700)# dot1q 700
all-1-1(config-interface-gei-1/1/2.700)# ipv4 address 192.168.25.251 24
```

The interface gei-1/1/2.700 configuration should look like this.

```
all-1-1(config-interface-gei-1/1/2.700)# show full
interface gei-1/1/2.700
 ipv4 address 192.168.25.251 24
 dot1q 700
exit
```

Create an IPoE Template

For the IPoE template, we need to configure the following.

- Choose "option" for authentication-type.
- Choose "option" for dhcp-v4 auth-on-up domain-type so the domain associated with IPoE access will be domain option60. See access domain definition description in section 4.3.2.**Error! Reference source not found.**
- Set up so that IPoE access user name comes from DHCP option 60.
- Set up IPoE user's password to be "ipoe-password-netElastic". Keep in mind that this password is shared among all users using this IPoE template.

Perform the following tasks to create an IPoE template.

```
all-1-1# config
Entering configuration mode terminal
all-1-1(config)# bras
all-1-1(config-bras)# ipoe template my_ipoe_template
all-1-1(config-bras-ipoe-template-my_ipoe_template)# authentication-type ipv4
dhcpv4 option
all-1-1(config-bras-ipoe-template-my_ipoe_template)# dhcp-v4 auth-on-up password-
type config config-password ipoe-password-netElastic
all-1-1(config-bras-ipoe-template-my_ipoe_template)# dhcp-v4 auth-on-up username-
type option60
all-1-1(config-bras-ipoe-template-my_ipoe_template)# dhcp-v4 auth-on-up domain-type
option
```

The complete IPoE template should look like the following:

```
all-1-1(config-bras-ipoe-template-my_ipoe_template)# show full
bras
 ipoe template my_ipoe_template
  authentication-type ipv4 dhcpv4 option
  authentication-type ipv6 dhcpv6 option
  dhcp-v4 auth-on-up password-type config config-password ipoe-password-netElastic
  dhcp-v4 auth-on-up username-type option60
  dhcp-v4 auth-on-up domain-type option
  dhcp-v6 auth-on-up password-type mac
  dhcp-v6 auth-on-up username-type mac
  dhcp-v6 auth-on-up domain-type optionparse
 exit
exit
```

Create a RADIUS authentication group

Now we need to create a RADIUS authentication group that matches the network diagram shown above. Here are the configuration steps.

```
all-1-1# config
all-1-1(config)# radius vendor-id 54268
all-1-1(config)# radius authentication group my_radius_grp
all-1-1(config-radius-authentication-group-my_radius_grp)# nas-ip-address
192.168.5.251
all-1-1(config-radius-authentication-group-my_radius_grp)# server 1 ipv4-address
192.168.25.9 port 1812 key my_radius_key
all-1-1(config-radius-authentication-group-my_radius_grp)#
```

Note: the Radius vendor ID for netElastic is 54268

The configured Radius authentication group "my_ipoe_radius_grp" configuration should look like this.

```
all-1-1(config-radius-authentication-group-my_radius_grp)# show full
radius authentication group my_radius_grp
 timeout        3
 retry-times    3
 nas-ip-address 192.168.5.251
 algorithm      master
 dead-time      5
 dead-count     10
 class-as-car   disable
 filter-id-type user-acl
 server 1 ipv4-address 192.168.25.9 port 1812 key my_radius_key
exit
```

Create Authentication Template

For Radius authentication, we need to specify authentication type to use Radius. Here are the configuration steps.

```
all-1-1(config-bras)# authentication my_authentication_template
all-1-1(config-bras-authentication-my_authentication_template)# authentication-type
radius
all-1-1(config-bras-authentication-my_authentication_template)# radius-
authentication-group my_radius_grp
all-1-1(config-bras-authentication-my_authentication_template)#commit
```

The configured authentication template "my_authentication_template" should look like this:

```
all-1-1(config-bras-authentication-my_authentication_template)# show full
bras
 authentication my_authentication_template
  authentication-type         radius
  radius-authentication-group my_radius_grp
  user-name-format            strip-domain
  nas-port-format             class1
  nas-port-id-format class1
  calling-station-id-format class1
  invalid-vlan-tag            0
 exit
exit
```

Create Authorization Template

We also need to create an authorization template to instruct the BNG how subscribers can be allocated with resources. In the authorization template, you will normally specify user services such as user ACL rules, NAT rules, QoS profiles etc. As a minimum, we need to configure

authorization-type to specify how you would like BNG to authorize services for subscribers. Here is a sample of a minimal authorization template configuration. The value mix-radius for **authorization-type** means use radius attributes first and then locally configured attributes when attributes are not available from radius. This is the most commonly used value for **authorization-type** as it provides the most flexibility.

```
bras
 authorization myAuthorization
  authorization-type mix-radius
  radius-nat-switch  disable
 exit
exit
```

Create an IP Pool

The IP Pool configuration for Radius authentication is exactly the same as that in test case 7.1. It is shown below again for quick reference.

```
all-1-1(config-ippool-group-my_ipoe_ippool)# show full
ippool group my_ippool
 gateway-ip 172.16.1.1 gateway-mask 255.255.255.0
 lease-time         3600
 ippool-status      unlock
 warning-threshold 80
 warning-exhaust    disable
 frame-ip lease manage disable
 section start-ip 172.16.1.1 end-ip 172.16.1.254
 exit
exit
```

Create a VGI interface

The IP Pool configuration for Radius authentication is exactly the same as that in test case 7.1. It is shown below again for quick reference.

```
all-1-1(config-interface-vgi1)# show full
interface vgi1
 ipv4 address 172.16.1.1 24
exit
all-1-1(config-bras-vgi-configuration)# show ful
bras
 vgi-configuration
  interface vgi1
  exit
 exit
exit
```

Create a domain

Now we need to create a domain to bind the authentication template, vgi interface, and ip pool all together. Although the content of some of the templates are different, how we are tying the pieces together to form the access domain is exactly the same as test case 7.1. Please refer to section 0 for configuration details. One difference here is that we need to name the domain name as option60. This is required as we have specified to use option60 domain name in the ipoe template configuration. If the domain named option60 does not exist, the system will revert back to the domain specified in the pre-domain field in the vci configuration. See section 0 for how vBNG looks up access domain with Radius authentication.

```
all-1-1(config-bras-domain-my_domain)# show full
bras
```

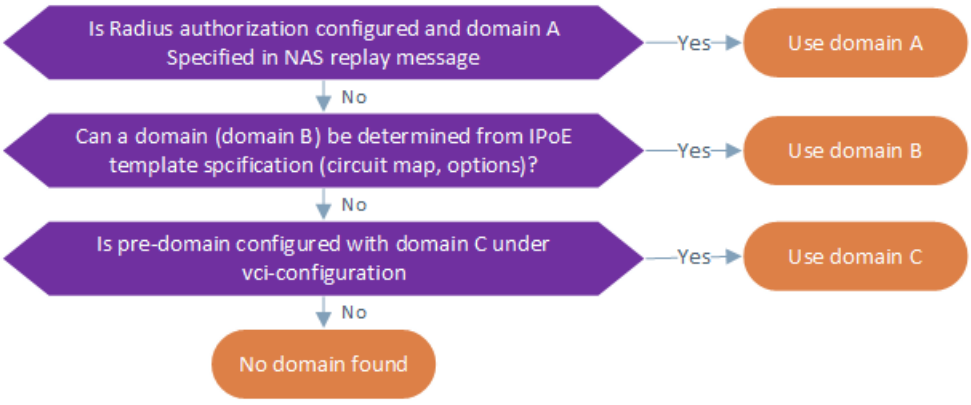
```
 domain option60
  bind authentication-template my_authentication_template
  bind authorization-template myAuthorization
  vgi                       vgi1
  domain-status             unlock
  user-routing-distribute disable
  tunnel-domain             disable
  flow-statistic            enable
  radius-attribute qos-acl-profile no-exist-policy offline
  quota-out offline
  bind-pool 1 my_ippool
 exit
exit
```

Create a VCI interface and bind with IPoE Template

Use vci-configuration to tie the access interface together with ipoe template and access domain. The configuration steps are exactly the same as that in test case 7.1. The vci-configuration should look like this.

```
all-1-1(config-bras-vci-configuration)# show full
bras
 vci-configuration
  interface gei-1/1/2.42
   ipoe template my_ipoe_template
   max-ipox-session           32000
   max-pppox-session          32000
   encapsulation              multi
   pre-domain                 my_domain
   ip-access-type             ipv4
   authentication-method-ipv6 ppp
  exit
 exit
exit
```

Note: In vci-configuration, we still assigned pre-domain with the defined my_ipoe_domain as the default access domain. As explained in section 0 , we have also specified to use domain60 as the access domain. Since we are using Radius for authentication, Radius can also replay with domain specification in the NAS reply message if Radius authorization is also enabled. The following flows chart shows how the access domain is determined by the vBNG.



Radius configuration

Since we are using Radius for authentication, we need to input user information (user name, password) into the Radius database.

7.4 IPoE Access with Static IP Assignments(IPhost)

Within the realm of IPoE access, there are cases when certain users want to have static IP assigned and while other users still have IP assigned by dhcp server dynamically. These static (IPhost) users will put static IPs on their devices connected to the vBNG. The vBNG will have to configured with corresponding provisions to acknowledge these users with static IPs, authenticate them, and give them appropriate resource authorization.

To enable IPhost access, the following configurations need to be added to related IPoE configurations.

Reserve IPhost IPs in the IP pool configuration.

The following sample shows an IP pool configuration with IPhost reserved IP highlighted in red.

```
ippool group IPoE-IPhost
 gateway-ip 105.105.1.1 gateway-mask 255.255.0.0
 lease-time          3600
 ippool-status       unlock
 warning-threshold 80
 warning-exhaust     disable
 frame-ip lease manage disable
 section start-ip 105.105.1.2 end-ip 105.105.30.255
  reserved-section reserved-start-ip 105.105.1.10 reserved-end-ip 105.105.20.255
  reserved-section reserved-start-ip 105.105.21.1 reserved-end-ip 105.105.21.255
  reserved-section reserved-start-ip 105.105.22.0 reserved-end-ip 105.105.22.0
  reserved-section reserved-start-ip 105.105.22.1 reserved-end-ip 105.105.30.255
 exit
```

Add IPhost users' IPs in their corresponding vgi-configuration.

The following sample shows a vgi-configuration with IPhost user entries highlighted in red. Note that you can enter either a singular entry or an IP range.

```
bras
 vgi-configuration
  interface vgi5
   ip-host 105.105.2.27 105.105.2.27 eth-trunk3.203 user-name certus domain-name
host password 123 first-vlan 203
   ip-host 105.105.21.1 105.105.60.255 eth-trunk3.203 domain-name host first-vlan
203
  exit
```

Note: If the iphost subscribers come in from a sub interface with dot1Q or QinQ vlan tags, it is VERY important to set the first-vlan and sec-vlan parameters to match the external and internal vlan tags.

Other related configurations such as authentication, authorization, domain, access template, and vci remain the same as their corresponding IPoE configurations.

To display IPhost access status, use the **show smgr-session detail user iphost** command.

```
domain# show smgr-session detail user iphost
smgr-session detail user iphost
 info
  mac-address       00:20:03:00:00:01
  ip-access-type    ipv4
  auth-type         none
  auth-status       accept
  user-name         ~
  domain-name       host
  author-domain     host
```

```
  create-time      "2019-09-30 13:15:13"
  online-times     17
  access-interface eth-trunk3.203
  vlan             203/0
  vgi-interface    vgi5
  vrf-name         host
  ippool-name      host
  ipv4-address     105.105.120.1
  gateway-address  105.105.1.1
  accounting-info  acct-type:none
  nat-info         "nat-type:none nat-domain: public-ip:0.0.0.0 start-port:0 end-
port:0 nat-interval:0"
  family-info      "family-id:0 family-qos-profile:"
  policy-name      "acl: qos: user-group:"
  timeout          "session-timeout:0(second) prepay:-(second) -(kbyte) idle-
timeout: 0(second) 0(KB)"
  webforce-info    "webforce-flag:0 adforce-flag:0 special-acl: http-url:
advertisement-url:"
  subcar-input     "cbs:0(B) cir:0(kbps) pbs:0(B) pir:0(kbps)"
  subcar-output    "cbs:0(B) cir:0(kbps) pbs:0(B) pir:0(kbps)"
  unicast-traffic  "update-time:2019-09-30 13:15:28.964 up-stream:0(byte) packets:0
down-stream:0(byte) packets:0"
  mru              0
```

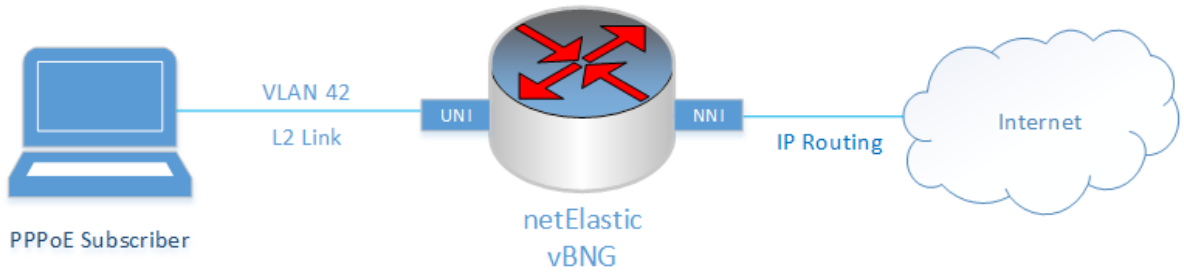
Note: IPhost is part of IPoE access. They share the same configurations such as vgi and vci except the two incremental configurations mentioned above. Iphost users will have to come in on the access interfaces where IPoE templates are bound in vci configuration.

Note: Please note the difference between IPhost and framed route. In both cases, the user's IP is statically assigned. The differences are:

- The IPs for IPhost users are assigned by the subscribers themselves. The vBNG learns their IP through ARP. Framed route IPs are assigned by the vBNG either dynamically assigned through Radius or statically configured on the vBNG for local subscribers as referenced in section 7.6
- IPhost is part of IPoE access scheme and does not apply to PPPoE. Framed route static IP assignment applies to both PPPoE and IPoE

7.5 PPPoE Access Without Authentication

Use Case Summary: In this use case, layer-2 connected PPPoE subscribers are connected to the vBNG access interface with VLAN 42. The DHCP server on the vBNG assigns IP addresses to IPoE subscribers. The subscribers will be connected to the vBNG without authentication. The following diagram shows the network topology:



The process of configuring PPPoE services on vBNG involves:

- Configuring access interface
- Creating an PPPoE template
- Creating a VGI
- Creating authentication template for no authentication
- Creating an IPPool

- Creating a domain
- Creating and configuring VCI

Create a sub-interface with VLAN 42

This configuration is exactly the same as the test case in section 7.1.
The configuration is shown here again for quick reference.

```
interface gei-1/1/2.42
 dot1q 42
exit
all-1-1(config)#
```

Create AAA Authentication Template

This configuration is exactly the same as the test case in section 7.1.
The configuration is shown here again for quick reference.

```
bras
 authentication my_authentication_template
  authentication-type none
  user-name-format    strip-domain
  nas-port-format     class1
  nas-port-id-format class1
  calling-station-id-format class1
  invalid-vlan-tag    0
 exit
exit
```

Create an IPPool

This configuration is exactly the same as the test case in section 7.1.
The configuration is shown here again for quick reference

```
ippool group my_ippool
 gateway-ip 172.16.1.1 gateway-mask 255.255.255.0
 lease-time         3600
 ippool-status      unlock
 warning-threshold 80
 warning-exhaust    disable
 frame-ip lease manage disable
 section start-ip 172.16.1.1 end-ip 172.16.1.254
 exit
exit
```

Create a VGI interface

This configuration is exactly the same as the test case in section 7.1.
The configuration is shown here again for quick reference

```
interface vgi1
 ipv4 address 172.16.1.1 24
exit
bras
 vgi-configuration
  interface vgi1
  exit
 exit
exit
```

Create a domain

This configuration is exactly the same as the test case in section 7.1.
The configuration is shown here again for quick reference

```
bras
```

```
 domain my_domain
  bind authentication-template my_authentication_template
  vgi                     vgi1
  domain-status           unlock
  user-routing-distribute disable
  tunnel-domain           disable
  flow-statistic          enable
  radius-attribute qos-acl-profile no-exist-policy offline
  quota-out offline
  bind-pool 1 my_ippool
 exit
exit
```

Create an PPPoE Template

Perform the following tasks to create a PPPoE template.

```
all-1-1(config-bras)# pppox template my_pppoe_template
all-1-1(config-bras-pppoe-template-my_pppoe_template)# ac-name netElastic-vBNG
all-1-1(config-bras-pppoe-template-my_pppoe_template)# default-domain my_domain
all-1-1(config-bras-pppoe-template-my_pppoe_template)# commit
```

The PPPoE-configured template should look like this.

```
all-1-1# show running-config bras pppoe template
bras
 pppox template my_pppoe_template
  check-magic-number enable
  ppp-authentication pap
  ac-name            netElastic-vBNG
  mru                1492
  service-name-omit  enable
  default-domain     my_domain
  quick-redial       disable
  keepalive-time     60
  keepalive-count    3
  check-ac-cookie    enable
 exit
exit
```

Create a VCI interface and bind with PPPoE Template

Perform the following steps to create a Virtual Circuit Interface (VCI) and bind it with PPPoE template.

```
all-1-1# config
Entering configuration mode terminal
all-1-1(config)# bras
all-1-1(config-bras)# vci-configuration
all-1-1(config-bras-vci-configuration)# interface gei-1/1/2.42
all-1-1(config-bras-vci-configuration-interface-gei-1/1/2.42)# pppoe template
my_pppoe_template
all-1-1(config-bras-vci-configuration-interface-gei-1/1/2.42)# commit
Commit complete.
```

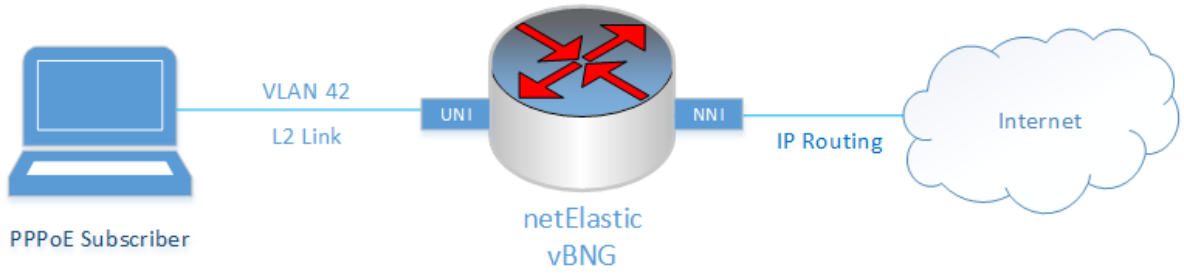
The vci-configuration should look like the following:

```
all-1-1(config-bras-vci-configuration-interface-gei-1/1/2.42)# show full
bras
 vci-configuration
  interface gei-1/1/2.42
   ipoe template my_ipoe_template
   pppoe template my_pppoe_template
   max-ipox-session           32000
   max-pppox-session          32000
   encapsulation              multi
   pre-domain                 my_domain
   ip-access-type             ipv4
   authentication-method-ipv6 ppp
  exit
 exit
```

```
exit
```

7.6 PPPoE Access With Local Authentication

This test case shows how to enable layer 2-connected PPPoE subscriber sessions with local authentication (on the vBNG). The test setup is shown as follows:



The process of provisioning a PPPoE session on vBNG involves:

- Configure access interface
- Configure a PPPoE template
- Configure a VGI
- Configure AAA (Authentication local only)
- Configure an IPPool
- Configure a domain
- Configure and configure VCI
- Configure local subscriber

Create a sub-interface with VLAN 42

This configuration is exactly the same as the test case in section 7.1. The configuration is shown here again for quick reference.

```
interface gei-1/1/2.42
 dot1q 42
exit
all-1-1(config)#
```

Create AAA Authentication Template

In the authentication template, we need to specify local authentication. Here are the configuration steps.

```
all-1-1# config
Entering configuration mode terminal
all-1-1(config)# bras
all-1-1(config-bras)# authentication my_authentication_template
all-1-1(config-bras-authentication-my_authentication_template)# authentication-type
local
all-1-1(config-bras-authentication-my_authentication_template)# commit
Commit complete.
```

The authentication template should look like the following:

```
all-1-1(config-bras-authentication-my_authentication_template)# show full
bras
 authentication my_authentication_template
  authentication-type local
  user-name-format    strip-domain
```

```
  nas-port-format     class1
  nas-port-id-format class1
  calling-station-id-format class1
  invalid-vlan-tag    0
 exit
exit
```

Create an IPPool

This configuration is exactly the same as the test case in section 7.1.
The configuration is shown here again for quick reference

```
ippool group my_ippool
 gateway-ip 172.16.1.1 gateway-mask 255.255.255.0
 lease-time         3600
 ippool-status      unlock
 warning-threshold 80
 warning-exhaust    disable
 frame-ip lease manage disable
 section start-ip 172.16.1.1 end-ip 172.16.1.254
 exit
exit
```

Create a VGI interface

This configuration is exactly the same as the test case in section 7.1.
The configuration is shown here again for quick reference

```
interface vgi1
 ipv4 address 172.16.1.1 24
exit
bras
 vgi-configuration
  interface vgi1
  exit
 exit
exit
```

Create a domain

This configuration is exactly the same as the test case in section 7.1.
The configuration is shown here again for quick reference

```
bras
 domain my_domain
  bind authentication-template my_authentication_template
  vgi                       vgi1
  domain-status             unlock
  user-routing-distribute disable
  tunnel-domain             disable
  flow-statistic            enable
  radius-attribute qos-acl-profile no-exist-policy offline
  quota-out offline
  bind-pool 1 my_ippool
 exit
exit
```

Create an PPPoE Template

This configuration is exactly the same as the test case in section 7.5.
The configuration is shown here again for quick reference.

```
bras
 pppox template my_pppoe_template
  check-magic-number enable
  ppp-authentication pap
  ac-name            netElastic-vBNG
  mru                1492
  service-name-omit  enable
```

```
  default-domain     my_domain
  quick-redial       disable
  keepalive-time     60
  keepalive-count    3
  check-ac-cookie    enable
 exit
exit
```

Create a VCI interface and bind with PPPoE Template

This configuration is exactly the same as the test case in section 7.57.4. The configuration is shown here again for quick reference.

```
bras
 vci-configuration
  interface gei-1/1/2.42
   ipoe template my_ipoe_template
   pppox template my_pppoe_template
   max-ipox-session           32000
   max-pppox-session          32000
   encapsulation              multi
   pre-domain                 my_domain
   ip-access-type             ipv4
   authentication-method-ipv6 ppp
  exit
 exit
exit
```

Create a local subscriber

Since we are using local authentication user, we need to create user entries on the vBNG that match the user name and password carried in the PPPoE packets. We can also specify the access domain for the PPPoE users in the local subscriber configuration. Below are the steps for creating local users on the vBNG:

```
all-1-1(config-bras)#  local-subscriber pppoe_user_1 domain my_domain
all-1-1(config-bras-local-subscriber-pppoe_user_1/my_domain)# password
pppoe_user_1_passwd
```

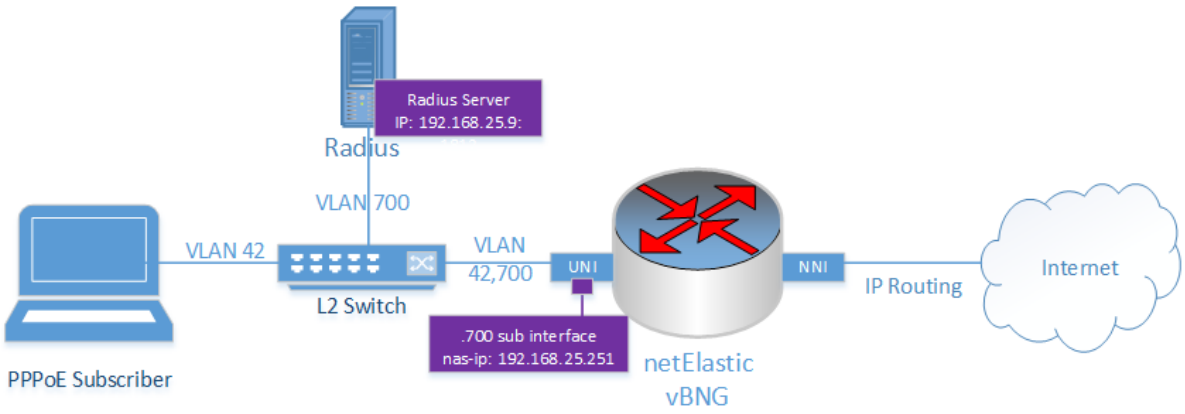
In the above example, we created a user with user name "pppoe_user_1" with password "pppoe_user_1_passwd". We also specified its access domain to be "my_domain". The complete configuration for this local user should look like this:

```
all-1-1(config-bras-local-subscriber-pppoe_user_1/my_domain)# show full
bras
 local-subscriber pppoe_user_1 domain my_domain
  password pppoe_user_1_passwd
 exit
exit
```

7.7 PPPoE Access With Radius AAA

This test case shows how to configure vBNG to work with PPPoE access with Radius AAA (authentication, authorization, and accounting).

The test setup is shown below.



The process of configuring PPPoE connections on the vBNG with Radius authentication, authorization and accounting involves:

- Configuring access interface
- Creating an PPPoE template
- Creating a VGI
- Creating Radius authentication group
- Creating Radius accounting group
- Creating authentication template
- Creating authorization template
- Creating accounting template and enable radius accounting
- Creating an IPPool
- Creating a domain
- Creating and configure VCI

Create a sub-interface with VLAN 42

This configuration is exactly the same as the test case in section 7.1. The configuration is shown here again for quick reference.

```
interface gei-1/1/2.42
 dot1q 42
exit
```

Create Radius Authentication Group

Radius authentication group configuration for PPPoE is exactly the same as that for IPoE as discussion for the test case in section 7.3. sRadius authentication group is used for Radius authorization as well. The configuration is shown here again for quick reference.

```
radius authentication group my_radius_grp
 timeout         3
 retry-times     3
 nas-ip-address 192.168.5.251
 algorithm       master
 dead-time       5
 dead-count      10
 class-as-car    disable
 filter-id-type user-acl
 server 1 ipv4-address 192.168.25.9 port 1812 key my_radius_key
exit
```

Enable Radius Accounting

To enable Radius accounting, we need to:

1. Enable Radius accounting at the config/radius level
2. Create a Radius accounting group. The Radius accounting normally shares the same server as the Radius authentication and authorization, but use a different port. Accounting usually uses port 1813 while authentication and authorization use port 1812.
3. Create an accounting template at the config/bras/accounting level and specify the Radius accounting group created in the accounting template.
4. Bind the accounting template in the appropriate domain.

Create a Radius Accounting Group. The Radius accounting group configuration should look like this

```
all-1-1(config-radius-accounting-group-my_radius_accounting_grp)# show full
radius accounting group my_radius_accounting_grp
 timeout     3
 retry-times 3
 nas-ip-address 192.168.5.251
 algorithm   master
 dead-time   5
 dead-count  10
 flow-unit   byte
 server 1 ipv4-address 192.168.25.9 port 1813 key my_radius_key
exit
```

We also need to enable Radius accounting under Radius configuration.

```
all-1-1(config)# radius accounting-on enable
```

Note: Radius accounting group is a separate Radius group from Radius authentication ad authorization group. The Radius configuration should contain two groups at this point as shown below.

```
all-1-1(config)# show full-configuration radius
radius vendor-id     54268
radius accounting-on enable
radius attribute-usermac-as mac
radius authentication group my_radius_grp
 timeout        3
 retry-times    3
 nas-ip-address 192.168.5.251
 algorithm      master
 dead-time      5
 dead-count     10
 class-as-car   disable
 filter-id-type user-acl
 server 1 ipv4-address 192.168.25.9 port 1812 key my_radius_key
exit
radius accounting group my_radius_accounting_grp
 timeout     3
 retry-times 3
 algorithm   master
 dead-time   5
 dead-count  10
 flow-unit   byte
 server 1 ipv4-address 192.168.25.9 port 1813 key my_radius_key
exit
```

Create Authentication Template

Radius authentication template configuration for PPPoE is exactly the same as that for IPoE as discussion for the test case in section 7.3. The configuration is shown here again for quick reference.

```
bras
 authentication my_authentication_template
  authentication-type         radius
  radius-authentication-group my_radius_grp
  user-name-format            strip-domain
  nas-port-format             class1
  nas-port-id-format class1
  calling-station-id-format class1
  invalid-vlan-tag            0
 exit
exit
```

Create Authorization Template

Radius authorization means vBNG will take authorization properties such as user's IP address, QoS plan, ACL rules, etc from the attributes carried in the Radius accept reply message instead of using locally configured properties. To achieve this, we need to create an authorization template from which to specify Radius authorization.

To create an authorization template to specify Radius authorization, follow these steps.

```
all-1-1# config
Entering configuration mode terminal
all-1-1(config)# bras
all-1-1(config-bras)# authorization my_authorization_template
all-1-1(config-bras-authorization-my_authorization_template)# authorization-type
radius
```

The configured authorization template should look like this:

```
all-1-1(config-bras-authorization-my_authorization_template)# show full
bras
 authorization my_authorization_template
  authorization-type radius
  nat-type           none
  radius-nat-switch  disable
 exit
exit
```

Of course, for all these to work, you need to configure corresponding attributes for the user on the Radius server (or your billing system) accordingly. Refer to section 5.2.2 for commonly used Radius authorization attributes.

Create Accounting Template

Perform the following tasks to create an Accounting template that binds the radius accounting group defined above

```
all-1-1# config
Entering configuration mode terminal
all-1-1(config)# bras
all-1-1(config-bras)# accounting my_accounting_template
all-1-1(config-bras-accounting-my_accounting_template)# accounting-type radius
all-1-1(config-bras-accounting-my_accounting_template)# first-radius-accounting-
group my_radius_accounting_grp
all-1-1(config-bras-accounting-my_accounting_template)# commit
```

The configured accounting template should look like the following:

```
all-1-1(config-bras-accounting-my_accounting_template)# show full
```

```
bras
 accounting my_accounting_template
  accounting-type               radius
  accounting-update             600
  first-radius-accounting-group my_radius_accounting_grp
  accounting-start-fail         online
  accounting-update-fail online
  accounting-update-immediately disable
  l2tp-accounting               vpdn-model
  user-name-format              strip-domain
  nas-port-format               class1
  nas-port-id-format class1
  calling-station-id-format class1
  invalid-vlan-tag              0
 exit
exit
```

Create an IPPool

This configuration is exactly the same as the test case in section 7.1. The configuration is shown here again for quick reference

```
ippool group my_ippool
 gateway-ip 172.16.1.1 gateway-mask 255.255.255.0
 lease-time        3600
 ippool-status     unlock
 warning-threshold 80
 warning-exhaust   disable
 frame-ip lease manage disable
 section start-ip 172.16.1.1 end-ip 172.16.1.254
 exit
exit
```

Create a VGI interface

This configuration is exactly the same as the test case in section 7.1. The configuration is shown here again for quick reference

```
interface vgi1
 ipv4 address 172.16.1.1 24
exit
bras
 vgi-configuration
  interface vgi1
  exit
 exit
exit
```

Create a domain

This configuration process is exactly the same as the test case in section 7.1. See section 4.3 for domain configuration details. In the domain definition, we need to bind all the authentication, authorization, and accounting templates that we created above. We also need to bind vgi and ippool to the domain as we did before. The complete configuration for the domain at this point should look like this:

```
all-1-1(config-bras-domain-my_domain)# show full
bras
 domain my_domain
  bind authentication-template my_authentication_template
  bind accounting-template my_accounting_template
  bind authorization-template my_authorization_template
  vgi                       vgi1
  domain-status             unlock
  user-routing-distribute disable
  tunnel-domain             disable
  flow-statistic            enable
  radius-attribute qos-acl-profile no-exist-policy offline
  quota-out offline
```

```
  bind-pool 1 my_ipoool
 exit
exit
```

Create an PPPoE Template

This configuration is exactly the same as the test case in section 7.5. The configuration is shown here again for quick reference.

```
bras
 pppox template my_pppoe_template
  check-magic-number enable
  ppp-authentication pap
  ac-name            netElastic-vBNG
  mru                1492
  service-name-omit  enable
  default-domain     my_domain
  quick-redial       disable
  keepalive-time     60
  keepalive-count    3
  check-ac-cookie    enable
 exit
```

Create a VCI interface and bind with PPPoE Template

This configuration is exactly the same as the test case in section 7.5. The configuration is shown here again for quick reference.

```
bras
 vci-configuration
  interface gei-1/1/2.42
   ipoe template my_ipoe_template
   pppox template my_pppoe_template
   max-ipox-session           32000
   max-pppox-session          32000
   encapsulation              multi
   pre-domain                 my_domain
   ip-access-type             ipv4
   authentication-method-ipv6 ppp
  exit
 exit
exit
```

Check accounting records on Radius

After users connect, vBNG begin to send their accounting records periodically at the interval set in the accounting template. Check your Radius or billing system database for the presence of accounting records.

7.8 PPPoE Access With Radius AAA, QoS,and NAT

In this example, we will present an example that closely mimics a real world pppoe subscriber management example. In the example:

- The vBNG router has two 10G interfaces. Users connect with PPPoE through a vlan 101 interface off physical interface 10gei-1/1/0.
- 10gei-1/1/1 is the vBNG router's upstream interface. We will set default route to route user traffic to the upstream router.
- Users will be authenticated on Radius with three credentials, user name, password, and calling-station-id carrying subscriber's mac address.
- Most users will get private IPs from a private IP pool and their traffic will be NATted.
- Some users will get statically assigned public IPs from radius and their traffic won't be NATted.

- Users' traffic will be controlled by QoS plans that set different rates at different times of the day.
- Some traffic will be placed on high priority queue while the remaining traffic will be on low priority queue at the interface level.
- Create white and black list IPs and ports and apply to the network interface to enhance security.

7.8.1 Create User QoS profiles for rate control

The requirement for User QoS is the following:

Package Name	Policy-2AM to 8AM			Policy-8AM to 9PM			Policy-9PM to 2AM		
	Download Mbps	Upload Mbps	CDN IP Mbps	Download Mbps	Upload Mbps	CDN IP Mbps	Download Mbps	Upload Mbps	CDN IP Mbps
Basic	24	24	24	12	12	12	9.6	9.6	9.6
Express	40	40	40	20	20	20	16	16	16

- The internet upload and download speeds are defined as in the table at the three different time frames.
- The CDN IP is 103.24.96.54/32. The traffic rates to this IP as shown in the table are symmetric up and down rates.

Here is our configuration flow:

1. User ACL to classify CDN IP traffic and everything else.
2. Create classmaps for CDN IP traffic and everything else based on defined ACL lists
3. Create three time ranges
4. Create CAR behaviors for different rates at different time ranges.
5. Creae QoS policies that ties classmaps and behaviours together.
6. Create QoS profiles the bind the upload and download policies together and name these QoS profiles to match the package name in the table.

Once the QoS profiles are defined, they can be referenced and activated by radius private attribute "NetElastic-Qos-Profile-Name" (VSA 31) as part of the radius reply message.

Here are the relevant configurations:

```
!define CDN IP and all others traffic flow ACL
access-list ALL-traffic-ACL
 rule 10 deny ip source 103.24.96.54/32 destination any
 rule 20 deny ip source any destination 103.24.96.54/32
 rule 30 permit ip source any destination any
exit
access-list CDN-IP-ACL
 rule 10 permit ip source 103.24.96.54/32 destination any
 rule 20 permit ip source any destination 103.24.96.54/32
 rule 30 deny ip source any destination any
exit

!define classmap for CDN IP traffic and all others.
class_map ALL-traffic match-way match-any
 match ipv4-access-list ALL-traffic-ACL
exit
class_map CDN-IP2-traffic match-way match-any
 match ipv4-access-list CDN-IP-ACL
exit

!time range definition
time-range TR_02-08
```

```
 daily start 02:00:00 end 08:00:00
exit
time-range TR_08-21
 daily start 08:00:00 end 21:00:00
exit
time-range TR_21-02
 daily start 21:00:00 end 02:00:00
exit

!define CAR behaviour for different rates
behavior Basic_CDN-IP
 item 1
  car cir 24000 pir 24000 cbs 3000000 pbs 3000000
  tr-name TR_02-08
 exit
 item 2
  car cir 12000 pir 12000 cbs 1500000 pbs 1500000
  tr-name TR_08-21
 exit
 item 3
  car cir 9600 pir 9600 cbs 1200000 pbs 1200000
  tr-name TR_21-02
 exit
exit
behavior Basic_INT_DOWN
 item 1
  car cir 24000 pir 24000 cbs 3000000 pbs 3000000
  tr-name TR_02-08
 exit
 item 2
  car cir 12000 pir 12000 cbs 1500000 pbs 1500000
  tr-name TR_08-21
 exit
 item 3
  car cir 9600 pir 9600 cbs 1200000 pbs 1200000
  tr-name TR_21-02
 exit
exit
behavior Basic_INT_UP
 item 1
  car cir 24000 pir 24000 cbs 3000000 pbs 3000000
  tr-name TR_02-08
 exit
 item 2
  car cir 12000 pir 12000 cbs 1500000 pbs 1500000
  tr-name TR_08-21
 exit
 item 3
  car cir 9600 pir 9600 cbs 1200000 pbs 1200000
  tr-name TR_21-02
 exit
exit
behavior Express_CDN-IP
 item 1
  car cir 40000 pir 40000 cbs 5000000 pbs 5000000
  tr-name TR_02-08
 exit
 item 2
  car cir 20000 pir 20000 cbs 2500000 pbs 2500000
  tr-name TR_08-21
 exit
 item 3
  car cir 16000 pir 16000 cbs 2000000 pbs 2000000
  tr-name TR_21-02
 exit
exit
behavior Express_INT_DOWN
 item 1
  car cir 40000 pir 40000 cbs 5000000 pbs 5000000
  tr-name TR_02-08
 exit
 item 2
  car cir 20000 pir 20000 cbs 2500000 pbs 2500000
  tr-name TR_08-21
 exit
 item 3
  car cir 16000 pir 16000 cbs 2000000 pbs 2000000
```

```
  tr-name TR_21-02
 exit
exit
behavior Express_INT_UP
 item 1
  car cir 40000 pir 40000 cbs 5000000 pbs 5000000
  tr-name TR_02-08
 exit
 item 2
  car cir 20000 pir 20000 cbs 2500000 pbs 2500000
  tr-name TR_08-21
 exit
 item 3
  car cir 16000 pir 16000 cbs 2000000 pbs 2000000
  tr-name TR_21-02
 exit
exit

!define policies
policy policy_Basic_DOWN
 class_map CDN-IP2-traffic behavior Basic_CDN-IP priority 5
 class_map ALL-traffic behavior Basic_INT_DOWN priority 1
exit
policy policy_Basic_UP
 class_map CDN-IP2-traffic behavior Basic_CDN-IP priority 5
 class_map ALL-traffic behavior Basic_INT_UP priority 1
exit
policy policy_Express_DOWN
 class_map CDN-IP2-traffic behavior Express_CDN-IP priority 5
 class_map ALL-traffic behavior Express_INT_DOWN priority 1
exit
policy policy_Express_UP
 class_map CDN-IP2-traffic behavior Express_CDN-IP priority 5
 class_map ALL-traffic behavior Express_INT_UP priority 1
exit

!define user QoS profiles
bras
 user-qos-profile Basic
  input-qos-policy  policy_Basic_UP
  output-qos-policy policy_Basic_DOWN
 exit
 user-qos-profile Express
  input-qos-policy  policy_Express_UP
  output-qos-policy policy_Express_DOWN
 exit
```

7.8.2 Create High and Low Traffic Classification and Related Queue Policies.

Here is the configuration flow:

1. Create ACL filter to identify the traffic flows for high and low priorities.
2. Create classmap and use the above defined ACL filters to classify traffic flows.
3. Create queuing behaviour for high and low priorities traffics
4. Create policies to tie classmap and queuing behaviour together.

```
! identify ICMP, DNS by ports.
access-list ICMP-DNS-ACL
 rule 10 permit specify 1 source any destination any
 rule 20 permit tcp source any gt 0 destination any eq domain
 rule 25 permit tcp source any eq domain destination any gt 0
 rule 30 permit udp source any gt 0 destination any eq domain
 rule 35 permit udp source any eq domain destination any gt 0
 rule 90 deny ip source any destination any
exit

! identify GAME(Players Unknown Battle Ground) by ports.
! (TCP 27015-27030,27036-27037 and UDP 4380,27000-27031,27036)
```

```
access-list PUBS-HTTP-ACL
 rule 10 permit tcp source any gt 0 destination any range 27015 27030
 rule 15 permit tcp source any range 27015 27030 destination any gt 0
 rule 20 permit tcp source any gt 0 destination any range 27036 27037
 rule 25 permit tcp source any range 27036 27037 destination any gt 0
 rule 30 permit udp source any gt 0 destination any eq 4380
 rule 35 permit udp source any eq 4380 destination any gt 0
 rule 40 permit udp source any gt 0 destination any range 27000 27031
 rule 45 permit udp source any range 27000 27031 destination any gt 0
 rule 50 permit udp source any gt 0 destination any eq 27036
 rule 55 permit udp source any eq 27036 destination any gt 0
 rule 60 permit udp source any gt 0 destination any eq 443
 rule 65 permit udp source any eq 443 destination any gt 0
 rule 70 permit udp source any gt 0 destination any eq 80
 rule 75 permit udp source any eq 80 destination any gt 0
 rule 90 deny ip source any destination any
exit

! define class maps for different flows based on ACL
class_map ICMP-DNS-traffic match-way match-any
 match ipv4-access-list ICMP-DNS-ACL
exit
class_map PUBS-HTTP-traffic match-way match-any
 match ipv4-access-list PUBS-HTTP-ACL
exit
class_map all match-way match-all
 match all
exit

! define priority queue behaviour
behavior queue_be
 item 1
  cbq queue be
 exit
exit
behavior queue_ef
 item 1
  cbq queue ef
 exit
exit

! define priority queue policy
policy policy_interface_queue
 class_map ICMP-DNS-traffic behavior queue_ef priority 8
 class_map PUBS-HTTP-traffic behavior queue_ef priority 7
 class_map all behavior queue_be priority 1
exit
```

At this point, we have priority queue policy "policy_interface_queue" defined. We can then apply it to the relevant interfaces as shown in section 7.8.5 and section 7.8.6.

7.8.3 Create NAT configuration

Create the NAT configuration involves the following steps:

1. Configure nat->user-policy where you specify nat-mode (nat algorithm), working-form (bras or standalone), single-user-max-entries (max per user sessions), and max-entries (max total sessions)
2. Turn the nat logging switch on/off to enable or disable nat logging.
3. Create portmap groups where the port size and starting port number are specified. Under portmap, you can also optionally create portrange-enable configuration enable SPR algorithm and set port allocation spec for each private IP
4. Create public IP pools to which private IPs will be natted.
5. Create ACL rules where you select the IPs to be NATted and exclude IPs from going through NAT.
6. Create NAT rules that bind portmap, public IP pool, and applicable ACL rules together.
7. Set "nat inside" on all related internal (access) vgi interfaces and set "nat outside" on all related outside (WAN) interfaces.

8. Set “nat-type inside” in the user’s associated authorization template.

Below are most of the nat related configurations. Other nat related configuration under interfaces and authorization template are annotated in their respective sections.

```
!define nat IP ACL, only permitted IPs will be natted
access-list PrivateIP-Filter
 rule 10 permit ip source 172.20.0.0/18 destination any
 rule 20 permit ip source any destination 172.20.0.0/18
 rule 30 deny ip source any destination any
exit

nat
 !nat policies
 user-policy
  nat-mode                     full-cone  !nat mode
  working-form                 bras       !bras or standalone mode
  max-entries                  4000000    !max nat sessions per router
  icmp-expire-time             20
  udp-expire-time              180
  tcp-expire-time              240
  tcp-fin-expire-time          30
  single-user-max-entries      2000       !max nat sessions per user
  alarm-enable                 disable
  alarm-total-entries-threshold 80
 exit
 !nat logging switch
 log
  switch    on
  log-style type3
 exit
 !define public ip pools
 ippool group PUBLIC_POOL_1
  section start-ip 103.93.218.0 end-ip 103.93.218.255
  section start-ip 119.152.100.0 end-ip 119.152.100.255
 exit
 !define port map
 portmap group my_nat_port_map_group
  start-port 6000
  size       50000
  portrange-enable 200 alarm-threshold 80 extend-port 400 extend-times 5
 exit
 !define nat rules that bind port map, public pool, and nat acl together
 rule group RULE_PUBLIC_POOL_1
  type            dynamic
  radius-origin   disable
  ip-alloc-random disable
  ippool-name PUBLIC_POOL_1 portmap-name my_nat_port_map_group acl-list-name
PrivateIP-Filter
 exit
exit
```

7.8.4 Create access related configurations

Creating the access related configuration involves the following:

1. Create private IP pool. These are the IPs that will be dynamically assigned to subscribers and these IPs will be NATted.
2. Create public IP pool. These are IPs that will be assigned to subscriber by radius and these IPs will be excluded from NAT.
3. Create a VGI interface that serves as the subscribers gateway for both the private IP users and public IP users. It is only possible to share the same VGI with IPs from different subnets with PPPoE.
4. Create radius authentication group where you specify external radius authentication server and server access secret.
5. Create radius accounting group where you specify external radius accounting server and server access secret.
6. Create access authentication template that binds radius authentication group created.

7. Create access accounting template that binds radius accounting group create.
8. Create access authorization template to specify to honor radius reply attributes.
9. Create an access domain that binds authentication, authorization, accounting, vgi and ip pools together.
10. Create pppox template and set the default-domain to be the domain defined above.
11. Create vgi-configuration to specify vgi interface.
12. Create vci-configuration and bind the pppox template defined above to the relevant interfaces.

Here are the configurations

```
!defien private ip pool
ippool group Nat_IPPools
 gateway-ip 172.20.0.1 gateway-mask 255.255.0.0
 lease-time        60
 dns-primary 103.24.96.146 secondary 103.24.96.6
 ippool-status     unlock
 warning-threshold 80
 warning-exhaust   disable
 frame-ip lease manage disable
 section start-ip 172.20.0.2 end-ip 172.20.60.250
 exit
exit
!define public ip pool
ippool group Public_IPPools
 gateway-ip 172.20.0.1 gateway-mask 255.255.255.255
 lease-time        60
 dns-primary 103.24.96.146 secondary 103.24.96.6
 ippool-status     unlock
 warning-threshold 80
 warning-exhaust   disable
 frame-ip lease manage disable
 section start-ip 119.152.102.168 end-ip 119.152.102.175
  reserved-section reserved-start-ip 119.152.102.168 reserved-end-ip
119.152.102.175
 exit
exit
!define vgi
interface vgi1
 nat inside
 ipv4 address 172.20.0.1 16
exit
!define radius authentication group
radius authentication group my_radius_authen_grp
 server-type    ipv4-server
 timeout        3
 retry-times    3
 nas-ip-address 172.17.1.98
 algorithm      master
 dead-time      5
 dead-count     10
 class-as-car   disable
 filter-id-type user-acl
 server 1 ipv4-address 103.24.96.142 port 1812 key netElastic
exit
!define radius accounting group
radius accounting group my_radius_acct_grp
 server-type    ipv4-server
 timeout        3
 retry-times    3
 nas-ip-address 172.17.1.98
 algorithm      master
 dead-time      5
 dead-count     10
 flow-unit      byte
 server 1 ipv4-address 103.24.96.142 port 1813 key netElastic
exit

bras
 !define authentication template
```

```
authentication radius_authen_template
 authentication-type         radius
 radius-authentication-group my_radius_authen_grp
 user-name-format            strip-domain
 nas-port-format             class5
 called-station-id-format    class2
 nas-port-id-format user-defined [ vlan ] format %d
 calling-station-id-format class1
 invalid-vlan-tag            0
exit
!define accounting template
accounting radius_acct_template
 accounting-type               radius
 accounting-update             600
 first-radius-accounting-group my_radius_acct_grp
 accounting-start-fail         online
 accounting-update-fail online
 accounting-update-immediately disable
 l2tp-accounting               vpdn-model
 user-name-format              strip-domain
 nas-port-format               class5
 called-station-id-format      class2
 nas-port-id-format user-defined [ vlan ] format %d
 calling-station-id-format class1
 invalid-vlan-tag              0
exit
!define authorization template
authorization radius_author_template
 authorization-type mix-radius
 bind nat-domain-name RULE_PUBLIC_POOL_1
 nat-type           inside
 radius-nat-switch  disable
exit
!define domain
domain my_domain
 bind authentication-template radius_authen_template
 bind accounting-template radius_acct_template
 bind authorization-template radius_author_template
 vgi                     vgi1
 domain-status           unlock
 user-routing-distribute disable
 tunnel-domain           disable
 flow-statistic          enable
 radius-attribute qos-acl-profile no-exist-policy offline
 quota-out offline
 bind-pool 1 Nat_IPPools
 bind-pool 2 Public_IPPools
exit
!define pppox template
pppox template my_pppoe_temp
 check-magic-number enable
 ppp-authentication pap
 ac-name            netElastic-vBNG
 mru                1492
 default-domain     my_domain
 quick-redial       disable
 keepalive-time     20
 keepalive-count    3
 check-ac-cookie    enable
 service-name-type  partial-match
 ppp-ncp-admit-any  disable
exit
!define vgi-configuration
vgi-configuration
 interface vgi1
 exit
exit
!define vci-configuration
vci-configuration
 interface 10gei-1/1/0.101
  pppox template my_pppoe_temp
  max-ipox-session  32000
  max-pppox-session 32000
  encapsulation     multi
  pre-domain        my_domain
  ip-access-type    ipv4
 exit
```

```
 exit
exit
```

7.8.5 Create a sub-interface with VLAN 101

The VLAN 101 interface configuration is show below where we

1. Define the dot1Q sub interface by specifying the dot1q vlan ID.
2. Apply traffic priority queue policy

```
interface 10gei-1/1/0.101
 bind qos in policy_interface_queue
 bind qos out policy_interface_queue
 dot1q 101
exit
```

7.8.6 Create the network (WAN) interface and apply QoS and security policies.

The network WAN interface configuration is show below where we

1. Configure an IP address on the interface.
2. Apply traffic priority queue policy
3. Create traffic security permit and deny ACL list and apply to the interface.
4. Specify “nat outside” to indicate this is the NAT outside (WAN) interface

```
access-list DNS-INT-ACL
 ! deny certain traffic and make exception for some IPs
 rule 10 permit ip source any destination 103.24.96.146/32
 rule 15 permit ip source 103.24.96.146/32 destination any
 rule 100 deny tcp source any gt 0 destination any eq 587
 rule 105 deny tcp source any eq 587 destination any gt 0
 rule 110 deny tcp source any gt 0 destination any eq 1723
 rule 115 deny tcp source any eq 1723 destination any gt 0
 ! deny external DNS, only allow specified DNS
 rule 200 permit ip source any destination 103.24.96.146/32
 rule 205 permit ip source 103.24.96.146/32 destination any
 rule 250 deny tcp source any gt 0 destination any eq domain
 rule 255 deny tcp source any eq domain destination any gt 0
 rule 260 deny udp source any gt 0 destination any eq domain
 rule 265 deny udp source any eq domain destination any gt 0
exit

! network interface definition
interface 10gei-1/1/1
 description "network interface"
 bind acl in ipv4 DNS-INT-ACL
 bind acl out ipv4 DNS-INT-ACL
 bind qos in policy_interface_queue
 bind qos out policy_interface_queue
 nat outside   !enable nat on this outside interface
 ipv4 address 172.17.1.98 30
exit
```