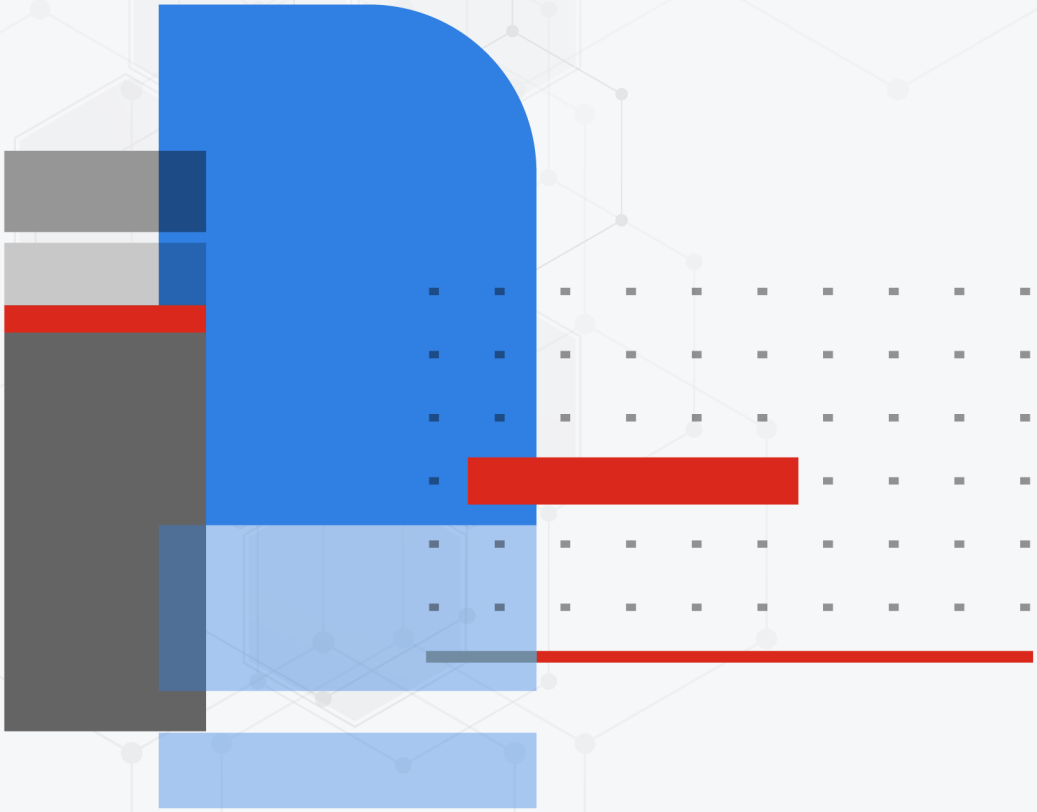
Fortinet Carrier Grade NAT Field Reference Architecture Guide

FortiOS 7.4.6



FORTINET DOCUMENT LIBRARY
https://docs.fortinet.com

FORTINET VIDEO LIBRARY
https://video.fortinet.com

FORTINET BLOG
https://blog.fortinet.com

CUSTOMER SERVICE & SUPPORT
https://support.fortinet.com

FORTINET TRAINING & CERTIFICATION PROGRAM
https://www.fortinet.com/training-certification

FORTINET TRAINING INSTITUTE
https://training.fortinet.com

FORTIGUARD LABS
https://www.fortiguard.com

END USER LICENSE AGREEMENT
https://www.fortinet.com/doc/legal/EULA.pdf

FEEDBACK
Email: techdoc@fortinet.com

February 14, 2025
FortiOS 7.4.6 Fortinet Carrier Grade NAT Field Reference Architecture Guide
01-746-984570-20250214

TABLE OF CONTENTS

Change log

Date	Change description
February 14, 2025	FortiOS 7.4.6 document release.

Abbreviations and Terminology

Abbreviation	Explanation
3GPP	3rd Generation Partnership Project is an umbrella for a number of standards organizations which develop protocols and guidance for mobile telecommunications.
5GC	5G Core network
CGNAT	Carrier-Grade NAT, also terms such as CG-NAT and CGN can be used.
CLAT	Customer Located Address Translation
CPE	Customer Premise Equipment
DNAT	Destination Network Address Translation, where the destination address of the connection is translated by the NAT system.
EIF	Endpoint Independent Filtering
EIM	Endpoint Independent Mapping
LSN	Large Scale NAT, another term for CGNAT although focused more at fixed-broadband services.
LTE-A	LTE Advanced, a major enhancement to LTE delivering increased downlink speeds.
LTE	Long Term Evolution, commonly referred to as 4G of 4G LTE.
MAP-E	Mapping of Address and Port with Encapsulation
MAP-T	Mapping of Address and Port with Translation
MNO	Mobile Network Operator
NAT	Network Address Translation
OAM	Operations And Management
PBA	Port Block Allocation
PGW	PDN (Packet Data Network) Gateway, logical component of the 4G Packet core decapsulating GTP-U tunnels and presenting UE traffic to the IP network.
PLAT	Provider Located Address Translation
SNAT	Source Network Address Translation, where the source address of the connection is translated by the NAT system.
SPA	Single Port Allocation.
UE	User Equipment, devices connecting to the Mobile network e.g. phones.
UPF	User Plane Function, logical component of the 5G Packet Core terminating GTP-U tunnels from UEs.

Abbreviation	Explanation
XLAT	General term covering both PLAT and CLAT.

Introduction

This guide provides technically-focused CGNAT solution details, guidance, and reference architecture for FortiOS Hyperscale CGNAT and Kernel CGNAT solutions. Hyperscale CGNAT is available on FortiGate hardware with NP7 processors with a hyperscale license. Kernel CGNAT is available on mainstream FortiOS running on FortiGate VM platforms and on high-end FortiGate models.

On NP7 platforms with hyperscale licenses, hyperscale CGNAT provides more functionality than Kernel CGNAT and includes hardware acceleration for session setup and logging. On NP7 platforms without hyperscale firewall licenses, CGNAT session setup and logging is not hardware accelerated, but data processing of established sessions is hardware accelerated.

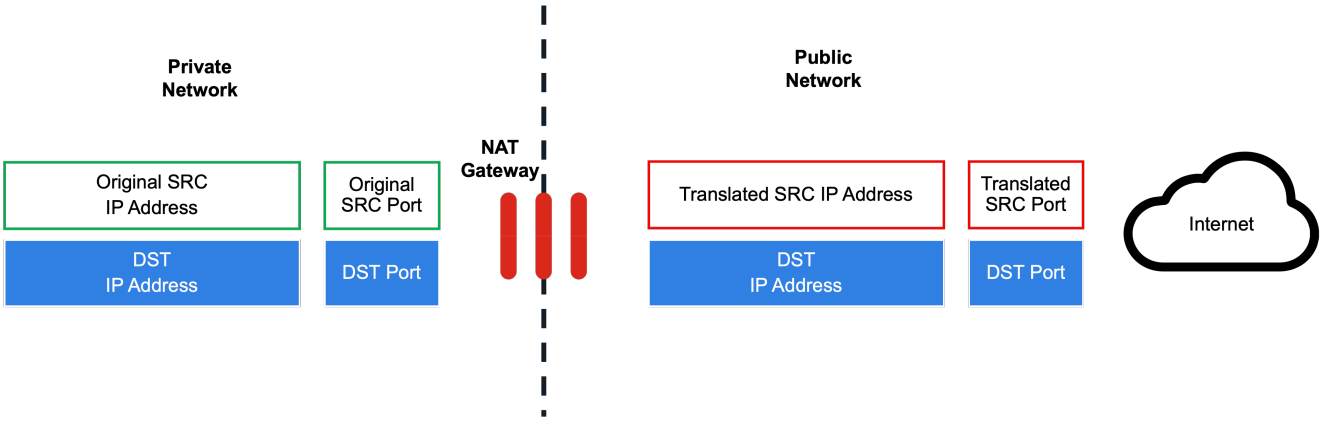
This document assumes the reader is comfortable and experienced with Fixed and Mobile Networking terms and definitions, as well as generic networking concepts for IPv4 and IPv6.

This document is based on software features available in FortiOS 7.2.5. Information about features available on more recent versions of FortiOS are also described.

The reader is expected to know and understand NAT as a generic function. This document can also serve as guidance for CGNAT design and deployment considerations in the pre-sale's engagement cycle. However, it is strongly recommended that Fortinet Professional Services is utilized for any formal project design or deployment.

The need for Carrier-Grade NAT

Carrier-Grade Network Address Translation (CGNAT, or CGN), and with it Large-Scale Network Address Translation (LSN), is by definition NAT that is used to translate many sources behind a smaller number of IP addresses (pool) for the purposes of accessing public resources. This is generally applied using Source NAT (SNAT), where the "private" client source IP address and source port is translated to a "public" source IP address and source port to aggregate a set of internal addresses behind a single IP address, or NAT pool of public address for potentially many subscribers.



Carrier-Grade NAT is a solution to the problem of exhaustion of IPv4 addresses, or simply overcomes overlapping IP addresses, or sometimes just provides access to public resources on the internet. Furthermore, Service Providers have

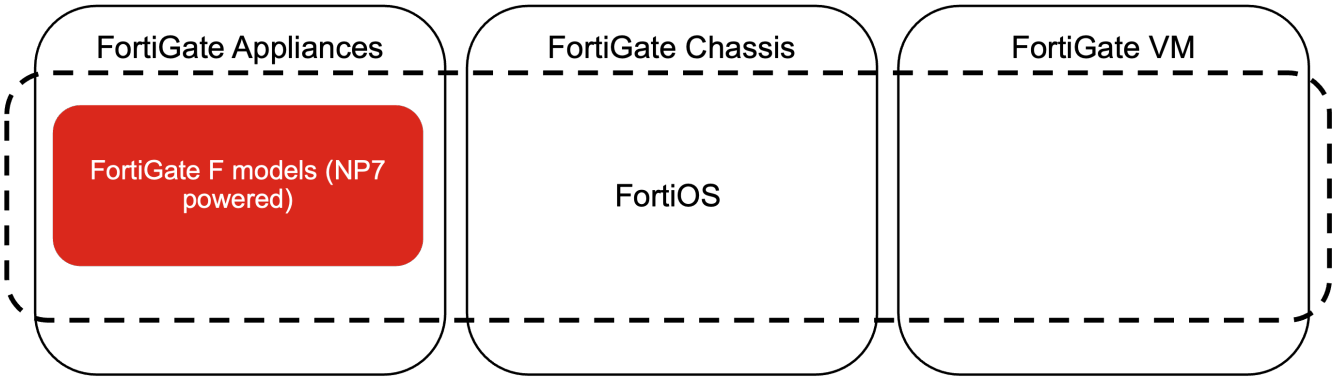
already been transitioning to IPv6 for some time, however the dilemma of how to access legacy IPv4 services on public networks, such as the internet, is valid and technologies such as NAT64 provide an effective solution to this IPv6 to IPv4 requirement. Fortinet provides hyperscale CGNAT solutions on NP7 hyperscale enabled FortiOS. FortiOS supplies Kernel CGNAT with mainstream FortiOS running on high-end FortiGate appliances or VM platforms. See Hyperscale and Kernel CGNAT feature comparison on page 20.

How to choose between Hyperscale or mainstream FortiOS CGNAT

Fortinet provides multiple CGNAT capable platforms (VMs, appliances, and chassis) and based on the required scale and translation features, you can decide for the different deployment options - NP7 hyperscale or mainstream FortiOS.

This document will focus on features available on NP7 hyperscale FortiGate systems (FG-180xF and higher) and their corresponding CGNAT features.

The FortiGate 7000 platform does not support hyperscale. For any guidance support or if in doubt, please engage with the Carrier CSE team.



So, why do I need a hyperscale license at all?

The hyperscale license unlocks NP7 hardware session setup, which provides higher session (CPS and CCS) capacity based on hardware policy offloading. Hyperscale CGNAT also includes NP7-accelerated log generation. This means that system bus or CPU are not involved in the log generation.

Hyperscale FortiOS also includes enhanced CGNAT features.

Other factors that would that influence whether to use NP7 hyperscale CGNAT include whether you need features that are not supported by hyperscale GCNAT. Commonly used features that are not supported by hyperscale CGNAT include security profiles and RSSO. For details about features that are not supported, see Hyperscale firewall 7.4.6 incompatibilities and limitations.

On a FortiGate with hyperscale enabled, hyperscale features are turned on per VDOM. So in the same FortiGate you can create hyperscale VDOMs for hyperscale features and normal VDOMs for features not supported in hyperscale VDOMs. IPSec is VPN is not supported by hyperscale VDOMs, it is possible to have non-hyperscale VDOMs on hyperscale enabled system and use IPSec in that VDOM.

On hyperscale systems the sessions that are matching session helpers/ALG will be established in CPU and the corresponding data will be offloaded. The logging for those sessions CPU based.

Most used CGNAT Technologies

FortiGate NP7 systems support the various CGNAT technologies (with or without hyperscale), however the most common CGNAT technologies that Service Providers are using are NAT44 and NAT64 (with, or without DNS64), as well as Dual-Stack.

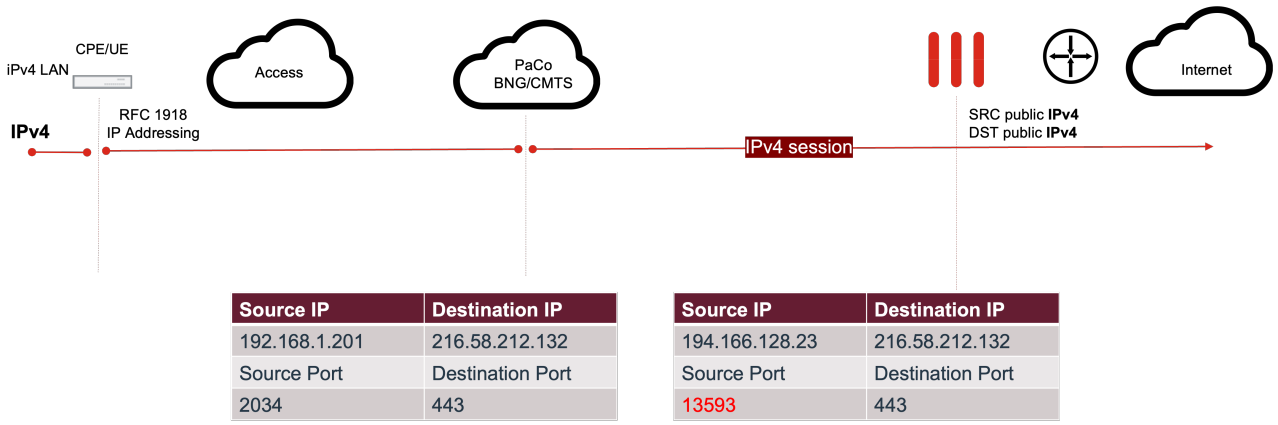
The table below represents a summary of the common CGNAT technologies:

CGNAT Type	Description
NAT44	In NAT44 the "private" IPv4 address is and source port are translated to a "public" IPv4 address and source port and this type of network translation is the most common CGNAT architecture used in mobile and fixed networks. The IPv4 translation can be used with different NAT methods e.g. PBA, Overload, deterministic, single port allocation or the hyperscale NAT methods like Port Block Allocation, Overload Port Block Allocation, Single Port Allocation, Overload Single Port Allocation or Fixed Allocation/Deterministic NAT.
NAT444	NAT444 is essentially double translation used in fixed networks, where the CPE translates the "private" IP addresses (of clients behind the CPE) into another set of "private" IP addresses towards the Service Provider Network which are then translated into "public" IP addresses towards Internet.
NAT64	NAT64 is SNAT translation between IPv6 and IPv4 hosts through a NAT64 gateway, which creates the mapping between the IPv6 and the IPv4 addresses using VIP with well known prefix 64:ff9b and the NAT pool via the policy.
DNS64	DNS64 is used with an IPv6/IPv4 translator to enable client-server communication between an IPv6-only client and an IPv4-only server without requiring any changes to either the IPv6 or the IPv4 node for applications that work through NAT. In this scenario the FortiGate translates the IPv6 to IPv4 and can act as a recursive DNS server using the DNS64 feature to translate IPv4 records (A) to IPv6 records (AAAA).
NAT46	NAT46 is the reverse of NAT64 using DNAT, which solves inbound IPv4 to IPv6 connectivity. NAT46 is used to translate IPv4 addresses to IPv6 addresses so that a client on an IPv4 network can communicate transparently with a server on an IPv6 network. The IPv6 address can be VIP or VIP group only. This type of translation is not so commonly used by service providers. Please refer to Technical Tip: Configure NAT46 on FortiGate for further details.
NAT66	NAT66 translates the private IPv6 to a public IPv6 address. NAT66 is not a common translation used by Service Providers but in some cases either source NAT66 (using IPv6 pool or single address) or destination NAT66 (using IPv6 VIP address) can be useful.
464XLAT	464XLAT provides IPv4 connectivity across an IPv6-only network. There are two types of 464XLAT translations: • CLAT is the Customer-side translator, which is stateless NAT46, integrated on the CPE side that translates 1:1 the "private" IPv4 addresses to IPv6 addresses and vice versa. The transport through the Service Provider network is IPv6 based. • PLAT is the Provider-side translator, which is stateful NAT64, that translates "public" IPv6 addresses to "public" IPv4 addresses and vice versa. The transport through the Service Provider network is IPv6 based. The FortiGate acts in the

CGNAT Type	Description
	case of PLAT as NAT64 translator.
DS-lite AFTR	DS-Lite AFTR allows applications using IPv4 to access the internet with IPv6. With DS-Lite, all the data from IPv4 clients (behind a CPE, on fixed networks) are encapsulated into IPv6 tunnel (originated from the IPv6 CPE) towards the the IPv6 tunnel endpoint/AFTR device. After decapsulating the tunnel the AFTR device is translating towards the internet using common CGNAT techniques (like PBA). DS-lite AFTR is not supported by FortiOS. This type of translation is not used by service providers in the mobile space.
4rd	4rd provides an IPv6 transition mechanism for deploying IPv6, while maintaining IPv4 service to customers. The CPE provides a dual stack 4rd IPv4 prefix. The transport between CPE and Border Relay is based on IPv6 tunnels. The Border Relay performs the NAT64 translation. CPE and Border Relay must be configured with Domain parameters related to mapping rules containing the mandatory IPv4 prefix, EA bits and IPv6 prefix data. Currently FortiGate supports MAP-E just as CE. This type of translation is not used by service providers in the mobile space due to lack of 4rd support on mobile devices.
MAP-E	Mapping of Address and Port with Encapsulation (MAP-E) - MAP-E uses IPv6 tunnel to encapsulate IPv4 traffic. Currently, MAP-BR is not supported in FortiOS FortiOS. MAP-E however is supported as customer edge device (CPE), which creates an IPv4-over-IPv6 tunnel between the FortiGate and third party border relay (BR) operating in an IPv6 network. This type of translation is not used by service providers in the mobile space.
MAP-T	MAP-T: Mapping of address and port using translation - For the FortiGate the MAP-T is NAT64 translation and the FortiGate should act as MAP-T Border Relay. Currently, MAP-T is not supported by the FortiGate. This type of translation is not used by SP's in the mobile space due to the lack of tunneling support on the mobile devices.

NAT44

NAT44 translates private IPv4 addresses and source ports into public IPv4 addresses and source ports and this type of network translation is kind the most common used by service providers.



The IPv4 translation can be used with hardware accelerated CGN resource allocation or kernel based FortiOS NAT pools. The differences between the NAT pools are explained in detail in Hyperscale CGNAT Pools on NP7 Systems on page 21 and Kernel CGNAT on page 45. To configure NAT44 you need a firewall policy and a NAT44 IP pool:

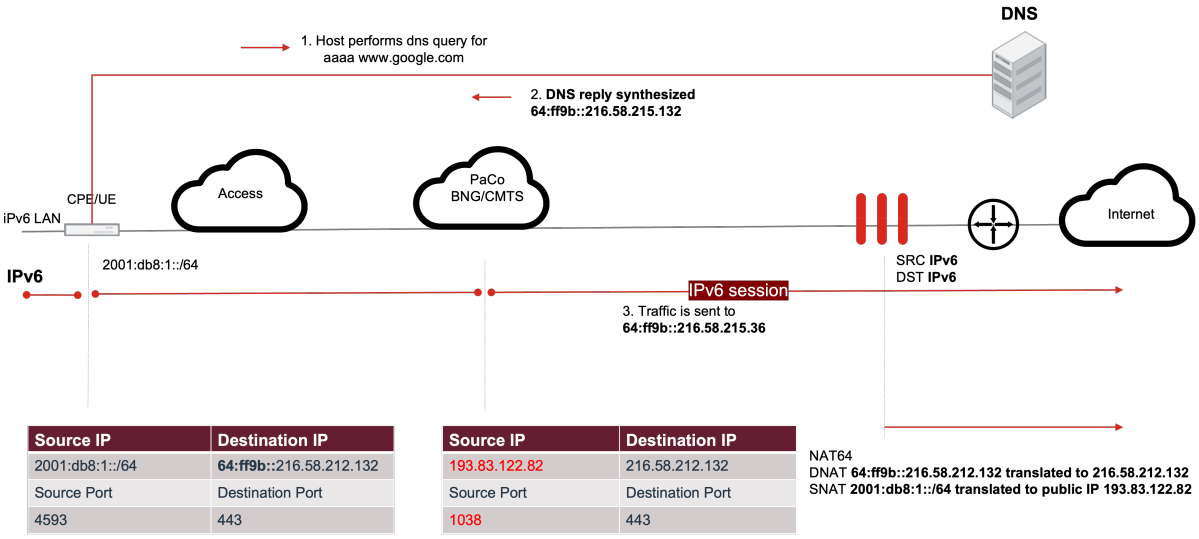
```
config firewall policy
   edit 1
      set name "NAT44"
      set srcintf "any"
      set dstintf "any"
      set action accept
      set srcaddr "all"
      set dstaddr "all"
      set schedule "always"
      set service "ALL"
      set logtraffic all
      set ippool enable (enables NAT)
      set poolname "PBA" (references to the NAT)
   end
```

In this example, a PBA NAT pool is used and it requires start and end IP Addresses, a block size, and number of blocks that will be provided for translations to the customer.

```
config firewall ippool
   edit "PBA"
      set type port-block-allocation (specifies the NAT pool type)
      set startip 194.166.128.0
      set endip 192.166.128.255
      set block-size 128 (the number of ports per block)
      set num-blocks-per-user 3 (the number of blocks per user)
      set pba-timeout 50
      set arp-reply enable
      set arp-intf ''
      set comments ''
   end
```

NAT64

NAT64 is translation between an IPv6 client and IPv4 hosts (usually on the internet) through a NAT64 gateway, which creates the mapping between the IPv6 and the IPv4 addresses. NAT64 is required because not all internet hosts are resolvable via IPv6. Therefore the client IPv6 address is translated to public IPv4 address, so that it can reach the IPv4 host on the internet.



In this example the IPv6 client is trying to reach www.google.com. The first thing that happens is name resolution. Let's assume for a second that www.google.com doesn't have IPv6 record. The IPv6 client 2001:db8:1::1 would need to get the AAAA record for the requested host from DNS.

If there is no IPv6 record for the requested www.google.com host, the DNS server will resolve it with and IPv4 address and will embed it into a AAAA response towards the client using the well known prefix of 64:ff9b. The resolved host will be 64:ff9b:216.58.212.132.

The IPv6 client will try to connect to 64:ff9b::d8:44:d4:84 (hex d8:44:d4:84 is 216.58.212.132), which will match the firewall policy containing the VIP6 destination and the FortiGate will utilize the VIP6 functionality and will translate/Destination NAT to 216.58.212.132 by looking into the last 32 bits of the IPv6 address. The client source IPv6 address 2001:db8:1::1 and source port 4593 will be translated with resource from the CGNAT pool to IPv4 address 193.83.122.82 and source port 5117.

FortiOS supports NAT64 with both hyperscale enabled systems and kernel/FortiOS. In NAT64 the firewall policy matches the VIP64 v6 destination to translate the client IPv6 address using IPv4 resources from the PBA64 pool so that a client on an IPv6 network can communicate transparently with a server on an IPv4 network.

In the example below the policy does not contain CGN resource allocation options, which means that kernel CGNAT is deployed:

```
config firewall policy
   edit 1
        set name "NAT64"
        set srcintf "any"
        set dstintf "any"
        set action accept
        set nat64 enable
        set srcaddr "all"
        set dstaddr "all"
        set srcaddr6 "all"
        set dstaddr6 "VIP64"
        set schedule "always"
        set service "ALL"
        set logtraffic all
        set ippool enable
        set poolname "PBA64"
     end
```

The vip6 destination is Virtual IP 64:ff9b::-64:ff9b::ffff:ffff with nat64 enabled, which uses embedded-ipv4-address of the resolved IPv6 host in the lower 32 bits of the external IPv6 address as mapped IPv4 address.

```
config firewall vip6
   edit "VIP64"
      set extip 64:ff9b::-64:ff9b::ffff:ffff
      set nat66 disable
      set nat64 enable
      set embedded-ipv4-address enable
   end
```

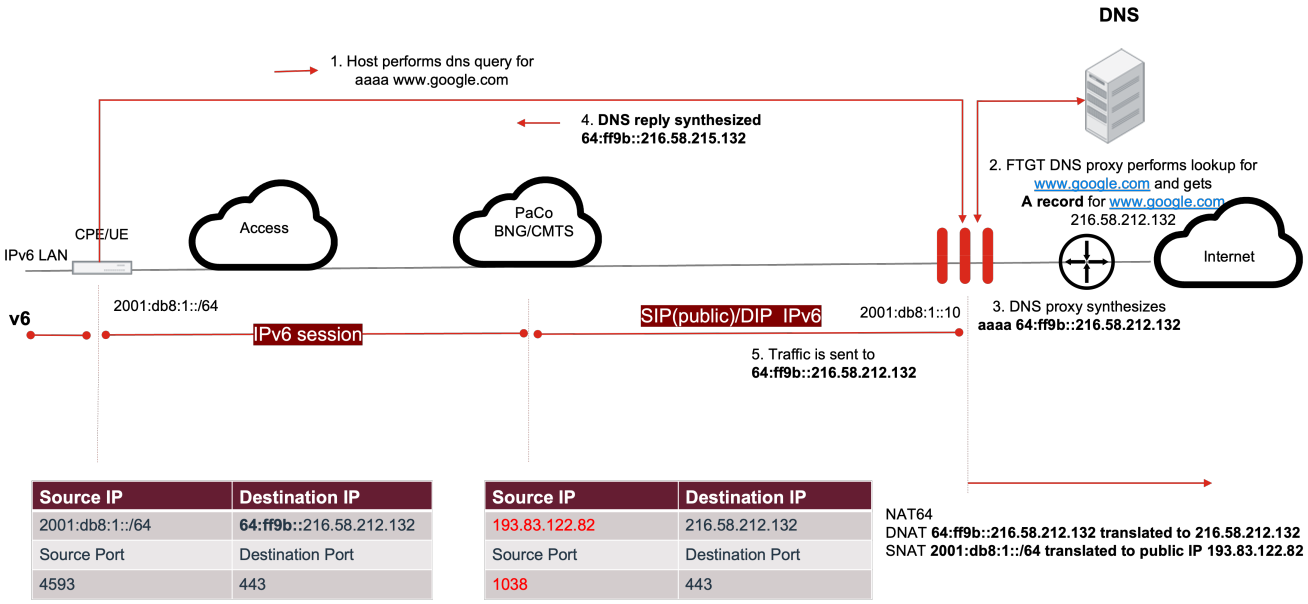
The PBA64 pool in this example is not hardware accelerated:

```
config firewall ippool
   edit "PBA64"
      set type port-block-allocation
      set startip 193.83.122.0
      set endip 193.83.122.255
      set block-size 128
      set num-blocks-per-user 3
      set pba-timeout 30
      set permit-any-host disable
      set nat64 enable
      set add-nat64-route enable
      set subnet-broadcast-in-ippool enable
   end
```

DNS64

DNS64 on FortiOS acts like a DNS server to the client and IPv6/IPv4 translator. The translator enables client-server communication between IPv6-only clients and an IPv4-only servers, without requiring any changes to either the IPv6 or the IPv4 node for applications that work through NAT. For this configuration to work correctly, the client has to use the FortiGate as a DNS server.

In this scenario the FortiGate will try to resolve the the server address (requested by the client) via the configured FortiGate DNS server. If the DNS server doesn't have an AAAA record and just an A record for the requested server, the FortiGate uses the well known prefix `64:ff9b::/96` and will embed the IPv4 address from the A record resolution into the DNS reply to the client. Once the client traffic is received by the FortiGate, the destination address is translated to IPv6 and the source IP address is translated with an IP address and port from the configured NAT pool.



Source IP	Destination IP
2001:db8:1::/64	64:ff9b::216.58.212.132
Source Port	Destination Port
4593	443

Source IP	Destination IP
193.83.122.82	216.58.212.132
Source Port	Destination Port
1038	443

The DNS64 policy matches vip6 addresses, similar to the NAT64 use case:

```
config firewall policy
   edit 3
      set name "dns64"
      set srcintf "port1"
      set dstintf "port2"
      set action accept
      set nat64 enable
      set srcaddr "all"
      set dstaddr "all"
      set srcaddr6 "all"
      set dstaddr6 "vip64"
      set service "ALL"
      set ippool enable
      set poolname "nat64_pba"
   end
```

The vip6 Virtual IP uses the same IPv6 prefix 64:ff9b::/96 configured as embedded type as NAT64 and the resolved IPv4 address is embedded into the vip6 address in the lower 32bits of the address:

```
config firewall vip6
   edit "vip64"
      set extip 64:ff9b::-64:ff9b::ffff:ffff
      set nat66 disable
      set nat64 enable
      set embedded-ipv4-address enable
   next
```

The DNS64 prefix must be configured as 64:ff9b::/96 and the aaaa record synthesis must be enabled:

```
config system dns64
   set status disable
   set dns64-prefix 64:ff9b::/96
   set always-synthesize-aaaa-record enable
end
```

The FortiGate is configured to accept DNS queries on the interface towards the client IPv6 network. In addition the FortiGate can resolve domains (dns-server must be configured).

The DNS server can translate A to AAAA records. An A record is responsible for translating a hostname to its corresponding IPv4 address, while the AAAA record is specifying the IPv6 address for a certain host.

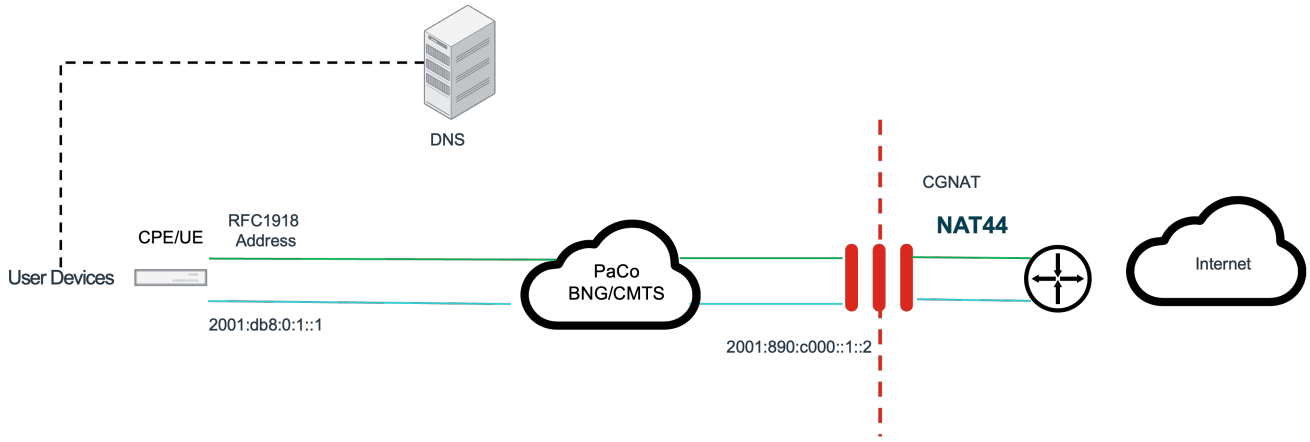
```
config system dns-server
   edit "port2"
      set mode forward-only
      set doh disable
   next
```

When the client tries to resolve some server name, it should point to the FortiGate interface:

```
nslookup
> server 192.168.194.230 <- This should be the IPv6 address of the Fortigate interface
default server: 192.168.194.230
Address: 192.168.194.230#53 > www.google.com
Server: 192.168.194.230
Address: 192.168.194.230#53
Non-authoritative answer:
Name: www.google.com
Address: 216.58.215.36
Name: www.google.com
Address: 64:ff9b::d83a:d724 <- The AAAA record should start with Prefix that is defined in
the DNS64 settings. Converted address is d8 (216), 3a (58), d7 (215), 24 (36). From this,
NAT64 device can extract IPv4 address as 216.58.215.36.
```

Dual Stack

In Dual Stack the client has two IP addresses - IPv4 and IPv6 and depending on the DNS resolution one or the other access will be used. From the FortiGate perspective the private IPv4 access is translated (NAT44) to a public address with the common methods and the public IPv6 can be protected by the FortiGate or bypass it.



As per above, the FortiGate has two IP addresses on the internal and external side:

```
config system interface
   edit "internal"
      set vdom "root"
      set ip 10.104.116.74 255.255.255.252
```

```
set allowaccess ping https ssh http
set type physical
   config ipv6
      set ip6-address fc00:10:104:116::14/127
      set ip6-allowaccess ping https ssh http
   end
next
edit "external"
   set vdom "root"
   set ip 194.166.128.23 255.255.255.252
   set allowaccess ping https ssh http
   set type physical
      config ipv6
         set ip6-address 2001:678:e90::14/127
         set ip6-allowaccess ping https http
      end
   end
```

The NAT44 policy translates the private access with resources from the PBA pool:

```
config firewall policy
   edit 1
      set name "NAT44"
      set srcintf "internal"
      set dstintf "external"
      set action accept
      set srcaddr "all"
      set dstaddr "all"
      set schedule "always"
      set service "ALL"
      set nat enable
      set ippool enable
      set poolname "PBA"
   next
end
```

The PBA NAT IP pool has a start and end IP Address, a block size, and the number of blocks that will be provided for translations to the client. The details about Port Block Allocation CGNAT are provided in the following chapters.

```
config firewall ippool
   edit "PBA"
      set type port-block-allocation
      set startip 150.0.0.1
      set endip 150.0.0.1
      set block-size 128
      set num-blocks-per-user 3
      set pba-timeout 50
      set arp-reply enable
      set arp-intf ''
      set comments ''
   next
end
```

The IPv6 traffic in this example goes through the FortiGate and it is not translated, it is just firewalled via the following policy:

```
config firewall policy
   edit 2
      set name "v6-Policy"
      set srcintf "internal"
      set dstintf "external"
      set action accept
      set srcaddr6 "all"
      set dstaddr6 "all"
      set schedule "always"
      set service "ALL"
   next
end
```

Hyperscale and Kernel CGNAT feature comparison

It is worth to mention that some of the CGNAT features work differently on hyperscale enabled systems compared with native FortiOS. Selecting hyperscale vs. non-hyperscale CGNAT shall be considered based on the required features as well:

Feature	Hyperscale CGNAT	Kernel CGNAT
NAT44 PBA	Supported	Supported
NAT44 Deterministic	Fixed Allocation (The size of the client network is limited to 64k IP addresses).	Fixed Port Allocation
NAT64 PBA	Supported	Supported
NAT64 Deterministic	Not Supported	Not Supported (supported by FortiOS 7.6)
DNS64	Supported	Supported
Overload/Port Reuse	Overload PBA, Overload SPA	PBA and fixed port allocation pools
Preserving the client src port	Not Supported	Supported, however if fixed-port-range is used, client SRC port traffic may be blocked due to contradiction.
NAT timers per protocol	VDOM and per policy session-ttl per policy takes precedence over the VDOM config	per VDOM, per policy with custom services session-ttl per policy takes precedence over the VDOM config
Session Refresh Direction	outgoing | incoming | both. Supported under `config system npu`.	Supported under `config system session-ttl`.
Session quota	In the policy	per IP shaper in the policy
Session Logging	Hyperscale logging and Syslogd via host logging (npu-server). Note that only FAZ-BD can parse HS logs.	Syslogd
PBA Logging	Hardware logging	Syslogd (event logs)
Log Filtering	Filtering on the syslogd only. Hyperscale logs can be configured per nat-mapping, per-session and per-session ending only.	Supported
Reliable Syslog	Support for syslog transmission using TCP.	Supported
Netflow v9 and v10	Supported	Netflow v9 only
EIM	Supported in the policy	Default behavior cannot be disabled
EIF	Supported in the policy configuration	PBA only, supported in the pool configuration

Feature	Hyperscale CGNAT	Kernel CGNAT
IP Fragmentation	Supported	Supported
NAT pool utilization alarm	SNMP trap in the policy	via fgTrapPoolUsage of FORTINET-FORTIGATE-MIB
Source and Destination address limits	150 unique IP addresses and 10 overlapping distributed and 9 single IP duplicate range addresses (startip and endip is the same) between the source and destination address fields. A policy with Fixed port range NAT pool does not allow more than 64k src/dst addresses. When more addresses are required, you can add multiple policies.	N/A
RSSO/FSSO	Not supported	Supported
Security profiles in the policy	Not Supported	Supported
Exclusion of IP Addresses from the NAT pools	Via the excludeip option in the NAT pool	Not Supported
NAT-Pooling/Grouping	Grouping of NAT pools of the same type only	Grouping of different NAT pools possible
Port randomization	Supported with PBA and SPA	Supported in FortiOS 7.6
Interim Logging	Support for long-lived sessions, new log fields `durationdelta`/`sentpktdelta`/ `rcvdpktdelta`.	

Hyperscale CGNAT Pools on NP7 Systems

When a FortiGate NP7 system is licensed with hyperscale, it provides HW resource allocation for the following CGNAT Methods:

Translation Methods	Description
Port Block Allocation	PBA provides predefined public IP ranges and port-blocks from the pool which are allocated to client's sessions as they are initiated. When all the client's connections are closed the port blocks are released. In case when the customer demands more ports and the port-block is exhausted and all ports in the block are allocated, PBA tries one more port-block and if available, the system will continue translating, however if the pool is exhaust and all port-blocks are allocated already, it won't translate further because there are no available resources in that pool. No session clash log will be generated in case the subscriber does not get resources for translation, however SNMP trap will be sent based on the configured in the pool

Translation Methods	Description
	alarm thresholds.
Overload Port Block Allocation	Overload PBA is similar to PBA, however the ports within a block can be reused or overloaded, which allows more connections before running out of ports. The overload is finite and it is based on the global npu IP pool overload settings, explained later in this document. No session clash log will be generated in case the subscriber does not get resources for translation, however SNMP trap will be sent based on the configured in the pool alarm thresholds.
Single Port Allocation	A SPA CGN resource allocation IP pool assigns single ports to client sessions from the the configured port range. The logging behavior is similar like the mentioned above: Per-session logging, per-nat mapping and per-session.ending. No session clash log will be generated in case the subscriber does not get resources for translation, however SNMP trap will be sent based on the configured in the pool alarm thresholds.
Overload Single Port Allocation	Overload SPA is similar to SPA and provides single ports to client sessions from the the configured port range with overload function in case the ports in the range are all exhausted. The log behavior is the same as the SPA. The overload is finite and it is based on the global npu IP pool overload settings, explained later in this document. No session clash log will be generated in case the subscriber does not get resources for translation, however SNMP trap will be sent based on the configured in the pool alarm thresholds.
Fixed Allocation	Fixed allocation is in fact deterministic NAT, where the fixed port CGN resource allocation IP pool creates mapping between external and internal IP addresses and port-block. In deterministic NAT the client address is always translated to the same external address. The provided port-block size indicates how many ports will be made available for each client translation. By default there is no log generation and the mapping is based on algorithm, which is reversible and can be used to derive the mapping for further investigation purposes. In case logs are required, this can be configured (per session, per-nat-mapping, per-session-ending. No session clash log will be generated in case the subscriber does not get resources for translation, however SNMP trap will be sent based on the configured in the pool alarm thresholds.
EIM	EIM NAT reuses the port mapping for subsequent packets sent from the same internal client (internal IP address and port) to any external IP address and port. The new sessions do not cause new resource allocation and the new sessions only count towards the session quota. EIM is configured per policy.
EIF	EIF allows different (external) servers to use an existing translation (public IP/port pair) to connect back to the client. This means that the NP7 will create new sessions by reusing the existing mapping (from previous established sessions). Sending traffic to the correct public IP address but different port will cause the communication to fail. EIF is configured per policy and works with non-overloading NAT pools only.

Kernel based NAT Pools

Kernel based NAT pools are available on mainstream and Hyperscale FortiOS.

Kernel based NAT pool types	Description
Overload	Overload NAT IP pools define one or more IP addresses or IP address ranges and their ports starting from 5117 to 65536 (per IP) for network translations. Kernel NAT with Overload NAT pools attempts to map the internal session SRC port with the external session SRC port and with the external/translated session SRC port. Due to the high number of available ports (5117 - 65536) a single client can generate many sessions. Port reuse/oversubscription with Overload NAT pools is supported. This is possible when the new session's 5-tuple is different. Before the resource can be reused, FortiOS determines if the session's 5-tuple is generating clash or not. Resource reuse permits FortiOS to have more than 65K sessions per IP address. A session clash/clash log message is generated if the client does not get resources for the new translation. The use of Overload NAT pools is recommended for networks where a high number of translations per client is required.
One-to-One	The name of this NAT pool explains more or less the nature of it. The client address is translated with a public IP address from the NAT pool where the source ports will be mapped one-to-one (for example, 92.168.1.1:10091 could be translated to 20.40.1.128:10091). A session clash log message is generated if the user does not get resources for translation (all ports per IP address and all addresses in the NAT pool are exhausted). The use of this pool is not recommended because the mapping is one to one, for every client one public IP address will be used.
Fixed Port Range	Fixed port range IP pool is deterministic NAT and requires the definition of both internal IP range and external IP range. The mapping between internal IP and port and external IP and port is based on a reversible algorithm, which can be used to determine the translation and used for further investigation purposes. The calculated single port block is in the range between 5117 and 65536. Port reuse/oversubscription with Fixed port range NAT pool is supported. This is possible when for the new session's the 5-tuple is different. Before the resource can be reused FortiOS tests to determine if the session 5-tuple is generating clash or not. If no resources are available, FortiOS generates a session clash event log indicating the clash.
Port Block Allocation	PBA in kernel NAT provides port-blocks for translations. The allocation is on demand and a port-block is dynamically allocated to each client. Once the block is exhausted, another block is assigned until the maximum configured number of blocks is reached. PBA in kernel NAT supports port reuse, so far the new session (initiated by the client) does not create clash. FortiOS will try to reuse the the original SRC port, however if the port does not fall in the port block range or if using the port will create a session clash, FortiOS will iterate through, starting at a random point in the port range block's space until it finds a non-clashing combination. If no non-clashing combination is found, FortiOS will stop translating and generate a log message.
Hairpining for kernel NAT	As described by Chapter 6 of RFC 4787, hair-pining allows two hosts behind a NAT to communicate even if they only use each other's external IP addresses and ports.

Kernel based NAT pool types	Description
Endpoint Independent Mapping (EIM)	With Endpoint Independent Mapping the client is translated using the IP address and port from the NAT pool and that existing mapping is reused to create new sessions. Some applications require EIM to work properly with NAT devices and for the new sessions no new resources will be allocated, hence EIM saves NAT resources. New sessions that reuse the mapping are counted against the session quota. With mainstream FortiOS, EIM is the default setting and cannot be disabled.

Endpoint Independent Filtering (EIF)	EIF allows different (external) servers to use an existing translation (public IP/port pair) and open new session by reusing the existing mapping.
	With Endpoint Independent Filtering, hair-pining is supported on mainstream FortiOS with PBA only. When a session is established from the internal side of the FortiGate, the FortiOS allows the existing mapping (from the first session) to be reused by externally initiated connections back to the internal endpoint. When traffic is supposed to reach a private host (IP/port), EIF will reuse the existing mapping (external NAT_IP/and NAT_Port), irrespective of the external IP/host that is originating the traffic.
	Fixed port range supports EIF in FortiOS 7.6, see Full cone NAT for fixed port range IP pools.
	The following mainstream FortiOS example allows EIF in the PBA NAT pool:
	`config firewall ippool`
	`edit "PBA"`
	`set type port-block-allocation`
	`set startip 150.0.0.101`
	`set endip 150.0.0.200`
	`set block-size 4096`
	`set num-blocks-per-user 8`
	`set pba-timeout 30`
	`set permit-any-host enable <--- Enables EIF`
	`set arp-reply enable`
	`set arp-intf ''`
	`end`
	EIF is not recommended due to security risk on the client side. To support EIF, FortiOS opens two additional expectation sessions for every public resource created and this could lead to high CPU rates and memory use to manage the additional sessions.

Hyperscale CGNAT

This chapter describes some hyperscale CGNAT features.

Load balancing (NP7 traffic distribution)

The traffic from the interfaces on hyperscale enabled system is load-balanced by the Internal Switch Fabric (ISF) based on hashing which decides how the client's sessions/traffic are distributed to the NPUs within the FortiGate. For hyperscale CGNAT, the ISF load-balancing uses the `src-ip` for the hash calculation so that sessions from a source IP address is distributed to the same NP7 processor.

The load-balancing hash calculation setting is global:

```
config global
   config system npu
      set hash-config src-ip
   end
```

Changing the `hash-config` setting requires mandatory system reboot. Also, since the load balancing/ hash-config setting is global it affects all VDOMs. Because the reverse traffic path can use on any of the NPUs, session information must be synchronized across all NPUs. A Reliable Link Terminal-RLT provides transportation service for cross communication of session data between NPUs on multi-NPU systems. In the hyperscale CGNAT use case, the IP pool resources (from the same NPU group) are present on all NPUs belonging to it. Once a resource is allocated, it cannot be reserved by different NPUs. Because all NPUs have the same session information it is not relevant which NPU terminates the return path. The resource quota is also accurately enforced because the client IP is anchored to the same NPU.

For the CGNAT use case we don't recommend configuring `5-tuple` for hash calculation for internal load-balancing because the ISF will spread session/traffic from single customer IP eventually to all system NPUs. Because of this effect, the NAT pool resources must be split among all system NPUs and with that the NPU resources usage is not optimal.

Load balancing and single-NP7 systems

Single NPU systems (for example the FortiGate 1800F and 2600F series) do not require setting the `hash-config` for the traffic distribution because no load-balancing is required.

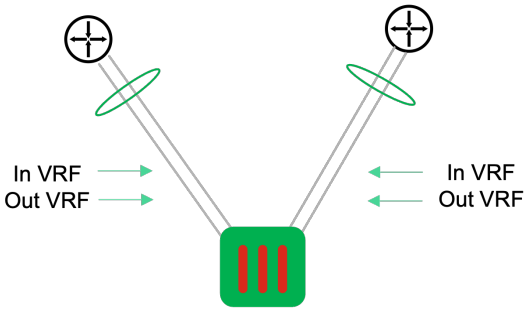
Load balancing and FortiGate 4800F series systems

The FortiGate 4800F series includes 16 NP7 NPUs and due to the limited number of RLT channels per NP7 processor the cross NPU connection does not support the required capacity by all NPUs. On 4800F hyperscale systems, the system must be split with four hyperscale VDOMs and four NPU groups (each with four NP7 processors). The group ID along with the NPU ID must be configured in each of the four hyperscale VDOMs:

```
config system settings
   set policy-offload-level full-offload
   set npu-group-id {0 | 1 | 2 | 3}
end
```

The connectivity architecture with four VDOMs 4800F systems must be carefully planned, due to the requirement for 800Gbps throughput per VDOM. On 4800F, inline setup with dual uplinks/downlinks (two routers upstream and two routers downstream) would provide 800Gbps in both directions. Such setup with all four VDOMs could become quite complex. This is why we recommend using on-a-stick architecture instead of inline architecture. In on-a stick architecture, uplinks/downlinks are represented with VLANs/VRFs within aggregated LAGs with 2x400GE interfaces per VDOM.

On a stick architecture



On FortiGate 4800Fsystems, hyperscale logging is impacted by the npu-group split. The hyperscale system can only send logs using interfaces in the same NP7 processor group as the NP7 processors that are handling the hyperscale sessions. This means that the logging servers in a logging server group must be using the same VDOM/same npu-group and the interfaces used for logging must be must be in the hyperscale VDOM deploying the same npu-group. This means that there will be four sets of logging servers/four logging server groups with 1:1 mapping of npu-group/npu mapping and logging server group with four VDOMs. The dedicated interfaces for logging (from the corresponding logging server group) will be in the same VDOM, deploying the particular npugroup/npu mapping.

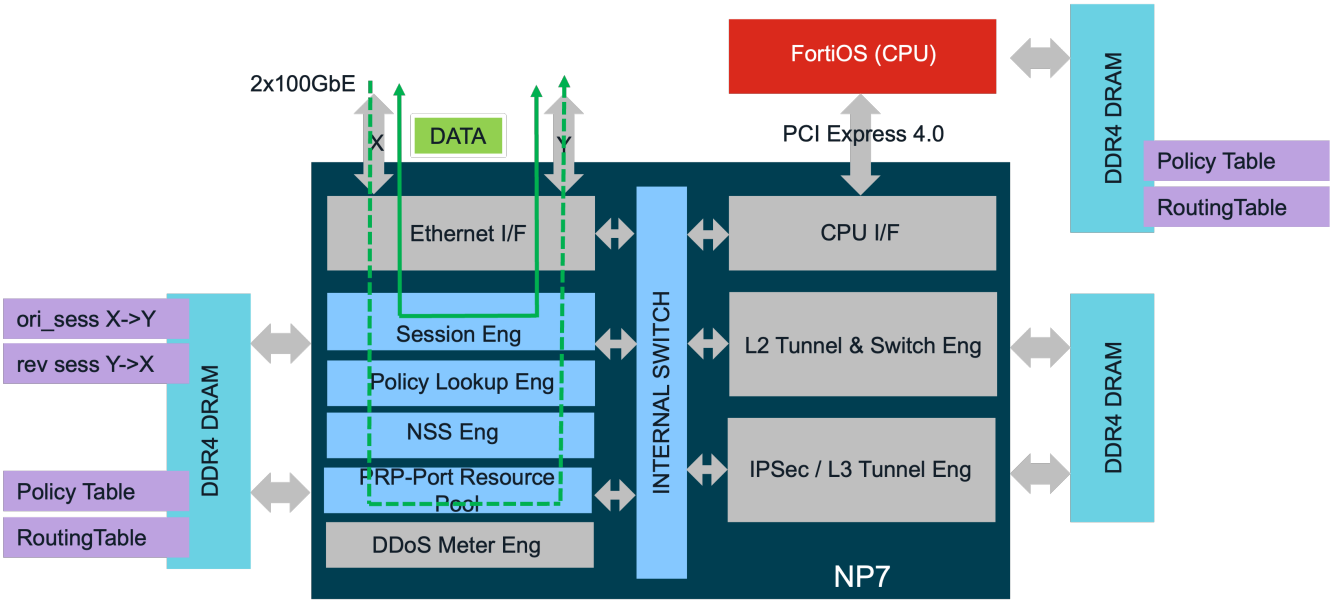
Hardware Session setup

Hardware session setup takes place entirely in the NP7 and its modules:

- NSS-NAPT Session Setup module, which works with the policy/route results from
- PLE-Policy Lookup Engine and applies address translations and inserts session entries to the
- SSE-Session Search Engine and DSE-Destination Search Engine NP7 modules. The NAPT is assisted by the
- PRP-Port Resource Pool module, which manages the Hyperscale NAT pools.

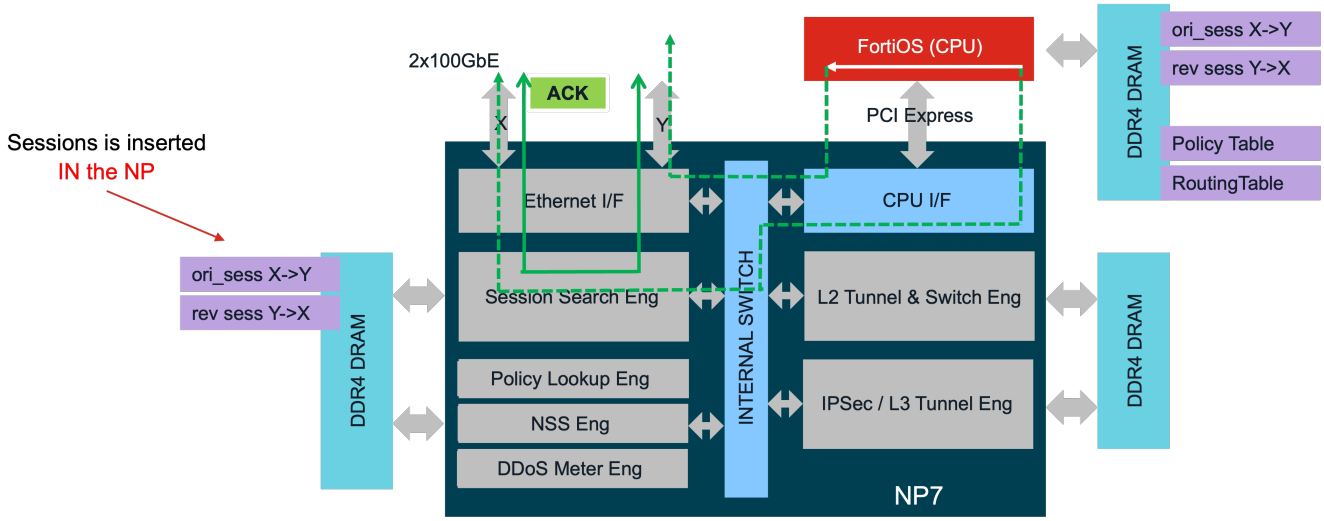
As you can see the CPU is not involved at all in the CGNAT Hyperscale Translations.

NP7 Hardware CGNAT Session Setup



Some applications are not compatible with hardware session setup. Typically these are applications that match the Application Layer Gateways (ALGs) and applications using different protocols. In this case those sessions would be established by the kernel and offloaded (when possible) to the NPUs.

NP7 Kernel CGNAT Session Setup



The resource reservations for sessions established by the NPU are made separately for TCP and UDP applications. There is further CPU and NPU NAT resources split within the same NAT pool and resource overlapping can happen, this however does not cause any traffic issues. The CPU resource allocation does not use further split for TCP and UDP.

Let's take the following NAT pool as example:

```
config firewall ippool
   edit "pba"
      set type cgn-resource-allocation
```

```
      set startip 150.0.0.0
      set endip 150.0.0.255
      set arp-reply enable
      set arp-intf ''
      set cgn-spa disable
      set cgn-overload disable
      set cgn-fixedalloc disable
      set cgn-block-size 256
      set cgn-port-start 1024
      set cgn-port-end 65530
      set utilization-alarm-raise 100
      set utilization-alarm-clear 80
   end
```

And check the ippool status:

```
diagnose firewall ippool list
list ippool info:(vf=cgn-hw1)
ippool pba: id=1, block-sz=256, num-block=8, fixed-port=no, use=2
     ip-range=150.0.0.1-150.0.0.255 start-port=5117, num-pba-per-ip=236
<--- kernel NAT, starting with 5117
     clients=0, inuse-NAT-IPs=0
     total-PBAs=472, inuse-PBAs=0, expiring-PBAs=0, free-PBAs=100.00%
     allocate-PBA-times=0, reuse-PBA-times=0
     grp=N/A, start-port=1024, end-port=65530
<--- NPU NAT, starting with 1024
     npu-clients=0, npu-inuse-NAT-IPs=0, total-NAT-IP=256
     npu-total-PBAs=65536, npu-inuse-PBAs=0/0, npu-free-PBAs=100.00%/100.00%
<--- the npu-inuse-PBA=0/0, first value is UDP second is TCP
     npu-tcp-sess-count=0, npu-udp-sess-count=0
```

When we try to list the users from the NAT pool (after there is traffic flowing) we can see that there is some resources overlapping:

```
diagnose firewall ippool list pba
…
user 172.18.0.10: 150.0.0.2 5633-6144, idx=9, use=1
user 172.18.0.5: 150.0.0.2 5629-6140, idx=1, use=5
…
```

As you can see for the address 150.0.0.2 the first and second port ranges are overlapping. The overlapping resource reservation/usage has impact on the logging and how subscribers can/shall be identified. This means that the request for user identification must be very specific and the protocol ID must be specified along with the public IP address and source port.

At this point it is also worth to mention some of the hyperscale incompatibilities and limitations.

Hyperscale basics

This section describes some basic hyperscale features.

Hyperscale license activation and creating a hyperscale VDOM

To activate a hyperscale license, see Applying the hyperscale firewall activation code or license key.

The following models can be licensed with hyperscale features:

- FortiGate 1800F
- FortiGate 2600F
- FortiGate 3000F
- FortiGate 3500F
- FortiGate 4200F
- FortiGate 4400F
- FortiGate 4800F

Be aware that after applying the hyperscale license the system must be rebooted.

The FortiGate-3200F, 3201F, 3700F and 3701F cannot be licensed for hyperscale firewall support.

The Hyperscale CGNAT setup requires a hyperscale VDOM. The hyperscale VDOM uses specific naming including VDOM name and a VDOM ID number. Before you create a hyperscale VDOM you need to set the maximum number (up to 250) of hyperscale VDOMs on the system first:

```
config system global
   set hyper-scale-vdom-num
end
```

Note that the range is between 1 and 250 and adding more than 10 VDOMs requires a VDOM license.

Enable multi-vdom mode:

```
config system global
   set vdom-mode multi-vdom
end
```

Create a **hyperscale VDOM**:

```
config vdom
   edit <string>-hw<vdom-id>
end
```

The `vdom-id` number must be in the range defined by the global `hyper-scale-vdom-num` setting and the string can contain alphanumeric upper or lower case characters and the - and _ characters.

Avoid using leading zeros in the to keep from accidentally creating duplicate IDs. The VDOM name, including the `-hw`, can be a up to a total of 11 characters long.

Next, system policy offloading must be enabled for full offload globally:

```
config global
   config system npu
      set policy-offload-level full-offload
   end
```

Along with the global setting, full policy offload must be enabled for all hyperscale VDOMs:

```
config vdom
   edit <string>-hw<vdom-id>
      config system settings
         set policy-offload-level full-offload
      end
```

In the hyperscale VDOM, hyperscale policies, configured with CGNAT pools must be also configured for offloading. The `policyoffload` option enables hyperscale session setup and `auto-asic-offload` enables NP7 acceleration/offloading in the policy:

```
config firewall policy
   edit 1
      set name "cgn44_pba"
      set srcintf "port1"
      set dstintf "port2"
      set action accept
      set srcaddr "all"
      set dstaddr "all"
      set service "ALL"
      set auto-asic-offload enable <-- Enables NP acceleration in the policy
      set policy-offload enable <-- Enables hyperscale sessions setup in the policy
      …
      set nat enable
      set ippool enable
      set poolname "pba"
   end
```

Port Overloading/Reuse

Some hardware accelerated CGNAT pools (Overload PBA and Overload SPA) can reuse ports if the new session does not generate clash.

Either sessions from the same client can get the same resources (NAT IP and NAT port), or sessions from different clients may be assigned with the same resources, so far the those sessions do not clash.

The thresholds for the port reuse/overload for hardware accelerated NAT pools can be configured globally:

```
config system npu
   set ippool-overload-low 150
   set ippool-overload-high 200
end
```

The high threshold is used to set the limit to stop overloading the port block (considered FULL when 200% of the block size is reached) and the low threshold is used to set the starting point to overload again (considered not FULL when the numbers of sessions reach 150% of the block size).

In the example above if the block size is 128 with 200% high overload, the maximum allocations in this block will be

```
2 * 128 = 256 ports
```

After reaching the threshold of 256 sessions, the block cannot be used, however when the usage falls below 150%

```
1.5 * 128 = 192 ports
```

the block will start taking new allocations again until it hits the 200% again.

If the NAT pool consists of multiple port blocks and the first port block is full (meaning it has reached `ippool-overload-high` limit of the port size), the next available port block will provide ports following the same logic (as described above).

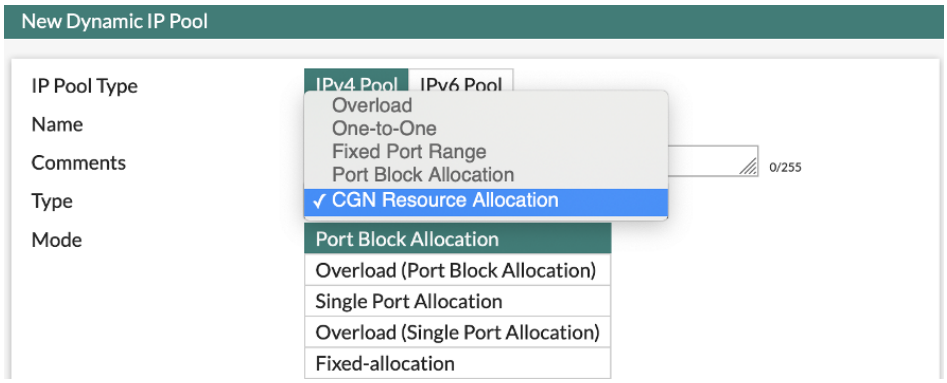
NAT Pooling/soft-APP

Hyperscale CGNAT supports the grouping of multiple NAT pools from the type in a CGN IP Pool Group and this group can be referenced in a policy with network translation. The pool order in the group is important because hyperscale FortiOS starts providing resources from the first listed pool in the group and once this pool gets exhausted, the next pool in the group will be used for ongoing translations.

Chapter 4 of RFC 7857 describes this situation and the FortiOS implementation is similar to “soft-APP”. When the resources from the first pool are exhausted the user will get different resources from the second pool in order. In this case the user will utilize two different public addresses.

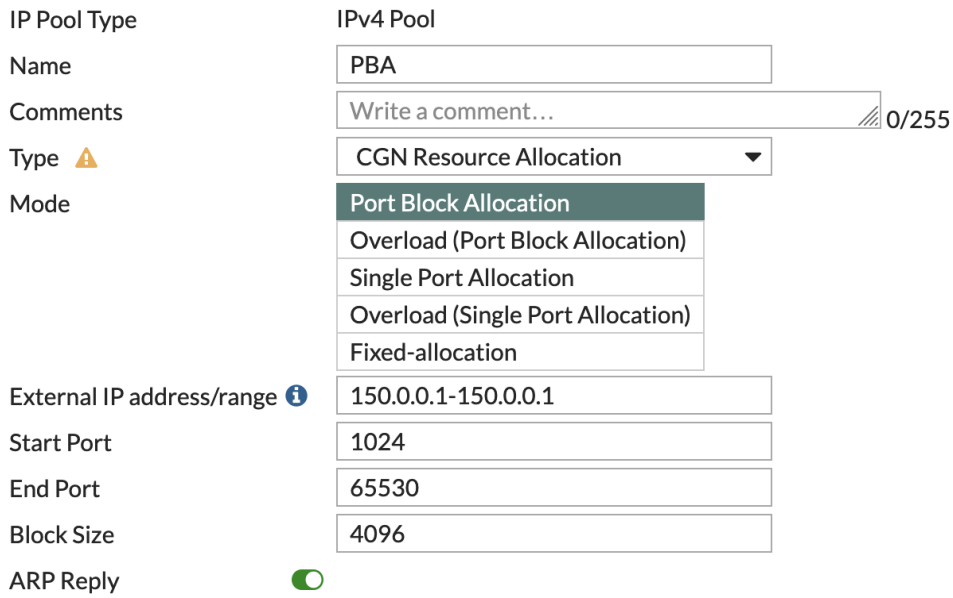
Hardware accelerated CGNAT pools - CGN Resource Allocation

If you select CGN Resource Allocation you can configure hardware accelerated Port Block Allocation, Overload Port Block Allocation, Single Port Allocation, Overload (Single Port Allocation) and Fixed Allocation. The most used CGNAT architectures are based on PBA and Fixed-Allocation deterministic NAT.



Port Block Allocation (PBA)

PBA provides predefined port blocks in a NAT IP address pool, which are allocated to user’s sessions as they are initiated.




```
config firewall ippool
   edit "pba"
      set type cgn-resource-allocation
      set startip 150.0.0.0
      set endip 150.0.0.255
      set arp-reply enable
      set arp-intf ''
      set cgn-spa disable
      set cgn-overload disable
      set cgn-fixedalloc disable
      set cgn-block-size 4096
      set cgn-port-start 1024
      set cgn-port-end 65530
      set utilization-alarm-raise 100
      set utilization-alarm-clear 80
   end
```

The PBA pool provides the option to configure the start `cgn-port-start` and end port `cgn-port-end` for translations. SNMP alarms will be generated when a `utilization-alarm` threshold is reached.

PBA address pools do not overload and when all the port-blocks are exhausted, PBA stops translating. If the overload function is required, you can use Overload PBA NAT pools.

Hyperscale PBA splits the resources for applications that support hyperscale session setup and applications, handled by the session helpers, that do not support accelerated session setup. The address pool is split and for sessions established in the kernel, Hyperscale PBA provides port blocks with with port resources starting with port 5117. For the rest of the applications, for which accelerated session setup is possible, Port Block Allocation NAT splits the resources for TCP and UDP and starts providing port blocks with ports starting at 1024.

In the following example, the PBA Hyperscale pool is split into two, for sessions for kernel NAT starting with 5117 and sessions for NPU NAT, starting with 1024. For TCP and UDP the Hyperscale PBA NAT pool provides different resources:

```
diagnose firewall ippool list
list ippool info:(vf=Test01-hw10)
ippool PBA-HS: id=1, block-sz=128, num-block=8, fixed-port=no, use=2
ip-range=30.0.0.0-30.0.0.255 start-port=5117, num-pba-per-ip=472      <-- Kernel NAT
clients=0, inuse-NAT-IPs=0
total-PBAs=120832, inuse-PBAs=0, expiring-PBAs=0, free-PBAs=100.00%
allocate-PBA-times=0, reuse-PBA-times=0
grp=N/A, start-port=1024, end-port=65530                              <-- Hyperscale NAT
npu-clients=0, npu-inuse-NAT-IPs=0, total-NAT-IP=256
npu-total-PBAs=65536, npu-inuse-PBAs=0/0, npu-free-PBAs=100.00%/100.00%
npu-tcp-sess-count=0, npu-udp-sess-count=0
```

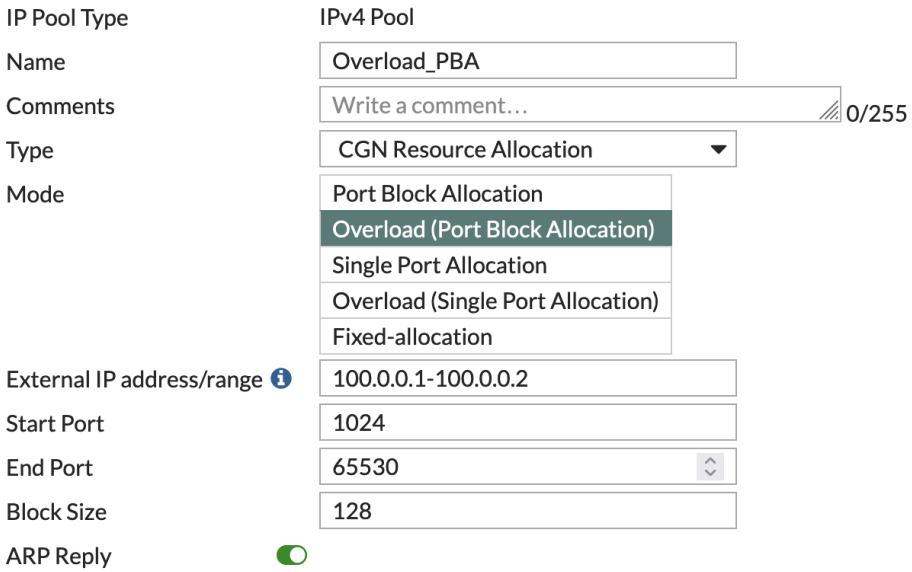
Further on, the PBA logs are quite small (about 350 bytes) and depending how hyperscale hardware logging is configured, PBA can generate logs per session, per-session ending and per-nat mapping and the log size would be different. For more information about logging, see CGNAT Logging on page 60.

Overload Port Block Allocation PBA

Overload PBA uses HW accelerated Port Block Allocation independent from the overload setting (ports can be reused, and the port reuse is determined by how much the pool is utilized). The overload depends on the 5-tuple of the new sessions. FortiOS checks the new session for clashes and if the new session doesn't clash, it may overload the existing mapping which again depends on how overloaded the pool is and what are the global overload NPU settings. If no resources are available, PBA will stop translating.

When a firewall policy is translating sessions with overload PBA NAT pool, only one block is allocated per client and the CGN Resource Quota (`cgn-resource-quota`) is ignored. This usually works because of resource reuse. Certain rare conditions such as session clash due to no available resources, may cause port allocation to fail.

Figure




```
config firewall ippool
   edit "Overload_PBA"
      set type cgn-resource-allocation
```

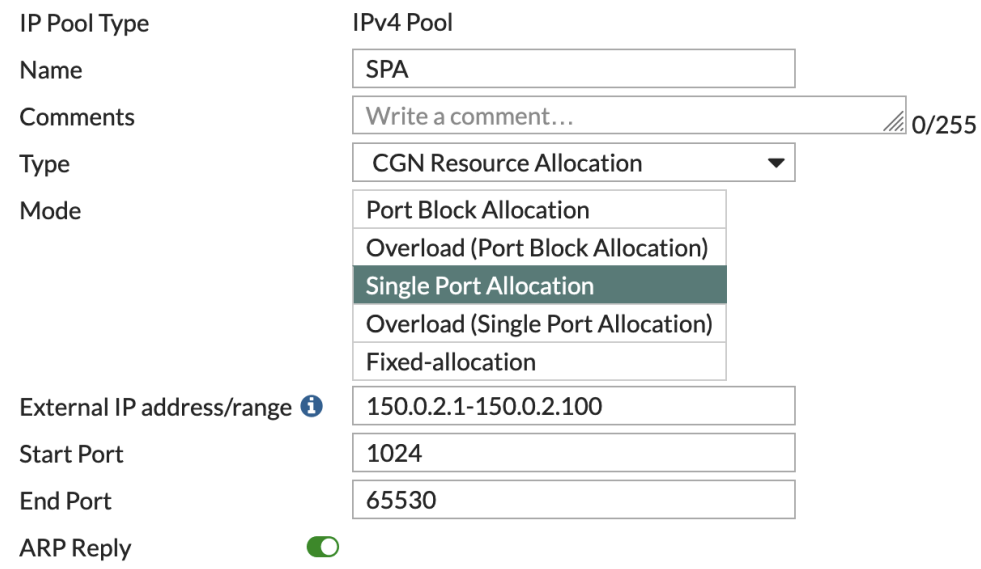
```
    set startip 100.0.0.1
    set endip 100.0.0.2
    set arp-reply enable
    set arp-intf ''
    set cgn-spa disable
    set cgn-overload enable
    set cgn-fixedalloc disable
    set cgn-block-size 128
    set cgn-port-start 1024
    set cgn-port-end 65530
    set utilization-alarm-raise 100
    set utilization-alarm-clear 80
    set comments ''
end
```

The Overload PBA pool provides the option to configure the start `cgn-port-start` and end port `cgn-port-end` for translations. The overload is finite and it is based on the global npu IP pool overload settings, mentioned earlier. SNMP alarms are generated when utilization alarm thresholds are reached. When overload is configured, `cgn-overload enable`, the use of EIF is not a viable option due to the fact that matching a single private IP for public inbound connection is not possible.

Again, there is resource split for kernel NAT starting with ports 5117 and Hyperscale NAT starting with port 1024 and there is the TCP and UDP resources split for the sessions processed by the NPU/hyperscale NAT. The type of logs, their size (and how logging can be configured for Overload PBA) is the same as for PBA. The level of overload is a global NPU configuration setting under `ippooloverload` low and high.

Single Port Allocation (SPA)

SPA assigns a single port block, instead of a ranges of port blocks like PBA. This method effectively reduces the port block size to 1. This method conserves ports to single port block size. When SPA is enabled, the FortiGate firewall will ensure that the same address is assigned from the SPA IP pool to the host for multiple concurrent sessions. When a packet comes in and the HW NAT module finds its private side IP is an “existing” one, meaning there are ongoing sessions alive, it will use the same public IP that already assigned to this private IP and choose a new port for the current packet/flow so that each client session gets a new port from the range of ports added to the IP pool that are available. In case there are no available resources for translation the system will stop translating, and **no clash logs will be generated**.




```
config firewall ippool
   edit "SPA"
      set type cgn-resource-allocation
      set startip 150.0.2.1
      set endip 150.0.2.100
      set arp-reply enable
      set arp-intf ''
      set cgn-spa enable
      set cgn-overload disable
      set cgn-port-start 1024
      set cgn-port-end 65530
      set utilization-alarm-raise 100
      set utilization-alarm-clear 80
      set comments ''
   end
```

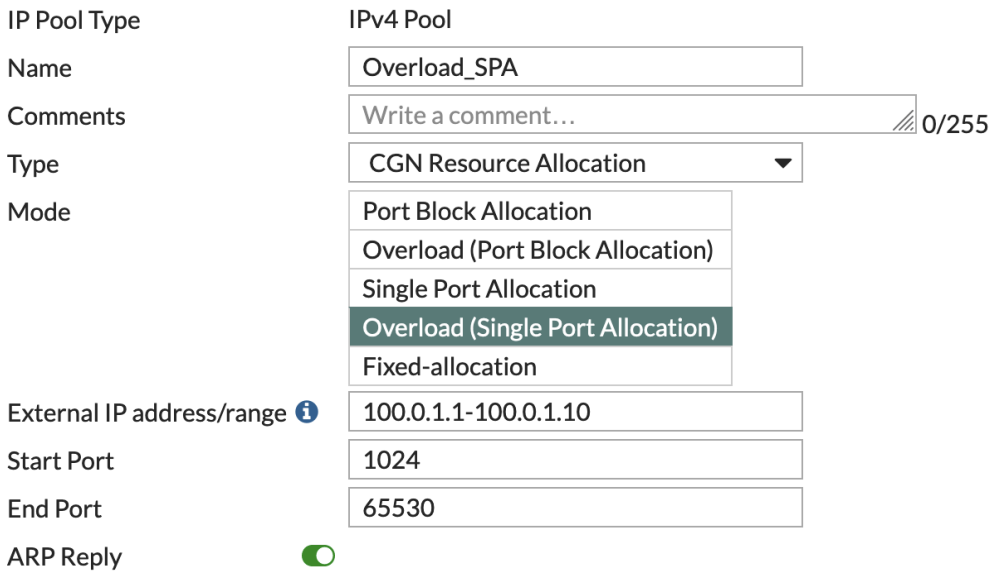
The SPA pool provides the option to configure the start `cgn-port-start` and end port `cgn-port-end` for translations. SNMP alarms will be generated when utilization alarm thresholds are reached.

There is again the spit between applications processed by session helpers and established in the kernel and sessions, for which accelerated session setup in the NPU is possible. There is no split for TCP and UDP because the algorithm reuses the port number and protocol and during translation single ports would be used without overload.

```
diagnose firewall ippool list
list ippool info:(vf=cgn01-hw10)
…
ippool SPA_HS: id=4, block-sz=128, num-block=8, fixed-port=no, use=2
     ip-range=20.0.0.0-20.0.0.255 start-port=5117, num-pba-per-ip=472 <-- Kernel NAT
     clients=0, inuse-NAT-IPs=0
     total-PBAs=120832, inuse-PBAs=0, expiring-PBAs=0, free-PBAs=100.00% <--Hyperscale NAT
     allocate-PBA-times=0, reuse-PBA-times=0
```

Overload Single Port Allocation

Overload Single Pool Allocation NAT is similar to Single Port Allocation with added overload function to reuse ports whenever possible (if the 5-tuple of the new session does not create clash). Again, the split between resources allocated for applications that are established in kernel and NPU established session is the same as for SPA. Port reuse is determined by how much the pool is utilized.




```
config firewall ippool
   edit "Overload_SPA"
      set type cgn-resource-allocation
      set startip 100.0.1.1
      set endip 100.0.1.10
      set arp-reply enable
      set arp-intf ''
      set cgn-spa enable
      set cgn-overload enable
      set cgn-client-ipv6shift 0
      set cgn-port-start 1024
      set cgn-port-end 65530
      set utilization-alarm-raise 100
      set utilization-alarm-clear 80
      set comments ''
   end
```

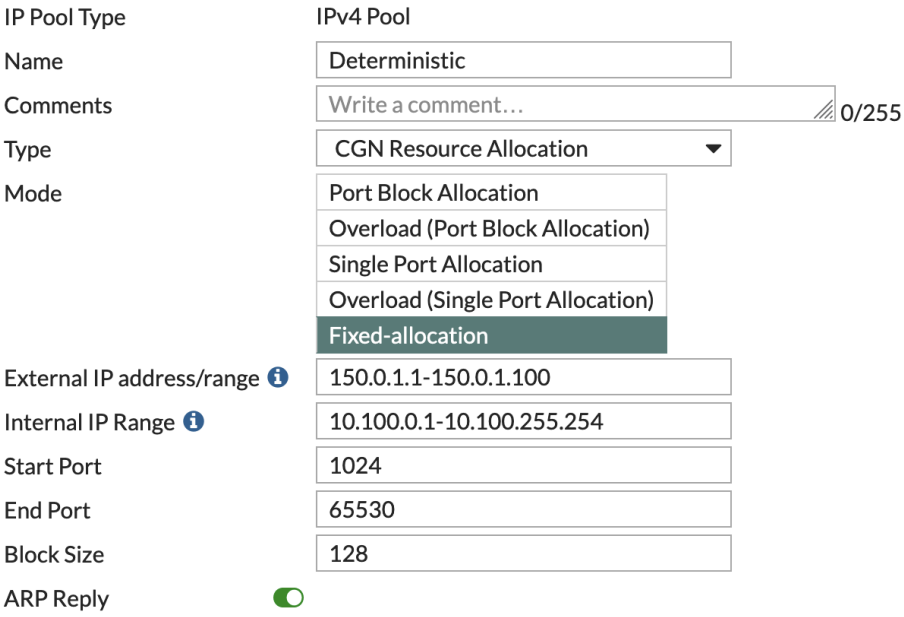
The Overload SPA pool provides the option to configure the start `cgn-port-start` and end port `cgn-port-end` for translations. Note that when overload is configured `cgn-overload enable`, the use of EIF is not viable option due to the fact that matching a single private IP for public inbound connection is not possible. The overload is finite and it is based on the global npu IP poll overload settings, mentioned earlier. SNMP alarm will be generated in case the utilization alarm thresholds are reached.

Fixed-allocation

The Hyperscale fixed allocation IP pool provides deterministic NAT and as such it does not require logging (it is configurable however) because it uses an algorithm that always allocates the internal IP to the same external IP address and port-block for NAT. The algorithm is similar to the one used in kernel fixed-port-allocation NAT. Basically, the algorithm calculates the number of clients on one public IP. Then it finds single port block (ports available for that public IP) and rounds the number to the closest to the configured block size. Due to the fact that resources are split between the NPUs the amount of resources per NPU is limited to 64k client (and public) addresses.

This is the main reason why Fortinet does not recommend to use Fixed allocation CGNAT pools unless this type of translation/use case must be supported by the FortiGate with smaller amount (64k) of addresses.

For this NAT pool you must configure the internal and external ranges, start/end ports as well as the block size and the utilization alarm settings. The internal and external ranges are mapped based on a reversible algorithm. A client gets a single port block, which is automatically calculated to the maximum possible block size (also considering the configured block size) using the external and internal IP ranges, and the start and end ports.




```
config firewall ippool
   edit "Deterministic"
      set type cgn-resource-allocation
      set startip 150.0.1.1
      set endip 150.0.1.100
      set arp-reply enable
      set arp-intf ''
      set cgn-spa disable
      set cgn-overload disable
      set cgn-fixedalloc enable
      set cgn-client-ipv6shift 0
      set cgn-block-size 128
      set cgn-client-startip "10.100.0.1"
```

```
      set cgn-client-endip "10.100.255.254"
      set cgn-port-start 1024
      set cgn-port-end 65530
      set utilization-alarm-raise 100
      set utilization-alarm-clear 80
      set comments ''
   end
```

If the port-block gets exhausted the system stops translating. There is no port overloading possible with fix-allocation IP pools. This means that if no further resources are available, no translations will be made and no clash logs will be generated. SNMP alarms will be generated if the utilization alarm thresholds are reached.

There is resource split for applications processed by session helpers and established in the kernel and sessions for which the accelerated setup in the NPU. There is no further split for TCP and UDP resource because no overload is possible with hyperscale fixed allocation NAT pools.

```
diagnose firewall ippool list
list ippool info:(vf=Test01-hw10)
…
ippool Deterministic_CGNAT: id=6, block-sz=128, num-block=8, fixed-port=no, use=2
        ip-range=117.1.1.0-117.1.1.255 start-port=5117, num-pba-per-ip=472
        clients=0, inuse-NAT-IPs=0
        total-PBAs=120832, inuse-PBAs=0, expiring-PBAs=0, free-PBAs=100.00%
        allocate-PBA-times=0, reuse-PBA-times=0
```

Firewall policies with Hyperscale CGNAT

A new version of the hyperscale firewall policy engine was added to FortiOS 7.4.3. The following limits have been introduced:

- Maximum amount of firewall policies with hyperscale NAT: 15,000.
- Maximum number of port-ranges specified by firewall addresses that can be added to a single hyperscale firewall policy: 1,000.
- Maximum number of port-ranges that can be added to the firewall policy database: 4,000.
- A single IPv4 hyperscale firewall policy can have up to 150 unique IP addresses distributed between the source and destination address fields.
- An IPv4 hyperscale firewall policy can have up to 150 unique IP addresses and 10 overlapping subnets distributed between the source and destination address fields.
- An IPv4 hyperscale firewall policy can have up to 150 unique IP addresses and 9 single IP duplicate range addresses distributed between the source and destination address fields.
- An IPv6 hyperscale firewall policy can have up to 20 IPv6 IP addresses distributed between the source and destination address fields.

Example firewall policy with hyperscale CGNAT pools:

```
config firewall policy
   edit 1
      set name "cgn44_pba"
      set srcintf "port1"
      set dstintf "port2"
      set action accept
      set srcaddr "all"
      set dstaddr "all"
```

```
    set service "ALL"
    set auto-asic-offload enable
    set policy-offload enable
    set cgn-session-quota 16777215
    set cgn-resource-quota 16
    set cgn-eif disable
    set cgn-eim disable
    set cgn-log-server-grp 'SG_CgNatLog_sess'
    set nat enable
    set ippool enable
    set poolname "pba"
  end
```

`policy-offload` must be enabled with hyperscale session processing. The following additional settings are available:

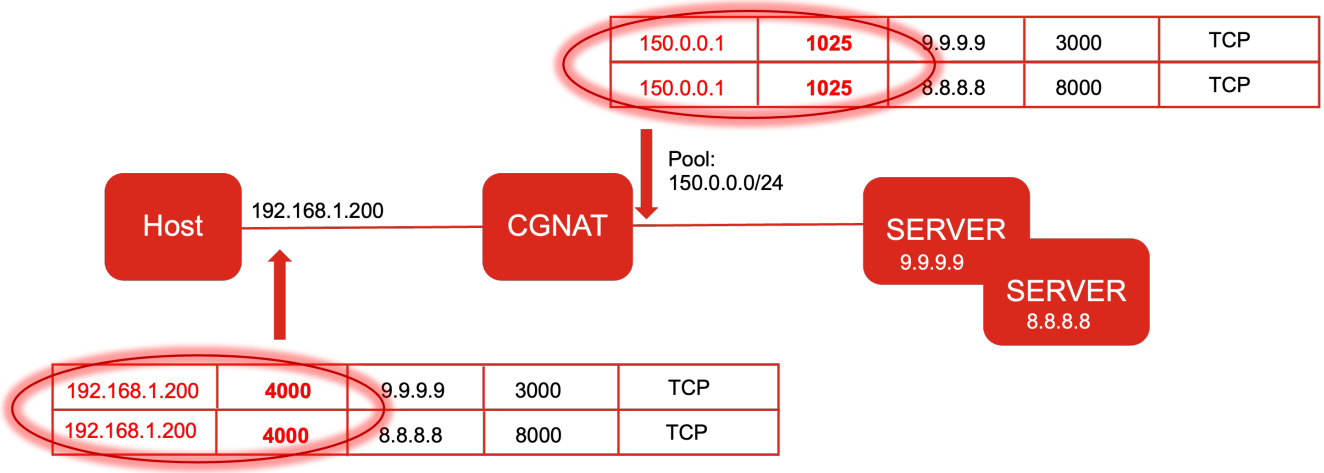
Session Quota

Sometimes operators want to limit the NAT resources and this setting can be configured in the policy. The session quota only applies to hardware sessions and does not apply to CPU sessions. Furthermore there is no quota limit of total software and hardware sessions. You can use per-ip traffic shaping to limit CPU sessions.

The `cgn-resource-quota` represents the number of blocks per client and `cgn-session-quota` represents the number of sessions per client. Depending on configured NAT pool type the number of blocks can be different (fixed allocation is single port block, vs. PBA, which can provide multiple port blocks).

Endpoint Independent Mapping

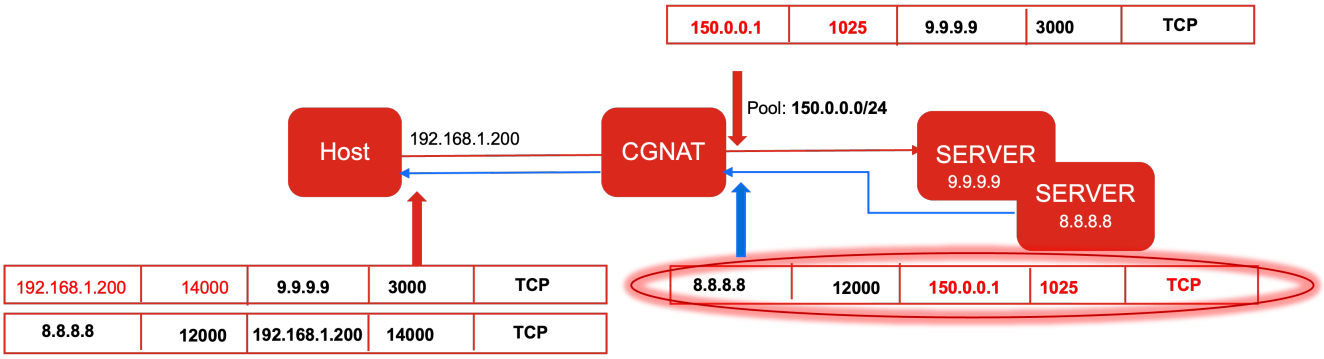
Endpoint Independent Mapping (EIM) (`set cgn-eim enable`) provides the same public resources (NAT_IP/SRC_ port) when the source/private IP and port is the same even if the destination address is different.



Endpoint-Independent Filtering

Endpoint-Independent Filtering (EIF) (`set cgn-eif enable`) allows traffic originating from Internet towards internal hosts through already opened pinhole/mapping. When traffic is supposed to reach a private host (IP/port), EIF will reuse

the existing mapping (external NAT_IP/port), irrespective of the external host (src IP/ src Port) that is originating the traffic. A NAT device deploying EIF will accept incoming traffic to a mapped public port from ANY external endpoint on the public network. It is good practice to define policy in which specific source/destinations are configured in order to limit the communication to only few servers and not opening the clients to the whole Internet because EIF is required.



Enabling EIF decreases the session capacity because for every outgoing session established by policy with EIF, two expectation sessions are created to enable the incoming sessions (one from the server, which can “tickle” the connection and another, which will be used by any host on Internet to connect to the client).

EIF does not work with Overload hyperscale type of pools (Overload PBA, Overload SPA), because overloading the NAT ports makes the mapping of those ports for (returning traffic) with EIF not possible.

Generally, EIF is not recommended to be deployed by service providers for security reasons (the client is unprotected from the internet and any vulnerable application could be exploited).

Hardware Logging

The log server group is also located in the firewall policy. The details related to configuring the hardware accelerated logging are discussed in CGNAT Logging on page 60.

IP address exclusion for Hyperscale CGNAT IP Pools

For some of the hardware accelerated CGNAT pools you can use the `excludeip` option to block the corresponding CGNAT pool from allocating one or more IP addresses. You may want to exclude an IP address from being allocated by a CGNAT pool if those IP addresses have been targeted by external attackers or they have been are blacklisted. To exclude individual IP addresses using the `excludeip` option in the IP pool, for example:

```
config firewall ippool
   edit <name>
      set type cgn-resource-allocation
      set excludeip <ip_address>, <ip_address>, <ip_address>
      ...
   end
```

Where `<ip-address>` is a single IP address, you cannot add IP address ranges. The limit (how many IP addresses can be excluded) depends on the FortiGate model.

You cannot exclude IP addresses in a fixed allocation CGN resource allocation IP pool.

Session Timers

Some MNOs will want to tune the session timers down for several reasons. The air interface and mobility of the UE leads to an increased chance of having sessions which are not terminated cleanly. Leaving the session and NAT resources in use for the session inactivity timer is potentially leading to a starvation of NAT resource. However, smart phones have a cloud connection which they maintain as up. Typically the smart phones run an algorithm to determine how often they need to “tickle” the connection to keep the state in the serving network active. Reducing this session timer has an impact on the battery life of the Smart phone as it needs to come out of low-power standby mode to send the keep-alive message more frequently.

Protocol timers

You can configure ports and refresh direction in the `session-ttl` per VDOM:

```
config vdom
   edit <vdom-name>
      config system session-ttl
         config port
            edit 1
               set protocol <protocol-number>
               set timeout <timeout>
               set refresh-direction {outgoing | incoming | both}
            end
         end
      end
```

The setting `refresh-direction {outgoing | incoming | both}` controls whether idle outgoing or incoming or both outgoing and incoming sessions are terminated when the timeout is reached. This is important security control and when set to `incoming` no internet servers can refresh the service ttl.

In addition to the VDOM `session-ttl` settings, you can also fine tune the session timeouts for individual hyperscale policies. You can use the following commands to create TCP and UDP session timeout profiles and then apply these profiles to individual hyperscale firewall policies.

Use the following command to create a TCP timeout profile:

```
config global
   config system npu
      config tcp-timeout-profile
         edit <tcp-profile-id>
            set tcp-idle <seconds>
            set fin-wait <seconds>
            set close-wait <seconds>
            set time-wait <seconds>
            set syn-sent <seconds>
            set syn-wait <seconds>
         end
      end
   end
```

Use the following command to create a UDP timeout profile:

```
config global
   config system npu
      config udp-timeout-profile
```

```
        edit <udp-profile-id>
           set udp-idle <seconds>
        end
     end
```

And apply a TCP and a UDP timeout profile to a hyperscale firewall policy:

```
config firewall policy
   edit 1
      ...
      set tcp-timout-pid <tcp-profile-id>
      set udp-timout-pid <udp-profile-id>
      ...
   end
```

To check whether there are sessions age-out (premature timeout/Aging success) you can use the following diagnose command:

```
diagnose npu np7 sse-stats -1 | grep ^age

agesucc       1536           1536           1536           1536           6144
agesucc       1440           1632           1440           1632           6144
agesucc       1536           1536           1536           1536           6144
agesucc       1536           1536           1536           1536           6144
```

For TCP the earlier timeout is not a problem because in most cases, there is traffic from the internet to refresh the connection, and because the TCP established state timeout is usually quite long, the client will update the session, which means that `session-ttl` effects TCP established state only.

For UDP the default timeout is 180 seconds and the recommendation is to configure a smaller value for custom use. This is required for the refresh of the UDP sessions without returning from internet traffic. Note that the software refresh period is global, so if a "shorter" UDP timeout is configured and the session has no traffic from internet, session timeout will "kick-in" and the session won't be refreshed.

The following requirements would come from the MNO and are subjective, but in the example below:

- Change the default session TTL to 5 minutes (30 minutes by default)
- Change the UDP idle timer to 1 min (3 minutes by default)
- Change the common HTTPS port to 10 minutes (banking/shopping carts may have problems at 5 min min
- Sets Smart phone cloud connections to 30 minutes
  - 5222 (XMPP) & 5223 (Apple Notification Service) – iPhone
  - 5228 – Android

It may also be true that you only want 5223/tcp to be 30 minutes towards Apple (17.0.0.0/8), in which case the TTL could be set in the firewall policy as opposed to VDOM wide.

Example of what may be appropriate:

```
config system session-ttl
   set default 300
      config port
         edit 17
            set protocol 17
            set timeout 60
            set end-port 65535
         next
         edit 443
            set protocol 6
               set timeout 600
```

```
            set start-port 443
            set end-port 443
        next
        edit 5222
            set protocol 6
            set timeout 3600
            set start-port 5222
            set end-port 5222
        next
        edit 5223
            set protocol 6
            set timeout 3600
            set start-port 5223
            set end-port 5223
        next
        edit 5228
            set protocol 6
            set timeout 3600
            set start-port 5228
            set end-port 5228
        end
    end
end
```

EIF DSE Timeout Configuration

DSE is the destination NAT engine, which is in charge of DST NAT. This setting is specific for hyperscale enabled systems. Fine tuning the values for the DSE timeout will allow EIF to work for as long as the related session is active. The default DSE timeout setting is 10 seconds. Adjust the dse-timeout with the same values as the maximum idle timers described in Session Timers on page 41.

```
config system npu
   ...
   set dse-timeout 3600
end
```

ALG/Session Helper Support

It is important to mention that the ALG's **can be configured and used on NP7 based systems**, but they are not compatible with NP7 hardware session setup. This means that the session setup will be processed by the CPU and not by the NP7 processors for traffic controlled by the ALG/Session Helpers.

NAT is well known to cause problems with certain legacy applications that carry IP addresses in the payload. In order to avoid such problems Fortinet provides session helper support for the following protocols for hyperscale enabled systems:

- FTP
- TFTP
- SIP
- MGCP
- H.323
- PPTP
- L2TP
- ICMP Error/IP-options
- PMAP
- TNS
- DCE-RPC
- RAS
- RSH

For example, the FortiOS SIP Application Layer Gateway allows SIP calls to pass through a FortiGate by opening pinholes and performing source and destination IP address and port translation inside the application payload. The SIP session helper will open the RTP/RTCP ports dynamically (based on previous SIP signaling) and when NAT is configured the session helper will make sure that the corresponding SIP and RTP/RTCP pinholes are created, using the NAT address/port. In same cases the NAT-ed port range for RTP/RTCP can be restricted. For further details refer to SIP pinholes.

It is important to mention that the session helpers/ALG's can be configured and used on NP7 based systems, but they are not compatible with the NP7 hardware session setup. This means that the sessions will be established by the CPU and not the NPU for sessions matching session helpers/ALG.

Depending on the session helper type, one or more CPUs can be involved in traffic processing, which may influence the inspection capacity. When using ALGs (and session helpers) on systems with enabled hyperscale license, the system must be configured to use source IP hashing traffic distribution (set `src-ip` to `hash-config`).

If the traffic processing requires different than `src-ip` traffic distribution, that traffic processing/session setup may not be compatible with the CGNAT configuration, because CGNAT mandates `src-ip` hashing, which is a global configuration and it impacts all VDOMs. This is also one of the reasons why security profiles in the policy are not supported on hyperscale systems.

Kernel CGNAT

Kernel CGNAT is available on mainstream and hyperscale FortiOS.

The session setup for Kernel CGNAT pools is not NPU accelerated. After the session is established, the data is offloaded to NPU. This is important because the setup rate and logging for those sessions will have impact on the overall CCS/CPS capacity.

Kernel CGNAT Firewall policies

Kernel CGNAT or mainstream FortiOS policies that include CGNAT pools (`set ippool enable` and `set pool name`) are similar to the hyperscale use case (hardware acceleration requires few more settings). Native FortiOS NAT pools definitions include:

- Overload
- One-to-One
- Fixed Port Range
- Port Block Allocation

Example native FortiOS CGNAT firewall policy:

```
config firewall policy
   edit 5
      set name "CGNAT"
      set srcintf "port1"
      set dstintf "port2"
      set action accept
      set srcaddr "all"
      set dstaddr "all"
      set schedule "always"
      set service "ALL"
      set logtraffic all
      set nat enable
      set ippool enable
      set poolname "PBA"
   end
```

This firewall policy will use the globally configured log settings:

```
config global
   config log syslogd setting
      set status enable
      set server "192.168.1.200"
      set mode udp
      set port 514
      set facility local7
      set source-ip ''
      set format default
      set priority default
      set max-log-rate 0
      set interface-select-method auto
```

```
   end
```

Native FortiOS CGNAT logging supports syslog and netflow v9. The details such as log format and logging options are described in CGNAT Logging on page 60.

Mainstream FortiOS CGNAT supports the use of multiple different NAT pool types in the policy. You can configure multiple NAT pools in the policy to scale the NAT resources, which means that once the first in order pool is exhausted, the next pool in order will start providing resources. The pooling is not about IP pool grouping, it is a policy with multiple CGNAT pools pools configured:

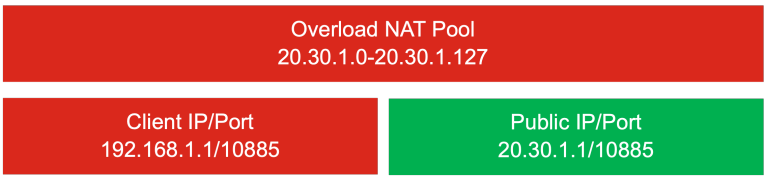
```
config firewall policy
   edit 5
      ...
      set nat enable
      set ippool enable
      set poolname "Overload" "PBA"
   end
```

The pool order in the policy is important. FortiOS will start translating with resources from the first NAT pool in order (in the policy) and once this pool is exhausted, the system will start providing resources from the next pool in order in the policy.

The NAT pooling functionality is similar to soft-APP in RFC7857 Chapter 4 with the caveat of NAT pooling, providing resources from two different NAT pools.

Overload CGNAT

Overload NAT pools map private IPs with IP address and port from the configured pool. Every connection from a private IP will be translated with public IP address and available port between 5117 and 65533.




```
config firewall ippool
   edit "Overload"
      set type overload
      set startip 20.30.1.0
      set endip 20.30.1.255
      set arp-reply enable
      set arp-intf ''
      set associated-interface ''
      set comments ''
      set nat64 disable
   end
```

Overload NAT pool translation tries to map the internal session SRC port with the external (translated) session SRC port:

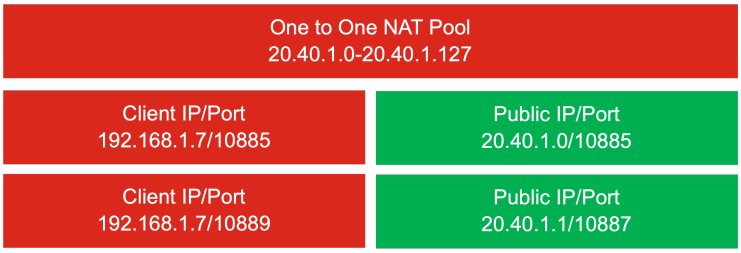
```
diagnose sys session list | grep dir=org
hook=post dir=org act=snat 192.168.1.1:10885->118.1.1.100:80(20.30.1.1:10885)
hook=post dir=org act=snat 192.168.1.1:10887->118.1.1.100:80(20.30.1.1:10887)
hook=post dir=org act=snat 192.168.1.1:10893->118.1.1.100:80(20.30.1.1:10893)
hook=post dir=org act=snat 192.168.1.1:10895->118.1.1.100:80(20.30.1.129:10895)
hook=post dir=org act=snat 192.168.1.1:10889->118.1.1.100:80(20.30.1.129:10889)
```

```
hook=post dir=org act=snat 192.168.1.1:10891->118.1.1.100:80(20.30.1.129:10891)
hook=post dir=org act=snat 192.168.1.1:10901->118.1.1.100:80(20.30.1.129:10901)
hook=post dir=org act=snat 192.168.1.1:10903->118.1.1.100:80(20.30.1.129:10903)
hook=post dir=org act=snat 192.168.1.1:10897->118.1.1.100:80(20.30.1.129:10897)
hook=post dir=org act=snat 192.168.1.1:10899->118.1.1.100:80(20.30.1.129:10899)
hook=post dir=org act=snat 192.168.1.1:10884->118.1.1.100:80(20.30.1.129:10884)
```

One-to-one CGNAT

One-to-One CGNAT type of IP pool maps the internal IP address and the external (translated) IP address to match one-to-one. The port address translation (PAT) is disabled when using this type of IP pool. In the example below, if a One-to-One type IP pool with 100 external IP addresses is defined (150.0.0.1 150.0.0.100), this IP pool only can handle 100 internal IP addresses:

```
config firewall ippool
   edit "One-to-one"
      set type one-to-one
      set startip 150.0.0.1
      set endip 150.0.0.100
      set arp-reply enable
      set arp-intf ''
      set comments ''
   end
```



Again the algorithm tries to use the same source port for the translation with IP address from the configured NAT pool:

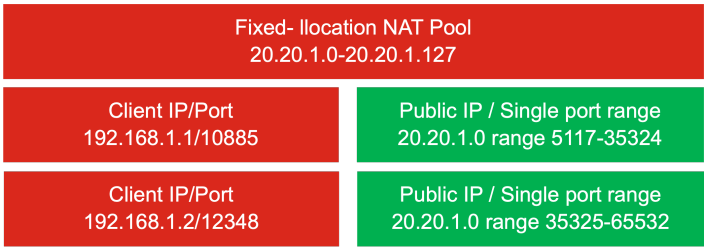
```
diagnose sys session list | grep hook=post
hook=post dir=org act=snat 192.168.1.1:10091->118.1.1.100:80(20.40.1.128:10091)
hook=post dir=org act=snat 192.168.1.3:10089->118.1.1.100:80(20.40.1.129:10089)
hook=post dir=org act=snat 192.168.1.7:10085->118.1.1.100:80(20.40.1.0:10085)
hook=post dir=org act=snat 192.168.1.7:10089->118.1.1.100:80(20.40.1.0:10089)
hook=post dir=org act=snat 192.168.1.6:10087->118.1.1.100:80(20.40.1.1:10087)
hook=post dir=org act=snat 192.168.1.6:10089->118.1.1.100:80(20.40.1.1:10089)
hook=post dir=org act=snat 192.168.1.6:10088->118.1.1.100:80(20.40.1.1:10088)
```

A session clash log will be generated if a client does not get resources for translation (all ports per IP address and all addresses in the NAT pool are exhausted). The use of one-to-one pools in busy networks is not recommended because the mapping is one to one and for every client one public IP address will be used.

Fixed Port Range

Fixed Port Range is kernel CGNAT deterministic NAT and it requires the definition of both the client IP range and public IP range. It uses a reversible algorithm to map the private IP to public IP.

The algorithm calculates the mapping between the client IP address with the public IP address and NAT ports (single port block).




```
config firewall ippool
   edit "Deterministic"
      set type fixed-port-range
      set startip 20.20.1.0
      set endip 20.20.1.127
      set source-startip 192.168.1.1
      set source-endip 192.168.1.254
      set port-per-user 30208
      set arp-reply enable
      set arp-intf ''
      set comments ''
   end
```

For FortiOS 7.4.x the Fixed port range is 5117 and end port is 65533 and these values cannot be changed . FortiOS 7.6.x supports configuring the Fixed port range. Also, fixed port range for NAT64 is available in FortiOS 7.6.x.

```
diagnose firewall ippool-fixed-range list natip 20.20.1.0
ippool name=Fixed_Port_Range, ip shared num=2, port num=30208
internal ip=192.168.1.1, nat ip=20.20.1.0, range=5117~35324
internal ip=192.168.1.2, nat ip=20.20.1.0, range=35325~65532
```

Fixed Port Range supports oversubscription, which means that the public IP address and NAT port can be reused by FortiOS so far the new session (initiated by the client) does not create clash. The current FortiOS overload behavior does not wait until all ports are used to decide which port to overload. The port reuse/overloading is different to other vendors. As long as the new connections do not create clashes (the tuples are different), the port reuse ensures that the users are not tightly restricted by IP pool size and can generate more sessions than configured in set `port-per-user`.

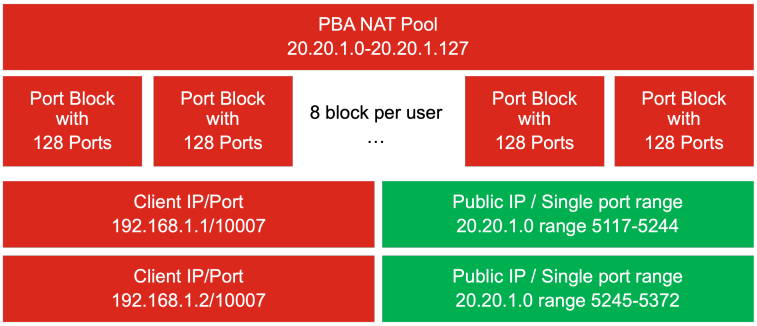
For port selection FortiOS will try to reuse the original SRC port, however if client source port does not fall in the “calculated” port range, or if the new tuples create a session that clashes, FortiOS will test ports iterating through, starting at a random point in the calculated port block. The port will be selected if it does not create a session that clashes (so overload could occur).

```
diagnose sys session list | grep -f '20.20.1.0'
hook=post dir=org act=snat 192.168.1.1:10029->118.1.1.100:80(20.20.1.0:10029)
hook=post dir=org act=snat 192.168.1.2:12348->118.1.1.100:80(20.20.1.0:35325)
```

In case FortOS has iterated over the ports in the calculated port-block without finding a port that doesn’t clash it will stop translating and will create log entry indicating the clash. In FortiOS 7.4 the port randomness is not ideal, because after the entry port is calculated, the next session will use the next iterated port. In FortiOS 7.6 the randomness can be configured.

Port Block Allocation

Port Block Allocation uses ports in port-blocks for translation. The allocation is on demand and a port-block is dynamically allocated to each client. The client sessions are translated using the provided resource IP address and ports in the allocated block. Once the block is exhausted, another block is provided (and so on) until the maximum configured blocks is reached. It is worth to mention that ports can be re-used/overloaded with PBA in kernel NAT. The behavior is the same as the mentioned above (`fixed-port-range`). FortiOS will try to reuse the the original SRC port, however if it does not fall in the port blocks range or if it creates a session that clashes, FortiOS will test ports iterating through starting at a random point in the port block until it finds non-clashing combination. If no available combinations are found, FortiOS will stop translating and will create log entry indicating the clash. If the new session is not clashing the resource (Src_Port, Src_NAT_IP) can be oversubscribed/overloaded.



In FortiOS 7.4 the port randomness is not ideal, because after the entry port in the port block is selected, the next session will use the next iterated port in that block. In FortiOS 7.6 the randomness can be configured.

```
config firewall ippool
   edit "PBA"
      set type port-block-allocation
      set startip 20.20.1.0
      set endip 20.20.1.127
      set block-size 128
      set num-blocks-per-user 8
      set pba-timeout 50
      set permit-any-host enable
      set arp-reply enable
      set arp-intf ''
      set comments ''
   end
```

For kernel NAT, the PBA start port is 5117 and end port is 65533. These values cannot be changed in FortiOS 7.4. Start/End port is configurable in FortiOS 7.6.

```
diagnose firewall ippool list pba
user 192.168.1.1, 20.20.1.0, 5117-5244, idx=0, use=6
user 192.168.1.2, 20.20.1.0, 5245-5372, idx=1, use=3
…
diagnose sys session list | grep hook=pre
hook=pre dir=reply act=dnat 118.1.1.100:80->20.20.1.0:5143(192.168.1.1:10007)
hook=pre dir=reply act=dnat 118.1.1.100:80->20.20.1.0:5139(192.168.1.1:10003)
..
hook=pre dir=reply act=dnat 118.1.1.100:80->20.20.1.0:5271(192.168.1.2:10007)
hook=pre dir=reply act=dnat 118.1.1.100:80->20.20.1.0:5268(192.168.1.2:10004)
…
```

Also Endpoint Independent Filtering (EIF) is available for PBA nat pools by configuring set `permit-any-host enable` in FortiOS 7.4. Fixed port range supports EIF in FortiOS 7.6.

NAT Pooling/soft-APP

Mainstream FortiOS CGNAT supports the use of multiple different NAT pool types in the policy. You can configure multiple NAT pools in the policy to scale the NAT resources, which means that once the first in order pool is exhausted, the next pool in order will start providing resources. The pooling is not about IP pool grouping, it is a policy with multiple CGNAT pools pools configured:

```
config firewall policy
   edit 5
      ...
      set nat enable
      set ippool enable
      set poolname "Overload" "PBA"
   end
```

The pool order in the policy is important. FortiOS will start translating with resources from the first NAT pool in order (in the policy) and once this pool is exhausted, the system will start providing resources from the next pool in order in the policy.

The NAT pooling functionality is similar to soft-APP in RFC7857 Chapter 4 with the caveat of NAT pooling, providing resources from two different NAT pools.

Endpoint Independent Mapping

EIM provides the same resource/mapping (SRC_NAT_IP/SRC__Port) when the source/private IP and source port for the new session are the same, under the assumption that the new session/tuples are not clashing. For kernel CGNAT, EIM is the default setting for all native FortiOS NAT pools. The overload behavior for PBA can be disabled or enabled:

```
config system npu
set pba-eim ?
disallow Disallow PBA(non-overload)/EIM combination in SNAT policy.
allow Allow PBA(non-overload)/EIM combination in SNAT policy.
```

Endpoint Independent Filtering

Endpoint Independent Filtering filters traffic originating from the internet towards internal hosts through already opened mapping. When traffic is supposed to reach a private host (IP/port), EIF will re-use the existing mapping (external NAT_ IP/and NAT_Port), irrespective of the external IP/host that is originating the traffic.

The native FortiOS CGNAT device deploying EIF will accept incoming **UDP traffic only** to a mapped public port from ANY external endpoint on the public network. EIF on native FortiOS is available in PBA CGNAT IP pools only.

```
config firewall ippool
   edit "PBA"
      set type port-block-allocation
      set startip 150.0.0.101
      set endip 150.0.0.200
      set block-size 4096
      set num-blocks-per-user 8
      set pba-timeout 30
```

```
    set permit-any-host enable <-- Enables EIF
    set arp-reply enable
    set arp-intf ''
    set comments ''
end
```

Enabling EIF decreases the session capacity because for every outgoing session established by policy with EIF, two expectation sessions are created to enable the incoming sessions (one from the server, which can “tickle” the connection and another, which will be used by any host on Internet to connect to the client).

EIF does not work with Overload hyperscale type of pools (Overload PBA, Overload SPA), because overloading the NAT ports makes the mapping of those ports for (returning traffic) with EIF not possible.

Generally, EIF is not recommended to be deployed by service providers for security reasons (the client is unprotected from the internet and any vulnerable application could be exploited).

```
diagnose sys session full-stat
session table: table_size=33554432 max_depth=9 used=4951223
misc info: session_count=3653362 setup_rate=34607 exp_count=7305862 reflect_count=0
clash=0 <-- Note the amount of exp sessions
memory_tension_drop=0 ephemeral=0/8469728 removable=0 extreme_low_mem=0
npu_session_count=290924
nturbo_session_count=0
delete=0, flush=13, dev_down=119/5167
session walkers: active=0, vf-97, dev-0, saddr-0, npu-0, wildcard-119
TCP sessions:
17 in ESTABLISHED state
firewall error stat:
…
```

Port Control Protocol (PCP) NAT

Port Control Protocol NAT allows an IPv4 client to manage specific NAT translations via a request/response mechanism using client/server architecture as specified in RFC6887.

The client/CPE can obtain resources (public IP and port/ports) for specific incoming/outgoing NAT translations and can control how the IPv4 packets are translated and forwarded by the CGNAT translator.

The FortiGate acts as a PCP server and CGNAT translator at the same time by managing the requests and responds to the PCP client and dynamically/statically translating the incoming/outgoing traffic.

FortiOS 7.4 supports PCP server, listening on IPv4 address and providing resources to IPv4 clients via PCP ver. 2 based on a request/response mechanism where the PCP identifies the type of the mapping in the OpCode (MAP or PEER) in the payload.

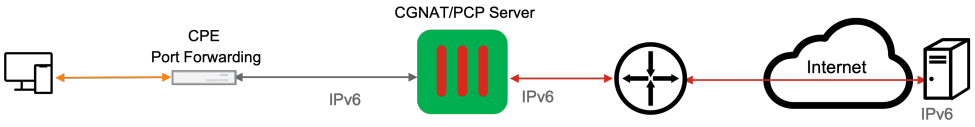
Note that PCP server reliability is not required due to the use of nonces. The PCP client is responsible for delivering the PCP request message. If the PCP client fails to receive an expected response from the server, the client must retransmit the message using the same nonce value. Applying this mechanism eliminates the need of protocol reliability and at the same time minimizes overload situations with the PCP server.

The MAP OpCode is used to create/renew dynamic mapping for inbound mapping to specific internal host (IP address and port). It allows an internal host to be reachable and receive inbound connections from any external/internet host. Once the resource is provided to the PCP Client, FortiOS translates the inbound session and creates a pinhole to allow communication towards the internal host.

The PEER OpCode is used to create/renew dynamic mapping for outbound mapping to a remote peer (IP address and port) and is used for client-initiated communications.

A PCP OpCode can be extended with one or more options. Options can be used in requests and responses. Different options are valid for different OpCodes. For example:

- The THIRD_PARTY option is valid for both MAP and PEER OpCodes. This option allows a third party, which doesn't support PCP, to request dynamic mapping via the PCP client. In this case the MAP request from the PCP client signifies that the mapping request is not for the own source IP address but the IP address of a third party.
- The FILTER option is valid only for the MAP OpCode (for the PEER OpCode it would have no meaning). The FILTER option indicates the permitted remote peer's source IP address and source port for packets coming from the internet and other traffic from other addresses should be blocked.
- The PREFER_FAILURE option is valid only for the MAP OpCode (for the PEER OpCode, similar semantics are automatically implied). This option is used for interworking with UPnP IGDv1 and indicates that if the PCP server is unable to map both the suggested external port and suggested external address, the PCP server should not create a mapping. This option will be deprecated in the future as more clients adopt PCP natively.



The PCP server must be enabled with a pool:

```
config system pcp-server
   set status enable
end
```

The `pcp-server` configuration should be enabled and configured with PCP pool definition for PCP operations:

```
config system pcp-server
   set status enable
      config pools
         edit "pcp-pool"
            set description "pcp_server"
            set id 1
            set client-subnet "192.168.1.0/24"
            set ext-intf "wan1"
            set arp-reply enable
            set extip 194.164.209.0-194.164.209.255
            set extport 3333
            set minimal-lifetime 120
            set maximal-lifetime 86400
            set client-mapping-limit 200
            set mapping-filter-limit 1
            set allow-opcode {map | peer | announce}
            set third-party disallow
            set multicast-announcement disable
            set intl-intf "internal"
            set recycle-delay 0
         end
```

`client-subnet` the IP address with subnet from which PCP requests are accepted.

`ext-intf` the external interface name.

`extip` the IP address or address range on the external interface to map to an address on the internal network.

`extport` the incoming port number or port range to map to a port number on the internal network.

`minimal-lifetime` the minimal lifetime of a PCP mapping, in seconds (60 - 300, default = 120).

`maximal-lifetime` the maximal lifetime of a PCP mapping, in seconds (3600 - 604800, default = 86400).

`client-mapping-limit` mapping limit per client (0 - 65535, default = 0, 0 = unlimited).

`client-subnet` the client subnet address.

`mapping-filter-limit` filter limit per mapping (0 - 5, default = 1).

`allow-opcode` defines the OppCodes map:

- `allow` MAP OpCode,
- `peer` allow PEER OpCode
- `announce` allow ANNOUNCE OpCode.

`third-party` allow/disallow the third-party option.

`third-party-subnet` the third party subnet when `third-party` is configured.

`multicast-announcement` enable/disable multicast announcements.

`announcement-count` set the number of multicast announcements (3 - 10, default = 3).

`intl-intf` the internal interface name.

`recycle-delay` the minimum delay the PCP server will wait before recycling mappings that have expired, in seconds (0 - 3600, default = 0).

The corresponding firewall policy, which uses specific PCP NAT direction must reference the `pcp-pool`:

```
config firewall policy
   edit 10
      set name "Overload_SNAT_192_168_1_0/24_PCP"
      set srcintf "internal"
      set dstintf "wan"
      set action accept
      set srcaddr "Net_192_168_1_0/24"
      set dstaddr "all"
      set schedule "always"
      set service "ALL"
      set logtraffic all
      set nat enable
      set pcp-outbound enable <-- Enables outbound PCP
      set pcp-poolname "pcp-pool" <-- Resources from this PCP pool will be used for outbound
           SNAT for pcp-signaled sessions
      set port-preserve disable
      set ippool enable
      set poolname "Overload" <-- Resources from this pool will be used for regular/outbound
           SNAT (non-pcp)
      set session-ttl 300
   end
```

The PCP direction depends how the policy is constructed.

If the `srcintf`` is the external interface, the possible direction configuration is `pcp-inbound`.

If the `srcintf` is a non-external interface, the possible direction configuration is `pcp-outbound`.

The policy must also completed with `pcp-pool` specifying the resources for `pcp-signaled` mappings.

Note that the FortiOS PCP implementation does not specify the PCP OpCode, it is rather working with the concept of direction, which is easier to understand from the operations perspective.

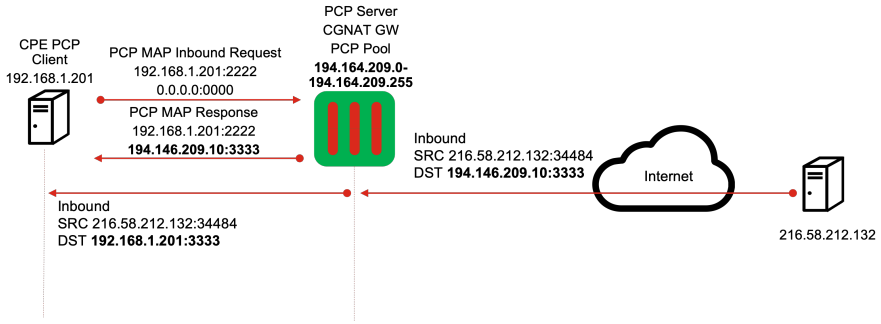
In the example above, when the PCP client sends a PCP outbound request, FortiOS will provide resources back to the client using the `pcp-pool` and will create new mapping for outbound traffic based on the PCP signalling.

If other sessions (that were not signaled by the PCP client via PCP) need to reach the internet, FortiOS uses the Overload pool and will create outbound source NAT mapping (private IP: source port to public IP:source port).

For inbound traffic another policy is required, which allows `pcp-inbound` with `pcp-pool`, used to allow inbound sessions towards the internal host/network. In this case:

```
config firewall policy
   edit 30
      set name "PCP_Inbound_to_192_168_1_201"
      set srcintf "wan1"
      set dstintf "internal"
      set action accept
      set srcaddr "all"
      set dstaddr "Host_192_168_1_202"
      set schedule "always"
      set service "ALL"
      set logtraffic all
      set pcp-inbound enable <-- Enables inbound PCP
      set pcp-poolname "pcp-pool" <-- Resources from the PCP pool for inbound PCP
   end
```

Due to existing `pcp-inbound` mapping, the incoming/inbound session towards the internal client/network will be allowed.



In this case, FortiOS uses dynamic resources for the PCP inbound mapping with IP address 194.146.209.100 and port 3333 from the `pcp-pool` mentioned in the PCP sever configuration.

Session Timers

FortiOS session timers in native FortiOS are available per VDOM under `config system session-ttl`. Also relevant for CGNAT, `refresh-direction` is supported by FortiOS 7.4.

The default timeout is 3600 seconds, however custom ports and related timeouts can be added:

```
config system session-ttl
   set default 3600
      config port
         edit 2123 <-- custom port
            set protocol 17 <-- proto
            set timeout 259200 <-- related timeout
            set start-port 2123
            set end-port 2123
         end
```

Session timeout can be configured under a service and applied in policy. These settings will take precedence over the VDOM config:

```
config firewall service custom
   edit "HTTP"
      ...
      set tcp-halfclose-timer 0
      set tcp-halfopen-timer 0
      set tcp-timewait-timer 0
      set tcp-rst-timer 0
      set udp-idle-timer 0
      set session-ttl 0
   end
      set schedule "always"
      set service "ALL"
      set nat enable
      set ippool enable
      set poolname "Deterministic"
      set per-ip-shaper "1k"
   next
```

Session Limits

For FortiOS (kernel CGNAT), the session limit can be configured alternatively by using per-ip-shaper.

```
config firewall shaper per-ip-shaper
   edit "1k"
      set max-concurrent-session 1000
      set max-concurrent-tcp-session 500
      set max-concurrent-udp-session 500
   end
```

In the following example the per-ip-shaper limiting max-concurrent sessions to 1000 is attached in the firewall policy, which is using deterministic NAT pool, based on fixed pool allocation with kernel CGNAT.

```
config firewall policy
   edit 1
      set status enable
```

```
      set name "cgn"
      set srcintf "any"
      set dstintf "any"
      set action accept
      set srcaddr "all"
      set dstaddr "all"
      set schedule "always"
      set service "ALL"
      set nat enable
      set ippool enable
      set poolname "Deterministic"
      set per-ip-shaper "1k"
   next
```

ALG/Session Helper Support

As already mentioned in the Hyperscale chapter, NAT causes problems with certain applications which carry IP addresses/ports in the payload. FortiOS provides resolution for this problem (for kernel NAT) by deploying session helpers/ALG which facilitate the network address translation by substituting the correct values for NAT source IP and source port in the protocol stack. There are number of session helpers available off-the-shelf in FortiOS:

```
config system session-helper
   edit 1
      set name pptp
      set protocol 6
      set port 1723
   next
   edit 2
      set name h323
      set protocol 6
      set port 1720
   next
   edit 3
      set name ras
      set protocol 17
      set port 1719
   next
   edit 4
      set name tns
      set protocol 6
      set port 1521
   next
   edit 5
      set name tftp
      set protocol 17
      set port 69
   next
   edit 6
      set name rtsp
      set protocol 6
      set port 554
   next
   edit 7
      set name rtsp
```

```
      set protocol 6
      set port 7070
    next
    edit 8
      set name rtsp
      set protocol 6
      set port 8554
    next
    edit 9
    set name ftp
    set protocol 6
    set port 21
    next
    edit 10
      set name mms
      set protocol 6
      set port 1863
    next
    edit 11
      set name pmap
      set protocol 6
      set port 111
    next
    edit 12
      set name pmap
      set protocol 17
      set port 111
    next
    edit 13
      set name sip
      set protocol 17
      set port 5060
    next
    edit 15
      set name rsh
      set protocol 6
      set port 514
    next
    edit 16
      set name rsh
      set protocol 6
      set port 512
    next
    edit 17
      set name dcerpc
      set protocol 6
      set port 135
    next
    edit 18
      set name dcerpc
      set protocol 17
      set port 135
    next
    edit 19
      set name mgcp
      set protocol 17
      set port 2427
```

```
   next
   edit 20
      set name mgcp
      set protocol 17
      set port 2727
   next
   edit 14
      set name dns-udp
      set protocol 17
      set port 53
   next
   end
```

When the FortiOS Carrier license is enabled two more helpers (PCP and GTP) are made available on top of the existing, listed above.

In FortiOS 7.0 flow-based SIP inspection was introduced, which is handled by the IPS Engine. When a VoIP profile is applied to a firewall policy, the inspection mode determines whether SIP ALG or flow based SIP is used.

Proxy-based SIP ALG is able to handle features such as pin-hole creation and NAT that flow-based SIP inspection cannot. On another hand flow-based SIP can handle features such as MSRP decoding and scanning that proxy-based SIP ALG cannot.

SIP message inspection and filtering was introduced in FortiOS 7.4. This features changes SIP message inspection and filtering behavior. The feature introduces a new IPS-based VoIP profile (ipsvoip-filter) that allows flow-based SIP to complement SIP ALG while working together.

```
config firewall policy
   edit <id>
      set ips-voip-filter <name>
   end
```

The `voip-profile` in the firewall policy can be selected regardless of the inspection-mode in the policy. Both options (`ips-flow` or `voipd-proxy` based inspection) are added now in the SIP configuration and previously these were part of the VoIP profile.

```
config voip profile
   edit <name>
      set feature-set {ips | voipd}
         config sip
            set call-id-regex <string>
            set call-id-regex <string>
         end
```

end

Two voip profiles (one with ips and another with voipd with the same functionality) shall configured:

```
config voip profile
   edit "voip_sip_alg"
      set feature-set voipd
      set comment "sip_alg_simple"
         config sip
            set log-violations enable
            set log-call-summary enable
         end
      next
      edit "voip_sip_ips"
         set feature-set ips
         set comment "ips_voip_blocking"
```

```
        config sip
           set block-invite enable
           set log-violations enable
        end
```

And used FortiOS 7.4 in the policy:

```
config firewall policy
   edit 1
      set srcintf "port1"
      set dstintf "port9"
      set action accept
      set srcaddr "all"
      set dstaddr "all"
      set srcaddr6 "all"
      set dstaddr6 "all"
      set schedule "always"
      set service "ALL"
      set utm-status enable
      set inspection-mode proxy
      set ips-sensor "g-default"
      set voip-profile "voip_sip_alg"
      set ips-voip-filter "voip_sip_ips"
      set logtraffic all
      set nat enable
   next
```

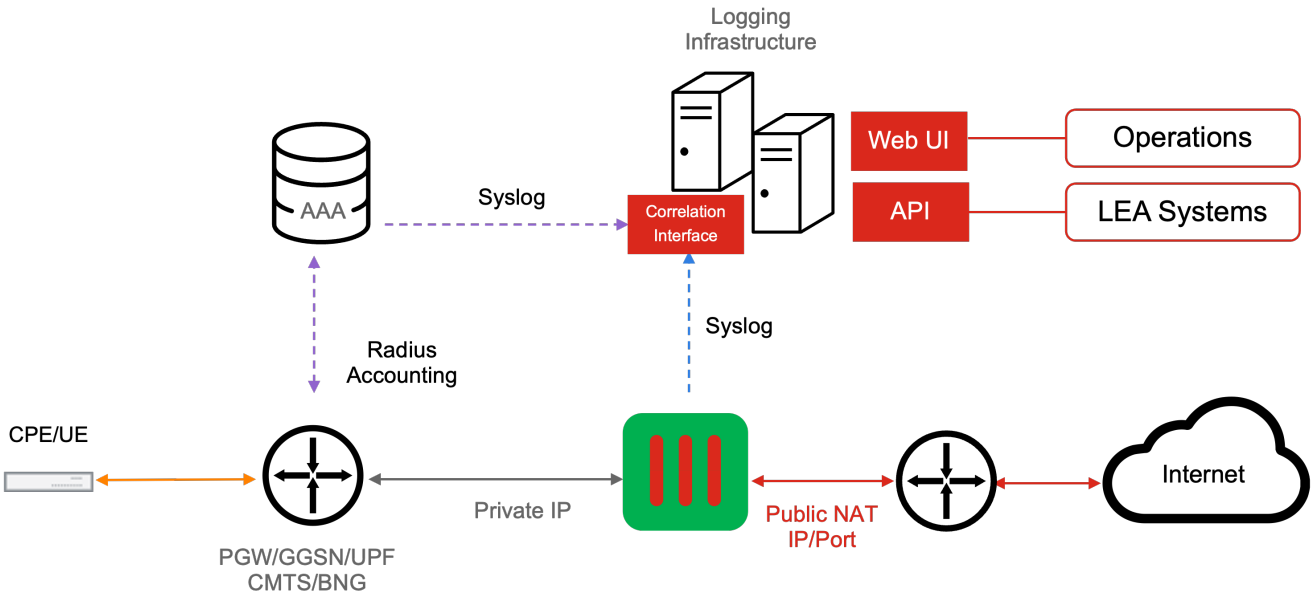
CGNAT Logging

CGNAT logging is very a important feature, guided by local legal and regulatory requirements to provide information about “private to public” mapping of IP addresses. Generally log processing is quite challenging and expensive task, especially when logs from multiple sources must be correlated to get the private/public IP address and the subscriber identity for legislative purposes. The main benefit of the hyperscale CGNAT features is the accelerated session setup and log generation in hardware. The CGNAT logs generated by the hyperscale systems are potentially at a very high rate, and depending on the CGNAT logging configuration also, high-volume.

Some operators are using syslog and others Netflow based logging infrastructure. The referenced logging architecture in the case of CGNAT is based on syslog, however exactly the same setup is also supported with Netflow as well.

Sometimes the service providers require to provide the information about IP address mapping along with the subscriber related information (User name, IMSI/MSISDN, sometimes location and RAT type etc.). Hyperscale CGNAT systems do not support RSSO (which is used to provide logs with correlated IP address mapping and user identity). In this case the external logging system provides the log correlation and reporting functionality.

Recommended Logging and Reporting Architecture



In this architecture the RADIUS server sends the mobile subscriber’s identity information (User name, IMSI/MSISDN, location, RAT Type, etc.) via Syslog (or Netflow) to the logging servers.

The log correlation between the two sources (RADIUS and CGNAT) is performed externally by the logging infrastructure. The related reporting function is provided by the external logging solution.

Fortinet offers logging solutions based on FortiAnalyzer or FortiSIEM either on appliances or virtual machines. FortiAnalyzer can parse Fortinet log file types only and cannot digest third party logs. FortiSIEM is multi-vendor and multi-protocol aware and in the case of external CGNAT and RADIUS logs correlation, FortiSIEM can be proposed as logging solution. This document does not discuss on the details related to logging systems and it is focusing on the FortiGate CGNAT configuration details only.

Hyperscale Logging Configuration

For best logging performance, make sure that you use dedicated system interfaces to send the logs from the system to the external logging servers. On some hyperscale systems you can configure `ports-using-npu` for hardware logging. When you add dedicated interface for hardware logging, the logs will be sent from NPU directly using the configured interface, bypassing the CPU:

```
config system npu
   config port-path-option
      set ports-using-npu <interfaces>
   end
```

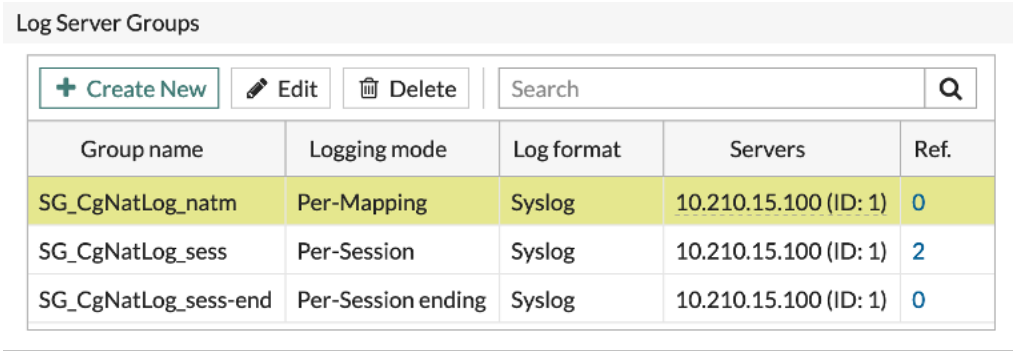
You can use multiple interfaces for hardware logging. Each interface must have an IP address and be able to communicate with your logging servers. You can pickup any of the available physical interfaces and configure them under `set ports-using-npu` as accelerated interfaces for hardware logging. It is recommended that the interfaces used for hardware logging should not be used for any other traffic.

However, it is noteworthy to mention that you can use VLAN interface for logging as well (multiple physical interfaces in link aggregation group). If your FortiGate is configured with multiple VDOMs, the `npu-server` is global configuration. The log servers are shared by all of the NPUs in the system and you can specify the corresponding VDOM for the logging servers:

```
config log npu-server
   set log-processor {hardware | host}
   set netflow-ver {v9 | v10}
      config server-info
         edit <index>
            set vdom <name>
            set ip-family {v4 | v6}
            set ipv4-server <ipv4-address>
            set ipv6-server <ipv6-address>
            set source-port <port-number>
            set dest-port <port-number>
         end
```

The `log-processor` selects whether to use NPU processors (hardware, the default) or the FortiGate CPUs (host) to generate traffic log messages for firewall sessions. Hardware logging is supported for IPv4, IPv6, NAT64, and NAT46. Setting up the `log-processor host` (host logging) can reduce the overall FortiGate performance because the FortiGate CPUs will handle the logging instead the NPU. User and event info is available via host logging as well.
You cannot address single server in the policy, instead you should use server-group. Logging servers must be grouped into `server-group`:

```
config server-group
   edit <group-name>
      set log-mode {per-session | per-nat-mapping | per-session-ending}
      set log-format {netflow | syslog}
      set server-number <number>
      set server-start-id <number>
   end
```



The hardware accelerated log format can be either Syslog or NetFlow (IPFix). On hyperscale devices NetFlow v10 (IPFix) and NetFlow v9 logging are supported.

Host logging on FortiOS 7.2 and older releases does not support Netflow v9. NetFlow v9 for session logging in hyperscale VDOMs has been added in FortiOS 7.4.2.

FortiOS 7.4.2 supports TCP log transmission. This is a significant improvement from the older versions, which only supported UDP. TCP provides stateful connection, ensuring no logs are lost during transmission.

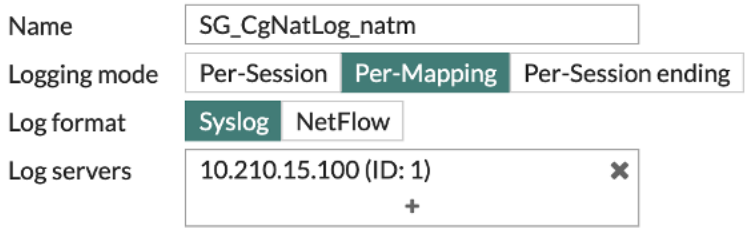
```
config server-info

set log-transport
udp Use UDP for log transport.
tcp Use TCP for log transport.

config server-info
   edit 1
      set vdom "CGN-hw1"
      set ip-family v6
      set log-transport tcp
      set ipv6-server <ipv6-address>
      set dest-port <server port>
   next
   edit 2
      set vdom "CGN-hw1"
      set log-transport tcp
      set ipv4-server <ipv4-address>
      set dest-port <server port>
   next
   end
```

The TCP Transport protocol option is not available for servers used in a group which log format is Netflow.

There are different options related how the log can be generated:

- **Per session** (two logs with start and end),
- **Per mapping** (two logs, with NAT mapping allocation and release),
- **Per session ending** (single log, when the session ends).

FortiOS supports sending the same (per-session, persession-ending, per-nat-mapping) logs (IPV4, NAT44, IPv6, NAT64) to at least two unicast IPv4 or IPv6 syslog servers at same time. For NAT44/46 the IPv4 logging server is used and for NAT66/64 the IPv6 logging server is used:



Once configured, you can then use those log servers or groups in the firewall policy.

```
config firewall policy
   edit 1
      set name "cgn44_pba"
      set srcintf "port1"
      set dstintf "port2"
      set action accept
      set srcaddr "all"
      set dstaddr "all"
      set service "ALL"
      set auto-asic-offload enable
      set np-acceleration enable
      set policy-offload enable
      set cgn-session-quota 16777215
      set cgn-resource-quota 16
      set cgn-eif disable
      set cgn-eim disable
      set cgn-log-server-grp 'SG_CgNatLog_sess' <---
      set nat enable
      set ippool enable
      set poolname "pba"
   end
```

Netflow/IPFix

The NP7 based systems support Netflow/IPFix logging in hardware (NP7 with hyperscale) but also software assisted (log2host) Netflow. NetFlow records are traditionally exported using User Datagram Protocol (UDP) and collected using a NetFlow collector.

The IP address of the NetFlow collector and the destination UDP port must be configured as logging server in the `config log npu-server` and/or `config server-group` sections. The standard value is UDP port 2055, but other values like 9555, 9025, or 9026 can also be used.

Note that the NP7 Netflow format supports both v9 and v10, and compared to Kernel Netflow logging, the supported version is v9. More information about the Kernel FortiOS templates can be found here: NetFlow templates.

Netflow for hyperscale NAT uses the following data template:

```
Cisco NetFlow/IPFIX
    Version: 10
    Length: 104
    Timestamp: Mar 12, 2023 22:17:24.000000000 CET
    FlowSequence: 0
```

```
    Observation Domain Id: 0
    Set 1 [id=2] (Data Template): 280
        FlowSet Id: Data Template (V10 [IPFIX]) (2)
        FlowSet Length: 88
        Template (Id = 280, Count = 20)
            Template Id: 280
            Field Count: 20
            Field (1/20): observationTimeMilliseconds
            Field (2/20): selectorName
            Field (3/20): observationDomainName
            Field (4/20): FLOW_EXPORTER
            Field (5/20): natEvent
            Field (6/20): natType
            Field (7/20): PROTOCOL
            Field (8/20): IP_SRC_ADDR
            Field (9/20): postNATSourceIPv4Address
            Field (10/20): L4_SRC_PORT
            Field (11/20): postNAPTSourceTransportPort
            Field (12/20): IP_DST_ADDR
            Field (13/20): postNATDestinationIPv4Address
            Field (14/20): L4_DST_PORT
            Field (15/20): postNAPTDestinationTransportPort
            Field (16/20): flowDurationMilliseconds
            Field (17/20): PKTS
            Field (18/20): BYTES
            Field (19/20): OUT_PKTS
            Field (20/20): OUT_BYTES
```

When the FortiGate sends NetFlow domain IDs information to the NetFlow server, the information includes the separate domain IDs for the FortiGate CPU and each NP7 processor.

Log messages from the FortiGate CPU and from each NP7 processor contain these domain IDs, allowing the NetFlow server to distinguish between FortiGate CPU traffic and traffic from each NP7 processor.

The format of the NP7 based NetFlow messages is like the following (example of session create `NAT44 session create`):

```
Cisco NetFlow/IPFIX
    Version: 10
    Length: 108
    Timestamp: Mar 12, 2023 22:13:16.000000000 CET
    FlowSequence: 1
    Observation Domain Id: 3
    Set 1 [id=280] (1 flows)
        FlowSet Id: (Data) (280)
        FlowSet Length: 92
        [Template Frame: 22 (received after this frame)]
        Flow 1
            Observation Time Milliseconds: Mar 12, 2023 22:13:17.309000000 CET
            Selector Name: FG420FTK20900039
            Observation Domain Name: CGNAT-hw250
            FlowExporter: 67108865
            Nat Event: NAT44 session create (4)
            Nat Type: 1
            Protocol: UDP (17)
            SrcAddr: 10.212.44.167
            Post NAT Source IPv4 Address: 193.110.55.24
```

```
            SrcPort: 47309 (47309)
            Post NAPT Source Transport Port: 50046
            DstAddr: 173.252.91.4
            Post NAT Destination IPv4 Address: 173.252.91.4
            DstPort: 2152 (2152)
            Post NAPT Destination Transport Port: 2152
            Duration: 0.000000000 seconds
            Packets: 0
            Octets: 0
            Post Packets: 0
            Post Octets: 0
```

And the session delete `NAT44 session delete`:

```
Cisco NetFlow/IPFIX
    Version: 10
    Length: 108
    Timestamp: Mar 12, 2023 22:15:16.000000000 CET
    FlowSequence: 2
    Observation Domain Id: 3
    Set 1 [id=280] (1 flows)
        FlowSet Id: (Data) (280)
        FlowSet Length: 92
        [Template Frame: 22 (received after this frame)]
        Flow 1
            Observation Time Milliseconds: Mar 12, 2023 22:15:16.598000000 CET
            Selector Name: FG420FTK20900039
            Observation Domain Name: CGNAT-hw250
            FlowExporter: 67108865
            Nat Event: NAT44 session delete (5)
            Nat Type: 1
            Protocol: UDP (17)
            SrcAddr: 10.212.44.167
            Post NAT Source IPv4 Address: 193.110.55.24
            SrcPort: 34364 (34364)
            Post NAPT Source Transport Port: 50045
            DstAddr: 173.252.91.4
            Post NAT Destination IPv4 Address: 173.252.91.4
            DstPort: 2152 (2152)
            Post NAPT Destination Transport Port: 2152
            Duration: 182.102000000 seconds
            Packets: 4
            Octets: 208
            Post Packets: 4
            Post Octets: 128
```

NPU Logging Examples

The logging examples use the following configuration for logging servers/groups and CGNAT IP pools:

```
config log npu-server
   set netflow-ver v10
     config server-info
        edit 1
           set vdom "CGNAT"
```

```
            set ip-family v4
            set ipv4-server 10.210.15.100
            set ipv6-server ::
            set source-port 514
            set dest-port 514
         next
      end
      config server-group
         edit "SG_CgNatLog_sess"
            set log-mode per-session
            set log-format syslog
            set server-number 1
            set server-start-id 1
         next
         edit "SG_CgNatLog_natm"
            set log-mode per-nat-mapping
            set log-format syslog
            set server-number 1
            set server-start-id 1
         next
         edit "SG_CgNatLog_sess-end"
            set log-mode per-session-ending
            set log-format syslog
            set server-number 1
            set server-start-id 1
         next
      end
   end
```

You can specify the number of log servers by configuring the `server-number` in the the log server group. The setting `server-start-id` the ID of one of the log servers. You can select the exact log server to add to a log server group by using the `server-number` and `server-start-id`.

For example, if you have used the config server-info command to create five log servers with IDs 1 to 5, you can add the first three of them (IDs 1 to 3) to a log server group by setting server-number to 3 and server-start-id to 1. This adds the log servers with ID 1, 2, and 3 to this log server group. To add the other two servers to a second log server group, set server-number to 2 and server-start-id to 4. This adds log servers 4 and 5 to the second log server group.

The following IP pools have been used for CGNAT (and related logging) in this document.

Fixed Allocation

```
edit "snat44_fa"
   set type cgn-resource-allocation
   set startip 193.110.55.128
   set endip 193.110.55.159
   set cgn-fixedalloc enable
   set cgn-block-size 1984
   set cgn-client-startip "10.212.44.0"
   set cgn-client-endip "10.212.47.255"
   set cgn-port-start 1043
```

Port Block Allocation

```
edit "snat44_pba"
   set type cgn-resource-allocation
   set startip 193.110.55.0
   set endip 193.110.55.31
```

PBA Overload

```
edit "snat44_pba_overload"
   set type cgn-resource-allocation
   set startip 193.110.55.32
   set endip 193.110.55.63
   set cgn-overload enable
```

Single Port Allocation

```
edit "snat44_spa"
   set type cgn-resource-allocation
   set startip 193.110.55.64
   set endip 193.110.55.95
   set cgn-spa enable
```

Overload Single Port Allocation

```
edit "snat44_spa_overload"
   set type cgn-resource-allocation
   set startip 193.110.55.96
   set endip 193.110.55.127
   set cgn-spa enable
   set cgn-overload enable
```

Port Block Allocation logging examples

The following PBA logging examples show "per-session", "per-session-ending" and “per-nat-mapping” logging modes. The logging mode is tied to the log server group definition.

Per-session example log messages

```
Sep 1 17:01:58 date=2022-09-01 time=16:01:57 sn=FG421FTK20900036 vd=CGNAT pid=1
type=sess act=start tran=snat proto=6 ipold=v4 ipnew=v4 sip=10.212.44.167
dip=173.252.91.4 sport=54405 dport=443 nsip=193.110.55.0 ndip=173.252.91.4 nsport=39461
ndport=443 sentp=0 sentb=0 rcvdp=0 rcvdb=0
```

```
Sep 1 17:03:10 date=2022-09-01 time=16:03:09 sn=FG421FTK20900036 vd=CGNAT pid=1
type=sess act=end tran=snat proto=6 ipold=v4 ipnew=v4 sip=10.212.44.167
dip=173.252.91.4 sport=54405 dport=443 nsip=193.110.55.0 ndip=173.252.91.4 nsport=39461
ndport=443 dur=71 sentp=814822 sentb=1222227253 rcvdp=409842 rcvdb=21366456
```

Two log entries are sent per firewall/NAT session to show the start and end of the session. Note that the information about the session is contained.

Per-session-ending example log message

```
Sep 1 17:10:01 date=2022-09-01 time=16:10:01 sn=FG421FTK20900036 vd=CGNAT pid=67108865
type=sess act=end tran=snat proto=6 ipold=v4 ipnew=v4 sip=10.212.44.167
dip=173.252.91.4 sport=59061 dport=443 nsip=193.110.55.2 ndip=173.252.91.4 nsport=36477
ndport=443 dur=12 sentp=17 sentb=1191 rcvdp=14 rcvdb=946
```

One log entry per firewall/NAT session at the end of the session. The duration field can be used to calculate the session start time.

Per-nat-mapping example log messages

```
Sep 2 15:10:31 date=2022-09-02 time=14:10:32 sn=FG421FTK20900036 vd=CGNAT pid=1
type=natm act=start mode=pbm proto=6 ipold=v4 ipnew=v4 sip=10.212.44.167
mip=193.110.55.7 mpbase=35069 mpbid=83 mpbsz=2
```

```
Sep 2 15:55:43 date=2022-09-02 time=14:55:42 sn=FG421FTK20900036 vd=CGNAT pid=1
type=natm act=end mode=pbm proto=6 ipold=v4 ipnew=v4 sip=10.212.44.167 mip=193.110.55.7
mpbase=35069 mpbid=83 mpbsz=2
```

The per NAT mapping setting takes advantage of only sending two log entries per PBA block used. Therefore, the information in the per-session logs cannot be reproduced as there is no log recording of each session, for example. The `mode` parameter is used to indicate which NAT type is in use; PBA in this case. Later, you will note this differs for SPA. So, from this log we can determine that sessions from original source IP 10.212.44.167 were translated to source 193.110.55.7 using ports 45693 to 45820.

- `mpbsz` is the cgn-block-size in units of 64, thus in this example 128 consecutive ports (the default value) are allocated for use with source IP 10.212.44.167
- `mpbid` is the CGN block identifier; it points to the first port, 83 x 128 = 10624
- `mpbase` is the port offset/base 35069 + 10624 = 45693

Matching the session as below:

```
session info: proto=6 proto_state=11 duration=364 expire=235 timeout=600 flags=00000000
sockflag=00000000 sockport=0 av_idx=0 use=1
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=255/255
state=new f18
statistic(bytes/packets/allow_err): org=524/8/0 reply=376/7/0 tuples=2
tx speed(Bps/kbps): 1/0 rx speed(Bps/kbps): 1/0
orgin->sink: org pre->post, reply pre->post dev=47->49/49->47
gwy=192.168.101.1/192.168.100.1
hook=post dir=org act=snat 10.212.44.167:40310->52.52.208.2:443
(193.110.55.7:45693)           <----
hook=pre dir=reply act=dnat 52.52.208.2:443->193.110.55.7:45693
(10.212.44.167:40310)
pos/(before,after) 0/(0,0), 0/(0,0)
misc=0 policy_id=1 auth_info=0 chk_client_info=0 vd=1
serial=00000bec tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id = 00000000 ngfwid=n/a
dd_type=0 dd_mode=0
  setup by offloaded-policy: origin=native
  O: npid=2/1, in: OID=120/VID=0, out: NHI=122/VID=0
  R: npid=1/2, in: OID=122/VID=0, out: NHI=120/VID=0
```

Overload (Port Block Allocation) logging examples

Logging is the same as PBA as this is a change only in how NAT resources are re-used.

Per-session example log messages

```
Sep 3 10:04:05 date=2022-09-03 time=09:04:05 sn=FG421FTK20900036 vd=CGNAT pid=67108865
type=sess act=start tran=snat proto=6 ipold=v4 ipnew=v4 sip=10.212.44.167
dip=172.217.21.3 sport=40642 dport=443 nsip=193.110.55.32 ndip=172.217.21.3
nsport=46592 ndport=443 sentp=0 sentb=0 rcvdp=0 rcvdb=0
```

```
Sep 3 10:05:35 date=2022-09-03 time=09:05:35 sn=FG421FTK20900036 vd=CGNAT pid=67108865
type=sess act=end tran=snat proto=6 ipold=v4 ipnew=v4 sip=10.212.44.167
dip=172.217.21.3 sport=40642 dport=443 nsip=193.110.55.32 ndip=172.217.21.3
nsport=46592 ndport=443 dur=89 sentp=1190943 sentb=1786410201 rcvdp=595807
rcvdb=31062892
```

Per-session-ending example log message

```
Sep 3 10:10:59 date=2022-09-03 time=09:10:59 sn=FG421FTK20900036 vd=CGNAT pid=1
type=sess act=end tran=snat proto=6 ipold=v4 ipnew=v4 sip=10.212.44.167
dip=172.217.21.3 sport=40650 dport=443 nsip=193.110.55.34 ndip=172.217.21.3
nsport=48386 ndport=443 dur=63 sentp=16 sentb=1139 rcvdp=14 rcvdb=946
```

Per-nat-mapping example log messages

```
Sep 3 10:13:21 date=2022-09-03 time=09:13:21 sn=FG421FTK20900036 vd=CGNAT pid=67108865
type=natm act=start mode=pbm proto=6 ipold=v4 ipnew=v4 sip=10.212.44.167
mip=193.110.55.36 mpbase=35069 mpbid=56 mpbsz=2
```

```
Sep 3 10:15:43 date=2022-09-03 time=09:15:44 sn=FG421FTK20900036 vd=CGNAT pid=67108865
type=natm act=end mode=pbm proto=6 ipold=v4 ipnew=v4 sip=10.212.44.167
mip=193.110.55.36 mpbase=35069 mpbid=56 mpbsz=2
```

Single Port Allocation

The following Single Port Allocation logging examples show "per-session", "per-session-ending" and “per-nat-mapping” logging modes. The logging mode is tied to the log server group definition.

Per-session example log messages

```
Sep 3 10:35:01 date=2022-09-03 time=09:35:02 sn=FG421FTK20900036 vd=CGNAT pid=1
type=sess act=start tran=snat proto=6 ipold=v4 ipnew=v4 sip=10.212.44.167
dip=52.52.208.2 sport=40466 dport=443 nsip=193.110.55.68 ndip=52.52.208.2 nsport=45275
ndport=443 sentp=0 sentb=0 rcvdp=0 rcvdb=0
```

```
Sep 3 10:36:04 date=2022-09-03 time=09:36:05 sn=FG421FTK20900036 vd=CGNAT pid=1
type=sess act=end tran=snat proto=6 ipold=v4 ipnew=v4 sip=10.212.44.167 dip=52.52.208.2
sport=40466 dport=443 nsip=193.110.55.68 ndip=52.52.208.2 nsport=45275 ndport=443
dur=62 sentp=16 sentb=1141 rcvdp=14 rcvdb=946
```

Per-session-ending example log message

```
Sep 3 10:40:13 date=2022-09-03 time=09:40:15 sn=FG421FTK20900036 vd=CGNAT pid=67108865
type=sess act=end tran=snat proto=6 ipold=v4 ipnew=v4 sip=10.212.44.167
dip=173.252.91.4 sport=46614 dport=443 nsip=193.110.55.70 ndip=173.252.91.4
nsport=37245 ndport=443 dur=62 sentp=16 sentb=1138 rcvdp=14 rcvdb=944
```

Per-nat-mapping example log messages

```
Sep 3 10:43:14 date=2022-09-03 time=09:43:15 sn=FG421FTK20900036 vd=CGNAT pid=1
type=natm act=start mode=spm proto=6 ipold=v4 ipnew=v4 sip=10.212.44.167
```

```
mip=193.110.55.73 sport=40504 mport=48101

Sep 3 10:45:17 date=2022-09-03 time=09:45:19 sn=FG421FTK20900036 vd=CGNAT pid=1
type=natm act=end mode=spm proto=6 ipold=v4 ipnew=v4 sip=10.212.44.167
mip=193.110.55.73 sport=40504 mport=48101
```

The `mode` parameter here shows SPA is in use and as such the fields are a little different to the PBA examples previously.

Overload (Single Port Allocation)

Logging should be the same as SPA as this is a change only in how NAT resources are re-used.

Per-session example log messages

```
Sep 3 10:52:20 date=2022-09-03 time=09:52:22 sn=FG421FTK20900036 vd=CGNAT pid=67108865
type=sess act=start tran=snat proto=6 ipold=v4 ipnew=v4 sip=10.212.44.167
dip=173.252.91.4 sport=46650 dport=443 nsip=193.110.55.97 ndip=173.252.91.4
nsport=46650 ndport=443 sentp=0 sentb=0 rcvdp=0 rcvdb=0

Sep 3 10:54:38 date=2022-09-03 time=09:54:39 sn=FG421FTK20900036 vd=CGNAT pid=67108865
type=sess act=end tran=snat proto=6 ipold=v4 ipnew=v4 sip=10.212.44.167
dip=173.252.91.4 sport=46650 dport=443 nsip=193.110.55.97 ndip=173.252.91.4
nsport=46650 ndport=443 dur=136 sentp=6162982 sentb=9244464549 rcvdp=3087475
rcvdb=160723648
```

Per-session-ending example log message

```
Sep 3 10:59:00 date=2022-09-03 time=09:59:02 sn=FG421FTK20900036 vd=CGNAT pid=1
type=sess act=end tran=snat proto=6 ipold=v4 ipnew=v4 sip=10.212.44.167 dip=52.52.208.2
sport=40524 dport=443 nsip=193.110.55.98 ndip=52.52.208.2 nsport=40524 ndport=443
dur=140 sentp=6486822 sentb=9730228701 rcvdp=3234462 rcvdb=168343612
```

Per-nat-mapping example log messages

```
Sep 3 11:00:22 date=2022-09-03 time=10:00:24 sn=FG421FTK20900036 vd=CGNAT pid=67108865
type=natm act=start mode=spm proto=6 ipold=v4 ipnew=v4 sip=10.212.44.167
mip=193.110.55.99 sport=46664 mport=46664

Sep 3 11:02:42 date=2022-09-03 time=10:02:42 sn=FG421FTK20900036 vd=CGNAT pid=67108865
type=natm act=end mode=spm proto=6 ipold=v4 ipnew=v4 sip=10.212.44.167
mip=193.110.55.99 sport=46664 mport=46664
```

Fixed-allocation

The following Fixed Allocation logging examples show "per-session", "per-session-ending" and “per-nat-mapping” logging modes. The logging mode is tied to the log server group definition. The key benefit of fixed allocation, or deterministic allocation, is that you do not need the logs to map an external address to an internal address and therefore do not need logging. However, you do have the option to log if desired, which can be beneficial for operational troubleshooting for example.

Per-session example log messages

```
Sep 3 13:32:51 date=2022-09-03 time=12:32:51 sn=FG421FTK20900036 vd=CGNAT pid=67108865
type=sess act=start tran=snat proto=6 ipold=v4 ipnew=v4 sip=10.212.44.167
dip=173.252.91.4 sport=46676 dport=443 nsip=193.110.55.133 ndip=173.252.91.4
nsport=32905 ndport=443 sentp=0 sentb=0 rcvdp=0 rcvdb=0
```

```
Sep 3 13:35:09 date=2022-09-03 time=12:35:10 sn=FG421FTK20900036 vd=CGNAT pid=67108865
type=sess act=end tran=snat proto=6 ipold=v4 ipnew=v4 sip=10.212.44.167
dip=173.252.91.4 sport=46676 dport=443 nsip=193.110.55.133 ndip=173.252.91.4
nsport=32905 ndport=443 dur=137 sentp=6162828 sentb=9244233845 rcvdp=3083791
rcvdb=16054635
```

Per-session-ending example log message

```
Sep Sep 3 13:50:32 date=2022-09-03 time=12:50:33 sn=FG421FTK20900036 vd=CGNAT pid=1
type=sess act=end tran=snat proto=6 ipold=v4 ipnew=v4 sip=10.212.44.167 dip=52.52.208.2
sport=40546 dport=443 nsip=193.110.55.133 ndip=52.52.208.2 nsport=33262 ndport=443
dur=121 sentp=4865964 sentb=7298940253 rcvdp=2437446 rcvdb=126930712
```

Per-nat-mapping example log messages

```
Sep Sep 3 13:56:26 date=2022-09-03 time=12:56:27 sn=FG421FTK20900036 vd=CGNAT
pid=67108865 type=natm act=start mode=pbm proto=6 ipold=v4 ipnew=v4 sip=10.212.44.167
mip=193.110.55.133 mpbase=29715 mpbid=7 mpbsz=7
```

```
Sep 3 13:58:29 date=2022-09-03 time=12:58:31 sn=FG421FTK20900036 vd=CGNAT pid=67108865
type=natm act=end mode=pbm proto=6 ipold=v4 ipnew=v4 sip=10.212.44.167
mip=193.110.55.133 mpbase=29715 mpbid=7 mpbsz=7
```

Endpoint Independent Mapping example log messages

The key thing for EIM is that multiple connections from the same client socket are NAT-ed to the same values:

```
Sep 3 16:38:13 date=2022-09-03 time=15:38:14 sn=FG421FTK20900036 vd=CGNAT pid=1
type=sess act=start tran=snat proto=17 ipold=v4 ipnew=v4 sip=10.212.44.167
dip=173.252.91.4 sport=48944 dport=3478 nsip=193.110.55.115 ndip=173.252.91.4
nsport=48944 ndport=3478 sentp=0 sentb=0 rcvdp=0 rcvdb=0
```

```
Sep 3 16:38:13 date=2022-09-03 time=15:38:14 sn=FG421FTK20900036 vd=CGNAT pid=1
type=sess act=start tran=snat proto=17 ipold=v4 ipnew=v4 sip=10.212.44.167
dip=173.252.91.204 sport=48944 dport=3478 nsip=193.110.55.115 ndip=173.252.91.204
nsport=48944 ndport=3478 sentp=0 sentb=0 rcvdp=0 rcvdb=0

Sep 3 16:42:52 date=2022-09-03 time=15:42:53 sn=FG421FTK20900036 vd=CGNAT pid=1
type=sess act=end tran=snat proto=17 ipold=v4 ipnew=v4 sip=10.212.44.167
dip=173.252.91.204 sport=48944 dport=3478 nsip=193.110.55.115 ndip=173.252.91.204
nsport=48944 ndport=3478 dur=276 sentp=1 sentb=56 rcvdp=1 rcvdb=96

Sep 3 16:42:53 date=2022-09-03 time=15:42:54 sn=FG421FTK20900036 vd=CGNAT pid=1
type=sess act=end tran=snat proto=17 ipold=v4 ipnew=v4 sip=10.212.44.167
dip=173.252.91.4 sport=48944 dport=3478 nsip=193.110.55.115 ndip=173.252.91.4
nsport=48944 ndport=3478 dur=278 sentp=5 sentb=280 rcvdp=1 rcvdb=96
```

EIM ensures that the same external address (193.110.55.115) and port (48944) will be assigned for all connections from a given host (10.212.44.167) if they use the same internal port (48944).

Endpoint Independent Filtering

The important aspect for EIF is the incoming connections, which are reusing the same binding:

```
Sep 16 12:34:16 date=2022-09-16 time=11:34:17 sn=FG421FTK20900036 vd=CGNAT pid=1
type=sess act=start tran=snat proto=17 ipold=v4 ipnew=v4 sip=10.212.44.167
dip=172.217.21.3 sport=43810 dport=443 nsip=193.110.55.64 ndip=172.217.21.3
nsport=47999 ndport=443 sentp=0 sentb=0 rcvdp=0 rcvdb=0

Sep 16 12:35:47 date=2022-09-16 time=11:35:48 sn=FG421FTK20900036 vd=CGNAT pid=1
type=sess act=start tran=dnat proto=17 ipold=v4 ipnew=v4 sip=52.52.208.2
dip=193.110.55.64 sport=35961 dport=47999 nsip=52.52.208.2 ndip=10.212.44.167
nsport=35961 ndport=43810 sentp=0 sentb=0 rcvdp=0 rcvdb=0

Sep 16 12:49:43 date=2022-09-16 time=11:49:44 sn=FG421FTK20900036 vd=CGNAT pid=1
type=sess act=end tran=snat proto=17 ipold=v4 ipnew=v4 sip=10.212.44.167
dip=172.217.21.3 sport=43810 dport=443 nsip=193.110.55.64 ndip=172.217.21.3
nsport=47999 ndport=443 dur=918 sentp=9046 sentb=74349932 rcvdp=1 rcvdb=32

Sep 16 12:57:36 date=2022-09-16 time=11:57:37 sn=FG421FTK20900036 vd=CGNAT pid=1
type=sess act=end tran=dnat proto=17 ipold=v4 ipnew=v4 sip=52.52.208.2
dip=193.110.55.64 sport=35961 dport=47999 nsip=52.52.208.2 ndip=10.212.44.167
nsport=35961 ndport=43810 dur=1296 sentp=53510 sentb=80157980 rcvdp=0 rcvdb=0
```

Kernel CGNAT logging

Kernel CGNAT supports remote logging to FortiAnalyzer, syslog v9 and netflow v9 servers. When VDOMs are configured on the FortiGate, multiple FortiAnalyzers and syslog servers can be added globally.

To configure remote logging to FortiAnalyzer:

```
config log fortianalyzer setting
```

```
   set status enable
   set server <server_IP>
   set upload option {store-and-upload | realtime | 1-minute | 5-minute}
end
```

Up to four syslog servers can be configured using the config log syslogd command and can send logs to syslog in CSV and CEF formats. To configure remote logging to a syslog server:

```
config log syslogd setting
   set status enable
   set server <syslog_IP>
   set format {default | cev | cef}
end
```

NetFlow collects IP network traffic statistics via samplers (TX, RX or both) which are configured per interface. Full NetFlow is supported through the information maintained in the firewall session. To configure NetFlow collector:

```
config system netflow
   set collector-ip <ip>
   set collector-port <port>
   set source-ip <ip>
   set active-flow-timeout <integer>
   set inactive-flow-timeout <integer>
   set template-tx-timeout <integer>
   set template-tx-counter <integer>
end
```

If you are in multi VDOM environment you can configure the collector per VDOM:

```
config vdom
   edit <vdom>
      config system vdom-netflow
         set vdom-netflow enable
         set collector-ip <ip>
         set collector-port <port>
         set source-ip <ip>
      end
```

And configure the interface NetFlow sampler:

```
config system interface
   edit <interface>
      set netflow-sampler {disable | tx | rx | both}
   end
```

Supported NetFlow templates

FortiOS supports these NetFlow templates. These templates are fixed and cannot be edited.

Log filters

You can set up log filters to determine which logs are sent to the FortiAnalyzer, FortiManager, and syslog servers. This allows certain logging levels and types of logs to be directed to specific log devices.

Reliable logging to FortiAnalyzer

Reliable logging to FortiAnalyzer prevents lost logs when the connection between FortiOS and FortiAnalyzer is disrupted. When reliable mode is enabled logs are cached in a FortiOS memory queue. FortiOS sends logs to FortiAnalyzer, and FortiAnalyzer uses seq_no to track received logs. After FortiOS sends logs to FortiAnalyzer, logs are moved to a confirm queue in FortiOS. FortiOS periodically queries FortiAnalyzer for the latest seq_no of the last log received, and clears logs from the confirm queue up to the seq_no. If the connection between FortiOS and FortiAnalyzer is disrupted, FortiOS resends the logs in the confirm queue to FortiAnalyzer when the connection is reestablished.

To enable reliable FortiAnalyzer mode:

```
config log fortianalyzer setting
   set reliable enable
end
```

The upload option allows FortiAnalyzer sending options to in different intervals:

```
config log fortianalyzer setting
   set upload-option
```

`store-and-upload` Log to hard disk and then upload to FortiAnalyzer.

`realtime` Log directly to FortiAnalyzer in real time.

`1-minute` Log directly to FortiAnalyzer at least every 1 minute.

`5-minute` Log directly to FortiAnalyzer at least every 5 minutes.

FortiAnalyzer log filters

Sometimes sending all possible logs to FortiAnalyzer would generate too many logs. You can configure log filters for FortiAnalyzer to reduce the amount of logs sent:

```
config log fortianalyzer filter
   set severity <level>
   set forward-traffic {enable | disable}
   set local-traffic {enable | disable}
   set multicast-traffic {enable | disable}
   set sniffer-traffic {enable | disable}
end
```

Syslog log filters

To configure log filters for a syslog server:

```
config log syslogd filter
   set severity information
   set forward-traffic enable
   set local-traffic enable
   set multicast-traffic disable
   set sniffer-traffic disable
      config free-style
         edit 1
            set category event
            set filter "logid 0100022015 0100022016"
            set filter-type include
```

```
        end
```

In the example above only event logs with logid 0100022015 - ippool-create and 0100022016 - ippool-close will be sent to the syslog server:

```
1: date=2024-11-15 time=14:38:35 eventtime=1731677915328417002 tz="+0100"
logid="0100022015" type="event" subtype="system" level="notice" vd="CGNAT"
logdesc="IP pool PBA created" action="ippool-create" saddr="10.63.128.1"
nat=17.0.16.0 portbegin=12029 portend=12156 poolname="pba" msg="IPpool create"

2: date=2024-11-15 time=14:41:51 eventtime=1731678111866768980 tz="+0100"
logid="0100022016" type="event" subtype="system" level="notice" vd="CGNAT"
logdesc="Deallocate IP pool PBA" action="ippool-close" saddr="10.34.0.1"
nat=17.0.16.0 portbegin=8829 portend=8956 poolname="pba" duration=211 msg="IPpool
close"
```

Setting up a backup FortiAnalyzer

FortiOS 7.4.1 supports switching to an alternate FortiAnalyzer if the main FortiAnalyzer is unavailable. Once the connectivity is restored, it will automatically fall back to the primary FortiAnalyzer.

```
config log fortianalyzer setting
   set status enable
   set server "172.16.200.250"
   set alt-server "172.16.200.251"
   set fallback-to-primary enable
   set serial "FAZ-VMTM22000000" "FAZ-VMTM23000003"
end
```

If the primary FortiAnalyzer server 172.16.200.250 goes down, FortiGate will automatically connect to the alternate FortiAnalyzer server 172.16.200.251. To manually switch from the primary to alternate FortiAnalyzer (and vice-versa):

```
execute log {fortianalyzer | fortianalyzer2 | fortianalyzer3} manual-failover
```

If the primary server is still up, the behavior resulting from running this command is based on the `fallback-to-primary` setting configured in the global FortiAnalyzer log settings. If `fallback-to-primary` is enabled (default), running `execute log fortianalyzer manual-failover` will switch to the alternate FortiAnalyzer, but it will switch back to the primary since it is not actually down. If `fallback-to-primary` is disabled, running `execute log fortianalyzer manual-failover` will switch to the alternate FortiAnalyzer, and it will not switch back to the primary.

In FortiOS 7.4.2 logging for long lived sessions has been added. Logging of long-live session statistics can be enabled or disabled in traffic logs.

```
config log setting
   set long-live-session-stat {enable | disable}
end
```

When enabled, traffic logs include the following new fields of statistics for long-live sessions:

Duration delta (durationdelta) Displays the time in seconds between the last session log and the current session log.

Sent packet delta (sentpktdelta) Displays the number of sent packets. When the number of packets reported in the sentpktdelta field matches the number of bytes reported in the sentpkt field, it shows no missing logs.

Received packet delta (rcvdpktdelta) Displays the number of received packets. When the number of packets reported in the rcvdpktdelta field matches the number of bytes reported in the rcvdpkt field, it shows no missing logs.

The following the log example shows the new fields in the session log:

```
1: date=2023-12-07 time=14:19:59 eventtime=1701987599439429340 tz="-0800"
logid="0000000020" type="traffic" subtype="forward" level="notice" vd="vdom1"
srcip=10.1.100.22 srcport=53540 srcintf="wan2" srcintfrole="undefined"
dstip=172.16.200.55 dstport=80 dstintf="wan1" dstintfrole="undefined"
srccountry="Reserved" dstcountry="Reserved" sessionid=296 proto=6 action="accept"
policyid=1 policytype="policy" poluuid="e538d622-53eb-51ee-8adc-f8fbb0f22fdd"
policyname="B-out" service="HTTP" trandisp="snat" transip=172.16.200.2
transport=53540 duration=120 sentbyte=10855 rcvdbyte=1397640 sentpkt=205 rcvdpkt=1130
appcat="unscanned" sentdelta=10855 rcvddelta=1397640 durationdelta=120
sentpktdelta=205 rcvdpktdelta=1130
```

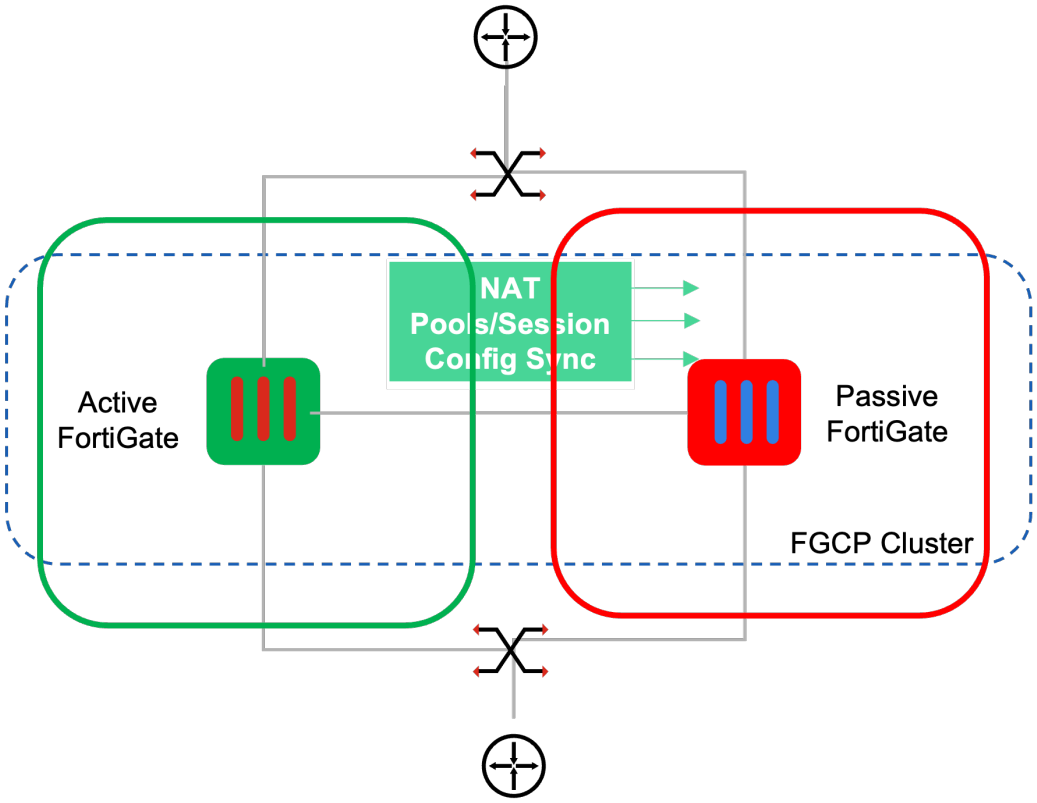
Reference Architectures

The most used translation types are NAT44, NAT64/DNS64 as well as dual-stack. In this chapter we will provide more details about these architectures can be deployed and to what extent the CGNAT service can be scaled with these architectures. We will start with general information about the supported clustering architectures.

FGCP Fortigate Clustering Protocol CGNAT

FGCP Active/Passive with CGNAT requires L2 connectivity and provides device, link and session failover. The configuration between the cluster members can be also synchronized using dedicated HW accelerated ports.

In the Active/Passive FGCP cluster (or vCluster) the HA hardware session synchronization copies the NAT sessions across (primary to the secondary node). FGCP A/P is supported for hyperscale and kernel CGNAT.



The NP7 systems provide dedicated interfaces for HA sync. These are the ha1 and ha2 interfaces. You could add more ports to the hardware session sync (and hardware logging) by adding those ports to the specific `set ports-using-npu` configuration listed below:

```
config system npu
   config port-path-option
      set ports-using-npu {ha1 ha2 aux1 aux2}
   end
```

The ports listed under the `set ports-using-npu` configuration shouldn't be used for other purposes. If you don't specify ports for HA hardware sync and session logging, the session synchronization will be sent via the internal switch fabric to the CPU and the HA sync (also logging) will be processed by the CPU and will not be accelerated.

Note that on FortiGate-1800F and 1801F there is limitation and on these systems you can only use port25 to port40 interfaces for hardware session synchronization.

The following configuration ha1 and ha2 are directly connected and used for heartbeat interfaces. Port aux1 is configured for session sync and it is hardware accelerated.

```
config system ha
   set hbdev ha1 100 ha2 100
   set session-pickup enable
   set hw-session-sync-dev aux1
end

config system npu
   config port-path-option
      set ports-using-npu aux1
   end
```

It is also possible to use other interface (or LAG) for hardware session synchronization interface, for example:

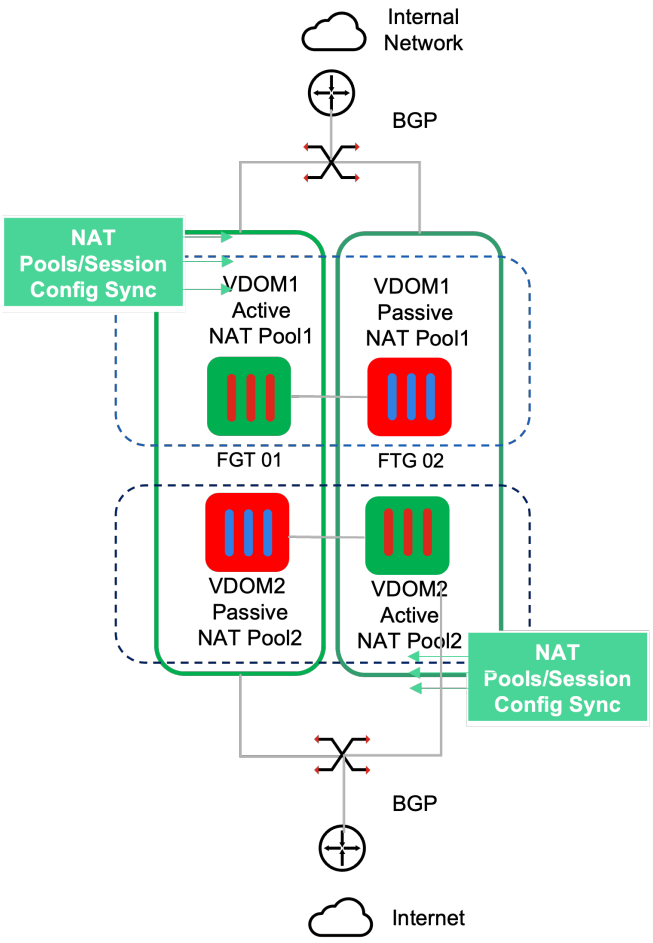
```
config system ha
   set hw-session-sync-dev port9 port10
end
```

No special configuration is required if you use a data interface. However, the data interface should not be used for any other traffic.

It is recommended that the primary and backup are directly connected, back-to-back, However, the session synchronization between two FortiGate can be stretched across data centers. The session sync interface must be L2 based and the delay on the link should be not more than 2-3 msec. Note that more than two members in FGCP cluster configuration is not supported. FGCP A/A is also not supported with hyperscale.

Virtual Cluster (vCluster) CGNAT

In a vCluster architecture there will be multiple VDOMs on both cluster nodes (active on one cluster node and passive on the other cluster node).



vCluster is supported for both hyperscale and Kernel CGNAT. In hyperscale vCluster CGNAT setup the HA hardware session synchronization copies the sessions from VDOMs processing traffic to VDOMs on the other (passive) FortiGate. Each "active" VDOM hosts own NAT pools and the sessions in this VDOM, which are synchronized with the corresponding VDOM on the "passive" node. In case of one node failure, the traffic will be redirected to the other device which means that this device has to be sized to support the traffic for both devices.

On a FortiGate FGCP cluster, the BGP router daemon process is only running on the Primary (Master) unit and when there is a failover, a new BGP process will be launched on the newly elected master. Even though the FortiGate has all the routes, if the peer sees the FortiGate as unresponsive, it will remove all the routes from its routing table and traffic will be interrupted. In to avoid traffic interruption BGP graceful restart is required on both peers.

Depending on how BGP is configured and which features are configured the following BGP timers can be fine tuned:

- **Holdtime-time**: number of seconds to mark peer as dead
- **Graceful-restart-time**: time needed for neighbors to restart(sec)

- **Graceful-stalepath-time**: time to hold stale paths of restarting neighbor(sec)
- **BFD**: the time how long the traffic should not pass after a failure.

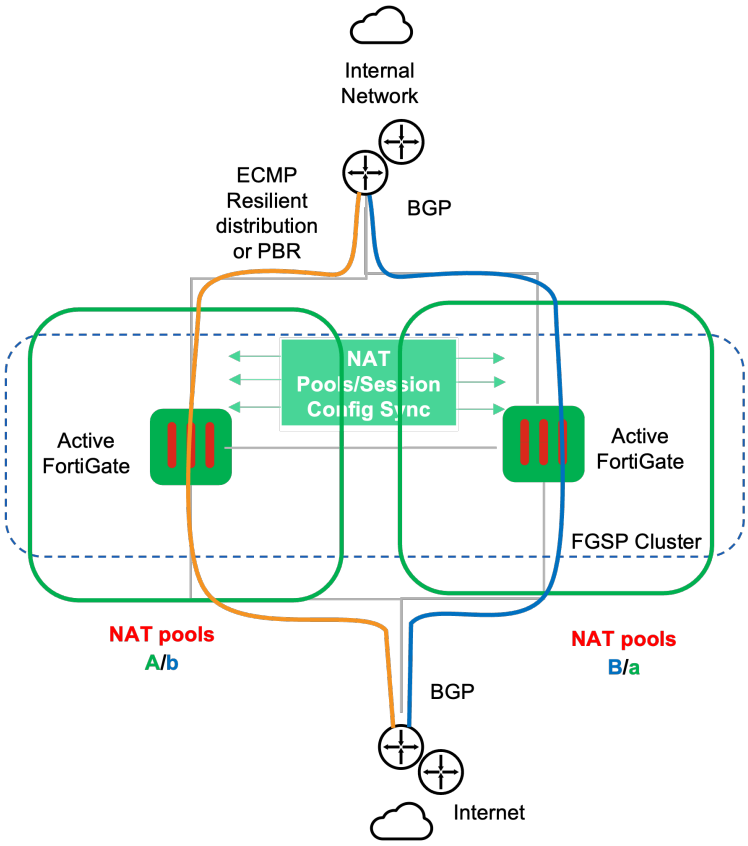
Recommendations for the optimal values for each setting is not part of this document because every network is different.

FortiGate Session Life Support Protocol (FGSP) CGNAT

FGSP - FortiGate Session Life Support Protocol is supported for both hyperscale and Kernel CGNAT. The recommended setup for FGSP is with two nodes because the sessions are synced between the nodes and the overall session cluster capacity is reduced due to this. A cluster of more than two nodes is possible but not recommended. Adding more nodes to the FGSP cluster does not increase the overall cluster session CPS/CCS capacity and only throughput is increased.

In hyperscale mode, FGSP cluster with two hyperscale peers, two FGCP clusters or one FortiGate and one FCGP cluster is supported.

In FGSP with two cluster nodes, both entities are active and are taking traffic, which is distributed externally. This could be load balancers or routers with ECMP (SRC IP Hashing) or Policy Based Routing. The external devices will distribute sessions among the FortiGate nodes and FGSP performs session synchronization of IPv4 and IPv6 sessions (including NAT - expectation sessions), so both entities are synchronized. In that architecture asymmetric traffic is supported due to the fact that the sessions are in sync on both nodes.



In case of one node failure, session failover occurs and active sessions will fail over to the peer that is still operating. The external routers or load balancers will detect the failover and re-distribute all sessions to the peer that is still operating. The routing and ECMP are greatly important to the solution. On the inside, the core router must receive the default gateway from the CGNAT firewall, which in turn the router advertises to the CGNAT firewall from the outside. To the outside, the CGNAT firewall needs to advertise prefixes to cover the NAT pools with differing BGP attributes to ensure the return traffic is routed amongst all FGSP members. The advertisements are controlled through route-maps and prefix-list configuration.

This architecture allows you to use one big NAT pool on both FortiGates and split it into two in the policy. The corresponding pool portion is announced with better priority either left or right. The order of NAT pools in the firewall policy is very important in the FGSP architecture.

```
config firewall policy
   edit 3
      set status enable
      set name "pba"
      set srcintf "Agg0.1000"
      set dstintf "Agg0.2000"
      set action accept
      set srcaddr "all"
      set dstaddr "all"
      set schedule "always"
      set service "ALL"
      set logtraffic disable
      set nat enable
      set ippool enable
```

```
      set poolname "PbaNAT-A" , "PbaNAT-B" <- This is the left Fortigate.
   next
   end
```

Note that on the right FortiGate the NAT pools order is opposite. The route maps `dgw` and `pool` and prefix lists `dgw`, `PoolA` and `PoolB` are controlling the routing advertisements.

```
config router bgp
   set as 65002
      set router-id 192.168.194.10
      set keepalive-timer 1
      set holdtime-timer 4
      set ebgp-multipath enable
      config neighbor
         edit "100.64.100.2"
            set bfd enable
            set soft-reconfiguration enable
            set as-override enable
            set remote-as 65001
            set route-map-out "dgw" <-- This map controls default route advertisement
         next
         edit "100.64.100.3"
            set bfd enable
            set soft-reconfiguration enable
            set as-override enable <-- This map controls default route advertisement
            set remote-as 65001
            set route-map-out "dgw"
         next
         edit "100.64.200.2"
            set bfd enable
            set soft-reconfiguration enable
            set remote-as 65001
            set route-map-out "pool" <-- This map controls the pools advertisement
         next
         edit "100.64.200.3"
            set bfd enable
            set soft-reconfiguration enable
            set remote-as 65001
            set route-map-out "pool" <-- This map controls the pools advertisement
         next
         end
         config redistribute "connected"
         end
         config redistribute "rip"
         end
         config redistribute "ospf"
         end
         config redistribute "static"
            set status enable
         end
         config redistribute "isis"
         end
      …
      end
end
```

And the route maps definitions:

```
config router route-map
```

```
   edit "dgw"
      config rule
         edit 1
            set match-ip-address "dgw"
            unset set-ip-prefsrc
         end

config router route-map
   edit "pool"
      config rule
         edit 1
            set match-ip-address "poolA"
            set set-aspath "65002"
            unset set-ip-prefsrc next
         edit 2
            set match-ip-address "poolB"
            set set-aspath "65002 65002"
            unset set-ip-prefsrc
         end

config router prefix-list
   edit "dgw"
      config rule
         edit 1
            set prefix 0.0.0.0 0.0.0.0
            unset ge
            unset le
         end

edit "poolA"
   config rule
      edit 1
         set prefix 17.0.0.0 255.255.252.0
         unset ge
         unset le
      end

config router static
   edit 2001
      set dst 17.0.0.0 255.255.252.0 <-- Blackhole as a pull-up route
      set blackhole enable
      set vrf 0
   end
```

The blackhole acts as pull-up route and in case default is not received from the Internet routers this route will not be advertised towards the core.

The following FGSP cluster configuration shows the required configuration:

```
config system standalone-cluster
   set standalone-group-id 14
   set group-member-id 1
   set layer2-connection available
   set session-sync-dev "port21" "port22"

   config cluster-peer
   edit 21
      set peerip 192.168.1.2
```

```
      set syncvd "CGNAT"
      set down-intfs-before-sess-sync "Agg0"
   next
   edit 22
      set peerip 192.168.2.2
      set syncvd "CGNAT"
      set down-intfs-before-sess-sync "Agg0"
   next
   end
```

In this setup ports 21 and 22 are used for FGSP communication:

```
config system interface
   edit "port21"
      set vdom "root"
      set ip 192.168.1.1 255.255.255.0
      set allowaccess ping
      set type physical
      set mediatype sr4
      set alias "FGSP1"
      set snmp-index 27
      set forward-error-correction cl91-rs-fec
      set speed 100Gfull
      set mtu-override enable
      set mtu 9216
   next
   edit "port22"
      set vdom "root"
      set ip 192.168.2.1 255.255.255.0
      set allowaccess ping
      set type physical
      set mediatype sr4
      set alias "FGSP2"
      set snmp-index 28
      set forward-error-correction cl91-rs-fec
      set speed 100Gfull
      set mtu-override enable
      set mtu 9216
   next
   end
```

The `layer2-connection` informs FGSP to use a layer 2 broadcast for session synchronization messages.

The `mtu`, `session-sync-dev`, and `sync-packet-balance` are optimizations as described in Optimizing FGSP session synchronization and redundancy.

The `down-intfs-before-sess-sync` leaves the listed interface(s) operationally down until synchronization is complete; this is useful for maintenance activities where the node should not take traffic until it has the session information it requires to handle traffic. This interface would be the hardware switch when deployed as recommended above.

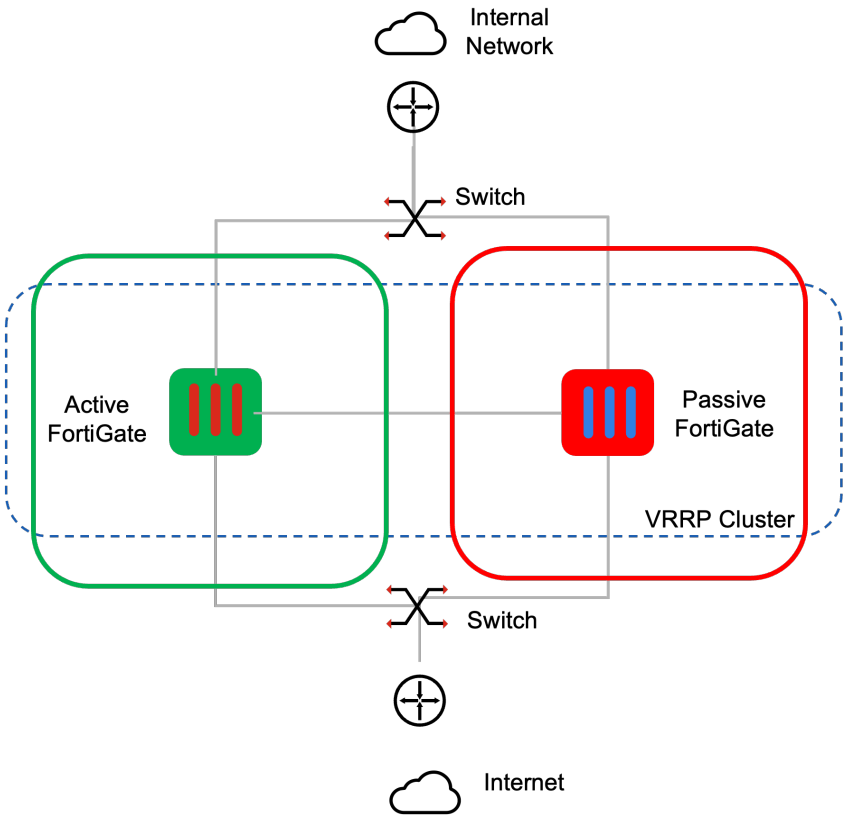
The NAT sessions are synced together with UDP and expectation sessions:

```
config system ha
   set sync-packet-balance enable
   set session-pickup enable
   set session-pickup-connectionless enable
   set session-pickup-expectation enable
   set session-pickup-nat enable
```

```
    set override disable
  end
```

Virtual Router Redundancy Protocol (VRRP) CGNAT

VRRP can be used as a high availability solution together with CGNAT. It guarantees resiliency without session sync. With VRRP, if a FortiGate unit fails, all traffic will be transparently failing over to another FortiGate that takes over the traffic from the failed FortiGate. If the failed FortiGate is restored, it will once again take over processing traffic for the network. VRRP v2 and v3 are supported with IPv4 and IPv6 VRRP and on the same interface.



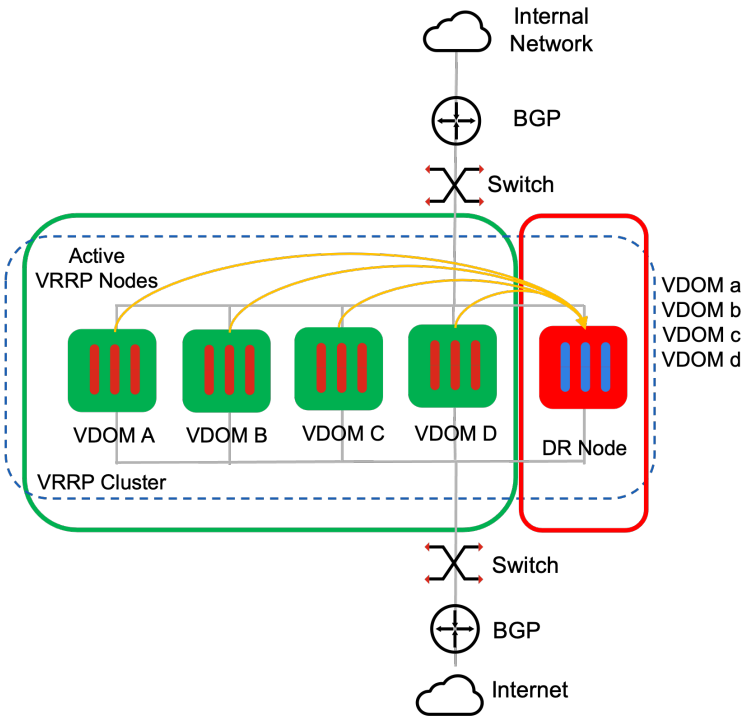
The VRRP virtual MAC address is a shared MAC address adopted by the primary router. If the primary router fails, the same virtual MAC address is picked up by the new primary router, allowing all devices on the network to transparently connect to the default route using the same virtual MAC address.

After a failover, the new primary unit sends gratuitous ARP packets to refresh the MAC forwarding tables of the switches connected to the cluster and than the switches start directing packets to the new primary unit. Some L3 switches may not update their table upon gratuitous ARP sent by the FortiGate. To solve that problem the FortiGate can shutdown all its interfaces for a second so that the switch can detect this failure and clear its MAC forwarding tables. FGSP is also compatible with FortiGate VRRP, in case session sync is required. The following article is describing the setup of VRRP cluster.

N+1 standalone nodes CGNAT

This architecture allows horizontal CGNAT scaling, by adding more FortiGate nodes, configured as VRRP cluster with a "DR" FortiGate . This design **does not provide session failover**, however if session redundancy is required, the N+1 architecture can be expanded with FGSP.

The DR FortiGate provides resiliency, meaning that the capacity of a "failed" FortiGate will be covered by the DR device. The DR FortiGate hosts multiple VDOMs and each "active" FortiGate builds up a VRRP cluster with the VDOM of the DR.



Note that the capacity or the DR FortiGate should be carefully planed. In a case of single node failure, the DR covers the capacity of a single "failed" FortiGate. In a case of multiple nodes failures the capacity of the DR FortiGate shall be correspondingly planned to cover the amount of "failed" FortiGates.

In the N+1 scenario with VRRP the "active" FortiGates are acting as default GW to the core router (the router will receive multiple default routes and will equally distribute sessions across with ECMP SRC IP hashing) and each node is translating customer traffic (and announcing the own NAT pools correspondingly).

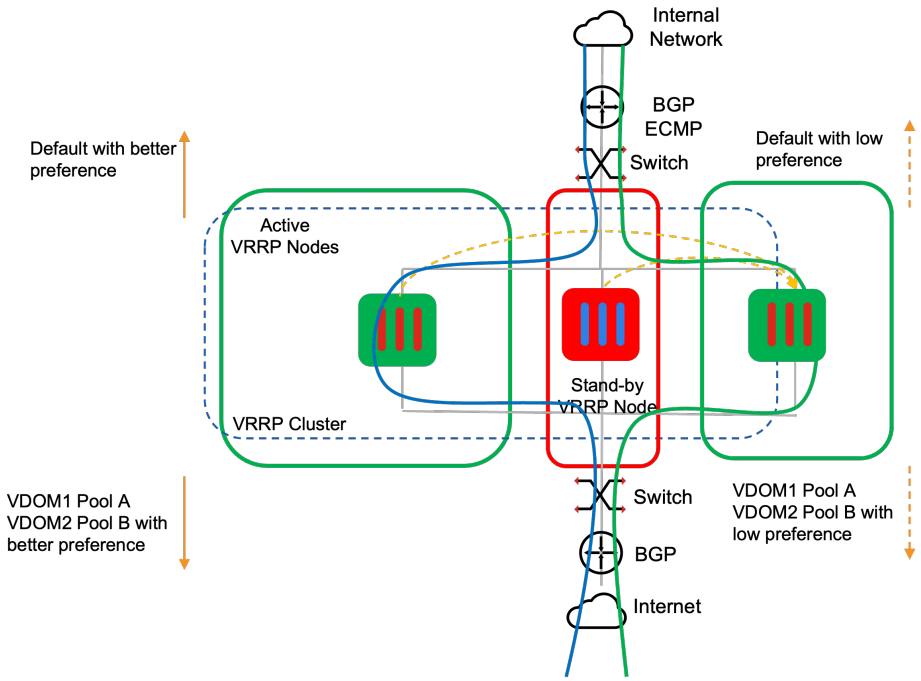
Similar N+1 architecture can be achieved without using VRRP by relying on dynamic routing protocol only.

In that case the DR node will announce default route towards the core with worse preference and the "active" FortiGate nodes will be announcing default route (0.0.0.0/0) with the same better preference compared to the DR FortiGate to achieve equal cost multi-path routing.

Equal cost routes will cause the core router to distribute traffic to all "active" FortiGates. In order to have persistent traffic spill-over, the core routers must be using SRC IP hashing ECMP.

The returning traffic (from Internet) is routed to the corresponding FortiGate, which is announcing the (own hosted) NAT pools. All "active" FortiGate nodes are announcing the own NAT pools with better priority and the DR FortiGate is announcing the same NAT pools (in different VDOMS) but with lower priority.

The effect of switching the routes is based on on conditional routing advertisement. In case where an “active” FortiGate is completely down (the middle device in the picture below), the corresponding pool on DR FortiGate will become valid (the conditional advertisement will trigger the DR path) and the DR VDOM routes will have better priority over the DR link.



Effectively the "active" FortiGate path failure will cause the traffic to flow through the corresponding VDOM on the DR node in both directions (Core and internet router) because the BGP sees the link down and immediately changes to advertise the routes to the DR path and the lower preference routes (default and NAT pools) will be activated. In case of failure the sessions on the failed node will be lost, however the majority of the client's sessions will recover based on built-in client recovery functionality.

Session Sync

Session synchronization for NAT is supported in the following clustering architectures for CGNAT on NP7 systems: FGCP A/P, Virtual Cluster and FGSP.

The following table provides short summary of the CGNAT clustering technologies with the session synchronization:

Clustering method	Hyperscale NAT Session Sync	Native FortiOS NAT Sync
FGCP A/P	Supported	Supported
FGCP A/A	Not Supported	Not Supported

Clustering method	Hyperscale NAT Session Sync	Native FortiOS NAT Sync
Virtual Cluster (FGCP) (FGCP based)	Supported	Supported
FGSP A/A	Supported (since FOS 7.0.6)	Supported
Standalone N+1	No session sync	No session sync
VRRP	No session sync , however FGSP can be added on top	No session sync , however FGSP can be added on top

Sometimes operators are looking to deploy another non-HW accelerated VDOM on a device running hyperscale CGNAT.

Note that HW-accelerated CGNAT requires **src-IP hashing**, which is global system setting. Be aware that utilizing another non-HW accelerated VDOM on the same device, which is running HW-accelerated CGNAT would impact the VDOM functionality and capacity.

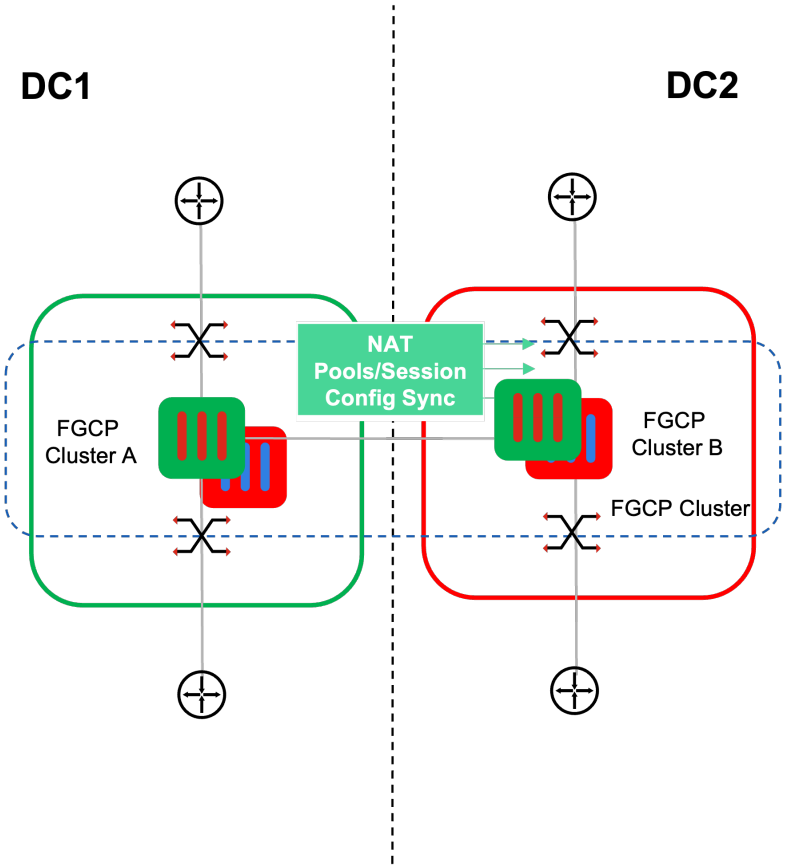
Note also, that deploying FGSP for the non-hyperscale VDOMs may cause the FGSP synchronizing sessions from other VDOMS (standalone or FGCP), despite putting `set syncvd`. In other words, currently the default FGSP sync is done automatically for all HW sessions in all hyperscale VDOMs and FGSP cluster-sync/syncvd (udp/708) is applied only for "standard/normal/non-hw" sessions/VDOMs. Future releases of FortiOS could provide per VDOM FGCP/FGSP sync definitions.

Note that session sync for NAT pools that deploy port blocks (PBA and fixed port range) do not synchronize the NAT pool state. Only the sessions (and expectation sessions) are synchronized:

```
config system ha
   set session-pickup enable
   set session-pickup-nat enable
   set session-pickup-connectionless enable
   set session-pickup-expectation enable
```

FGCP A/P CGNAT geo redundancy

In this architecture the customer traffic is routed to the "active" FortiGate node on each site and passive nodes are not taking traffic. Each "active" FortiGate node is using own NAT pools for translations.



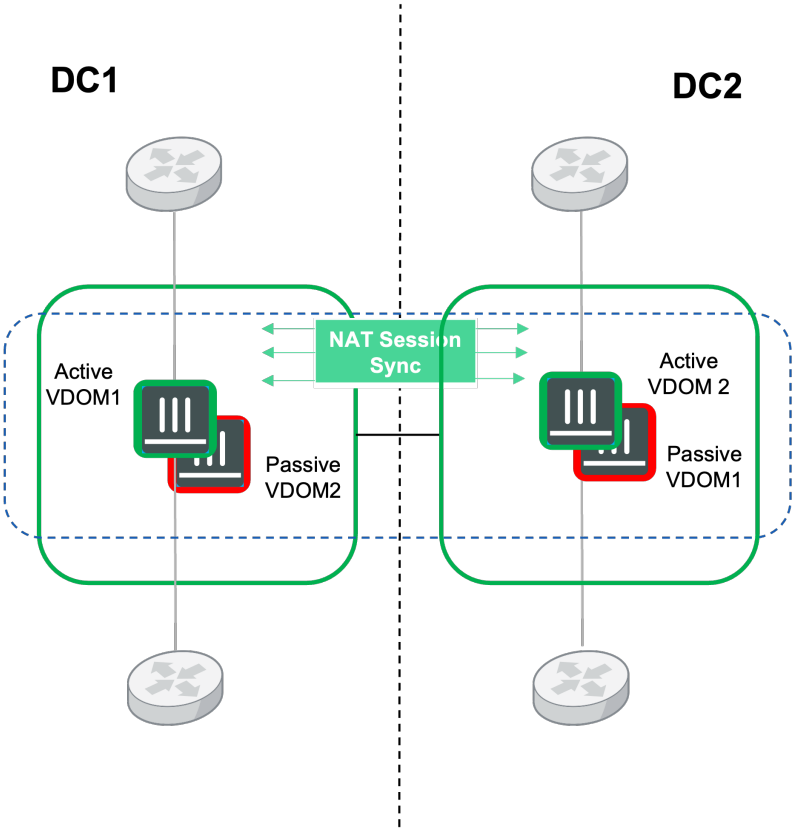
Synchronizing sessions between FGCP clusters is useful when data centers in different locations can be used for load-balancing, however the session synchronization must be fast enough (faster than the server reply) in order not to come to session setup irregularities. You can check the following link for further details.

Otherwise the FGCP cluster can be stretched across two data centers.
The Active node can be located on one data center and the Passive node on the other data center side.

vCluster CGNAT geo redundancy

In this architecture both data centers and both FortiGates (on each site) are active. The FortiGate with "active" VDOM1 (with own NAT pools) is taking traffic on DC1 and synchronizes the own sessions with the passive VDOM1 on DC2. The FortiGate with the active VDOM2 (with own NAT pools) is taking traffic on DC2 and synchronizes the own sessions with

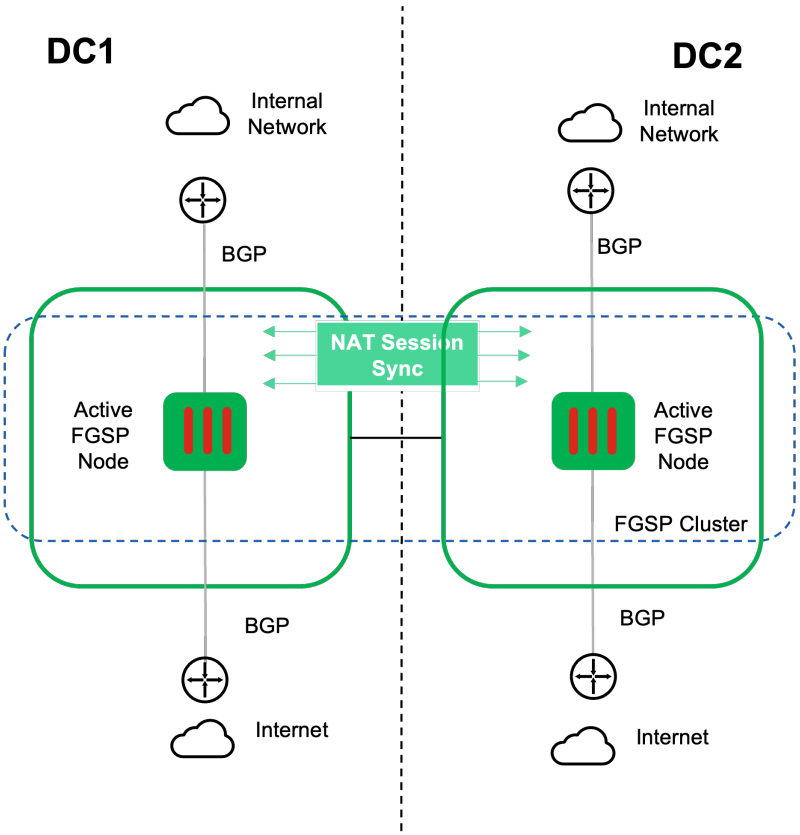
the passive VDOM2 on DC1. The session sync links between the data centers are L2 based.



In this architecture the core router must use ECMP SRC IP hashing or Policy Based Routing in order to guarantee session persistence to one of the FortiGate nodes. The article Check HA synchronization status discusses the setup of two nodes vCluster.

FGSP CGNAT geo redundancy

In this architecture both data centers and both FortiGates (on each side) are active and are taking traffic. If one of the FortiGate nodes fails, session failover occurs, and active sessions fail over to the peer that is still operating. This failover occurs without any loss of data and the traffic from the failed site will transverse to the healthy FortiGate on the other data center side. The session sync links between the data centers can be both L2 or L3 based.

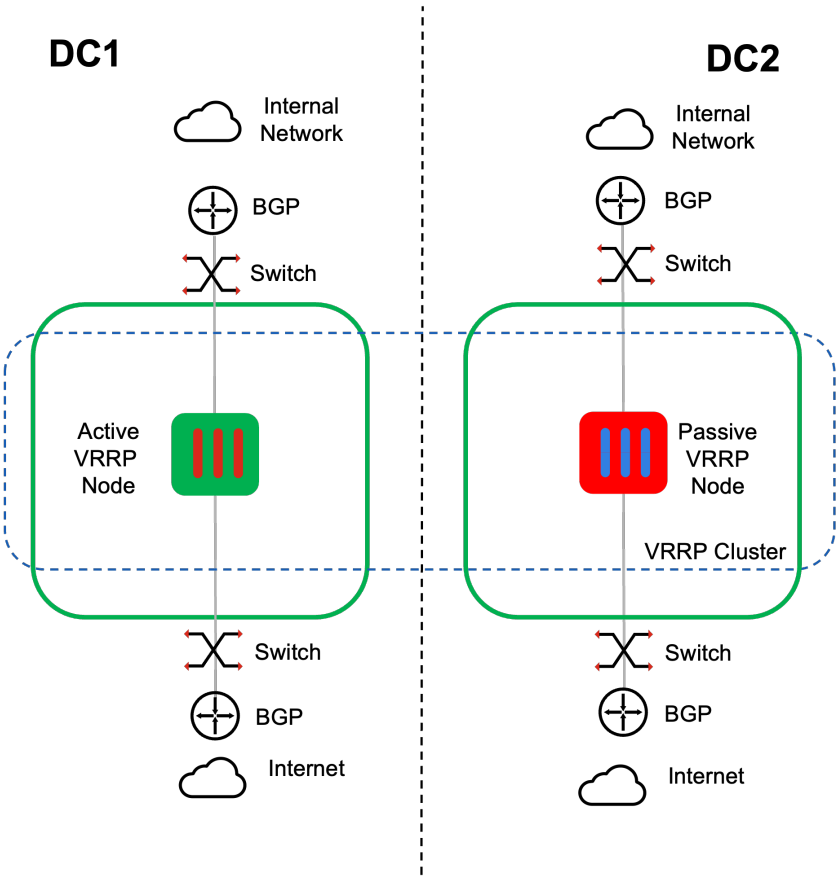


The core router shall be using ECMP SRC IP hash load balancing, so that the load sharing between the FGSP nodes is working flawlessly.

A FGSP cluster with four nodes (Deterministic NAT and PBA) has been successfully tested in lab environment, however the session capacity of such cluster is limited to the capacity of a single FortiGate node.

VRRP CGNAT geo redundancy

In this architecture the customer traffic is routed to the "active" FortiGate (which is holding the virtual MAC) and the passive FortiGate is located on a DR data center and it is not taking traffic. No sessions are synchronized between the active and passive nodes between the data centers. The connection between the data centers must be L2. In case session sync is required, FGSP can be configured on top of VRRP.

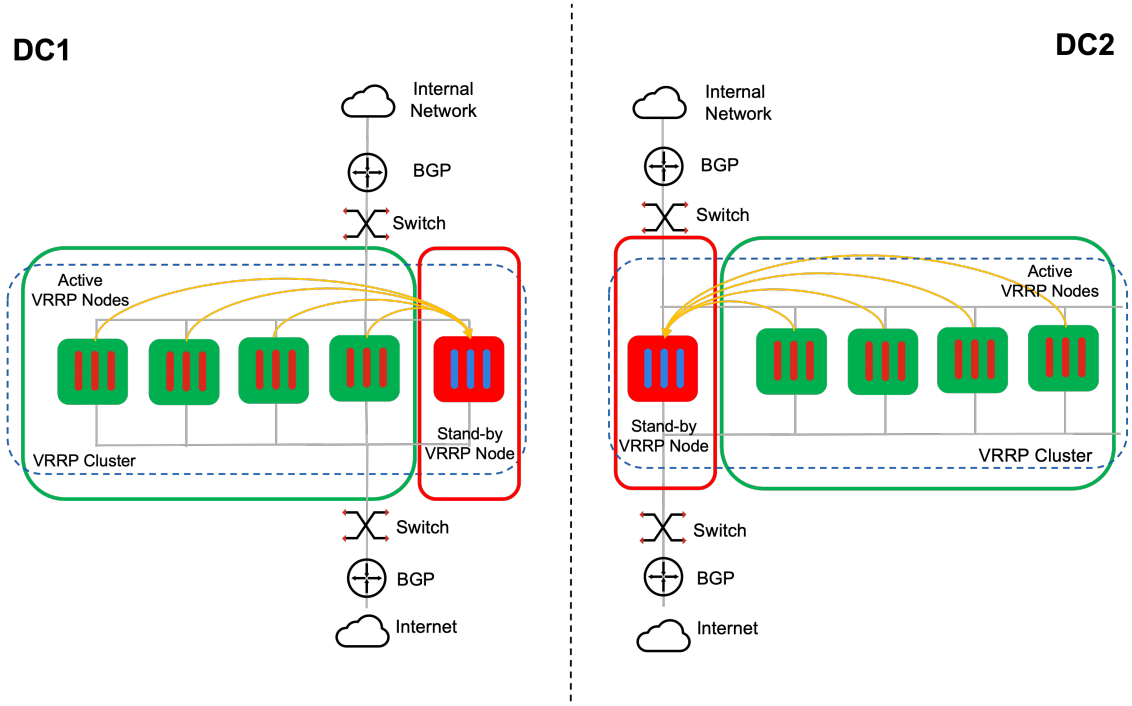


This architecture is based on single-domain VRRP and consists of a VRRP domain with two FortiGates, however multiple variations can be deployed, such as N+1 with several VRRP domains, where multiple "active/primary FortiGates" are located on active data center and the backup FortiGate is on DR data center.

N+1 Standalone nodes CGNAT geo redundancy

In this architecture there are multiple active stand-alone nodes with a single local stand-by on one data center and the same setup (multiple active stand-alone and single local stand-by) on the other data center. The local redundancy can be achieved by using simple routing technique, namely exporting the default route with better priority to the local data center's router, hence BGP will always prefer best route and build ECMP between identical routes towards the "active" FortiGate's. For the local stand-by device default is exported with worse priority, hence this route will be activated only in case of failure (conditional routing). The returning from Internet traffic (for the NAT pools from the DC1) is routed to the corresponding device, according the NAT pools announcement. The stand-by device is hosting the same NAT pools like the active devices but it is announcing those with worse preference, hence these routes will be activated only in case of failure (conditional routing).

The geo-redundancy towards DC2 is achieved again with the same routing technique (default route is exported with second worse priority towards DC2) and the NAT pools are announced with second worse preference and will be activated only in case of failure (conditional routing).



It is important to mention that in case of full site failure, the nodes in the other DC will be taking double the traffic from the failed DC and the node's capacity shall be correspondingly sized.

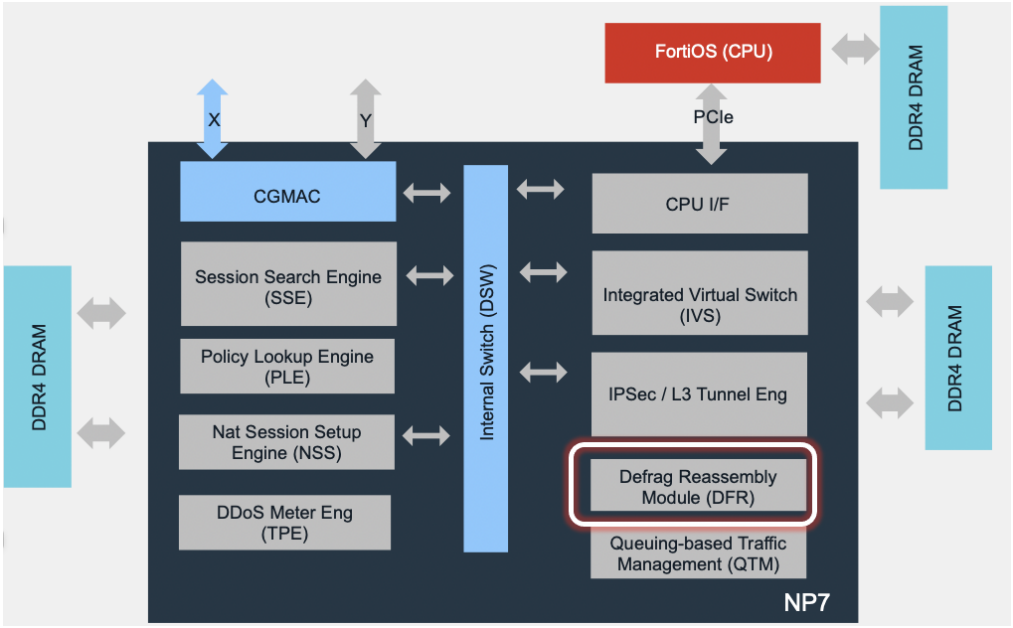
Also a dual failure scenario (which is unlikely) - a full DC and any of the "active" FortiGate nodes failure on the current DC could be potentially foreseen and implemented via 2nd/3rd lower preference NAT pools announcement, however in such scenario all the nodes must be over-provisioned for the triple amount of traffic that usually a single node would be carrying.

NP7 Specific Operational Topics

This chapter includes a collection of topics relevant to operating a CGNAT system.

Defrag Reassembly Module in NP7 (DFR)

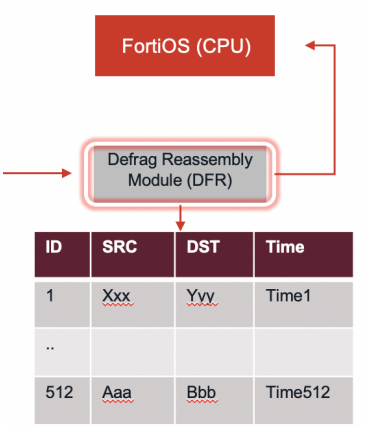
The NP7 processors uses the defrag/reassembly (DFR) module to accelerate and re-assemble fragmented packets. This offload functionality is supported for packets that have been fragmented into two packets (1 header and 1 packet fragment). Traffic that has been fragmented into more than two packets is re-assembled by the CPU.



Each NPU DFR uses de-fragmentation table with size of 512 entries/contexts, which is used as a buffer and every time a fragmented packet is entered into the table as de-frag context with Source IP, Destination IP and context ID.

DFR Timers

Depending on the amount of NPU processors available in the system, the overall de-fragmentation table size will be different.



If there is no match and the table is not full the context is stored and pending `min_timeout` and `max_timeout` timers are started. These timers are configurable under:

```
config system npu
   config ip-reassembly
      set status {disable | enable}
      set min_timeout <micro-seconds>
      set max_timeout <micro-seconds>
   end
```

Where:

- `min_timeout` is the minimum timeout value for IP reassembly in the range 5 to 600,000,000 µs (micro seconds). The default `min-timeout` is 64 µs.
- `max_timeout` is the maximum timeout value for IP reassembly 5 to 600,000,000 µs. The default `max-timeout` is 1000 µs.

In case there is match in the table the the packet is assembled with the pending fragment, defragmented and sent with normal flow.

If the table is full and there is a match the `min_timeout` will remove the de-frag context and will send it to the CPU to leave space for new packets to come in. If the table is not full but `max_timeout` elapses, the fragment is removed from the table and is sent to CPU for further processing.

In the other situation, when the table is full and there is no match, the fragment is sent to the CPU for further re-assembly.

Under heavy fragmentation attacks this behavior could cause CPU spikes and lead potentially to system overload. Therefore host protection engine (HPE) shall be configured to protect the platform from DoS attacks by categorizing the incoming packets based on packet rate and processing cost and applying packet shaping to packets that can cause DoS attacks.

Host Protection Engine (HPE)

The host protection engine (HPE) is providing protection to the FortiGate CPU from DoS attacks by categorizing incoming packets based on packet rate and processing cost and applying packet shaping to packets that can cause DoS attacks.

For this the HPE is using queues, in which the corresponding packets are put. By configuring the following the HPE DoS protection can be turned on and off and the corresponding queues can be adjusted:

```
config system npu
   config hpe
      set tcpsyn-max <packets-per-second>
      set tcp-max <packets-per-second>
      set udp-max <packets-per-second>
      set icmp-max <packets-per-second>
      set sctp-max <packets-per-second>
      set esp-max <packets-per-second>
      set ip-frag-max <packets-per-second>
      set ip-others-max <packets-per-second>
      set arp-max <packets-per-second>
      set l2-others-max <packets-per-second>
      set pri-type-max <packets-per-second>
      set enable-shaper {disable | enable}
   end
```

HPE queues

```
diagnose npu np7 hpe 0 <- NPU ID0
[NP7_0]
Queue  Type         NPU-min   NPU-max  CFG-min(pps) CFG-max(pps) Pkt-credit
------------------------------------------------------------------------------
0      all-protocol 39731     39731    40000        40000        0
*The all-protocol is active. The config of individual protocols won't apply.
0      high-priority39731     39731    40000        40000        0
0      IP-Frag      39731     39731    40000        40000        0
0      TCP-syn      39731     39731    40000        40000        0
0      TCP-synack   39731     39731    40000        40000        0
0      TCP-finrst   39731     39731    40000        40000        0
0      TCP          39731     39731    40000        40000        0
0      UDP          39731     39731    40000        40000        0
0      ICMP         4966      4966     5000         5000         0
0      SCTP         4966      4966     5000         5000         0
0      ESP          4966      4966     5000         5000         0
0      IP_others    4966      4966     5000         5000         0
0      ARP          4966      4966     5000         5000         0
0      l2_others    4966      4966     5000         5000         0
------------------------------------------------------------------------------
HPE HW pkt_credit:13947 , tsref_inv:42500, tsref_gap:32, hpe_refskip:0 , hif->nr_ring:16
```

The `NPU-min` and `NPU-max` the register reading of max and min value for each queue.

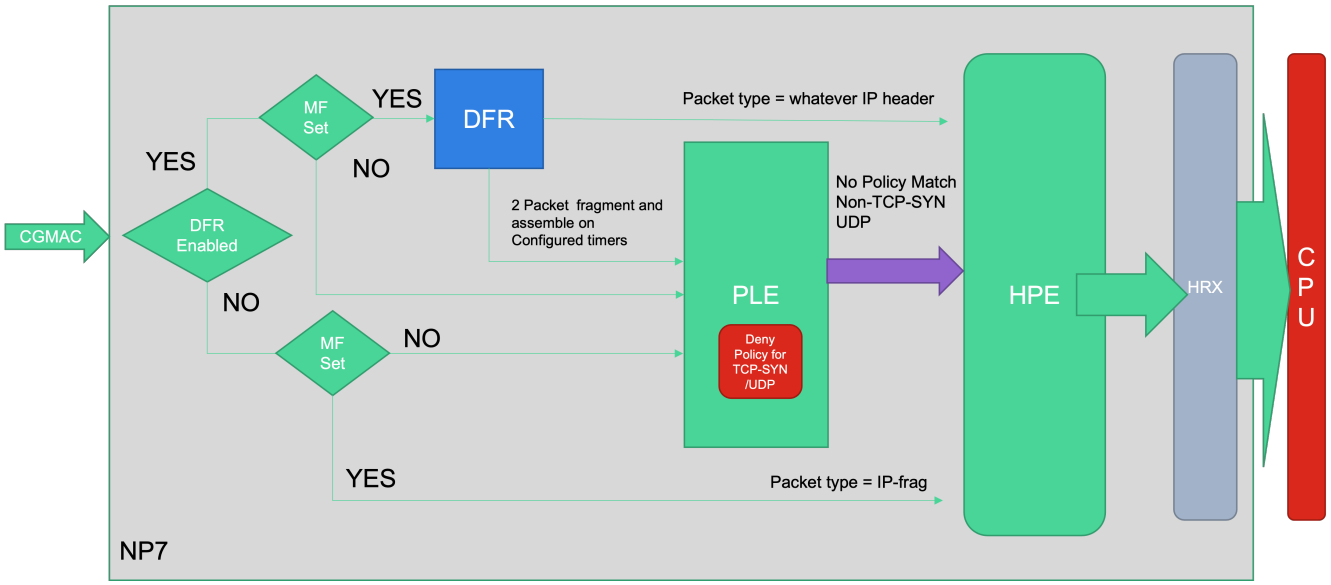
`CFG-min(pps` the setting value of hpe configuration in CLI command and it is packet per second rate limit for each host rx queue.

`CFG-max(pps)` the value is `CFG-min` of the HPE configuration.

`Pkt-credit` the register reading of type shaping credit in HPE engine.

Note the NPU-min and NPU-max queue values and the configured min and max pps values for each queue in the NPU.

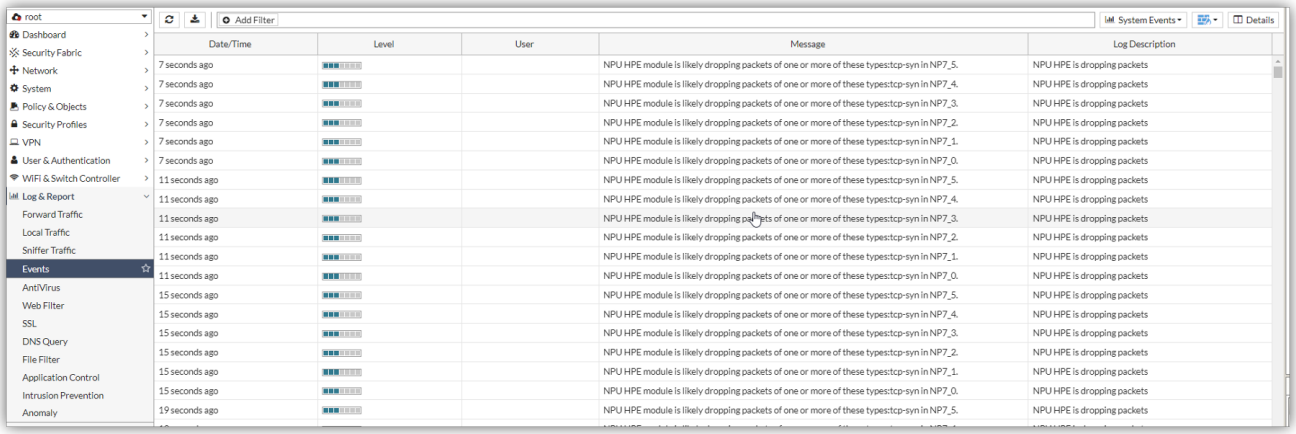
There is a major difference how the HPE is treating the packets based on the fact if DFR is configured or not. The following diagram shows the behavior:



When the DFR is reassembling under normal operations (two fragments and configured DFR timers) the hyperscale firewall default policy action shall be configured for dropping TCP-SYN and UDP packets. The setting for `drop-on-hardware` is default, where the NP7 processors drop TCP-SYN and UDP packets. Non TCP-SYN and non-UDP packets (also packets, with no policy match) are forwarded the CPU (correspondingly to the HPE). With this option the number of packets sent to the CPU is reduced. All other packet types (for example, ICMP packets) that don't match the hyperscale firewall policy are sent to the CPU, also packets accepted by session helpers are also sent to the CPU/HPE.

In case DFR is forwarding packets to the CPU for reassembly (more than two fragments or DFR fragments taken out from the DFR table due to timers elapsed) the HPE feature sees the packet type and and puts those packets in the corresponding queues (TCP-SYN, TCP-ACK, TCP-SYNACK, UDP etc.). The events in the FortiGate GUI will show the HPE drops with the corresponding packets with packet type in the different NPUs:

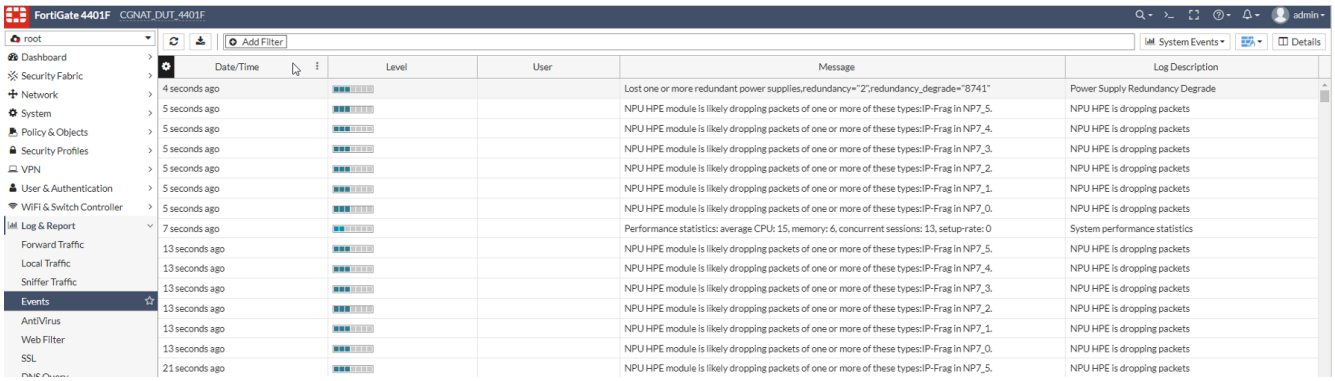
FortiGate Events GUI with DFR configured



If the DFR is not configured the HPE feature treats the packets as IP-Frag packets and only this queue is used, which means that the HPE feature will be dropping much faster compared to the previous case if the DFR is configured.

The events in the FortiGate GUI will show the HPE drops as IP-Frag in the different NPUs:

FortiGate Events GUI without DFR configured



Denial of Service protection

Hosts (located in the DMZ or other segments) or client NAT pools can protected by the FortiGate from DoS attacks. A Denial of Service (DoS) policy examines network traffic arriving at a FortiGate interface for anomalous patterns, which usually indicates an attack.

DoS policies are processed **before** security policies, preventing attacks from triggering more resource intensive security protection and slowing down the FortiGate.

It is **not recommended** to operate DoS protection in monitor mode (that is DoS policies with Action set to Monitor) on a hyperscale FortiGate.

Enabling the monitor mode shall be done for debugging DoS protection only and during normal operation, monitor mode should be disabled.

Hyperscale `per-session` hardware logging is not compatible with session-count DoS anomalies. When configuring hardware logging server groups, if `log-mode` is set to `per-session` you must delete any session-count DoS anomalies that you have been added to DoS policies. If not, for some processes resource usage can reach 100% and some processes might become stuck or crash.

Rate-based DoS anomalies are compatible with hyperscale per-session hardware logging. Session count based DoS anomalies have session in their name (for example, `tcp_src_session` and `tcp_dst_session`).

For kernel NAT monitor for DoS policies can be used for a shoe time to figure out the traffic base. During normal operation monitor mode should be disabled. Monitor mode can cause NP7 processors to become unresponsive when processing large amounts of traffic.

NAT pool exhaustion attacks can be stopped by configuring the services session refresh direction to outgoing when the some system on the internet side tries to keep the sockets open:

```
config system session-ttl
   config port
   edit 1
      set protocol <protocol-number>
```

```
    set timeout <timeout>
    set refresh-direction outgoing
  end
```

When `outgoing` is configured as refresh direction, only the client can keep the session open and after the session expires the resource is released.

DoS protection on Hyperscale Systems

FortiGates with NP7 processors provide support for hardware accelerated DoS protection. The hyperscale features on NP7 systems must be configured to offload the DoS policy sessions to the NPU:

```
config system settings
  set policy-offload-level full-offload
end
```

`full-offload` enables hyperscale features (including offloading DoS policy session to the NP7 processors) for the FortiGate (or the current VDOM). All sessions (except session helpers) are processed by the NPUs and bypass the CPUs. This option is available when the system is licensed for hyperscale processing.

The options for the DoS policy hardware acceleration must be also configured:

```
config system npu
  config dos-options
    set npu-dos-meter-mode {global | local}
  end
end
```

Setting `npu-dos-meter-mode to global` configures DoS metering across all NP7 processors and it should only be used along with CGNAT. This mode configures the anomaly threshold to total across all available NP7 processors.

There are number of predefined IPv4 L3 and L4 (as well as IPv6) anomalies:

```
tcp-syn-fin          TCP SYN flood SYN/FIN flag set anomalies.
tcp-fin-noack        TCP SYN flood with FIN flag set without ACK setting anomalies.
tcp-fin-only         TCP SYN flood with only FIN flag set anomalies.
tcp-no-flag          TCP SYN flood with no flag set anomalies.
tcp-syn-data         TCP SYN flood packets with data anomalies.
tcp-winnuke          TCP WinNuke anomalies.
tcp-land             TCP land anomalies.
udp-land             UDP land anomalies.
icmp-land            ICMP land anomalies.
icmp-frag            Layer 3 fragmented packets that could be part of layer 4 ICMP
                     anomalies.
ipv4-land            Land anomalies.
ipv4-proto-err       Invalid layer 4 protocol anomalies.
ipv4-unknopt         Unknown option anomalies.
ipv4-optrr           Record route option anomalies.
ipv4-optssrr         Strict source record route option anomalies.
ipv4-optlsrr         Loose source record route option anomalies.
ipv4-optstream       Stream option anomalies.
ipv4-optsecurity     Security option anomalies.
ipv4-opttimestamp    Timestamp option anomalies.
ipv4-csum-err        Invalid IPv4 IP checksum anomalies.
tcp-csum-err         Invalid IPv4 TCP checksum anomalies.
udp-csum-err         Invalid IPv4 UDP checksum anomalies.
icmp-csum-err        Invalid IPv4 ICMP checksum anomalies.
```

```
ipv6-land            Land anomalies.
ipv6-proto-err       Layer 4 invalid protocol anomalies.
ipv6-unknopt         Unknown option anomalies.
ipv6-saddr-err       Source address as multicast anomalies.
ipv6-daddr-err       Destination address as unspecified or loopback address anomalies.
ipv6-optralert       Router alert option anomalies.
ipv6-optjumbo        Jumbo options anomalies.
ipv6-opttunnel       Tunnel encapsulation limit option anomalies.
ipv6-opthomeaddr     Home address option anomalies.
ipv6-optnsap         Network service access point address option anomalies.
ipv6-optendpid       End point identification anomalies.
ipv6-optinvld        Invalid option anomalies.Invalid option anomalies.
```

Not all the traffic can be offloaded to the NPUs for DoS policy anomalies in the Hyperscale CGNAT use case. For more information, see DoS policy hardware acceleration.

Fortinet recommends protecting the CPU by enabling the Host Protection Engine (HPE) (seeHost Protection Engine (HPE) on page 98) to minimize the DoS attacks impact caused by some sensors processing in the CPU (refer to DoS policy hardware acceleration). The HPE shapers efficiently drop ICMP, UDP and SCTP sessions which are beyond the safe limit (DoS Attacks). In fact the HPE is NP7 module implemented in the NP7 HIF - Host Interface connecting the NP7 and CPU. Any sessions sent to the CPU would be inspected by the HIF and hardware accelerated, and with that the CPU is protected.

The anomalies can be used in different DoS policies. The DoS policies are applied to the ingress network traffic at a FortiGate interface and inspected before security policies are applied to the traffic

```
config firewall DoS-policy
   edit 1
      set status enable
      set name "DoS_policy"
      set comments ''
      set interface "port22"
      set srcaddr "all"
         config anomaly
            edit "tcp_syn_flood"
               set status disable
               set log disable
               set action pass
               set quarantine none
               set threshold 2000
            next
            ...
            edit "udp_flood"
               set status enable
               set log enable
               set action block
               set quarantine none
               set threshold 5000
            next
            ...
            edit "sctp_dst_session"
               set status disable
               set log disable
               set action pass
               set quarantine none
               set threshold 5000
            end
```

In the example above `udp_flood` protection is enabled and the configuration blocks UDP flooding when the amount of UDP packets per second is beyond 5000.

DoS protection on kernel NAT Systems

Denial of Service protection in kernel CGNAT works pretty much the same way as described in the hyperscale use case. A Denial of Service (DoS) policy examines network traffic arriving at a FortiGate interface and the DoS policies are processed **before** security policies, preventing attacks from triggering more resource intensive security protection and slowing down the FortiGate.

On devices with NP7 processors the policy offload level must be configured for `dos-offload`.

```
config system settings
   set policy-offload-level dos-offload
end
```

`dos-offload` This command enables DoS policy hardware acceleration for the FortiGate or for the current VDOM if multiple VDOMs are enabled. You can also use the following command to configure some DoS policy hardware acceleration options:

```
config system npu
   config dos-options
      set npu-dos-meter-mode {global | local}
      set npu-dos-tpe-mode {disable | enable}
   end
```

`npu-dos-meter-mode` global is the default setting and with that the DoS metering/anomaly threshold is activated for all NP7 processors. npu-dos-meter-mode local set the threshold per every available NP7 processor in system so that the configured anomaly threshold is the sum of all NP7 thresholds in the system. For example threshold is 400, which means that the sensor will be activated at value of 1600 on system with four NPUs.

`npu-dos-tpe-mode` enable is the default value and it inserts the dos meter ID into the session table. When this is enabled UDP_FLOOD and ICMP_FLOOD DoS protection applies to offloaded sessions. If it is disabled, UDP_FLOOD and ICMP_FLOOD DoS protection will not apply to offloaded sessions.

Not all the traffic can be offloaded to the NPUs for DoS policy anomalies. For more information, see DoS policy hardware acceleration.

The DoS policy and anomalies are following the same logic and are configured the same way as in the hyperscale example above. HPE is deployed/configured in a similar way as in the hyperscale use case.

General Diagnose Commands

The following diagnose commands can be used to troubleshoot ongoing issues.

To get overview of the overall system performance status you can use the `get sys performance status` command. See Technical Tip: Explaining the 'get system performance status' output:

```
get system performance status | grep 'HW-setup'
Average HW-setup sessions: 4 sessions in last 1 minute, 4 sessions in last 10 minutes, 4
sessions in last 30 minutes
```

If you want to capture traffic on the hyperscale FortiGate, you can use the `diagnose npu-sniffer` command. See NP7 packet Sniffer:

```
diagnose npu sniffer filter intf port21
diagnose npu sniffer filter dir 2
diagnose npu sniffer start
diagnose sniffer packet npudbg
```

To get session information you can use the diagnose sys npu-session command, see Displaying information about NP7 hyperscale firewall hardware sessions:

```
diagnose sys npu-session list 44
session info: proto=6 proto_state=01 duration=64721 expire=0 timeout=3600 flags=00000000
sockflag=00000000 sockport=0 av_idx=0 use=1
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=255/255
state=new f18
statistic(bytes/packets/allow_err): org=3620/40/0 reply=0/0/0 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=22->23/0->0 gwy=10.100.200.1/10.160.21.191
hook=post dir=org act=snat 192.168.10.12:49698->52.230.222.68:443(10.3.3.5:5128)
hook=pre dir=reply act=dnat 52.230.222.68:443->10.3.3.5:5128(192.168.10.12:49698)
pos/(before,after) 0/(0,0), 0/(0,0)
misc=0 policy_id=0 auth_info=0 chk_client_info=0 vd=0
serial=000163ff tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id = 00000000 ngfwid=n/a
dd_type=0 dd_mode=0
setup by offloaded-policy: origin=native
O: npid=255/0, in: OID=76/VID=0, out: NHI=77/VID=0
R: npid=0/0, in: OID=0/VID=0, out: NHI=0/VID=0
```

Also, `diagnose sys npu-session` can be used with filters/filter-options.

Use `diagnose sys npu-session stat` to show stats for IPv4 NP7 hardware sessions after adding an IPv4 filter:

```
diagnose sys npu-session stat verbose 44
misc info: session_count=10000 tcp_session_count=10000 udp_session_count=0
snat_count=10000 dnat_count=0 dual_nat_count=0
3T_hit_count=0 accounting_enabled_count=0
TCP sessions:
10000 in ESTABLISHED state
Session filter:
vd: 2
sintf: 10
proto: 6-6
3 filters
```

diagnose npu np7 cgmac-stats 0 displays the TX/RX/Error counters. Error counters can also be displayed when using action:

`0|b|brief`: Show non-zero counters,
`1|v|verbose`: Show all the counters,
`2|c|clear`: Clear counters

If **`CG_FULL`** indicates a different value than 0, this means that congestion is occurring, and the corresponding packets cannot be sent to further internal modules of NP7.

When a packet enters the NPU, the complete packet is first copied to a central buffer called Packet Buffer (PBUF). At a second step, a Packet Descriptor (compiles the packet L2, L3 & L4 Headers only). If for some reason packet buffer or Packet Description Queue is not freed when packet has left the NP or has been dropped, a **PBA leak occurs**.

In case of packet burst, or busy module, packet descriptors are likely to increase in the queues. When a queue gets full, packets will be dropped. Such drops are accounted in the NP 'drop table' of the Drop Counter Engine (DCE) available with command **`diagnose npu np7 dce-drop-all`**.

Another diagnose command is `diagnose npu np7 pmon`, which provides quite good NPU performance monitoring is the **PMON**. This tool can be used to get the NPU modules load information, for the EIF Ethernet interfaces (in the NPU).

The interfaces load is provided as a % of usage (last column). The Ingress flow and Egress flow load are separated (_ IGR and _EGR):

```
diagnose npu np7 pmon 0 v

[NP7_0]
Index Name            Counter    Sample_ver Usage%
----- --------------- ---------- ---------- ------
0     EIF_IGR0        7           0          1
1     EIF_IGR1        0           0          0
2     EIF_IGR2        14          0          1
3     EIF_IGR3        0           0          0
4     EIF_EGR0        0           0          0
5     EIF_EGR1        0           0          0
6     EIF_EGR2        0           0          0
7     EIF_EGR3        0           0          0
8     EIF_IGR4        0           0          0
9     EIF_IGR5        0           0          0
10    EIF_IGR6        7           0          1
11    EIF_IGR7        7           0          1
12    EIF_EGR4        0           0          0
13    EIF_EGR5        0           0          0
14    EIF_EGR6        0           0          0
15    EIF_EGR7        0           0          0
../..
241   L2P_EIF0        10          0          1
242   L2P_EIF1         7          0           1
```

IP Pool Diagnose commands

You can also display the hyperscale NAT IP pool usage using the `diagnose firewall ippool` command, see Displaying IP pool usage information commands from a hyperscale firewall VDOM. The output is including client IP addresses, PBA blocks, and public IP addresses currently in use.

```
diagnose firewall ippool {list {pba | nat-ip | user} | stats}
diagnose firewall ippool {list {pba | nat-ip | user} | stats | get-priv | get-pub | get-
pub6}

diagnose firewall ippool stats
ippool stats:
Total 6 ippools are allocated.
Total 0 client host is online.
Total 0 natip is allocated.
```

```
Total 0 PBA is allocated.
Approximate 0 PBA is allocated in 1 second before.

NPU ippool stats:
Total 3 client hosts are online.
Total 1 natip is allocated.
Total 3 PBA(s) are allocated.
Approximate 0 PBA is allocated in 1 second before.
```

Note that the the first section of the stats will be kernel pools (for traffic that is established in CPU) and the second part **NPU ippool stats** will be pool statistics for sessions established by the the NPUs.

`diagnose firewall ippool list` displays the names, configuration details and current usage information for all of the CGN and non-CGN IP pools in the current VDOM. For CGN IP pools that have been added to hyperscale firewall policies, IP pool usage information consists of two parts:

- Kernel CGNAT IP pool usage information (basically placeholder information that doesn't represent actual CGN IP pool usage).
- Hyperscale CGNAT IP pool usage information).

```
diagnose firewall ippool list
list ippool info:(vf=cgn-hw1)
ippool cgn-pool1: id=1, block-sz=64, num-block=8, fixed-port=no, use=2
ip-range=203.0.113.2-203.0.113.3 start-port=5117, num-pba-per-ip=944
clients=0, inuse-NAT-IPs=0
total-PBAs=1888, inuse-PBAs=0, expiring-PBAs=0, free-PBAs=100.00%
allocate-PBA-times=10, reuse-PBA-times=0
grp=cgn_pool_grp1, start-port=5117, end-port=65530
npu-clients=1, npu-inuse-NAT-IPs=1, total-NAT-IP=0
npu-total-PBAs=0, npu-inuse-PBAs=16/0, npu-free-PBAs=0.00%/-nan%
npu-tcp-sess-count=1024, npu-udp-sess-count=0
ippool cgn-pool2: id=2, block-sz=64, num-block=8, fixed-port=no, use=2
ip-range=203.0.113.4-203.0.113.5 start-port=5117, num-pba-per-ip=944
clients=0, inuse-NAT-IPs=0
total-PBAs=1888, inuse-PBAs=0, expiring-PBAs=0, free-PBAs=100.00%
allocate-PBA-times=0, reuse-PBA-times=0
grp=cgn_pool_grp1, start-port=5117, end-port=65530
npu-clients=1, npu-inuse-NAT-IPs=1, total-NAT-IP=0
npu-total-PBAs=0, npu-inuse-PBAs=16/0, npu-free-PBAs=0.00%/-nan%
npu-tcp-sess-count=1024, npu-udp-sess-count=0
```

The IP pools information related to the pool id, the block size and number of blocks, if fixed port is configured or not for that NAT pool, the public IP range and starting port, number of port blocks per public IP, clients and in use NAT IP.

Note the example with first values relate to the kernel NAT and the second values, after `grp=` relate to the NAT in the network processor. This is required because some of the sessions cannot be established in NPU but in CPU.

The general information related to the pool usage is represented by IP range **`ip-range=`**, start port - **`start-port=`**, end port **`**end-port=`**, the number of clients using this pool **`npu-clients=`**, the IP addresses in use **`npu-inuse-NAT-IPs=`**, the total IP addresses used for translations **`total-NAT-IP=, npu-total-PBAs=`**. Also the `npu-inuse-PBAs=16/0`,represent the port blocks used by TCP and UDP sessions and `npu-free-PBAs=0.00%/-nan%`the free blocks session utilization percentage (the first is TCP and the second UDP).
The session count is represented by **`npu-tcp-sess-count=`** and **`npu-udp-sess-count=`**.

If the IP pool has not been added to a firewall policy (hyperscale VDOM), then only the kernel firewall information will be shown in the output of the `diagnose firewall ippool list` command above.

If pools are configured in a group, the grouped IP pools information is the same as for individual IP pools, except that the grp field includes an IP pool group name (as in the example above). The information displayed for each IP pool in the group is actually the usage information for the entire IP pool group and not for each individual IP pool in the group.

If you want to get overview of the ongoing translations with Port Block Allocation, (showing the client IP with the NAT IP and allocated port blocks) you can use `diagnose firewall ippool list pba` command:

```
diagnose firewall ippool list pba
user 20.0.0.2: 192.168.215.100 34685-35388, idx=42, use=1
user 20.0.0.2: 192.168.215.100 49469-50172, idx=63, use=1
user 20.0.0.3: 192.168.215.100 50173-50876, idx=64, use=1
Total pba in NP: 3
```

The `idx=` represent the port block index and `use=1` is a reference counter in the kernel and the `Total pba in NP` is the amount of accelerated port blocks.

When the `diagnose firewall ippool list` command is used with `user`, the client IP is represented with the number of blocks allocated and the number of users in the network processor:

```
diagnose firewall ippool list user

User-IP 100.64.0.2: pba=1, use=1
User-IP 100.64.0.3: pba=1, use=1
User-IP 100.64.0.4: pba=1, use=1
User-IP 100.64.0.5: pba=1, use=1
User-IP 100.64.0.8: pba=1, use=1
User-IP 100.64.0.9: pba=1, use=1
…
User-IP 100.64.3.229: pba=1, use=1
User-IP 100.64.3.241: pba=1, use=1
User-IP 100.64.3.252: pba=1, use=1
User-IP 100.64.3.253: pba=1, use=1
Total user in NP: 218
```

The `pba` indicates the number of port blocks assigned to user and `use` is a kernel reference.

Further, the same diagnose command can be used with `list nat-ip`:

```
diagnose firewall ippool list nat-ip
NAT-IP 203.0.113.9: pba=256, use=3
Total nat-ip in NP: 1
```

Where the NAT IP is public IP address from the NAT pool, `pba=265` represents the number of port blocks for that NAT-IP and `use` is the number of port blocks in use.

The following command query the mapping of the running traffic with either private or public IP:

```
# diagnose  firewall ippool get-priv <public-ip> <public-port>
# diagnose  firewall ippool get-pub <private-ip>

diagnose firewall ippool get-priv 203.0.113.9
Query public IP 203.0.113.9
np-0 policy-3 pool-33(PBA) private IP: 100.64.0.248 port: 5118-5181
np-0 policy-3 pool-33(PBA) private IP: 100.64.0.249 port: 5182-5245
np-0 policy-3 pool-33(PBA) private IP: 100.64.0.254 port: 5246-5309
np-0 policy-3 pool-33(PBA) private IP: 100.64.0.255 port: 5310-5373

diagnose firewall ippool get-priv 203.0.113.9 5118
```

```
Query public IP 203.0.113.9, port 5118
np-0 policy-3 pool-33(PBA) private IP: 100.64.0.248 port: 5118-5181
```

The `np-0` represents the NPU index, the `policy-3` is the Firewall policy ID, `pool-33(PBA)` is the pool ID on that NPU, `private IP: 100.64.0.248` is the client IP, and `port: 5118-5181` is the public port block range.

Fixed Allocation IP Pool Calculation

The `diagnose firewall ippool-fixed-range list natip` command provides the mapping between internal and external IP addresses and port-ranges for fixed allocation IP pools:

```
diagnose firewall ippool-fixed-range list natip 150.0.0.1
ippool name=FixedPort, ip shared num=656, port num=92
internal ip=10.1.0.1, nat ip=150.0.0.1, range=5117~5208
.
.
internal ip=10.1.2.144, nat ip=150.0.0.1, range=65377~65468
```

The output is self explanatory, however `ippool name` is the pool name, `ip shared num` is the amount of clients behind the NAT IP and `num` is the amount of ports provided per customer.

Displaying PRP/NAT pool resources per NPU

PRP is the Port Resource Pool module in the NP7 processor and it is managing the NAT pool resources. You can check the NAT pool resources on each NPU using the `fnsysctl` utility, which more detail about the allocated NAT pool per NPU and the current usage:

```
fnsysctl cat /proc/net/np7/np7_2/prp
tx-cmd:        00000000  tx-cmd_err:    00000000
rx-rsp:        00000000  rx-rsp_err:    00000000
rx-unsolicited:00000000
2 os-pools, 2 hw-pools, 2 ranges:

    pool-id=13, type=PBA(:), vdom=500, pool_name=pba:
    alarm=(80, 100)
    Usage: CLIENT=0, IP=0, PBA=(0,0)
    pub-ranges(1):
    id=13, 95.109.101.156~95.109.101.157, 2 IP's
    port_start=1024, port_end=62463, per_np_rsc=3/3

    pool-id=14, type=PBA(:), vdom=500, pool_name=nat64_pba:
    alarm=(80, 100)
    Usage: CLIENT=0, IP=0, PBA=(0,0)
    pub-ranges(1):
    id=14, 172.16.101.2~172.16.101.3, 2 IP's
    port_start=5117, port_end=65404, per_np_rsc=117/117

add     : 7             0
del     : 5             0
deasy   : 5             0
```

```
ipadd   : 6833          6
ipdel   : 6             0
tcpalm  : 47            47
poolchk : 817           817

ent = 0
```

For CGNAT the internal traffic distribution requires src-ip hashing, which setting will cause sessions to be distributed by source IP address to all available NPUs in the system. With that configuration applied a session (and its data) from specific source IP address will be processed by the same NP7 processor.

On hyperscale systems the NAT pools are divided across all available NPUs equally, which has impact of the minimal port block size (with PBA NAT pool) of 256 on systems with single NPU, for two/three-NPU based systems, the block size can be 128, for more than three NPU systems, the minimum block size is 64.

Another configuration aspect with NP7 is the size of client IP range and public IP range, which is currently limited to 64K in NP7 hardware ip-range table. In FortiOS 7.0.6 and later when the client and public IP ranges are bigger than 64 the system divides the larger ranges into smaller ones to be used by each NP.

IP Pool Statistics SNMP Monitoring

The FortiGate MIB file can be obtained from Customer Service & Support (https://support.fortinet.com), or directly from the appliance. The FORTINET-FORTIGATE-MIB file provides definitions for the FortiGate CGNAT devices and in particular the specific IP pools OID - Firewall IP pool statistics table:

```
fgFwIppStatsEntry OBJECT-TYPE
    SYNTAX      FgFwIppStatsEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "An entry containing an ippool statistics."
    INDEX       { fgFwIppStatsStartIp, fgFwIppStatsEndIp }
    ::= fgFwIppStatsTable 1 }

FgFwIppStatsEntry ::= SEQUENCE {
    fgFwIppStatsName          DisplayString,
    fgFwIppStatsType          DisplayString,
    fgFwIppStatsStartIp       IpAddress,
    fgFwIppStatsEndIp         IpAddress,
    fgFwIppStatsTotalSessions Gauge32,
    fgFwIppStatsTcpSessions   Gauge32,
    fgFwIppStatsUdpSessions   Gauge32,
    fgFwIppStatsOtherSessions Gauge32
}
fgFwIppStatsName OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Name of the ippool."
    ::= { fgFwIppStatsEntry 1 }

    fgFwIppStatsType OBJECT-TYPE
```

```
    SYNTAX      DisplayString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
      "Type of the ippool."
    ::= { fgFwIppStatsEntry 2 }

fgFwIppStatsStartIp OBJECT-TYPE
    SYNTAX      IpAddress
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Startip of the ippool."
    ::= { fgFwIppStatsEntry 3 }

fgFwIppStatsEndIp OBJECT-TYPE
    SYNTAX      IpAddress
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "Endip of the ippool."
    ::= { fgFwIppStatsEntry 4 }

fgFwIppStatsTotalSessions OBJECT-TYPE
    SYNTAX      Gauge32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Total session number of the ippool."
    ::= { fgFwIppStatsEntry 5 }

    fgFwIppStatsTcpSessions OBJECT-TYPE
    SYNTAX      Gauge32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Tcp session number of the ippool."
    ::= { fgFwIppStatsEntry 6 }

fgFwIppStatsUdpSessions OBJECT-TYPE
    SYNTAX      Gauge32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Udp session number of the ippool."
    ::= { fgFwIppStatsEntry 7 }

fgFwIppStatsOtherSessions OBJECT-TYPE
    SYNTAX      Gauge32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Other session number of the ippool."
    ::= { fgFwIppStatsEntry 8 }

fortinet.fnFortiGateMib.fgFirewall.fgFwGtp
```

The fgTrapPoolUsage SNMP trap is also available for IP pool utilization:

```
fgTrapPoolUsage NOTIFICATION-TYPE
    OBJECTS { fnSysSerial, sysName, fgFwIppTrapType, fgFwIppStatsName,
       fgFwIppStatsGroupName, fgFwTrapPoolUtilization, fgFwIppTrapPoolProto } STATUS current
       DESCRIPTION
        "A trap for ippool." ::= { fgTrapPrefix 1401 }
```

Policy statistics via SNMP

You can use the following MIB fields to send SNMP queries for hyperscale firewall policy information. These MIB fields support IPv4 and IPv6 hyperscale firewall policies and are available from the latest FORTINET-FORTIGATE-MIB.mib.

Path: FORTINET-FORTIGATE-MIB:fortinet.fnFortiGateMib.fgFirewall.fgFwPolicies.fgFwPolTables

NPU monitoring via SNMP

Hyperscale enabled systems can be monitored via SNMP and the corresponding OID is available in the FORTINET-FORTIGATE-MIB file:

```
fortinet.fnFortiGateMib.fgNPU

fgNPU OBJECT IDENTIFIER
    ::= { fnFortiGateMib 20 }

fortinet.fnFortiGateMib.fgNPU.fgNPUInfo

FgNPUIndex ::= TEXTUAL-CONVENTION
    DISPLAY-HINT "d"
    STATUS      current
    DESCRIPTION
        "data type for NPU indexes"
    SYNTAX      Integer32 (0..255)

fgNPUInfo OBJECT IDENTIFIER
    ::= { fgNPU 1 }

fgNPUNumber OBJECT-TYPE
    SYNTAX      Integer32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of NPUs in NPUTable"
    ::= { fgNPUInfo 1 }

fgNPUName OBJECT-TYPE
    SYNTAX      DisplayString (SIZE(0..64))
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
```

```
        "Name of the NPU"
    ::= { fgNPUInfo 2 }

fgNPUDrvDriftSum OBJECT-TYPE
    SYNTAX      Integer32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Summation of driver session drift counters(fgNPUDrvDrift)"
    ::= { fgNPUInfo 3 }

fortinet.fnFortiGateMib.fgNPU.fgNPUTables

fgNPUTables OBJECT IDENTIFIER
    ::= { fgNPU 2 }

fortinet.fnFortiGateMib.fgNPU.fgNPUTables.fgNPUTable

fgNPUTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF FgNPUEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A table of NPUs in the device"
    ::= { fgNPUTables 1 }

fgNPUEntry OBJECT-TYPE
    SYNTAX      FgNPUEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "An entry containing information applicable
         to a particular NPU"
    INDEX       { fgNPUEntIndex }
    ::= { fgNPUTable 1 }

FgNPUEntry ::= SEQUENCE {
    fgNPUEntIndex    FgNPUIndex,
    fgNPUSessionTblSize Gauge32,
    fgNPUSessionCount Gauge32,
    fgNPUDrvDrift Integer32
}

fgNPUEntIndex OBJECT-TYPE
    SYNTAX      FgNPUIndex (0..255)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "NPU index used to uniquely identify NPU in the system."
    ::= { fgNPUEntry 1 }

fgNPUSessionTblSize OBJECT-TYPE
    SYNTAX      Gauge32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
```

```
        "Size of session table in the NPU"
    ::= { fgNPUEntry 2 }

fgNPUSessionCount OBJECT-TYPE
    SYNTAX      Gauge32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Number of active sessions in the NPU"
    ::= { fgNPUEntry 3 }

fgNPUDrvDrift OBJECT-TYPE
    SYNTAX      Integer32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Difference of session count between driver and hardware"
    ::= { fgNPUEntry 4 }
```

The following SNMP OIDs can be used for polling critical Port Block Allocations (PBAs) IP pool statistics including total PBAs, in use PBAs, expiring PBAs, and free PBAs:

```
.fgFwIppStatsTotalPBAs 1.3.6.1.4.1.12356.101.5.3.2.1.1.9
.fgFwIppStatsInusePBAs 1.3.6.1.4.1.12356.101.5.3.2.1.1.10
.fgFwIppStatsExpiringPBAs 1.3.6.1.4.1.12356.101.5.3.2.1.1.11
.fgFwIppStatsFreePBAs 1.3.6.1.4.1.12356.101.5.3.2.1.1.12
```

SNMP Logging Monitoring

Logging on NP7 systems can be monitored via SNMP and the corresponding OID is available in the FORTINET-FORTIGATE-MIB file:

```
fortinet.fnFortiGateMib.fgLog.fgLogDevices.fgLogDeviceTable

fgLogDeviceTable OBJECT-TYPE
    SYNTAX       SEQUENCE OF FgLogDeviceEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "Table of log devices on the fortigate"
       ::= { fgLogDevices 1 }

fgLogDeviceEntry OBJECT-TYPE
    SYNTAX      FgLogDeviceEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "An entry containing information about a specific log device"
    INDEX       { fgLogDeviceEntryIndex }
           ::= { fgLogDeviceTable 1 }
        FgLogDeviceEntry ::= SEQUENCE {
        fgLogDeviceEntryIndex     FgLogDeviceIndex,
        fgLogDeviceEnabled        Integer32,
        fgLogDeviceName           DisplayString,
```

```
        fgLogDeviceSentCount      Counter32,
        fgLogDeviceRelayedCount   Counter32,
        fgLogDeviceCachedCount    Gauge32,
        fgLogDeviceFailedCount    Counter32,
        fgLogDeviceDroppedCount   Counter32
}
fgLogDeviceEntryIndex OBJECT-TYPE
    SYNTAX      FgLogDeviceIndex (0..255)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Log device index in the list of reported log devices"
    ::= { fgLogDeviceEntry 1 }

fgLogDeviceEnabled OBJECT-TYPE
    SYNTAX      Integer32
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Indicated whether the log device is enabled"
    ::= { fgLogDeviceEntry 2 }

fgLogDeviceName OBJECT-TYPE
    SYNTAX      DisplayString (SIZE(0..64))
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Name of the log device"
    ::= { fgLogDeviceEntry 3 }

fgLogDeviceSentCount OBJECT-TYPE
    SYNTAX       Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "Number of logs which have been sent"
    ::= { fgLogDeviceEntry 4 }

fgLogDeviceRelayedCount OBJECT-TYPE
    SYNTAX       Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "Number of logs which have been relayed"
    ::= { fgLogDeviceEntry 5 }

fgLogDeviceCachedCount OBJECT-TYPE
    SYNTAX       Gauge32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "Number of logs which are cached for later sending"
    ::= { fgLogDeviceEntry 6 }

fgLogDeviceFailedCount OBJECT-TYPE
    SYNTAX       Counter32
```

```
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "Number of logs which have failed to send"
    ::= { fgLogDeviceEntry 7 }

fgLogDeviceDroppedCount OBJECT-TYPE
    SYNTAX       Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "Number of logs which have been dropped"
    ::= { fgLogDeviceEntry 8 }
```

Netflow troubleshooting

If data are not seen on the NetFlow collector after it has been configured, use the following sniffer commands to verify if the FortiGate and the collector are communicating:

By collector port:

```
diagnose sniffer packet 'port <collector-port>' 6 0 a
```

By collector IP address:

```
diagnose sniffer packet 'host <collector-ip>' 6 0 a
```

NetFlow uses the sflow daemon. The current NetFlow configuration can be viewed using test level 3 or 4:

```
diagnose test application sflowd 3
diagnose test application sflowd 4
Netflow Cache Stats:
vdoms=1 Collectors=1 Cached_intf=2 Netflow_enabled_intf=1 Live_sessions=0 Session
cache max count:71950
```

REST API for Monitoring

The FortiOS REST API offers monitoring functionality on the NP7 based FortiGate appliances. Using the monitoring API you can retrieve dynamic data related to system resources (NPU) and NAT pools. For details how to generate API token and make API requests using the web browser, place a request in the correct VDOM, please refer to FNDN.

The following API request is providing further details about NPU sessions in interval of 10 min in the root VDOM:

```
https://ip_address/api/v2/monitor/system/resource/usage?resource=npu_
session&scope=global&interval=10-min

{
"http_method":"GET",
"results":{
  "npu_session":[
    {
      "current":0,
      "historical":{
```

```
        "10-min":{
          "values":[
            [ ...
            ],
            "max":0,
            "min":0,
            "average":0,
            "start":1684228119000,
            "end":1684228697000
     }
    }
   }
 ]
},
"vdom":"root",
"path":"system",
"name":"resource",
"action":"usage",
"status":"success",
"serial":"...",
"version":"...",
"build":...
}
```


This API request provides the firewall policy details, including the CGN bytes and CGN packets:


```
https://ip_address/api/v2/monitor/firewall/ippool?vdom=CGNAT-hw01
{
  "http_method":"GET",
  "results":[
    ...
    {
      "policyid":1,
      "uuid":"d2833834-c4f7-51ed-5358-15b4f5ed21a5",
      "active_sessions":0,
      "bytes":305572575867,
      "packets":1471982427,
      "software_bytes":278324263443,
      "software_packets":1212072212,
      "asic_bytes":27248312424,
      "asic_packets":259910215,
      "cgn_bytes":0,
      "cgn_packets":0,
      "cgn_hit_count":0,
      "nturbo_bytes":0,
      "nturbo_packets":0,
      "last_used":1679316152,
      "first_used":1679078695,
      "hit_count":128736108,
      "session_last_used":1679079361,
      "session_first_used":1679078695,
      "session_count":0,
      "1_week_ipv4":{
       "hit_count":[
          ...
       ],
       "bytes":[
```

```
            ...
         ],
         "packets":[
            ...
         ],
         "software_bytes":[
            ...
         ],
         "software_packets":[
            ...
         ],
         "asic_bytes":[
            ...
         ],
         "asic_packets":[
            ...
         ],
         "nturbo_bytes":[
            ...
         ],
         "nturbo_packets":[
            ...
         ]
      }
   }
],
"vdom":"CGNAT-hw01",
"path":"firewall",
"name":"policy",
"action":"",
"status":"success",
"serial":"...",
"version":"..",
"build":..
}
```


The following API call provides the NAT pool details:


```
https://ip_address/api/v2/monitor/firewall/ippool??vdom=CGNAT-hw01
{
  "http_method":"GET",
  "results":{
    "cgn_44_pba_pool_1":{
      "name":"cgn_44_pba_pool_1",
      "blocks":8,
      "block_size":4096,
      "fixed_port":false,
      "pba_per_ip":15,
      "group_name":"CGN_44_PBA_1",
      "natip_total":65535
    },
    "cgn_44_pba_pool_2":{
      "name":"cgn_44_pba_pool_2",
      "blocks":8,
      "block_size":4096,
      "fixed_port":false,
      "pba_per_ip":15,
```

```
        "group_name":"CGN_44_PBA_2",
        "natip_total":65535
      }
    },
    "vdom":"CGNAT-hw01",
    "path":"firewall",
    "name":"ippool",
    "action":"",
    "status":"success",
    "serial":"FG181FTK20900033",
    "version":"v7.0.11",
    "build":489
}
```


Upgrading Hyperscale Systems

Upgrading the software from older to newer releases (starting with FortiOS 7.0.5) does not support upgrading NAT64 and NAT46 firewall policies or VIP46 and VIP64 firewall policies. After upgrading, you should manually reconfigure all NAT64 and NAT46 firewall policies and all VIP64 and VIP46 firewall policies.

Having the old configuration reviewed and stored before the upgrade can save you lot of time of troubleshooting and outages. After FortiOS upgrade to newer releases (starting with 7.0.5) the NP queue priority configuration may be incorrect.

The default NP queue priority configuration should provide optimal performance, however empty or incorrect NP queue priority configuration can cause BGP flapping when a lot of IP traffic and/or non-SYN TCP traffic is processed by the CPU.

Here the default NP queue configuration:

```
config system npu
   config np-queues
      config ethernet-type
         edit "ARP"
            set type 806
            set queue 9
         next
         edit "HA-SESSYNC"
            set type 8892
            set queue 11
         next
         edit "HA-DEF"
            set type 8890
            set queue 11
         next
         edit "HC-DEF"
            set type 8891
            set queue 11
         next
         edit "L2EP-DEF"
            set type 8893
            set queue 11
         next
         edit "LACP"
            set type 8809
```

```
      set queue 9
   next
end
config ip-protocol
   edit "OSPF"
      set protocol 89
      set queue 11
   next
   edit "IGMP"
      set protocol 2
      set queue 11
   next
   edit "ICMP"
      set protocol 1
      set queue 3
   next
end
config ip-service
   edit "IKE"
      set protocol 17
      set sport 500
      set dport 500
      set queue 11
   next
   edit "BGP"
      set protocol 6
      set sport 179
      set dport 179
      set queue 9
   next
   edit "BFD-single-hop"
      set protocol 17
      set sport 3784
      set dport 3784
      set queue 11
   next
   edit "BFD-multiple-hop"
      set protocol 17
      set sport 4784
      set dport 4784
      set queue 11
   next
   edit "SLBC-management"
      set protocol 17
      set dport 720
      set queue 11
   next
   edit "SLBC-1"
      set protocol 17
      set sport 11133
      set dport 11133
      set queue 11
   next
   edit "SLBC-2"
      set protocol 17
      set sport 65435
      set dport 65435
```

```
        set queue 11
    end
```

Also the hyperscale policies (from older FortiOS 6.2 and 6.4) have been removed. Reworking the policy would require precise attention.

Enriched logging using RSSO

The main benefit of the NP7 platforms is the NP7 accelerated session setup and log generation. The CGN logs generated by the NP7 systems are at high-rate and depending on the CGN configuration also, high-volume. The log processing is quite a challenging and expensive task, especially when logs from multiple sources must be correlated to get the private/public IP address with the subscriber identity.

Note that RSSO is not supported in a VDOM with configured hyperscale support.

If for any reason enriched logs are required, this can be achieved, using dedicated VDOM without configured hyperscale support. RSSO is supported on non-hyperscale systems though.

On native FortiOS, RSSO can be used together with CGNAT and the logs will be enriched with the user identity information.

www.fortinet.com