Hyperscale Firewall Guide

FortiOS 7.0.19

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

FORTINET

January 30, 2026
FortiOS 7.0.19 Hyperscale Firewall Guide
01-7019-688354-20260130

TABLE OF CONTENTS

Change log

Date	Change description
January 30, 2026	FortiOS 7.0.19 document release.
November 5, 2025	FortiOS 7.0.18 document release.
October 3, 2025	The following information about hyperscale FortiOS and asymmetric sessions added to Hyperscale firewall 7.0.19 incompatibilities and limitations. • In an FGSP configuration, if FGSP session synchronization is enabled, software sessions can support asymmetric session paths. To support asymmetric session paths, each FGSP cluster member synchronizes session state changes to other peers in the FGSP cluster. This configuration is not recommended because FGSP session synchronization can cause overall performance reduction. • Asymmetric session paths are not supported for hardware sessions.
August 14, 2025	Added more information about how hyperscale deny log messages are sent to NetFlow or syslog servers and not to FortiAnalyzer, see Hardware logging on page 48.
July 31, 2025	Hyperscale firewall 7.0.19 incompatibilities and limitations on page 13 has been updated with the following: • Firewall virtual IP (VIP) features `srcfltr`, `srcintf-fltr`, `service`, `arp-reply`, `nat-source-vip`, and `portforwarding` are not supported by hyperscale firewall policies even though they are visible from the CLI or GUI when configuring IPv4 and IPV6 firewall VIPs in a hyperscale firewall VDOM.
January 14, 2025	FortiOS 7.0.17 document release.
October 22, 2024	FortiOS 7.0.16 document release.
September 26, 2024	Changes to Hyperscale firewall policy engine limitations and mechanics on page 44. Moved the former section "CGN resource allocation firewall policy source and destination address limits" to Per hyperscale policy limits on page 45. The number of firewall policies that can be added to a Hyperscale firewall VDOM is limited to 15,000. For more information, see About the 15,000 policy per hyperscale VDOM limit on page 44. Hyperscale firewall VDOMs do not support the FortiOS Internet Service Database (ISDB), IP Reputation Database (IRDB), and IP Definitions Database (IPDB) features, see Hyperscale firewall 7.0.19 incompatibilities and limitations on page 13.
August 23, 2024	If your FortiGate has multiple NP7 processors, depending on whether or not you are enabling EIF in hyperscale firewall policies, you may want to use the `nss-threads-option` of the `config system npu` command to optimize performance, see Optimizing NP7 network session setup (NSS) engine performance on page 79. You should not operate DoS protection in monitor mode on a FortiGate licensed for hyperscale firewall, for more information in this limitation, see Hyperscale firewall 7.0.19 incompatibilities and limitations on page 13.

Date	Change description
May 28, 2024	Corrected information about firewall VIP support for hyperscale firewall VDOMs in Hyperscale firewall 7.0.19 incompatibilities and limitations on page 13.
April 4, 2024	FortiOS 7.0.15 document release.
March 20, 2024	Changes to Recommended NP7 traffic distribution for optimal CGNAT performance on page 66. New section Carrier-Grade NAT Architecture Guide on page 15. Per-session hardware logging is not compatible with session-count DoS anomalies, see Hyperscale firewall 7.0.19 incompatibilities and limitations on page 13 for more information.
February 8, 2024	FortiOS 7.0.14 document release.
December 22, 2023	More information about the FortiGate 4800F and 4801F and hyperscale hardware logging, see Enabling hyperscale firewall features on page 17 and Configuring hardware logging on page 49. More information added to Hardware logging on page 48 and Configuring hardware logging on page 49.
October 26, 2023	FortiOS 7.0.13 document release.
August 25, 2023	New section: Hyperscale and standard FortiOS CGNAT feature comparison on page 26.
June 27, 2023	Added information about hardware logging sending multiple session start log messages if `log-processor` is set to `hardware` and `log-mode` is set to `per-session` to Hyperscale firewall 7.0.19 incompatibilities and limitations on page 13.
June 8, 2023	FortiOS 7.0.12 document release.
March 16, 2023	FortiOS 7.0.11 document release. Added more information about the NP7 `hash-config` option to Recommended NP7 traffic distribution for optimal CGNAT performance on page 66.
March 7, 2023	NP7 hardware logging must use interfaces connected to NP7 processors to communicate with the remove log servers. This information was added to Configuring hardware logging on page 49.
February 23, 2023	FortiOS 7.0.10 document release. Corrections to SNMP queries for hardware session counts on page 69.
February 8, 2023	Improvements to Displaying IP pool usage information on page 86.
November 22, 2022	FortiOS 7.0.9 document release.
November 21, 2022	Corrections to Creating hyperscale firewall VDOMs on page 16.
October 13, 2022	FortiOS 7.0.8 document release. For information about new features, see What's new for hyperscale firewall for FortiOS 7.0.8 on page 11.
October 6, 2022	FortiOS 7.0.7 document release. More information and explanation added to Hyperscale firewall policy engine limitations and mechanics on page 44.
August 16, 2022	New sections: FGSP HA hardware session synchronization on page 64 and Basic FGSP HA hardware session synchronization configuration example on page 64. Changes to FGCP HA hardware session synchronization on page 60.
August 9, 2022	Changes to Configuring FGCP HA hardware session synchronization on page 61.

Date	Change description

	New section:
	• Optimizing FGCP HA hardware session synchronization with data interface LAGs on page 62.
	Revised information about using the FortiGate-4200F/4201F and 4400F/4400F HA1, HA2, AUX1 and AUX2 interfaces. Using the following command is no longer recommended. Instead see the new sectionRecommended interface use for an FGCP HA hyperscale firewall cluster on page 63.
	`config system npu`
	`  config port-path-option`
	`    set ports-using-npu {ha1 ha2 aux1 aux2}`
	`  end`

July 21, 2022	Improvements to Hyperscale firewall VDOM session timeouts on page 74. New section Session timeouts for individual hyperscale policies on page 75.
June 15, 2022	Added information about how FortiOS 7.0.19 hyperscale policies are no longer separated from normal firewall policies and how hyperscale firewall policies are converted during the upgrade process to Upgrading hyperscale firewall features to FortiOS 7.0.19 on page 9.
June 14, 2022	Misc. changes throughout the document.
June 7, 2022	FortiOS 7.0.6 document release. See What's new for hyperscale firewall for FortiOS 7.0.6 on page 11. New section: Allowing packet fragments for NP7 NAT46 policies when the DF bit is set to 1 on page 80.
May 17, 2022	Added more information about the verbose option to diagnose sys npu-session stat [verbose [{44 | 66 | 64 | 46}]] on page 83.
April 5, 2022	Improvements to Displaying information about NP7 hyperscale firewall hardware sessions on page 81.
April 4, 2022	FortiOS 7.0.5 document release.

What's new

This section describes new Hyperscale firewall features for FortiOS 7.0 releases.

What's new for hyperscale firewall for FortiOS 7.0.19

Hyperscale firewall for FortiOS 7.0.19 includes the bug fixes described in Resolved issues.

Upgrading hyperscale firewall features to FortiOS 7.0.19

If your FortiGate is currently running FortiOS firmware and is licensed for hyperscale firewall features, you can follow a normal firmware upgrade process to upgrade to FortiOS 7.0.19.

If you are currently operating a FortiGate with NP7 processors without a hyperscale firewall license, you can use the upgrade path to upgrade to FortiOS 7.0.19. Once you have upgraded to 7.0.19 you can activate your hyperscale firewall license and set up your hyperscale firewall configuration.

The firmware upgrade code does not support upgrading NAT64 and NAT46 firewall policies or VIP46 and VIP64 firewall policies to 7.0.15. After upgrading, you should review all NAT64 and NAT46 firewall policies and all VIP64 and VIP46 firewall policies added prior to upgrading.

In FortiOS 7.0.19, you apply hyperscale firewall features by creating normal firewall policies in hyperscale firewall VDOMs. FortiOS 7.0.19 no longer has hyperscale firewall policies in a separate hyperscale firewall policy list, as supported by FortiOS 6.2 and 6.4.

The FortiOS 7.0.19 upgrade process converts FortiOS 6.2 and 6.4 hyperscale firewall policies to normal firewall policies and adds them to the normal policy list in their hyperscale firewall VDOMs. During the conversion process, the policy IDs of the hyperscale firewall policies may be changed when they are converted to normal firewall policies.

What's new for hyperscale firewall for FortiOS 7.0.18

Hyperscale firewall for FortiOS 7.0.18 includes the bug fixes described in Resolved issues.

What's new for hyperscale firewall for FortiOS 7.0.17

Hyperscale firewall for FortiOS 7.0.17 includes the bug fixes described in Resolved issues.

What's new for hyperscale firewall for FortiOS 7.0.16

Hyperscale firewall for FortiOS 7.0.16 includes the bug fixes described in Resolved issues.

What's new for hyperscale firewall for FortiOS 7.0.15

This section describes the new NP7 hyperscale firewall feature added to FortiOS 7.0.15.

- If your FortiGate has multiple NP7 processors, depending on whether or not you are enabling EIF in hyperscale firewall policies, you may want to use the `nss-threads-option` of the `config system npu` command to optimize performance, see Optimizing NP7 network session setup (NSS) engine performance on page 79.

Hyperscale firewall for FortiOS 7.0.15 also includes the bug fixes described in Resolved issues.

What's new for hyperscale firewall for FortiOS 7.0.14

Hyperscale firewall for FortiOS 7.0.14 includes the bug fixes described in Resolved issues.

What's new for hyperscale firewall for FortiOS 7.0.13

Hyperscale firewall for FortiOS 7.0.13 includes the bug fixes described in Resolved issues.

What's new for hyperscale firewall for FortiOS 7.0.12

Hyperscale firewall for FortiOS 7.0.12 includes the bug fixes described in Resolved issues.

What's new for hyperscale firewall for FortiOS 7.0.11

Hyperscale firewall for FortiOS 7.0.11 includes the bug fixes described in Resolved issues.

For more information about FortiOS 7.0.11 hyperscale firewall, see the FortiOS 7.0.11 Hyperscale Firewall Release Notes.

What's new for hyperscale firewall for FortiOS 7.0.10

Hyperscale firewall for FortiOS 7.0.10 includes the bug fixes described in Resolved issues.

For more information about FortiOS 7.0.10 hyperscale firewall, see the FortiOS 7.0.10 Hyperscale Firewall Release Notes.

What's new for hyperscale firewall for FortiOS 7.0.9

Hyperscale firewall for FortiOS 7.0.9 includes the bug fixes described in Resolved issues.

For more information about FortiOS 7.0.9 hyperscale firewall, see the FortiOS 7.0.9 Hyperscale Firewall Release Notes.

What's new for hyperscale firewall for FortiOS 7.0.8

This section lists the new NP7 hyperscale firewall features added to FortiOS 7.0.8.

- New FGCP HA hardware session synchronization timers, see FGCP HA hardware session synchronization timers on page 62.
- You can only use a static mode LAG as the hardware session synchronization interface, see Configuring FGCP HA hardware session synchronization on page 61.
- Hardware logging for hyperscale firewall policies that block sessions, see Hardware logging for hyperscale firewall policies that block sessions on page 58.
- View summary information about IPv4 and IPv6 sessions offloaded to NP7 processors, see diagnose sys npu-session list-brief [{44 | 46}] on page 82 and diagnose sys npu-session list-brief6 [{66 | 64}] on page 83.

For more information about FortiOS 7.0.8 hyperscale firewall, see the FortiOS 7.0.8 Hyperscale Firewall Release Notes.

What's new for hyperscale firewall for FortiOS 7.0.7

Hyperscale firewall for FortiOS 7.0.7 includes the bug fixes described in Resolved issues.

For more information about FortiOS 7.0.7 hyperscale firewall, see the FortiOS 7.0.7 Hyperscale Firewall Release Notes.

What's new for hyperscale firewall for FortiOS 7.0.6

This section lists the new NP7 hyperscale firewall features added to FortiOS 7.0.6. For new FortiOS 7.0.6 NP7 features, see What's new in FortiOS 7.0.6.

- Enhanced hyperscale firewall policy engine, see Hyperscale firewall policy engine limitations and mechanics on page 44.
- Support for using a LAG as the HA hardware session synchronization interface, see Configuring FGCP HA hardware session synchronization on page 61.
- FGSP support for HA hardware session synchronization, see FGSP HA hardware session synchronization on page 64.

- New SNMP PBA statistics fields for polling total PBAs, in use PBAs, expiring PBAs, and free PBAs, see IP pool MIB and trap fields on page 67.
- Enhanced hyperscale firewall blackhole and loopback routing, see Viewing and changing NP7 hyperscale firewall blackhole and loopback routing on page 89.
- Including user information in hardware log messages, see Include user information in hardware log messages on page 57.
- Adding event logs to hardware logging, see Adding event logs to hardware logging on page 57.

For more information about FortiOS 7.0.6 hyperscale firewall, see the FortiOS 7.0.6 Hyperscale Firewall Release Notes.

What's new for hyperscale firewall for FortiOS 7.0.5

This section lists the new NP7 hyperscale firewall features added to FortiOS 7.0.5.

- NP7 hardware logging supports multicast-mode logging, see Configuring hardware logging on page 49.

For more information about FortiOS 7.0.5 hyperscale firewall, see the FortiOS 7.0.5 Hyperscale Firewall Release Notes.

Getting started with NP7 hyperscale firewall features

This section provides an overview of FortiOS NP7 hyperscale firewall support. Hyperscale firewall features include:

- NP7 hardware session setup takes place entirely on the NP7 policy and NAT engine (called the Session Search Engine or SSE) without any involvement of the system bus or CPU. Hardware session setup is also called hardware policy offload.
- IPv4 and NAT64 firewall policies includes support for carrier-grade NAT (CGNAT) features.
- Hardware logging (syslog and IPFIX) offloads syslog or NetFlow messages for all offloaded sessions.
- Hardware session synchronization supports HA session sync for hyperscale firewall HA clusters.
- Hyperscale firewall features are enabled per VDOM.
  - Hyperscale firewall VDOMs only support hyperscale firewall policies.
  - Hyperscale firewall VDOMs do not support UTM or NGFW firewall features.
  - Hyperscale firewall VDOMs do not support Central NAT.
  - You must use a special naming convention when creating a hyperscale firewall VDOM, see Creating hyperscale firewall VDOMs on page 16 for details.

Hyperscale firewall 7.0.19 incompatibilities and limitations

Hyperscale firewall for FortiOS 7.0.19 has the following limitations and incompatibilities with FortiOS features:

- Proxy or flow based inspection is not supported. You cannot include security profiles in hyperscale firewall policies.
- Single-sign-on authentication including FSSO and RSSO is not supported. Other types of authentication are supported.
- IPsec VPN is not supported. You cannot create hyperscale firewall policies where one of the interfaces is an IPsec VPN interface.
- The number of firewall policies that can be added to a Hyperscale firewall VDOM is limited to 15,000. For more information, see About the 15,000 policy per hyperscale VDOM limit on page 44.
- Hyperscale firewall VDOMs do not support Central NAT.
- Hyperscale firewall VDOMs do not support Policy-based NGFW Mode.
- Hyperscale firewall VDOMs must be NAT mode VDOMs. Hyperscale firewall features are not supported for transparent mode VDOMs.
- Hyperscale firewall VDOMs do not support traffic shaping policies or profiles. Only outbandwidth traffic shaping is supported for hyperscale firewall VDOMs.
- Hyperscale firewall VDOMs do not support the FortiOS Internet Service Database (ISDB), IP Reputation Database (IRDB), and IP Definitions Database (IPDB) features.
- Traffic shaping with queuing using the NP7 QTM module is not compatible with carrier-grade NAT and hyperscale firewall features. See NP7 traffic shaping.
- Traffic that requires session helpers or ALGs is processed by the CPU and not by NP7 processors (for example, FTP, TFTP, SIP, MGCP, H.323, PPTP, L2TP, ICMP Error/IP-options, PMAP, TNS, DCE-RPC, RAS, and RSH). For more information, see ALG/Session Helper Support.
- Active-Active FGCP HA and FGSP do not support HA hardware session synchronization. Active-passive FGCP HA and virtual clustering do support FGCP HA hardware session synchronization.

- In an FGSP configuration, if FGSP session synchronization is enabled, software sessions can support asymmetric session paths. To support asymmetric session paths, each FGSP cluster member synchronizes session state changes to other peers in the FGSP cluster. This configuration is not recommended because FGSP session synchronization can cause overall performance reduction.
- Asymmetric session paths are not supported for hardware sessions.
- ECMP usage-based load balancing is not supported. Traffic is not directed to routes with lower spillover-thresholds.
- The Sessions dashboard widget does not display hyperscale firewall sessions.
- Interface device identification should not be enabled on interfaces that send or receive hyperscale firewall traffic.
- The `proxy` action is not supported for DoS policy anomalies when your FortiGate is licensed for hyperscale firewall features. When you activate a hyperscale firewall license, the `proxy` option is removed from the CLI of both hyperscale VDOMs and normal VDOMs.
- Access control list (ACL) policies added to a hyperscale firewall VDOM that is processing traffic may take longer than expected to become effective. During a transition period, traffic that should be blocked by the new ACL policy will be allowed.
- During normal operation, UDP sessions from protocols that use FortiOS session helpers are processed by the CPU. After an FGCP HA failover, when the UDP session helper sessions are re-established, they will not be identified as session helper sessions and instead will be offloaded to the NP7 processors.
- When operating an FGCP HA cluster with session synchronization enabled, some of the sessions accepted by an IPv4 or a NAT64 hyperscale firewall policy with an overload IP pool may not be synchronized to the secondary FortiGate. Some sessions are not synchronized because of resource conflicts and retries. The session loss rate depends on the percentage of resource retries during session setup. You can reduce the session loss by making sure the IP pool has as many IP addresses and ports as possible.
- If hardware logging is configured to send log messages directly from NP7 processors (`log-processor` is set to `hardware`) (also called log2hw) and the log server group is configured to send log messages at the start and end of each session (`log-mode` is set to `per-session`), hardware logging may send multiple session start log messages, each with a different start time. Creating multiple session start log messages is a limitation of NP7 processor hardware logging, caused by the NP7 processor creating extra session start messages if session updates occur. You can work around this issue by:
  - Setting `log-mode` to `per-session-ending`. This setting creates a single log message when the session ends. This log message records the time the session ended as well as the duration of the session. This information can be used to calculate the session start time.
  - Setting `log-processor` to `host` (also called log2host). Host hardware logging removes duplicate log start messages created by the NP7 processor. Host logging may reduce performance.
- Firewall virtual IP (VIP) features `srcfltr`, `srcintf-fltr`, `service`, `arp-reply`, `nat-source-vip`, and `portforwarding` are not supported by hyperscale firewall policies even though they are visible from the CLI or GUI when configuring IPv4 and IPV6 firewall VIPs in a hyperscale firewall VDOM.

  Even though the `arp-reply` CLI option is not supported for IPv4 and IPv6 firewall VIPs, responding to ARP requests for IP addresses in a virtual IP is supported. What is not supported is using the `arp-reply` option to disable responding to an ARP request.

- 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 `per-session` hardware logging. Session-count based DoS anomalies have `session` in their name (for example, `tcp_src_session` and `tcp_dst_session`). For information about DoS anomalies, see DoS policy.

- Because of how NP7 hyperscale hardware-based session setup logic works, you should not operate DoS protection in monitor mode (that is DoS policies with Action set to Monitor) on a FortiGate licensed for hyperscale firewall. You can enable monitor mode for debugging your DoS protection setup. But during normal operation, operating DoS protection in monitor mode can cause NP7 processors to become unresponsive when processing large amounts of traffic.

Carrier-Grade NAT Architecture Guide

The Carrier-Grade NAT Architecture Guide provides technically-focused CGNAT solution details, guidance, and reference architecture for FortiOS Hyperscale CGNAT and Kernel CGNAT solutions. The guide includes detailed information on Kernel and Hyperscale CGNAT features, as well as information on hyperscale CGNAT hardware logging, The guide also includes a Reference Architectures section that provides more details about CGNAT architectures that can be deployed and to what extent the CGNAT service can be scaled with these architectures.

Applying the hyperscale firewall activation code or license key

To activate hyperscale firewall features for your FortiGate, you must purchase a hyperscale firewall license and obtain an activation code or license key from the Fortinet Support website. You can use the following command to apply your hyperscale firewall activation code or license key to activate hyperscale firewall features for your FortiGate:

```
execute hscalefw-license {<activation-code> | <license-key>}
```

After you enter this command, the FortiGate restarts with hyperscale firewall features available. Check the Licenses dashboard widget to verify that the FortiGate has been successfully licensed for hyperscale firewall features.

If you are operating an HA cluster, all FortiGates in the cluster must have a hyperscale firewall license.

You can also use the `get system status` command to verify that your hyperscale firewall license is enabled:

```
get system status
...
Hyperscale license: Enabled
...
end
```

You can now create hyperscale firewall configurations for your FortiGate. To apply hyperscale firewall features, your FortiGate must be operating in multi VDOM mode. You cannot use the root VDOM for hyperscale firewall features. Instead you must create new hyperscale firewall VDOMs for the traffic that you want to apply hyperscale firewall features to. You can also use the root VDOM for other traffic or create other VDOMs for other traffic.

Creating hyperscale firewall VDOMs

VDOMs in which you will be enabling hyperscale firewall features must be created with a special VDOM name that also includes a VDOM ID. The VDOM ID is used by FortiOS to create a kernel VDOM_ID for the VDOM that NP7 processors use to track hyperscale firewall sessions for that VDOM.

The number of hyperscale firewall VDOMs that you can create depends on your hyperscale firewall license and is controlled by the following configuration option:

```
config system global
   set hyper-scale-vdom-num <vdom-id-num>
end
```

By default <vdom-id-num> is set to the maximum number of hyperscale VDOMs that the FortiGate is licensed for. You can manually change the <vdom-id-num> if you want to limit the number of hyperscale VDOMs that can be created. The <vdom-id-num> range is 1 to 250.

Use the following syntax to create a hyperscale firewall VDOM:

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

Where:

<name> is a string that can contain any alphanumeric upper or lower case characters and the - and _ characters. The name cannot contain spaces and you should not use -hw in the name.

<vdom-id> a VDOM ID number in the range from 1 to <vdom-id-num>. For example, if your FortiGate is licensed for 250 hyperscale firewall VDOMs, if you haven't used the hyper-scale-vdom-num option to change the number of hyperscale firewall VDOMs, <vdom-id> can be from 1 to 250. Each hyperscale firewall VDOM must have a different <vdom-id>.

When you add a new hyperscale firewall VDOM with a <vdom-id>, FortiOS calculates the kernel VDOM_ID using the following formula:

```
kernel VDOM_ID = 501 - <vdom-id>
```

If you include leading zeros in the <vdom-id>, the kernel removes them when creating the ID. So avoid using leading zeros in the <vdom-id> to keep from accidentally creating duplicate IDs.

The VDOM name, including the <string>, -hw, and <vdom-id> can include up to 11 characters. For example, the VDOM name CGN-1-hw23 is valid but CGN-1234-hw23 is too long.

When you create a new hyperscale firewall VDOM, the CLI displays an output line that includes the VDOM name followed by the kernel VDOM_ID. For example:

```
config vdom
   edit Test-hw150
   current vf=Test-hw150:351
```

In this example, the kernel VDOM_ID is 351.

Another example:

```
config vdom
   edit Test02-hw2
   current vf=Test02-hw2:499
```

In this example, the kernel VDOM_ID is 499.

When you create a VDOM from the CLI, the new hyperscale VDOM becomes the current VDOM. The new hyperscale firewall VDOM may not appear in the VDOM list on the GUI until you log out of the GUI and then log back in.

Enabling hyperscale firewall features

Use the following global command to enable hyperscale firewall features for your FortiGate:

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

On a FortiGate-4800F or 4801F, in addition to enabling `full-offload` for the hyperscale firewall VDOM, you also need to assign an NP7 processor group to the hyperscale firewall VDOM:

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

You need to assign the NP7 processor group before adding any interfaces to the hyperscale firewall VDOM. Assigning an NP7 processor group is required because of the NP7 configuration of the FortiGate 4800F and 4801F. For more information, see Assigning an NP7 processor group to a hyperscale firewall VDOM.

On a FortiGate 4800F or 4801F, hyperscale hardware logging can only send logs to interfaces in the same NP7 processor group as the NP7 processors that are handling the hyperscale sessions.

This means that hyperscale hardware logging servers must include a hyperscale firewall VDOM. This VDOM must be assigned the same NP7 processor group as the hyperscale firewall VDOM that is processing the hyperscale traffic being logged. This can be the same hyperscale VDOM or another hyperscale firewall VDOM that is assigned the same NP7 processor group. For more information, see NP7 processor groups and hyperscale hardware logging.

Once you have enabled global hyperscale firewall features, you must edit each hyperscale firewall VDOM and use the following command to enable hyperscale firewall features for that VDOM.

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

The following options are available for this command:

`disable` disable hyperscale firewall features and disable offloading DoS policy sessions to NP7 processors for this VDOM. All sessions are initiated by the CPU. Sessions that can be offloaded are sent to NP7 processors. This is the default setting.

`dos-offload` offload DoS policy sessions to NP7 processors for this VDOM. All other sessions are initiated by the CPU. Sessions that can be offloaded are sent to NP7 processors.

`full-offload` enable hyperscale firewall features for the current hyperscale firewall VDOM. This option is only available if the FortiGate is licensed for hyperscale firewall features. DoS policy sessions are also offloaded to NP7 processors. All other sessions are initiated by the CPU. Sessions that can be offloaded are sent to NP7 processors.

For more information about DoS policy hardware acceleration and how it varies depending on the policy offload level, see DoS policy hardware acceleration.

Hyperscale firewall GUI changes

A hyperscale firewall VDOM has the following GUI changes:

Firewall policies include hyperscale options

To add a hyperscale firewall policy, go to **Policy & Objects > Firewall Policy** and select **Create New** and configure the hyperscale firewall policy as required.

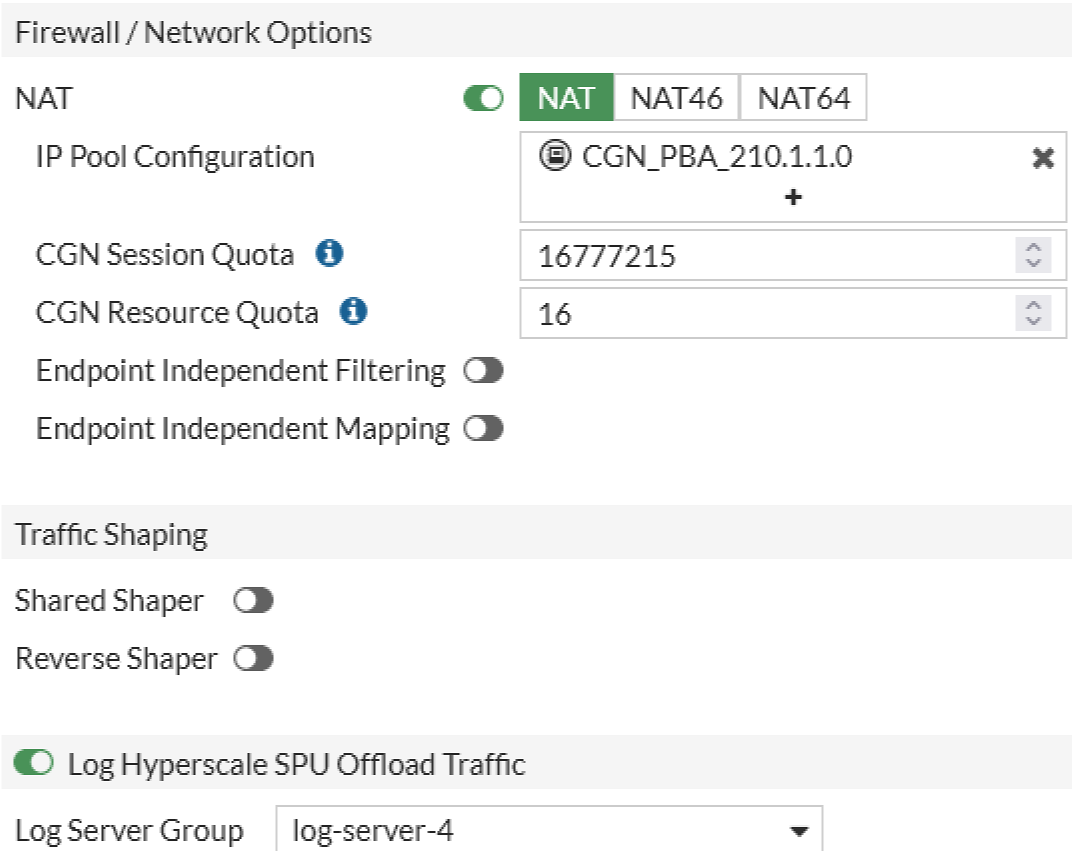
You can select **Log Hyperscale SPU Offload Traffic** to enable hyperscale firewall logging for all of the traffic accepted by the policy that is offloaded to NP7 processors.

IPv4 and NAT64 NAT hyperscale firewall policies can include CGN resource allocation IP Pools and other CGN options.

Firewall policies in Hyperscale VDOMs do not support UTM or NGFW features.

The number of firewall policies that can be added to a hyperscale firewall VDOM is limited to 15,000. For more information, see About the 15,000 policy per hyperscale VDOM limit on page 44.

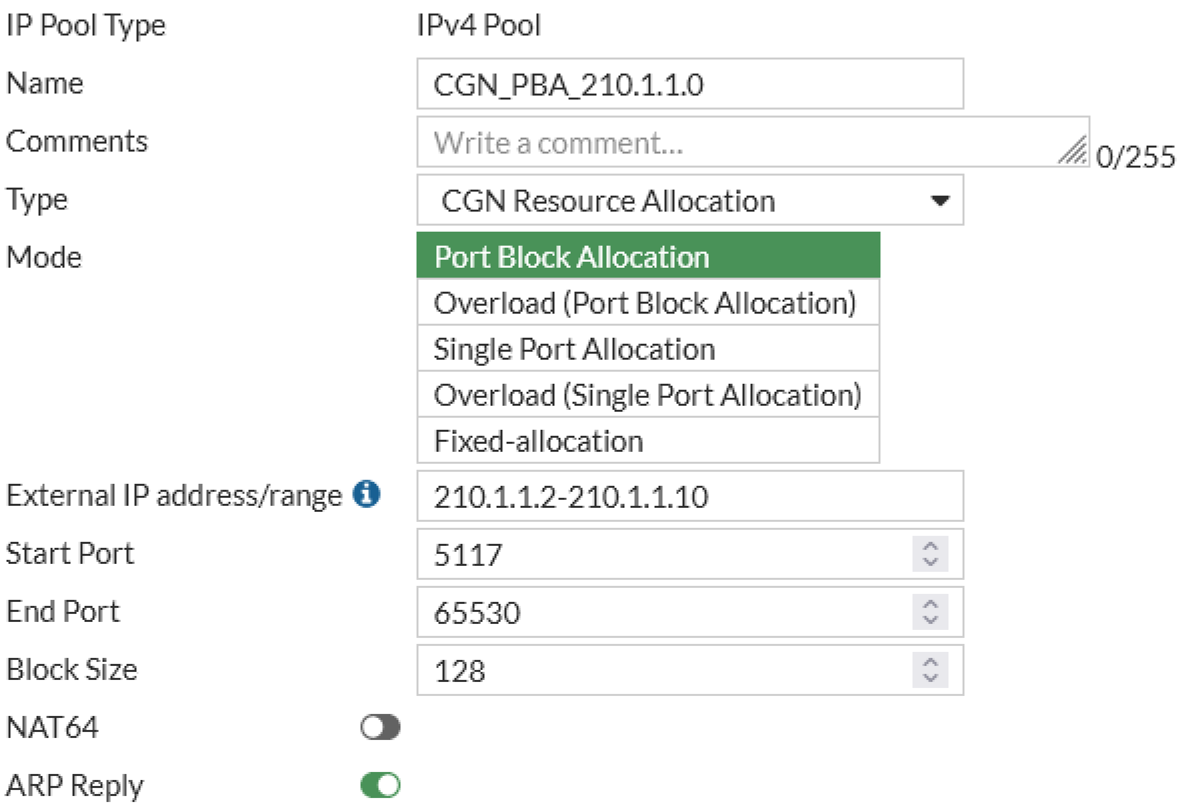
CGN and hardware logging options in a hyperscale firewall policy



IPv4 CGN resource allocation IP pools and groups

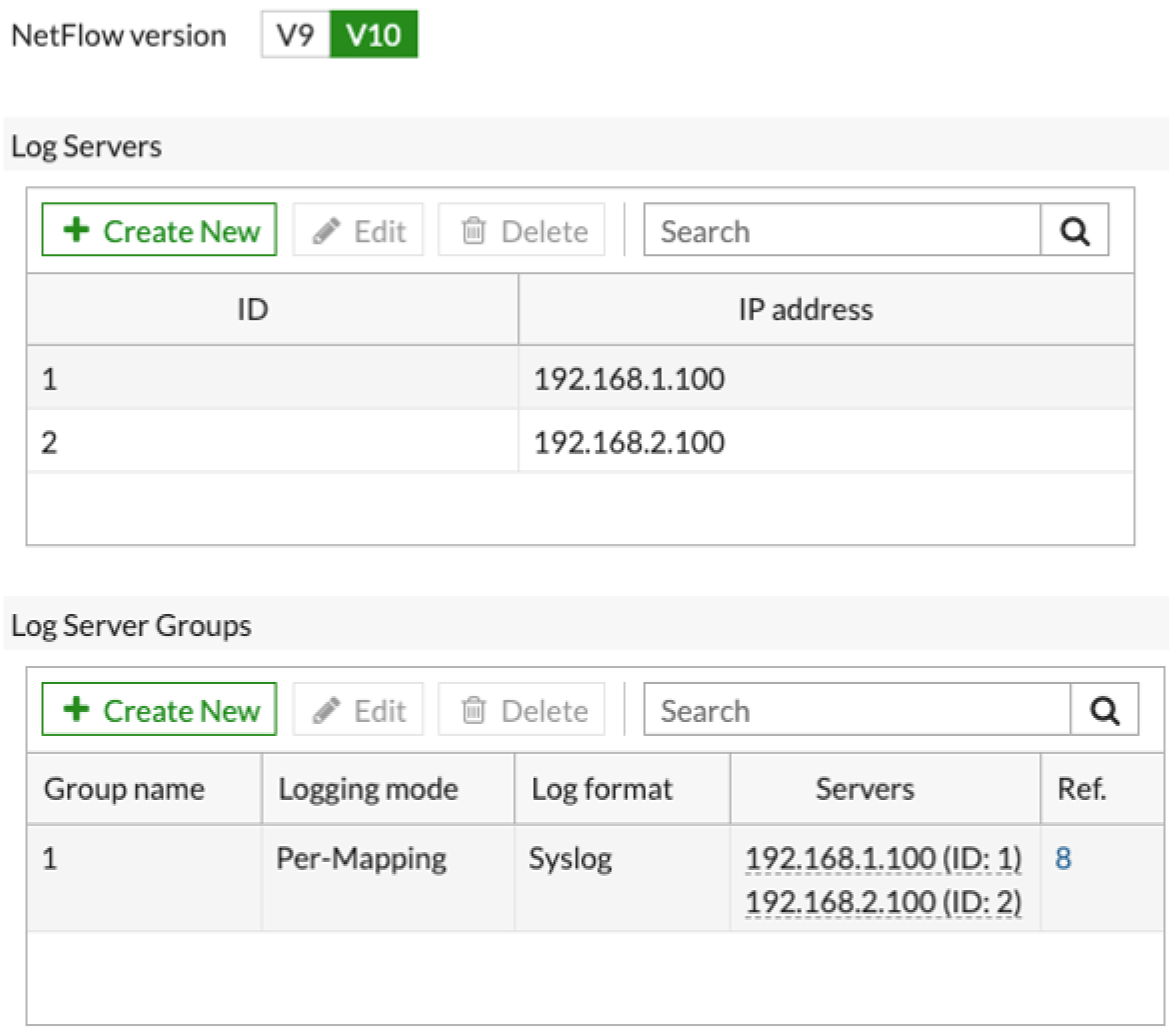
You can configure CGN resource allocation IP pools to add carrier grade NAT features to IPv4 or NAT64 hyperscale firewall policies. Go to **Policy & Objects > IP Pools**, select **Create New > IP Pool**, and set **IP Pool Type** to **IPv4 IP Pool**. Then set **Type** to **CGN Resource Allocation** and select a **Mode**.

You can also create CGN IP pool groups by going to **Create New > CGN IP Pool Group**.



Hyperscale hardware logging servers

You can set up multiple hyperscale hardware logging servers and add them to server groups. This is a global feature and all hyperscale VDOMs can use these globally configured servers. To configure hardware logging, from the Global GUI, go to **Log & Report > Hyperscale SPU Offload Log Settings**.



Hyperscale firewall CLI changes

The following hyperscale firewall CLI commands are available:

Enable hyperscale firewall features

Use the following global command to enable hyperscale firewall features:

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

Use the following command to enable hyperscale firewall features for a VDOM:

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

Special hyperscale firewall VDOM naming convention

VDOMs in which you will be enabling hyperscale firewall features must be created with a special VDOM name that also includes a VDOM ID number.

The following option can be used to set the VDOM ID range:

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

By default this option is set to 250, allowing you to configure up to 250 hyperscale firewall VDOMs by setting the VDOM in the range of 1 to 250.

Use the following syntax to create a hyperscale firewall VDOM from the global CLI:

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

For information about how to name hyperscale firewall VDOMs, see Creating hyperscale firewall VDOMs on page 16.

Firewall policies include hyperscale options

For any firewall policy in a hyperscale firewall VDOM, you can use the `cgn-log-server-grp` option to enable hyperscale firewall logging for all of the traffic accepted by the policy that is offloaded to NP7 processors.

> The number of firewall policies that can be added to a hyperscale firewall VDOM is limited to 15,000. For more information, see About the 15,000 policy per hyperscale VDOM limit on page 44.

IPv4 and NAT64 NAT hyperscale firewall policies can include the following CGN resource allocation options. You can also add CGN resource allocation IP pools to these policies.

```
config firewall policy
   edit 1
      set cgn-session-quota <quota>
      set cgn-resource-quota <quots>
      set cgn-eif {disable | enable}
      set cgn-eim {disable | enable}
   end
```

Firewall policies in Hyperscale VDOMs do not support UTM or NGFW features.

CGN Resource allocation IP pools

You can use the following command to configure CGN Resource allocation IP pools:

```
config firewall ippool
   edit <name>
      set type cgn-resource-allocation
      set startip <ip>
      set endip <ip>
      set arp-reply {disable | enable}
      set arp-intf <interface-name>
      set cgn-spa {disable | enable}
      set cgn-overload {disable | enable}
```

```
      set cgn-fixedalloc {disable | enable}
      set cgn-block-size <number-of-ports>
      set cgn-client-startip <ip>
      set cgn-client-endip <ip>
      set cgn-port-start <port>
      set cgn-port-end <port>
      set utilization-alarm-raise <usage-threshold>
      set utilization-alarm-clear <usage-threshold>
   end
```

CGN Resource allocation IP pool groups

You can use the following command to create CGN Resource Allocation IP pool groups:

```
config firewall ippool_grp
   edit <name>
      set member <cgn-ippool> ...
   end
```

Hardware logging

The following hardware logging commands are available:

```
config log npu-server
   set log-processor {hardware | host}
   set log-processing {may-drop | no-drop}
   set netflow-ver {v9 | v10}
   set syslog-facility <facility>
   set syslog-severity <severity>
      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>
            set template-tx-timeout <timeout>
         end
      config server-group
         edit <group-name>
            set log-mode {per-session | per-nat-mapping | per-session-ending}
            set log-format {netflow | syslog}
            set log-tx-mode {roundrobin | multicast}
            set sw-log-flags {tcp-udp-only | enable-all-log | disable-all-log}
            set log-user-info {disable | enable}
            set log-gen-event {disable | enable}
            set server-number <number>
            set server-start-id <number>
         end
```

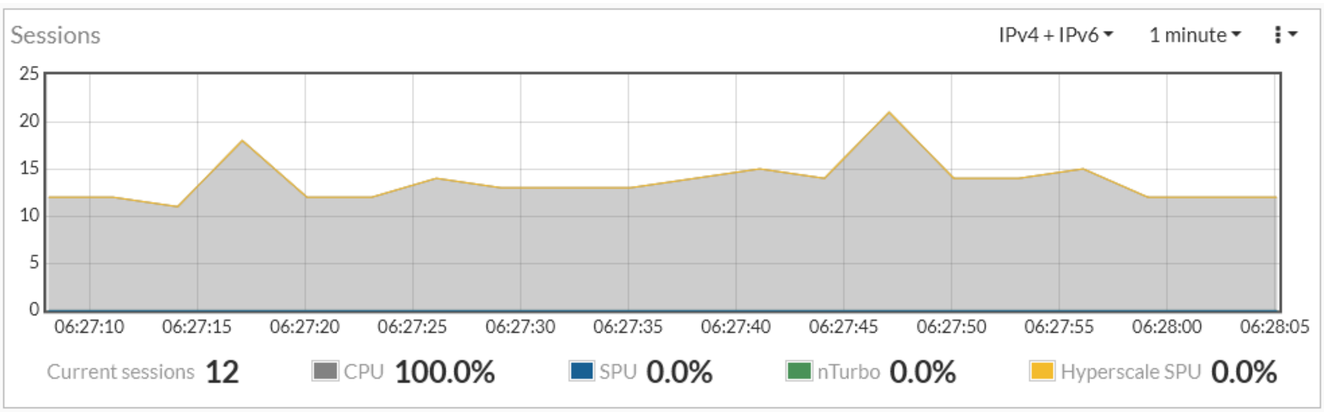
Hyperscale firewall inter-VDOM link acceleration

You apply NP7 acceleration to inter-VDOM link traffic by creating inter-VDOM links with the `type` set to `npupair`. For example:

```
config system vdom-link
   edit <name>
      set type npupair
   end
```

Hyperscale sessions dashboard widget (NP7 performance monitoring)

On a FortiGate with a hyperscale firewall license, the Sessions dashboard widget shows Hyperscale sessions as well as CPU, offloaded SPU and nTurbo sessions. **Current sessions** shows the total number of sessions, **CPU** shows the percent of sessions handled by the CPU, **SPU** shows the percentage of these sessions that are SPU sessions, and **Nturbo** shows the percentage that are nTurbo sessions. **Hyperscale SPU** shows the percentage of the total sessions that are hyperscale firewall sessions.



Hyperscale, SPU, and NTurbo sessions are displayed on the dashboard widget if NP7 per-session accounting is enabled. By default NP7 per-session accounting is enabled for all traffic accepted by firewall policies that have traffic logging enabled. You can use the following command to disable NP7 per-session accounting or enable per-session accounting for all traffic offloaded by NP7 processors.

```
config system npu
   set per-session-accounting {disable | enable | traffic-log-only}
end
```

`enable` enables per-session accounting for all traffic offloaded by NP7 processors.

`disable` turns off per-session accounting.

`traffic-log-only` (the default) turns on NP7 per-session accounting for traffic accepted by firewall policies that have traffic logging enabled.

Enabling per-session accounting can affect NP7 offloading performance.

Enabling or disabling per-policy accounting for hyperscale firewall traffic

Per-policy accounting records hit counts for packets accepted or denied by hyperscale firewall policies and makes this information available from the firewall policy GUI and from the CLI.

Per-policy accounting for hyperscale firewall policies can reduce hyperscale firewall performance.You can use the following command to enable or disable hyperscale firewall per-policy accounting for all hyperscale traffic:

```
config system npu
   set per-policy-accounting {disable | enable}
end
```

Per-policy accounting is disabled by default. When per-policy accounting is enabled, you can see hyperscale firewall policy hit counts on the GUI and CLI. If you disable per-policy-accouting for hyperscale firewall traffic, FortiOS will not collect hit count information for traffic accepted or denied by hyperscale firewall policies.

Enabling or disabling per-policy accounting deletes all current sessions, disrupting traffic. Changing the per-policy accounting configuration should only be done during a quiet period.

Hardware accelerated Carrier Grade NAT

Hyperscale firewall Carrier Grade NAT (CGN) features can be used to accelerate dynamic SNAT resource management for IPv4 and NAT64 traffic. Using carrier grade NAT features, FortiOS is capable of managing SNAT resources for complex networks containing large numbers of devices with private IPv4 addresses. Hyperscale CGN uses an enhanced implementation of FortiOS IP Pools to apply these CGN resource management features to traffic as it passes through the FortiGate.

For information about FortiOS IP pools, see Dynamic SNAT.

Start a hyperscale firewall carrier grade NAT configuration by creating one or more CGN resource allocation IP pools. These IP pools are variations on an overload IP pool that define how the firewall manages source addresses and source ports. Then you create a hyperscale firewall policy and add the CGN resource allocation IP pools to the firewall policy.

If you add multiple CGN resource allocation IP pools to a hyperscale firewall policy, the IP pools must all have the same CGN mode (none, overload, single port allocation, or fixed-allocation) and their IP ranges must not overlap.

Instead of adding multiple IP pools to a hyperscale firewall policy, you can create a CGN IP pool group and add multiple CGN IP pools to the group. Then add the CGN IP pool group to the firewall policy. All of the CGN IP pools in a CGN IP pool group must have the same CGN mode and their IP ranges must not overlap.

Hyperscale and standard FortiOS CGNAT feature comparison

In many cases, standard FortiOS can provide many carrier grade NAT (CGNAT) features and, depending on the hardware platform, excellent CGNAT performance. Hyperscale FortiOS supports CGNAT with much higher connections per second performance, hardware session logging, and more CGNAT features but does not support these features for UTM traffic.You can license a FortiGate for Hyperscale, use hyperscale firewall VDOMs for non-UTM traffic and normal VDOMs for UTM traffic.

Hyperscale FortiOS also supports a few more CGNAT features than standard FortiOS. The following table breaks down the CGNAT features supported by hyperscale FortiOS and standard FortiOS:

CGNAT Feature	Hyperscale FortiOS	Standard FortiOS
PBA with no overloading	Yes Port block allocation CGN IP pool on page 30.	No. FortiOS PBA re-uses addresses.
PBA with overloading • Dynamic IP consistency • Port block	Yes Overload with port-block-allocation CGN IP pool on page 32.	Yes Port block allocation

CGNAT Feature	Hyperscale FortiOS	Standard FortiOS
allocation • Port reuse within block • Deterministic NAT		
PBA with NAT64	Yes Overload with port-block-allocation CGN IP pool on page 32.	No
Single port allocation (SPA) • Dynamic IP consistency • No port reuse • Deterministic NAT	Yes Single port allocation CGN IP pool on page 34.	No
Single port allocation (SPA) with overload • Dynamic IP consistency • Port reuse within the entire port range • Deterministic NAT	Yes Overload with single port allocation CGN IP pool on page 36.	No
PBA. fixed allocation • Static IP consistency • Static port block allocation • No port reuse • Deterministic NAT	Yes Fixed allocation CGN IP pool on page 37.	Yes Fixed port range
IP pool groups • Streamlines hyperscale firewall policy configuration.	Yes CGN resource allocation IP pool groups on page 39.	No
Port starting number	5117	5117
Bi-directional session TTL refresh timers	Yes You can control whether idle outgoing or incoming or both outgoing and incoming sessions are terminated when the TTL is reached. See Hyperscale firewall VDOM session timeouts on page 74.	No
Endpoint Independent Mapping (EIM)	Yes	Yes

CGNAT Feature	Hyperscale FortiOS	Standard FortiOS
	You can enable or disable EIM in a hyperscale firewall policy CGN resource allocation hyperscale firewall policies on page 40.	EIM + overloading (Reuse) is always enabled
Endpoint Independent Filtering (EIF)	Yes You can enable or disable EIF in a hyperscale firewall policy CGN resource allocation hyperscale firewall policies on page 40.	Partially • PBA IP pools support EIF by enabling `permit-any-host` • Fixed port range IP pools do not support EIF.

CGN resource allocation IP pools

CGN resource allocation IP pools are variations on overload IP pools that take advantage of NP7 hardware acceleration to apply Carrier Grade NAT (CGN) features to IPv4 or NAT64 hyperscale firewall policies. CGN resource allocation IP pools manage the allocation of IPv4 source ports, addresses, and system resources used for logging.

You create CGN resource allocation IP pools from the GUI by going to **Policy & Objects > IP Pools > Create New > IP Pool**. Set the **IP Pool Type** to **IPv4 IP Pool**, set **Type** to **CGN Resource Allocation**, select a **Mode** (or type), and edit settings for the selected mode.

From the CLI, you create CGN resource allocation IP pools by creating an IP pool and setting the `type` to `cgn-resource-allocation`. You can then enable or disable `cgn-spa`, `cgn-overload`, and `cgn-fixedalloc` to select a CGN IP pool type and then edit settings for the selected type. You can enable `nat64` to make this a NAT64 IP pool

```
config firewall ippool
   edit <name>
      set type cgn-resource-allocation
      set startip <ip>
      set endip <ip>
      set arp-reply {disable | enable}
      set arp-intf <interface-name>
      set cgn-spa {disable | enable}
      set cgn-overload {disable | enable}
      set cgn-fixedalloc {disable | enable}
      set cgn-block-size <number-of-ports>
      set cgn-client-startip <ip>
      set cgn-client-endip <ip>
      set cgn-port-start <port>
      set cgn-port-end <port>
      set utilization-alarm-raise <usage-threshold>
      set utilization-alarm-clear <usage-threshold>
      set nat64 {disable | enable}
   end
```

Five different types or modes of CGN resource allocation IP pool modes are available. The following table summarizes each type and the following sections describe the GUI and CLI configuration for each type.

IP pool type (mode)	GUI option	CLI options	Supported CGNAT Features

Port Block Allocation (PBA)	Port Block Allocation	`set cgn-spa disable`
		`set cgn-overload disable`
		`set cgn-fixedalloc disable`
			• Dynamic IP consistency
			• Port block allocation
			• No port reuse
			• Deterministic NAT

Overload with port block allocation (PBA, overload)	Overload (Port Block Allocation)	`set cgn-spa disable` `set cgn-overload enable`	• Dynamic IP consistency • Port block allocation • Port reuse within block • Deterministic NAT
Single port allocation (SPA)	Single Port Allocation	`set cgn-spa enable` `set cgn-overload disable`	• Dynamic IP consistency • No port reuse • Deterministic NAT
Overload with single port allocation (SPA, overload)	Overload (Single Port Allocation)	`set cgn-spa enable` `set cgn-overload enable`	• Dynamic IP consistency • Port reuse within the entire port range • Deterministic NAT
Fixed allocation, (also called Port block allocation with fixed NAT or Deterministic NAT) (PBA, fixed NAT)	Fixed-allocation	`set cgn-spa disable` `set cgn-fixedalloc enable`	• Static IP consistency • Static port block allocation • No port reuse • Deterministic NAT

Static IP consistency

If more than one public IP address is available, static IP consistency makes sure that sessions from a given client are always assigned the same public source IP address.

Dynamic IP consistency

The first time a client starts a new session, the session gets any one of the available public IP addresses. New sessions started by the same client use the same public IP address, so all currently active sessions from a client will have the same public IP address. If all sessions from a client time out, the next time the client starts a new session, the session can again get any one of the available public IP addresses.

Port reuse within block

Sessions from the same client may be assigned duplicate public source ports.

Port reuse within whole port range

Sessions from different clients may be assigned the same public source ports.

Port block allocation

A block of source ports is dynamically allocated to each client. Sessions started by a client can use any one of the ports in their allocated block. Whether ports can be re-used and how they are re-used depends on what other features are active.

Static port block allocation

Blocks of ports are assigned to clients exclusively and deterministically. When a block of ports is assigned to a client, all sessions started by that client use the assigned ports and sessions started by other clients cannot use those ports.

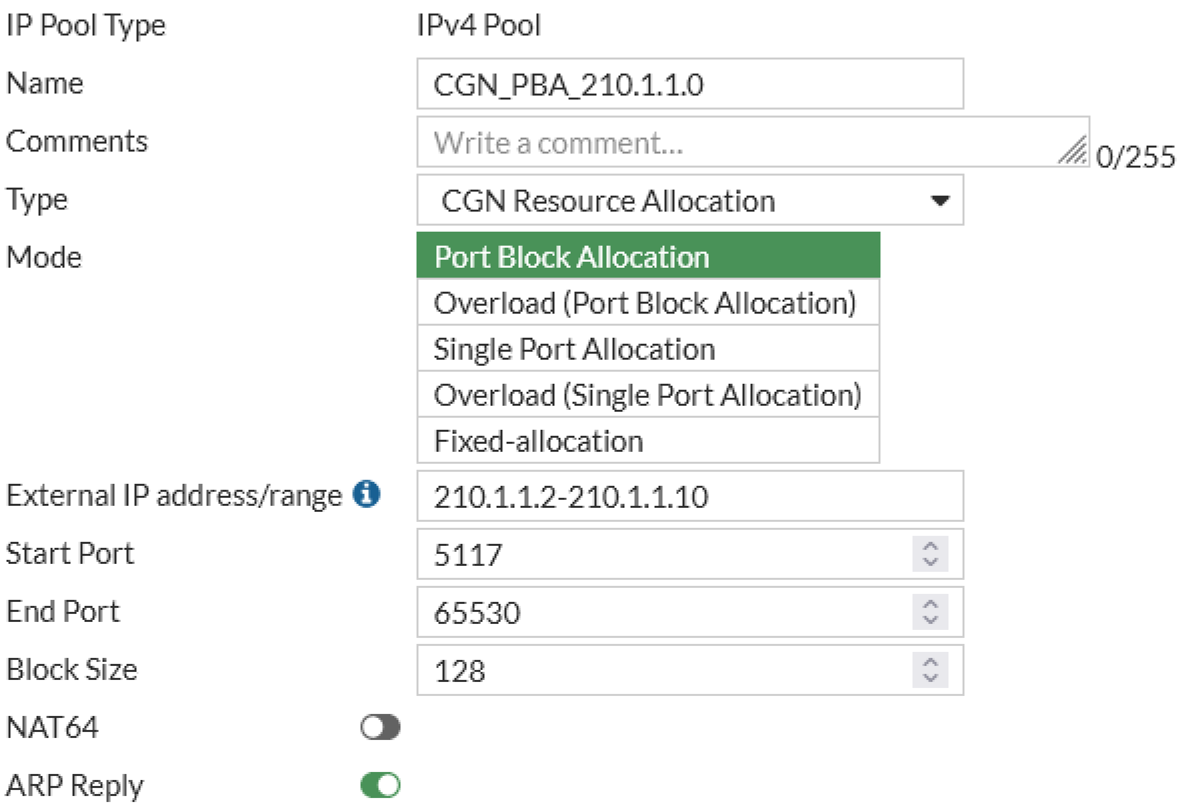
Deterministic NAT

Creates a one to one mapping between external and internal IP addresses. You add matching external and internal address ranges to the configuration, and a given internal address is always translated to the same external address. The number of clients that can use a deterministic NAT pool is limited by the number of IP addresses in the pool.

Port block allocation CGN IP pool

This is the default CGNAT IP pool configuration.

On the GUI go to **Policy & Objects > IP Pools > Create New > IP Pool**. Set **IP Pool Type** to **IPv4 IP Pool**, set **Type** to **CGN Resource Allocation**, and set **Mode** to **Port Block Allocation**. You can enable **NAT64** to make this a NAT64 IP pool.



On the CLI:

```
config firewall ippool
   edit <name>
      set type cgn-resource-allocation
      set startip <ip>
      set endip <ip>
      set arp-reply {disable | enable}
      set arp-intf <interface-name>
      set cgn-spa disable
      set cgn-overload disable
      set cgn-fixedalloc disable
      set cgn-block-size <number-of-ports>
      set cgn-port-start <port>
      set cgn-port-end <port>
      set utilization-alarm-raise <usage-threshold>
      set utilization-alarm-clear <usage-threshold>
      set nat64 {disable | enable}
   end
```

Port block allocation (PBA) reduces CGNAT logging overhead by creating a log entry only when a client first establishes a network connection and is assigned a port block. The number of log entries are reduced because a log entry is created when the port block is assigned, and not for each client connection.

When all of the client sessions have ended, FortiOS releases the port block and writes another log message. You can also configure logging to only write a log message when the port block is released. See Configuring hardware logging on page 49.

In general, because each customer environment is different, different configurations may be required to achieve optimal performance.

PBA allocates a contiguous set of source translation endpoints called port blocks. These port blocks are associated to a client by one IP address and a block of ports. Port blocks are allocated on-demand and have a fixed size.

Choose these settings carefully to adequately and efficiently service clients that may require a different number of simultaneous connections. Careful analysis and testing is required to find optimal values for the traffic conditions on your network.

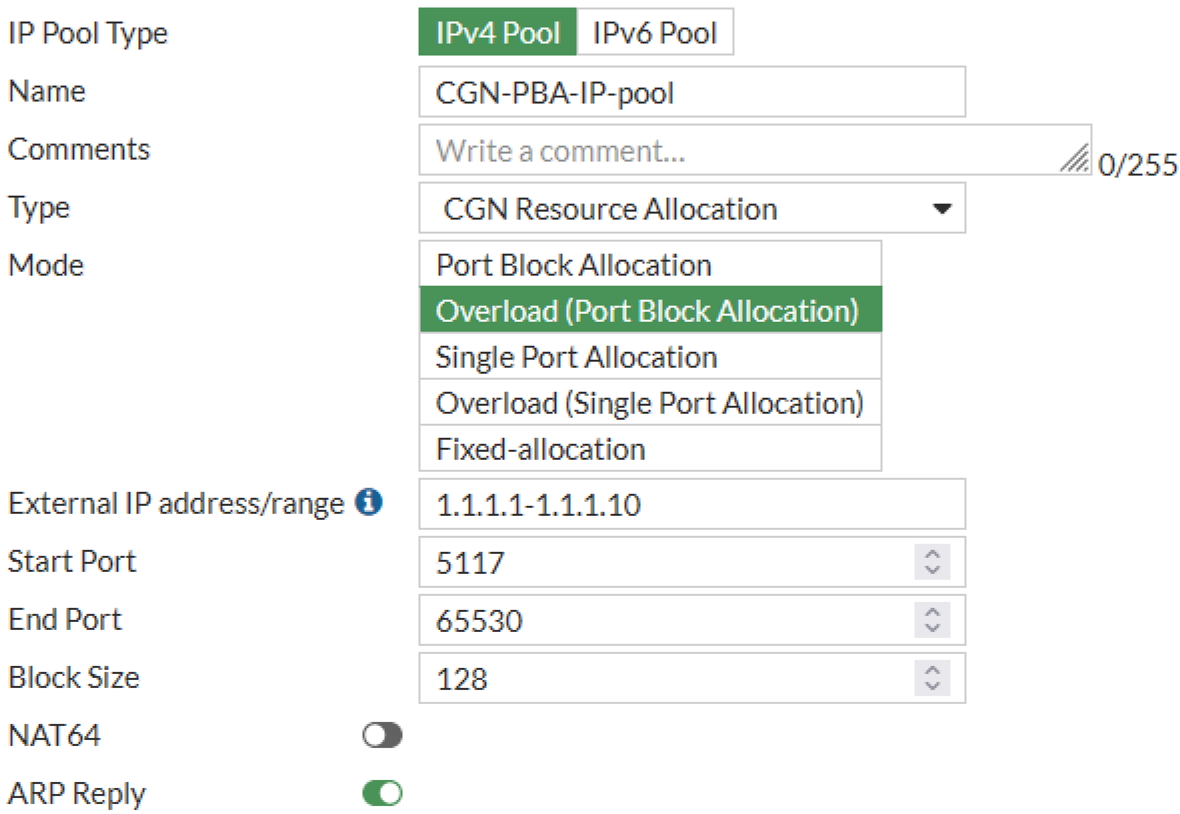
You can define a port-block allocation IP pool by configuring the following:

- External IP address range (`start-ip` and `end-ip`). Specifies the set of translation IP addresses available in the pool as a collection of IP prefixes with their prefix lengths. These are typically public-side addresses.
- Start port (`cgn-port-start`). The lowest port number in the port range. The range is 1024 to 65535. The default value is 5117.
- End port (`cgn-port-end`). The highest possible port number in the port range. The default value is 65530.
- Block size (`cgn-block-size`). The number of ports allocated in a block. The block size can be from 64 to 4096 in increments of 64 (for example, 64, 128, 192,..., 4096). The default value is 128. Use a smaller port block size to conserve available ports.
- Enable or disable ARP reply (`arp-reply`) to reply to ARP requests for addresses in the external address range.
- Optionally specify the interface (`arp-intf`) that replies to ARP requests.
- Generate an SNMP trap when the usage of the resources defined by an IP pool exceeds a threshold (`utilization-alarm-raise`). The range is 50 to 100 per cent.
- Generate an SNMP trap when the usage of the resources defined by an IP pool falls below a threshold (`utilization-alarm-clear`). The range is 40 to 100 per cent.
- You can enable `nat64` to make this a NAT64 IP pool.

You can also configure PBA with overload. Overload causes FortiOS to re-use ports within a block, allowing for more possible connections before running out of ports. To configure PBA with overload, see Overload with port-block-allocation CGN IP pool on page 32.

Overload with port-block-allocation CGN IP pool

On the GUI go to **Policy & Objects > IP Pools > Create New > IP Pool**. Set **IP Pool Type** to **IPv4 IP Pool**, set **Type** to **CGN Resource Allocation**, and set **Mode** to **Overload (Port Block Allocation)**. You can enable **NAT64** to make this a NAT64 IP pool.



On the CLI:

```
config firewall ippool
  edit <name>
    set type cgn-resource-allocation
    set startip <ip>
    set endip <ip>
    set arp-reply {disable | enable}
    set arp-intf <interface-name>
    set cgn-spa disable
    set cgn-overload enable
    set cgn-fixedalloc disable
    set cgn-block-size <number-of-ports>
    set cgn-port-start <port>
    set cgn-port-end <port>
    set utilization-alarm-raise <usage-threshold>
    set utilization-alarm-clear <usage-threshold>
    set nat64 {disable | enable}
  end
```

Overload with Port block allocation (PBA) reduces CGNAT logging overhead by creating a log entry only when a client first establishes a network connection and is assigned a port block. The number of log entries are reduced because a log entry is created when the port block is assigned, and not for each client connection. Overload causes FortiOS to re-use ports within a block, allowing for more possible connections before running out of ports.

When all of the client sessions have ended, FortiOS releases the port block and writes another log message. You can also configure logging to only write a log message when the port block is released. See Configuring hardware logging on page 49.

In general, because each customer environment is different, different configurations may be required to achieve optimal performance.

PBA allocates a contiguous set of source translation endpoints called port blocks. These port blocks are associated to a client by one IP address and a block of ports. Port blocks are allocated on-demand and have a fixed size.

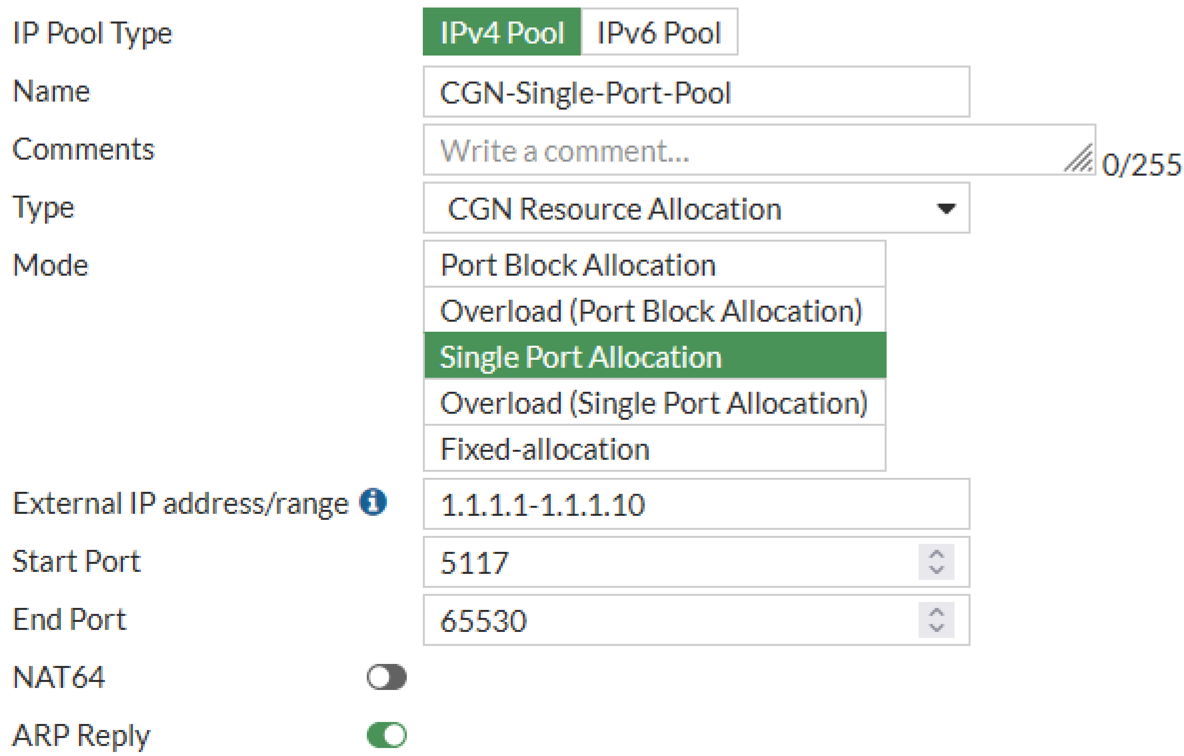
Choose these settings carefully to adequately and efficiently service clients that may require a different number of simultaneous connections. Careful analysis and testing is required to find optimal values for the traffic conditions on your network.

You can define an overload port-block allocation IP pool by configuring the following:

- External IP address range (`start-ip` and `end-ip`). Specifies the set of translation IP addresses available in the pool as a collection of IP prefixes with their prefix lengths. These are typically public-side addresses.
- Start port (`cgn-port-start`). The lowest port number in the port range. The range is 1024 to 65535. The default value is 5117.
- End port (`cgn-port-end`). The highest possible port number in the port range. The default value is 65530.
- Block size (`cgn-block-size`). The number of ports allocated in a block. The block size can be from 64 to 4096 in increments of 64 (for example, 64, 128, 192,..., 4096). The default value is 128. Use a smaller port block size to conserve available ports.
- Enable or disable ARP reply (`arp-reply`) to reply to ARP requests for addresses in the external address range.
- Optionally specify the interface (`arp-intf`) that replies to ARP requests.
- Generate an SNMP trap when the usage of the resources defined by an IP pool exceeds a threshold (`utilization-alarm-raise`). The range is 50 to 100 per cent.
- Generate an SNMP trap when the usage of the resources defined by an IP pool falls below a threshold (`utilization-alarm-clear`). The range is 40 to 100 per cent.
- You can enable `nat64` to make this a NAT64 IP pool.

Single port allocation CGN IP pool

On the GUI go to **Policy & Objects > IP Pools > Create New > IP Pool**. Set **IP Pool Type** to **IPv4 IP Pool**, set **Type** to **CGN Resource Allocation**, and set **Mode** to **Single Port Allocation**. You can enable **NAT64** to make this a NAT64 IP pool.



On the CLI:

```
config firewall ippool
   edit <name>
      set type cgn-resource-allocation
      set startip <ip>
      set endip <ip>
      set arp-reply {disable | enable}
      set arp-intf <interface-name>
      set cgn-spa enable
      set cgn-overload disable
      set cgn-port-start <port>
      set cgn-port-end <port>
      set utilization-alarm-raise <usage-threshold>
      set utilization-alarm-clear <usage-threshold>
      set nat64 {disable | enable}
   end
```

A single port allocation CGN resource allocation IP pool assigns single ports instead of ranges of ports. This type of CGN IP pool conserves ports by effectively reducing the port block size to 1. Since blocks of ports are not assigned to each client, this CGN IP Pool type works better for networks with large numbers of clients that start fewer individual sessions.

Since this is not an overload IP pool, ports are not re-used. Each client session gets a new port from the range of ports added to the IP pool that are available.

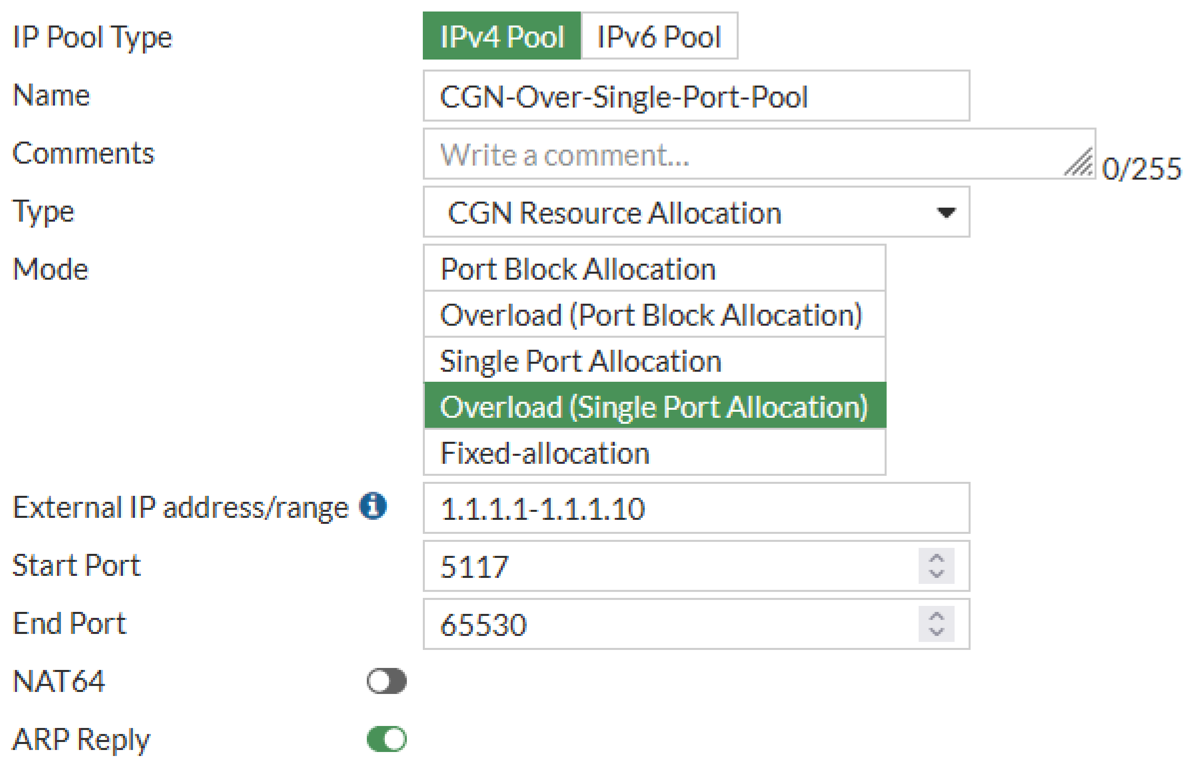
You can define a single port allocation IP pool by configuring the following:

- External IP address range (`start-ip` and `end-ip`). Specifies the set of translation IP addresses available in the pool as a collection of IP prefixes with their prefix lengths. These are typically public-side addresses.
- Start port (`cgn-port-start`). The lowest port number in the port range. The range is 1024 to 65535. The default value is 5117.

- End port (`cgn-port-end`). The highest possible port number in the port range. The default value is 65530.
- Enable or disable ARP reply (`arp-reply`) to reply to ARP requests for addresses in the external address range.
- Optionally specify the interface (`arp-intf`) that replies to ARP requests.
- Generate an SNMP trap when the usage of the resources defined by an IP pool exceeds a threshold (`utilization-alarm-raise`). The range is 50 to 100 per cent.
- Generate an SNMP trap when the usage of the resources defined by an IP pool falls below a threshold (`utilization-alarm-clear`). The range is 40 to 100 per cent.
- You can enable `nat64` to make this a NAT64 IP pool.

Overload with single port allocation CGN IP pool

On the GUI go to **Policy & Objects > IP Pools > Create New > IP Pool**. Set **IP Pool Type** to **IPv4 IP Pool**, set **Type** to **CGN Resource Allocation**, and set **Mode** to **Overload (Single Port Allocation)**. You can enable **NAT64** to make this a NAT64 IP pool.



On the CLI:

```
config firewall ippool
   edit <name>
      set type cgn-resource-allocation
      set startip <ip>
      set endip <ip>
      set arp-reply {disable | enable}
      set arp-intf <interface-name>
      set cgn-spa enable
      set cgn-overload enable
      set cgn-port-start <port>
```

```
    set cgn-port-end <port>
    set utilization-alarm-raise <usage-threshold>
    set utilization-alarm-clear <usage-threshold>
    set nat64 {disable | enable}
end
```

An overload single port allocation CGN resource allocation IP pool assigns single ports instead of ranges of ports. This type of CGN IP pool conserves ports by effectively reducing the port block size to 1. Since this is an overload IP pool, ports are re-used. A client session can get any port from the range of ports added to the IP pool that are available.

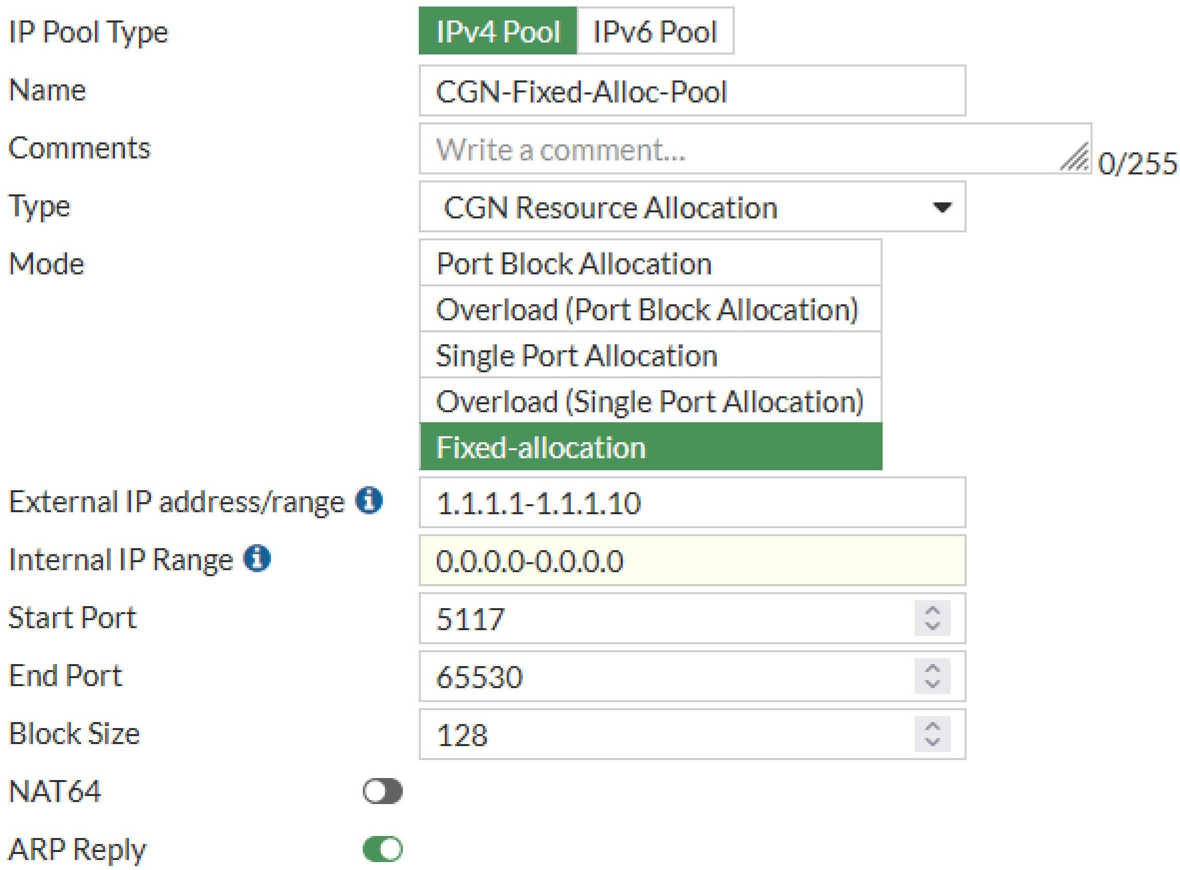
Since blocks of ports are not assigned to each client and ports are re-used, there are no limits on the number of ports that a client IP address can use. Port re-use is determined by how much the pool is utilized. This IP pool type works for networks with large numbers of clients where those clients may start many individual sessions.

You can define an overload single port allocation IP pool by configuring the following:

- External IP address range (`start-ip` and `end-ip`). Specifies the set of translation IP addresses available in the pool as a collection of IP prefixes with their prefix lengths. These are typically public-side addresses.
- Start port (`cgn-port-start`). The lowest port number in the port range. The range is 1024 to 65535. The default value is 5117.
- End port (`cgn-port-end`). The highest possible port number in the port range. The default value is 65530
- Enable or disable ARP reply (`arp-reply`) to reply to ARP requests for addresses in the external address range.
- Optionally specify the interface (`arp-intf`) that replies to ARP requests.
- Generate an SNMP trap when the usage of the resources defined by an IP pool exceeds a threshold (`utilization-alarm-raise`). The range is 50 to 100 per cent.
- Generate an SNMP trap when the usage of the resources defined by an IP pool falls below this threshold (`utilization-alarm-clear`). The range is 40 to 100 per cent.
- You can enable `nat64` to make this a NAT64 IP pool.

Fixed allocation CGN IP pool

On the GUI go to **Policy & Objects > IP Pools > Create New > IP Pool**. Set **IP Pool Type** to **IPv4 IP Pool**, set **Type** to **CGN Resource Allocation**, and set **Mode** to **Fixed-allocation**. You can enable **NAT64** to make this a NAT64 IP pool.



On the CLI:

```
config firewall ippool
   edit <name>
      set type cgn-resource-allocation
      set startip <ip>
      set endip <ip>
      set arp-reply {disable | enable}
      set arp-intf <interface-name>
      set cgn-spa disable
      set cgn-fixedalloc enable
      set cgn-block-size <number-of-ports>
      set cgn-client-startip <ip>
      set cgn-client-endip <ip>
      set cgn-port-start <port>
      set cgn-port-end <port>
      set utilization-alarm-raise <usage-threshold>
      set utilization-alarm-clear <usage-threshold>
      set nat64 {disable | enable}
   end
```

Also called deterministic NAT, a fixed allocation CGN resource allocation IP pool causes FortiOS to find the maximum possible block size, given the configured NAT resources and gives one block to each client.

The number of clients that can use a fixed allocation CGN resource allocation IP pool is limited by the number of IP addresses in the pool. Since this is not an overload IP pool, ports are not re-used.

You can define a fixed allocation IP pool by configuring the following:

- External IP address range (`start-ip` and `end-ip`). Specifies the set of translation IP addresses available in the pool as a collection of IP prefixes with their prefix lengths. These are typically public-side addresses.
- Internal or client IP address range (`cgn-client-startip` and `cgn-client-endip`). The range of internal addresses. This range must match or be a subset of the available source IP addresses.
- Start port (`cgn-port-start`). The lowest port number in the port range. The range is 1024 to 65535. The default value is 5117.
- End port (`cgn-port-end`). The highest possible port number in the port range. The default value is 65530
- Block size (`cgn-block-size`). When `cgn-fixedallc` is enabled, the `cgn-block-size` configuration is ignored because FortiOS calculates a block-size to find the maximum possible block size and gives one block to each client.
- Enable or disable ARP reply (`arp-reply`) to reply to ARP requests for addresses in the external address range.
- Optionally specify the interface (`arp-intf`) that replies to ARP requests.
- Generate an SNMP trap when the usage of the resources defined by an IP pool exceeds a threshold (`utilization-alarm-raise`). The range is 50 to 100 per cent.
- Generate an SNMP trap when the usage of the resources defined by an IP pool falls below this threshold (`utilization-alarm-clear`). The range is 40 to 100 per cent.
- You can enable `nat64` to make this a NAT64 IP pool.

CGN resource allocation IP pool groups

You can configure CGN resource allocation IP pool groups to group together related CGN resource allocation IP pools to be able to add multiple IP pools to the same firewall policy. All of the CGN IP pools in a CGN IP pool group must have the same CGN mode and their IP ranges must not overlap.

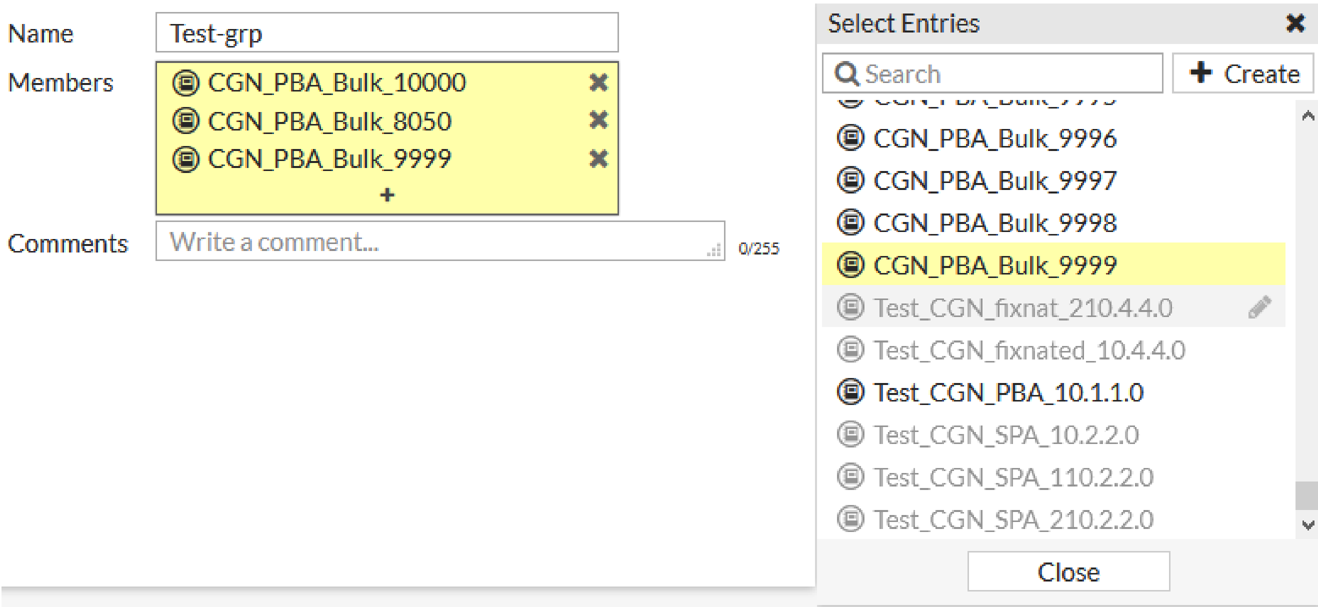
Use the following command to create an CGN resource allocation IP pool group:

```
config firewall ippool_grp
  edit <name>
    set member <cgn-ippool> ...
  end
```

`member` select the names of the CGN IP pools to add to the CGN IP pool group.

Use the following steps to configure CGNAT IP pool groups from the GUI:

1. Go to **Policy & Objects > IP Pools**.
2. Select **Create New > CGN IP Pool Group**.
3. Select CGN IP pools to add to the **Members** list.



CGN resource allocation hyperscale firewall policies

The number of firewall policies that can be added to a hyperscale firewall VDOM is limited to 15,000. For more information, see About the 15,000 policy per hyperscale VDOM limit on page 44.

Use the following options to add a IPv4 CGN resource allocation hyperscale firewall policy to a hyperscale firewall VDOM:

```
config firewall policy
   edit <id>
      set action accept
      set dstaddr <address>
      set nat enable
      set ippool enable
       set poolname {<cgn-ippool> | <cgn-ippool-group>}...
      set cgn-session-quota <quota>
      set cgn-resource-quota <quota>
      set cgn-eif {enable| disable}
      set cgn-eim {enable| disable}
      set cgn-log-server-grp <group-name>
   end
```

Use the following options to add a NAT64 CGN resource allocation hyperscale firewall policy to a hyperscale firewall VDOM:

```
config firewall policy
   edit <id>
      set action accept
      set dstaddr <address>
      set nat64 enable
      set ippool enable
```

```
     set poolname {<cgn-ippool> | <cgn-ippool-group>}...
    set cgn-session-quota <quota>
    set cgn-resource-quota <quota>
    set cgn-eif {enable| disable}
    set cgn-eim {enable| disable}
    set cgn-log-server-grp <group-name>
  end
```

`poolname` select one or more CGN IP pools or IP pool groups to apply CGN resource allocation IP pools to the firewall policy. To be able to add IP pools, `nat` or `nat64`, and `ippool` must be enabled and the addresses in the IP pools must overlap with the `dstaddr` address.

`cgn-session-quota` limit the number of concurrent sessions available for a client IP address (effectively the number of sessions per user). The range is 0 to 16777215 (the default). The default setting effectively means there is no quota.

`cgn-resource-quota` set a quota for the number port blocks available for a client IP address (effectively the number of port blocks per client IP address). Only applies if the firewall policy includes CGN IP pools with port block sizes. The range is 1 to 16 and the default is 16. If your FortiGate has multiple NP7 processors, the resource quota should be set differently depending on the `hash-config` used by the internal switch fabric (ISF). See Recommended NP7 traffic distribution for optimal CGNAT performance on page 66 for details.

`cgn-eif` enable or disable Endpoint Independent Filtering (EIF). Disabled by default. if another server attempts to connect to a public IP and port which is used by an existing session, when EIF is enabled, the NP7 will create the session and reuse the mapping for the existing session. When EIF is not enabled, the server attempts to connect to the public IP and port will fail. This practice is recommended in RFC 4787 for client applications that require this behavior.

For example, Client-A has an existing session, {A.a, B.b, S.s}. When another server S1.s1 attempts to connect to public address and port B.b, when EIF is enabled, the NP7 creates a new session as {A.a, B.b, S1.s1}. When EIF is disabled, such connection will be checked in full-policy and probably dropped.

If your FortiGate has multiple NP7 processors, depending on whether or not you are enabling EIF in hyperscale firewall policies, you may want to use the `nss-threads-option` of the `config system npu` command to optimize performance, see Optimizing NP7 network session setup (NSS) engine performance on page 79.

`cgn-eim` enable or disable Endpoint Independent Mapping (EIM). If a client uses an existing source port to connect to a different server, the NP7 reuses the existing mapping to create new sessions. This practice is more compatible for some applications to work with NAT devices, also it is more efficient. A new resource allocation counts towards the resource quota. If EIM is triggered, the new session does not cause new resource allocation and the new session only counts towards the session quota.

For example, Client-A has an existing session, represented as {A.a, B.b, S.s}, where A.a is the client IP and port, B.b is the mapped IP and port, and S.s is the server IP and port. When EIM is enabled, if the client uses A.a to connect to another server S1.s1, the NP7 reuses the public IP and port at B.b to create session that can be represented as {A.a, B.b, S1.s1}.

About hairpinning

You can use EIF to support hairpinning. A hairpinning configuration allows a client to communicate with a server that is on the same network as the client, but the communication takes place through the FortiGate because the client only knows the external address of the server.

To set up a hyperscale firewall hairpinning configuration, you need to enable EIF in the firewall policy. As well, the IP pool added to the policy should include addresses that overlap with the firewall policy destination address. In many cases you can do this by setting the firewall policy destination address to all.

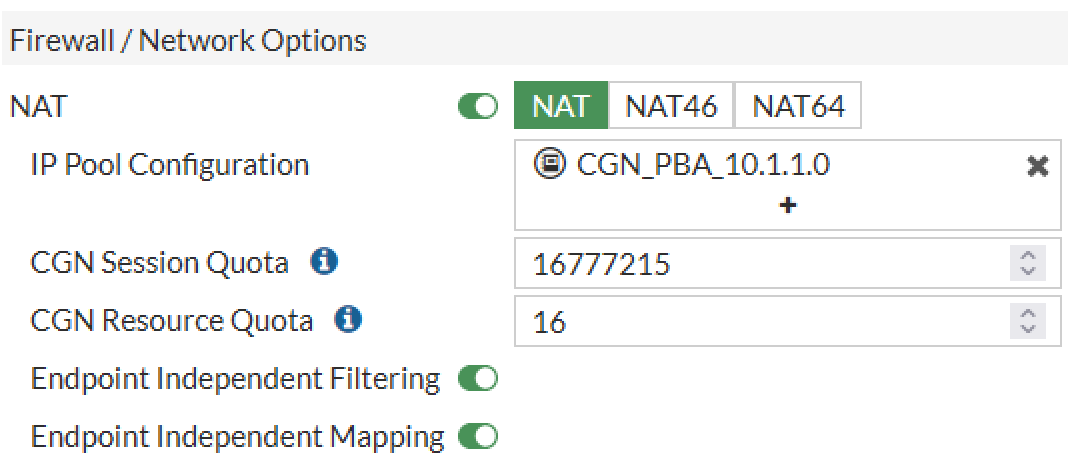
If the policy uses a specific address or address range for the destination address, then this destination address and the IP pool address range should have some overlap.

`cgn-log-server-grp` the name of the hardware logging server group. See Hardware logging on page 48.

From the GUI

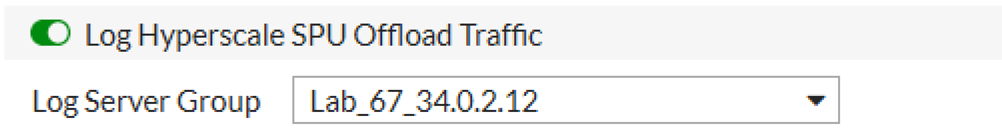
Use the following steps to add CGNAT firewall policies to a hyperscale firewall VDOM from the GUI:

1. Go to **Policy & Objects** and select **Firewall Policy**.
2. Configure source and destination interfaces and addresses and other standard firewall options as required.
3. If you are configuring an IPv4 or NAT64 hyperscale firewall policy you can also configure the following CGN resource allocation options:
   - **IP Pool Configuration** select one or more CGN resource allocation IP pools or CGN resource allocation IP pool groups. All of the IP pools or IP pool groups must have the same mode and their source IP addresses must not overlap.
   - **CGN Session Quota** to limit the concurrent sessions available for a source IP address.
   - **CGN Resource Quota** to limit the number of port blocks assigned to a source IP address.
   - Enable or disable **Endpoint Independent Filtering**.
   - Enable or disable **Endpoint Independent Mapping**.



4. Optionally enable hardware logging by selecting **Log Hyperscale SPU Offload Traffic** and selecting a **Log Server**

Group.



Overload PBA port-reuse limitation for traffic between a single source and destination IP address

Because of an NP7 hardware limitation, port-reuse does not work as expected when processing multiple sessions between a single client IP address and a single server IP address when using an overload with port-block-allocation CGN IP pool. The hardware limitation prevents the NP7 processor from establishing all of the required sessions and some sessions will time out sooner than expected.

It is very unlikely for this condition to occur. A client is most likely to always be connecting to many different servers. If they are connecting multiple times to the same server, they are most likely using multiple server ports.

Here are three possible ways to resolve the issue:

- Use a non-overload PBA CGN IP pool.
- Use an overload PBA CGN IP pool but reduce the `ippool-overload-high` threshold:
  ```
  config system npu
     set ippool-overload-high <threshold>
  end
  ```
  The default `<threshold>` is 200, for example, you could reduce the threshold to 100.
- Change the network to increase the number of client or server IP addresses.

Overload PBA resource quota limitation

Because of an NP7 hardware limitation, for CGN traffic accepted by a hyperscale firewall policy that includes an overload with port block allocation CGN IP Pool, only one block is allocated per client. The setting of the hyperscale firewall policy CGN Resource Quota (`cgn-resource-quota`) is ignored.

Because of this limitation, under certain rare conditions (for example, only a single server side IP address and port are being used for a large number of sessions), port allocation may fail even if the block usage of the client is less than its quota.

Here are two possible ways to resolve this issue:

- In cases such as this, if the client has traffic towards some other servers or ports, additional port allocation can become successful.
- You can also work around this problem by increasing the IP Pool block size (`cgn-block-size`).

Hyperscale firewall policy engine limitations and mechanics

The NP7 hyperscale firewall policy engine is also called the Policy Lookup Engine (PLE). The PLE handles processing of all hyperscale firewall policies in all hyperscale firewall VDOMs. When the hyperscale firewall policy configuration changes, the PLE compiler creates a new hyperscale policy database (also called a policy set) that is used by each NP7 processor to apply hyperscale firewall and carrier grade NAT (CGN) features to offloaded traffic. The hyperscale policy database includes all of the hyperscale firewall policies and the firewall objects added to those policies.

Based on internal testing and testing in customer environments, Fortinet has developed some functional limitations for the number of firewall policies, address ranges, and port ranges that can be loaded into the hyperscale policy database and compiled by the PLE. The functional limitations described in this section are for guidance purposes and are not enforced by software. These functional limitations are also independent of the tablesize or maximum values for your FortiGate. Standard tablesize limits for firewall objects apply to hyperscale firewall VDOMs and FortiGates licensed for hyperscale firewall. The tablesize system imposes hard limits on the number of firewall objects that you can create.

Table size limits or maximum values that refer to "hyperscale-policy" are no longer valid since FortiOS no longer distinguishes hyperscale policies from non-hyperscale policies.

In addition, FortiOS limits the number of firewall policies that can be added to a hyperscale VDOM to 15,000. This limit applies to all firewall policies and is imposed by preventing you from adding a firewall policy with a policy ID higher than 15,000. The number 15,000 was selected based on optimizing performance of the PLE and hyperscale policy database and could be changed in the future.

Because the tablesize values are independent of hyperscale firewall database limitations, and because the 15,000 policy limit is per-hyperscale firewall VDOM, you may be able to create enough firewall objects in hyperscale firewall policies to exceed hyperscale firewall database functional limitations. When you make changes to the hyperscale firewall configuration, a new hyperscale policy database is compiled. It's possible that the database may not compile successfully if some limitations are exceeded. If this happens, FortiOS writes an error message and you can contact Fortinet Support to help troubleshoot the problem.

The limitations described in this section are hardware limitations imposed by NP7 processors and software limitations imposed by the PLE. Because the entire hyperscale firewall database has to be handled by one NP7 processor, these limitations apply to all FortiGates licensed for hyperscale firewall. The limitations are independent of FortiGate model, max values, system memory, CPU capacity, and number of NP7 processors.

Hyperscale firewall VDOMs do not support the FortiOS Internet Service Database (ISDB), IP Reputation Database (IRDB), and IP Definitions Database (IPDB) features.

About the 15,000 policy per hyperscale VDOM limit

The limit of 15,000 policies per hyperscale VDOM is enforced by not allowing you to create a firewall policy with an ID higher than 15,000. If you attempt to add a firewall policy to a hyperscale firewall VDOM with a policy ID of higher than 15000, an error message similar to the following appears.

```
# config firewall policy
# edit 15003
```

```
Maximum policy ID is 15000 in hyperscale policy!
node_check_object fail! for policyid 15003

value parse error before '15003'
Command fail. Return code -651
```

The limit of 15,000 is per hyperscale firewall VDOM and applies to all firewall policies in a hyperscale firewall VDOM; whether or not those firewall policies are hyperscale firewall policies. This limit is the same for all FortiGate models.

The number 15,000 was selected based on optimizing performance of the PLE and hyperscale policy database and could be changed in the future. The 15,000 policy limit is not part of the tablesize/max values system.

Per hyperscale policy limits

The following are per hyperscale policy limits for the numbers of IP addresses and subnets that are supported by the hyperscale firewall database. These limitations have been tested for FortiOS 7.0.19 and may be changed in the future as a result of ongoing and future optimizations.

- The maximum number of port-ranges specified by firewall addresses that can be added to a single hyperscale firewall policy: 1,000.
- The 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. Example subnet: 5.2.226.0/24
- 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. Example duplicate range IP address: start-ip/end-ip 118.1.1.152.
- An IPv6 hyperscale firewall policy can have up to 20 IPv6 IP addresses distributed between the source and destination address fields.

Examples of overlapping subnets:

The subnet 5.2.227.0/24 can also be written as 5.2.227.0 - 5.2.227.255.

Subnets 5.2.227.0/24 5.2.227.0 /28 are examples of overlapping subnets. The subnet 5.2.227.3 /32 overlaps with 5.2.227.0/24 and 5.2.227.0 /28 as well as including unique IP addresses that don't overlap with the other two subnets.

IP range overlapping: The IP range 5.2.226.3 - 5.2.227.10 partially overlaps with 5.2.227.0 - 5.2.227.255.

In general, the more overlaps like these that are present the more complex the compiled hyperscale firewall database becomes and the more likely that the PLE will be unable to compile the database.

Global hyperscale policy limits

The following are global limits for all hyperscale VDOMs and are not per individual VDOM. These limitations have been tested for FortiOS 7.0.19 and may be changed in the future as a result of ongoing and future optimizations.

- The maximum number of hyperscale firewall policies allowed in the hyperscale policy database: 20,000.
- The maximum number of IP address ranges specified by firewall addresses that can be added to a single hyperscale firewall policy: 2,000. There is no limitation on the number of IP addresses in the address ranges and the

sizes of the address ranges does not affect the maximum number of 2,000.
- The maximum number of IP address ranges that can be added to the hyperscale policy database: 32,000.

Additional considerations

The factors that affect whether a hyperscale policy database can be compiled or not includes but are not limited to:

- The total number of hyperscale firewall policies.
- The total number of IP address ranges and port-ranges as defined by firewall addresses added to hyperscale firewall policies in the firewall policy database.
- The relationship between policies, such as how IP address ranges are distributed among hyperscale firewall policies.

It is possible to create a hyperscale policy database that is within the limitations described in this section, but that cannot be compiled. If this happens, FortiOS will create an error message when the policy database is compiled. When this happens, the new hyperscale policy database cannot be used so the previous hyperscale policy database remains in operation. If you receive an error message during policy compilation, contact Fortinet Support for assistance diagnosing and correcting the problem.

You can also create a policy database that exceeds some or all of the limits listed in this section, but can be successfully compiled. If you plan to create a configuration with one or more parameters close to or above their maximum values, you should contact Fortinet Support to review your configuration before deploying it.

It is a best practice to restart your FortiGate after making significant changes to a hyperscale policy database, especially if one or more parameters are close to or above the limitations described in this section.

Hyperscale policy database complexity and performance

The complexity of your hyperscale firewall policy database affects how long it takes for your FortiGate to start up. In general, more complex hyperscale policy databases result in longer start up times.

The complexity of your hyperscale firewall policy database also affects your FortiGate's hyperscale connections per second (CPS) performance. In general, more complex policy databases result in lower CPS performance.

How hyperscale policy database changes are implemented while the FortiGate is processing traffic

The complexity of your hyperscale firewall policy database affects how long it takes after inputting a policy change before the updated policy database can be applied to new and established sessions. This period of time is called the preparation time.

During the preparation time, new sessions are evaluated with the current hyperscale policy database.

Access control list (ACL) policies added to a hyperscale firewall VDOM that is processing traffic may take longer than expected to become effective. During a transition period, traffic that should be blocked by the new ACL policy will be allowed.

After the preparation time, new sessions are evaluated with the new hyperscale policy database. Established sessions are also re-evaluated with the new hyperscale policy database. The time required to re-evaluate established sessions is called the transition time. CPS performance can be reduced during the transition time.

The transition time is affected by hyperscale policy database complexity, the total number of established sessions to be re-evaluated, and by the rate that the system is receiving new sessions.

During the transition time, FortiOS terminates an established session if:

- The session is matched with a policy that has a different policy search key (for example, a different source IP range) or policy action.
- The session is matched with the same policy but the policy includes a resource, such as an IP pool, that dynamically assigns a value (for example, an IP address) to the session and now it has to be returned because of the policy change.

Hardware logging

You can configure NP7 processors to create traffic or NAT mapping log messages for hyperscale firewall sessions and send them to remote NetFlow or Syslog servers. Hardware logging is supported for IPv4, IPv6, NAT64, and NAT46 hyperscale firewall policies. Full NetFlow is supported through the information maintained in the firewall session.

Hardware logging also handles hyperscale VDOM software session logs (that is hyperscale VDOM sessions handled by the kernel/CPU). As part of the hardware logging configuration, you can configure software session logging to log TCP and UDP software sessions or all software sessions. You can also disable software session logging. Software session logging uses `per-session` logging, which creates two log messages per session, one when the session is established and one when the session ends. Software session logging supports NetFlow v10 and syslog log message formats.

Hardware logging features include:

- On some FortiGate models with NP7 processors you can configure hardware logging to either use the NP7 processors to create and send log messages or you can configure hardware logging to use FortiGate CPU resources to create and send hardware log messages. Using the NP7 processors to create and send log messages improves performance. Using the FortiGate CPU for hardware logging is called host logging. Each option has some limitations, see Configuring hardware logging on page 49.
- Per session logging creates two log messages per session; one when the session is established and one when the session ends.
- Per session ending logging creates one log message when the session ends. This log message includes the session duration, allowing you to calculate the session start time. Per session ending logging may be preferable to per session logging because fewer log message are created, but the same information is available.
- Per NAT mapping logging, creates two log messages per session, one when the session allocates NAT mapping resources and one when NAT mapping resources are freed when the session ends.
- By default, log messages are sent in NetFlow v10 format over UDP. NetFlow v10 is compatible with IP Flow Information Export (IPFIX).
- NetFlow v9 logging over UDP is also supported. NetFlow v9 uses a binary format and reduces logging traffic.
- Syslog logging over UDP is also supported.
- To configure hardware logging, you create multiple log server groups to support different log message formats and different log servers.
- Round-robin load balancing distributes log messages among the log servers in a log server group to reduce the load on individual log servers. A log server group can contain up to 16 log servers. All messages generated by a given session are sent to the same log server.
- You can also configure multicast-mode hardware logging to send all log messages to multiple log servers.
- Hyperscale deny log messages are generated by hardware logging and not by the CPU and are sent to the same servers as other hardware log messages. Depending on your hardware logging configuration, this can be netflow or syslog servers. Hardware deny log messages are not sent to FortiAnalyzer.
- Hardware logging log messages are similar to most FortiGate log messages but there are differences that are specific to hardware logging messages. For example, the `dur` (duration) field in hardware logging messages is in milliseconds (ms) and not in seconds.
- Hardware logging is supported for protocols that use session helpers or application layer gateways (ALGs). If hyperscale firewall policies accept session helper or ALG traffic, for example, ICMP traffic, hardware log messages for these sessions are created and sent according to the hardware logging configuration for the policy. For more information, see ALG/Session Helper Support.

Configuring hardware logging

Use the global `config log npu-server` command to configure global hardware logging settings, add hardware log servers, and create log server groups. The hardware logging configuration is a global configuration that is shared by all of the NP7s and is available to all hyperscale firewall VDOMs.

```
config log npu-server
   set log-processor {hardware | host}
   set log-processing {may-drop | no-drop}
   set netflow-ver {v9 | v10}
   set syslog-facility <facility>
   set syslog-severity <severity>
      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>
            set template-tx-timeout <timeout>
         end
      config server-group
         edit <group-name>
            set log-mode {per-session | per-nat-mapping | per-session-ending}
            set log-format {netflow | syslog}
            set log-tx-mode {roundrobin | multicast}
            set sw-log-flags {tcp-udp-only | enable-all-log | disable-all-log}
            set log-user-info {disable | enable}
            set log-gen-event {disable | enable}
            set server-number <number>
            set server-start-id <number>
         end
```

Global hardware logging settings

Global hardware logging settings control how hardware logs are handled (by NP7 processors or by the CPU) and control global log settings such as the NetFlow version.

```
config log npu-server
   set log-processor {hardware | host}
   set log-processing {may-drop | no-drop}
   set netflow-ver {v9 | v10}
   set syslog-facility <facility>
   set syslog-severity <severity>
end
```

`log-processor` select whether to use NP7 processors (`hardware`, the default) or the FortiGate CPUs (`host`) (called host logging) to generate traffic log messages for hyperscale firewall sessions.

If you set this option to `hardware`, the following limitations apply:

- The interface through which your FortiGate communicates with the remote log server must be connected to your FortiGate's NP7 processors. Depending on the FortiGate model, this usually this means you can't use a management or HA interface to connect to the remote log server. See FortiGate NP7 architectures for information about the interfaces that are connected to NP7 processors and the interfaces are not for your FortiGate model.

- The interface through which your FortiGate communicates with the remote log server can be in any VDOM and does not have to be in the hyperscale VDOM that is processing the traffic being logged.
- The interface through which a FortiGate 4800F or 4801F communicates with the remote log server must be in a hyperscale firewall VDOM. This can be the VDOM that is processing the traffic being logged, or another VDOM assigned to the same NP7 processor group, see NP7 processor groups and hyperscale hardware logging.
- The `vd=` field in generated traffic log messages includes the VDOM name followed by trailing null characters. If possible, you can configure your syslog server or NetFlow server to remove these trailing null characters.
- Normally the `PID=` field in traffic log messages contains the policy ID of the firewall policy that generated the log message. But, if the policy that generated the traffic log message has recently changed, the `PID=` field can contain extra information used by the NP7 policy engine to track policy changes. You can extract the actual policy ID by converting the decimal number in the `PID=` field to hexadecimal format and removing all but the last 26 bits. These 26 bits contain the policy ID in hexadecimal format. You can convert this hex number back to decimal format to generate the actual policy ID.
- If log-mode is set to `per-session`, NP7 hardware logging may send multiple session start log messages, each with a different start time. Creating multiple session start log messages is a limitation of NP7 processor hardware logging, caused by the NP7 processor creating extra session start messages if session updates occur. You can work around this issue by using host logging or by setting `log-mode` to `per-session-ending`. This setting creates a single log message when the session ends. This log message records the time the session ended as well as the duration of the session. This information can be used to calculate the session start time.

If you set this option to `host`, all hardware logging functions are supported. There are no restrictions on the interface through which your FortiGate communicates with the remote log server. With host logging enabled, NP7 processors send session information to the CPU, which maintains a session table of NP7 sessions and software sessions in software and system memory. Host logging then uses this session table to create log messages. Host logging has the following limitations:

- Setting `log-processor` to `host` can reduce overall FortiGate performance because the FortiGate CPUs handle hardware logging instead of offloading logging to the NP7 processors.
- Host logging may not provide the NHI, stats, OID, gateway, expiration, and duration information for short-lived sessions.
- Host logging does not support Netflow v9.

`log-processing {may-drop | no-drop}` change how the FortiGate queues CPU or host logging packets to allow or prevent dropped packets. This option is only available if `log-processor` is set to host. In some cases, hyperscale firewall CPU or host logging packets can be dropped, resulting in lost log messages and incorrect traffic statistics.

- `may-drop` the default CPU or host log queuing method is used. Log message packet loss can occur if the FortiGate is very busy.
- `no-drop` use an alternate queuing method that prevents packet loss.

`netflow-ver {v9 | v10)` select the version of NetFlow that this log server is compatible with. `v10`, which is compatible with IP Flow Information Export (IPFIX), is the default. `host` hardware logging does not support NetFlow v9.

`syslog-facility` set the syslog facility number added to hardware log messages. The range is 0 to 255. The default is 23 which corresponds to the local7 syslog facility.

`syslog-severity` set the syslog severity level added to hardware log messages. The range is 0 to 255. The default is 5, which corresponds to the notice syslog severity.

Hardware logging servers

You can use the config server-info command to add up to 16 log servers. The log server configuration includes the information that the FortiGate uses to communicate with a log server. This includes the name of the VDOM through which the FortiGate can communicate with the log server, and the IPv4 or IPv6 IP address of the log server. Once you have added log servers, you can add them to one or more log server groups.

```
config log npu-server
  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>
      set template-tx-timeout <timeout>
    end
```

edit <index> create a log server. <index> is the number of the log server. You use this number when you add the log server to a server group. <index> can be 1 to 16. You must specify the number, setting <index> to 0 to select the next available number is not supported.

vdom the virtual domain that contains the FortiGate interface that you want to use to communicate with the log server. If log-processor is set to hardware, the VDOM must include an interface connected to NP7 processors, you must use an interface connected to an NP7 processor for hardware logging. Usually this means you cannot select a management VDOM. If log-processor is set to host, you can select any VDOM.

On a FortiGate 4800F or 4801F, hyperscale hardware logging servers must include a hyperscale firewall VDOM. This VDOM must be assigned the same NP7 processor group as the hyperscale firewall VDOM that is processing the hyperscale traffic being logged. This can be the same hyperscale VDOM or another hyperscale firewall VDOM that is assigned the same NP7 processor group.

For more information, see Enabling hyperscale firewall features on page 17 and NP7 processor groups and hyperscale hardware logging.

ip-family the IP version of the remote log server. v4 is the default.

ipv4-server the IPv4 address of the remote log server.

ipv6-server the IPv6 address of the remote log server.

source-port the source UDP port number added to the log packets in the range 0 to 65535. The default is 514.

dest-port the destination UDP port number added to the log packets in the range 0 to 65535. The default is 514.

template-tx-timeout the time interval between sending NetFlow template packets. NetFlow template packets communicate the format of the NetFlow messages sent by the FortiGate to the NetFlow server. Since the message format can change if the NetFlow configuration changes, the FortiGate sends template updates at regular intervals to make sure the server can correctly interpret NetFlow messages. The timeout range is from 60 to 86,400 seconds. The default timeout is 600 seconds.

Hardware logging server groups

Configure hardware logging server groups to group the hardware logging servers that receive logs from traffic accepted by a hyperscale firewall policy. To add hardware logging for hyperscale firewall traffic, you add a log server group to a hyperscale firewall policy.

You also use the log server group to configure the number of log messages sent for each session, the log format (NetFlow or syslog), how software sessions are logged, whether log messages are distributed to the log servers in the server group or simultaneously sent to all log servers in the server group, and to select the log servers added to the log server group.

```
config log npu-server
   config server-group
      edit <group-name>
         set log-mode {per-session | per-nat-mapping | per-session-ending}
         set log-format {netflow | syslog}
         set log-tx-mode {roundrobin | multicast}
         set sw-log-flags {tcp-udp-only | enable-all-log | disable-all-log}
         set log-user-info {disable | enable}
         set log-gen-event {disable | enable}
         set server-number <number>
         set server-start-id <number>
      end
```

`log-mode` select a log mode:

- `per-session` (the default) create two log messages per session, one when the session is established and one when the session ends. If `log-processor` is set to `hardware`, NP7 processors may incorrectly create multiple session start messages due to a hardware limitation.
- `per-nat-mapping` create two log messages per session, one when the session allocates NAT mapping resources and one when NAT mapping resources are freed when the session ends.
- `per-session-ending` create one log message when a session ends. This log message includes the session duration, allowing you to calculate the session start time. `per-session-ending` logging may be preferable to `per-session` logging because fewer log message are created, but the same information is available.

`log-format {netflow | syslog}` select the log message format. You can select `netflow` or `syslog`. If you select `netflow`, the global hardware log `netflow-ver` setting determines the NetFlow version (v9 or v10) of the log messages.

`log-tx-mode {roundrobin | multicast}` select `roundrobin` (the default) to load balance log messages to the log servers in the server group. Select `multicast` to enable multicast-mode logging. Multicast-mode logging simultaneously sends log messages to all of the log servers in the server group.

`sw-log-flags {tcp-udp-only | enable-all-log | disable-all-log}` configure how software session logs are handled by the log server group. Software session logging uses `per-session` logging, which creates two log messages per session, one when the session is established and one when the session ends. Software session logging supports NetFlow v10 and syslog log message formats.

- `tcp-udp-only` log only TCP and UDP software sessions (the default).
- `enable-all-log` log all software sessions.
- `disable-all-log` disable software session logging for this log server group.

`log-user-info` enable to include user information in log messages. This option is only available if `log-format` is set to `syslog`.

`log-gen-event` enable to add event logs to hardware logging. This option is only available if `log-format` is set to `syslog` and `log-mode` is set to `per-nat-mapping` to reduce the number of log messages generated.

`server-number` the number of log servers, created using `config server-info`, in this log server group. The range is 1 to 16 and the default is 0 and must be changed.

`server-start-id` the ID of one of the log servers in the `config server-info` list. The range is 1 to 16 and the default is 0 and must be changed.

Use `server-number` and `server-start-id` to select the log servers to add to a log server group. You can add the same log server to multiple log server groups.

For example, if you have created five log servers with IDs 1 to 5:

```
config server-info
   edit 1
      set vdom Test-hw12
      set ipv4-server 10.10.10.20
   end
   edit 2
      set vdom Test-hw12
      set ipv4-server 10.10.10.21
   end
   edit 3
      set vdom Test-hw12
      set ipv4-server 10.10.10.22
   end
   edit 4
      set vdom Test-hw12
      set ipv4-server 10.10.10.23
   end
   edit 5
      set vdom Test-hw12
      set ipv4-server 10.10.10.24
   end
```

You can add the first three log servers (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.

```
config server-group
   edit test-log-11
      set server-number 3
      set server-start-id 1
   end
```

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 this log server group.

```
config server-group
   edit test-log-12
      set server-number 2
      set server-start-id 4
   end
```

To add all of the log servers to a third log server group, set `server-number` to 5 and `server-start-id` to 1. This adds log servers 1 to 5 to the this log server group.

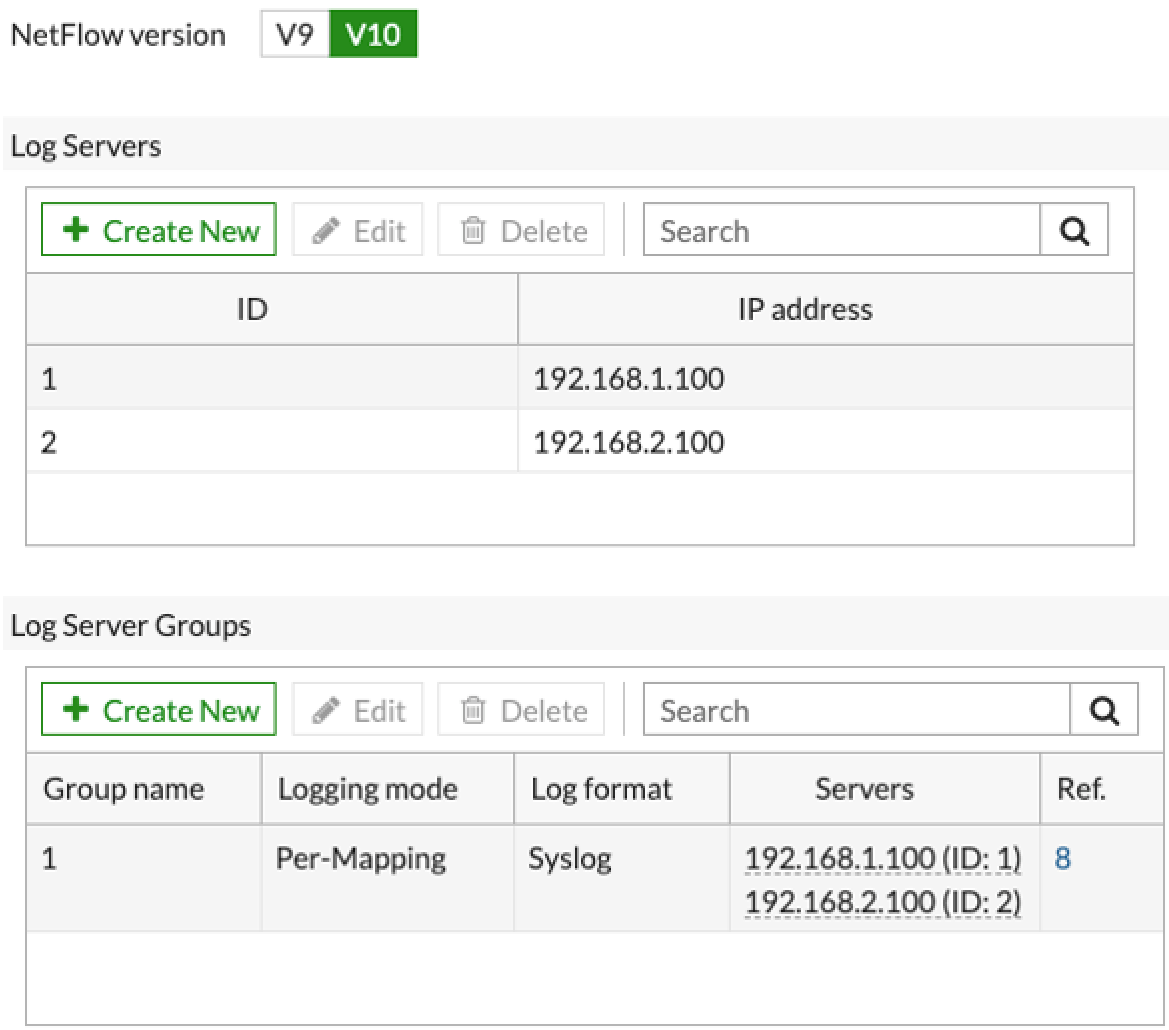
```
config server-group
   edit test-log-13
      set server-number 5
      set server-start-id 1
```

```
end
```

From the GUI

You can configure hardware logging from the Global GUI.

1. Go to **Log & Report > Hyperscale SPU Offload Log Settings**.
2. If `log-processor` is set to `hardware` you can select the **Netflow version**.
3. Under **Log Servers**, select **Create New** to create a log server.
4. Select the **Virtual Domain** containing the interface that can communicate with the log server.
5. Select the **IP version** supported by the log server and enter the log server **IP address** or **IPv6 address**.
6. Enter the **Source port** and **Destination port** to be added to the log message packets.
7. Set the **Template transmission timeout**, or the time interval between sending NetFlow template packets.
8. Select **OK** to save the log server.
9. Repeat to add more log servers.
10. Under **Log Server Groups** select **Create New** to add a log server group.
11. Enter a **Name** for the log server group.
12. Select the **Logging Mode** and **Log format**.
13. Add one or more **Log servers**.
14. Select **OK** to save the log server group.
15. Select **Apply** to apply your hardware logging changes.



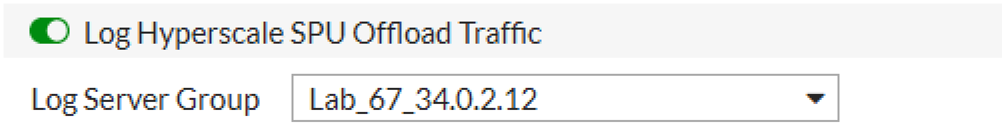
Adding hardware logging to a hyperscale firewall policy

Use the following command to enable hardware logging in a hyperscale firewall policy and assign a hardware logging server group to the firewall policy.

```
config firewall policy
   edit <id>
      set policy-offload {enable | disable}
      set cgn-log-server-grp <group-name>
   end
```

From the GUI:

1. Go to **Policy & Objects > Firewall Policy** and create a new or edit a firewall policy.
2. While configuring the policy, select **Log Hyperscale SPU Offload Traffic**.
3. Select a **Log Server Group**.



When configuring hardware logging, the recommended or required IP addresses of the hardware logging servers that you can use with hyperscale firewall policies are the following:

- You should only use logging servers that have IPv4 addresses with IPv4 hyperscale firewall policies. Logging servers with IPv6 IP addresses can be used but are not recommended.
- You should only use logging servers that have IPv6 addresses with IPv6 hyperscale firewall policies. Logging servers with IPv4 IP addresses can be used but are not recommended.
- You can only use logging servers that have IPv6 addresses with NAT64 hyperscale firewall policies.
- You can only use logging servers that have IPv4 addresses with NAT46 hyperscale firewall policies.

You can add hardware logging to hyperscale policies with action set to deny. Hyperscale deny log messages are generated by hardware logging and not by the CPU and are sent to the same servers as other hardware log messages. Depending on your hardware logging configuration, this can be netflow or syslog servers. Hyperscale deny log messages are not sent to FortiAnalyzer. For more information, see Hardware logging for hyperscale firewall policies that block sessions on page 58.

Multicast-mode hardware logging example

You can use multicast-mode hardware logging to simultaneously send hardware log messages to multiple remote syslog or NetFlow servers.

Enable multicast-mode logging by creating a log server group that contains two or more remote log servers and then set `log-tx-mode` to `multicast`:

```
config log npu-server
   set log-processor {hardware | host}
      config server-group
         edit "log_ipv4_server1"
            set log-format {netflow | syslog}
            set log-tx-mode multicast
         end
```

The following example shows how to set up two remote syslog servers and then add them to a log server group with multicast-mode logging enabled. This configuration is available for both NP7 (hardware) and CPU (host) logging.

```
config log npu-server
   set log-processor {hardware | host}
      config server-info
```

```
      edit 1
         set vdom "root"
         set ipv4-server <server-ip>
         set source-port 8055
         set dest-port 2055
         set template-tx-timeout 60
      next
      edit 2
         set vdom "root"
         set ipv4-server <server-ip>
         set source-port 8055
         set dest-port 2055
         set template-tx-timeout 60
      end
   end
config server-group
   edit "Example-Multicast"
      set log-format syslog
      set log-tx-mode multicast
      set server-number 2
      set server-start-id 1
   end
```

Include user information in hardware log messages

You can configure host hardware logging to include user information in hardware log messages to record information about logged in users accessing hyperscale firewall features.

Only host hardware logging supports including user information in hardware log messages. As well, this feature is only supported for syslog messages.

CLI syntax to add user information to hardware log messages. Enable `log-user-info` to include user information in log messages

```
config log npu-server
   set log-processor host
      config server-group
         edit <group-name>
            set log-mode {per-session | per-nat-mapping | per-session-ending}
            set log-format syslog
            set log-user-info enable
         end
```

Adding event logs to hardware logging

Only CPU or host hardware logging supports adding event logs to hardware log messages. As well, event log messages are only supported when the log mode is set to per NAT mapping. Per NAT mapping creates two log messages per session, one when the session allocates NAT mapping resources and one when NAT mapping resources are freed when the session ends.

CLI syntax to add event logs to hardware logging. Enable `log-gen-event` to add event logs to hardware logging. This option is only available if `log-format` is set to `syslog` and `log-mode` is set to `per-nat-mapping` to reduce the number of log messages generated.

```
config log npu-server
   set log-processor host
      config server-group
         edit <group-name>
            set log-mode per-nat-mapping
            set log-format syslog
            set log-gen-event enable
         end
```

Hardware logging for hyperscale firewall policies that block sessions

Hardware logging supports the following features related to hyperscale firewall policies that block sessions, that is hyperscale firewall policies with action set to deny:

- You can enable hardware logging for hyperscale firewall policies with action set to deny. Hardware logging creates a log message for each session that is blocked.
- Hardware session information includes information about whether the session blocked traffic. For example, when displaying session information from the CLI, a field similar to the following appears to indicate that the session blocked traffic: `Session action (DROP/TO-HOST): DROP`.
- Because these hyperscale deny log messages are generated by hardware logging and not by the CPU, they are sent to the same servers as other hardware log messages. Depending on your hardware logging configuration, this can be netflow or syslog servers. Hardware deny log messages are not sent to FortiAnalyzer.

Hardware log messages indicate if the session accepted or denied traffic. For example:

- Example log messages for a policy that accepts traffic:

  ```
  Oct 5 23:29:33 172.16.200.26 date=2022-10-06 time=02:29:32 sn=F2K61FTK21900840 vd=cgn-hw1 pid=805306369 type=sess act=start tran=snat proto=6 ipold=v4 ipnew=v4 sip=10.1.100.11 dip=172.16.200.155 sport=40836 dport=80 nsip=172.16.201.182 ndip=172.16.200.155 nsport=8117 ndport=80 sentp=0 sentb=0 rcvdp=0 rcvdb=0
  ```

  ```
  Oct 5 23:29:36 172.16.200.26 date=2022-10-06 time=02:29:35 sn=F2K61FTK21900840 vd=cgn-hw1 pid=805306369 type=sess act=end tran=snat proto=6 ipold=v4 ipnew=v4 sip=10.1.100.11 dip=172.16.200.155 sport=40836 dport=80 nsip=172.16.201.182 ndip=172.16.200.155 nsport=8117 ndport=80 dur=2936 sentp=6 sentb=398 rcvdp=4 rcvdb=1307
  ```

  Decimal version of the pid = 805306369

  Binary version of the pid = 0**0**11 0000 0000 0000 0000 0000 0000 0001

  pid[30] is '**0**' for accept action (count from bit0 to bit31 and right to left)

- Example log messages for a policy that blocks or denies traffic:

  ```
  Oct 5 23:31:49 172.16.200.26 date=2022-10-06 time=02:31:49 sn=F2K61FTK21900840 vd=cgn-hw1 pid=1946157057 type=sess act=start tran=none proto=6 ipold=v4 ipnew=v4 sip=10.1.100.11 dip=172.16.200.155 sport=40837 dport=80 nsip=10.1.100.11 ndip=172.16.200.155 nsport=40837 ndport=80 sentp=0 sentb=0 rcvdp=0 rcvdb=0
  ```

```
Oct 5 23:32:02 172.16.200.26 date=2022-10-06 time=02:32:01 sn=F2K61FTK21900840
vd=cgn-hw1 pid=1946157057 type=sess act=end tran=none proto=6 ipold=v4 ipnew=v4
sip=10.1.100.11 dip=172.16.200.155 sport=40837 dport=80 nsip=10.1.100.11
ndip=172.16.200.155 nsport=40837 ndport=80 dur=12719 sentp=2 sentb=120 rcvdp=0
rcvdb=0
```

Decimal version of the pid = 1946157057

Binary version of the pid = 0**1**11 0100 0000 0000 0000 0000 0000 0001

pid[30] is ‘**1**’ for deny action (count from bit0 to bit31 and right to left)

FGCP HA hardware session synchronization

When configuring active-passive FortiGate Clustering Protocol (FGCP) HA or active-passive FGCP virtual clustering for two FortiGates with hyperscale firewall support, you can use FGCP HA hardware session synchronization to synchronize NP7 sessions between the FortiGates in the cluster. FGCP HA hardware session synchronization is only supported between two FortiGates.

In an active-passive FGCP cluster, HA hardware session synchronization copies sessions from the primary FortiGate to the secondary FortiGate. Both FortiGates maintain their own session tables with their own session timeouts. FGCP HA hardware session synchronization does not compare FortiGate session tables to keep them synchronized. In some cases you may notice that the secondary FortiGate in the HA cluster may have a lower session count than the primary FortiGate. This is a known limitation of FGCP HA hardware session synchronization. Normally the difference in session count is relatively minor and in practice results in very few lost sessions after a failover.

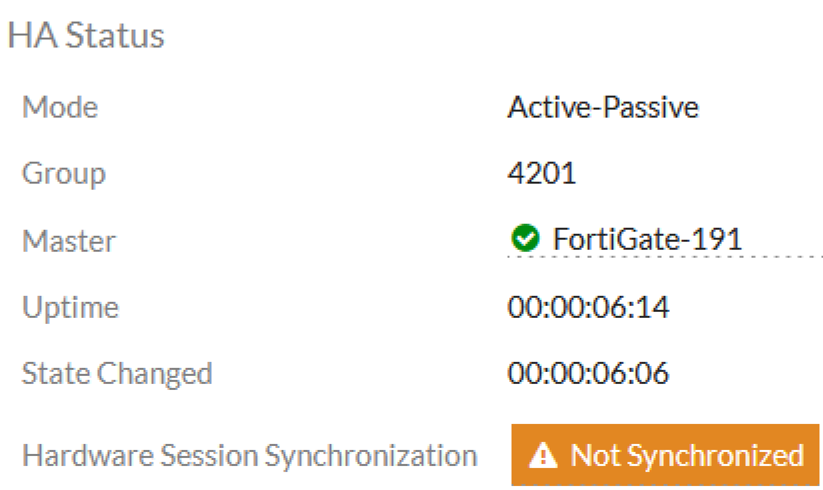
In an active-passive FGCP virtual clustering configuration, FGCP HA hardware session synchronization copies sessions from VDOMs processing traffic to VDOMs on the other FortiGate in the virtual cluster that are not processing traffic. All VDOM instances maintain their own session tables with their own session timeouts. FGCP HA hardware session synchronization does not compare VDOM session tables between FortiGates to keep them synchronized.

FGCP HA hardware session synchronization packets are the same as standard session synchronization packets. For FGCP HA they are layer 2 TCP and UDP packets that use destination port 703. FGCP HA does not require you to add IP addresses to the interfaces that you use for HA hardware session synchronization.

HA hardware session synchronization is not supported for active-active HA.

FGSP HA hardware session synchronization is supported, see FGSP HA hardware session synchronization on page 64.

The HA Status dashboard widget shows hardware session synchronization status.



Configuring FGCP HA hardware session synchronization

Use the following command to configure FGCP HA hardware session synchronization.

```
config system ha
   set session-pickup enable
   set hw-session-sync-dev <interface>
end
```

`session-pickup` must be enabled for FGCP HA hardware session synchronization.

`hw-session-sync-dev` select an interface to use to synchronize hardware sessions between the FortiGates in an FGCP cluster. Fortinet recommends using a data interface or a data interface LAG as the FGCP HA hardware session synchronization interface. The interface or LAG can only be used for FGCP HA hardware session synchronization. See Recommended interface use for an FGCP HA hyperscale firewall cluster on page 63.

Use the following configuration to create a data interface LAG. The members of the LAG can be any data interfaces that can be added to LAGs as supported by your FortiGate model.

```
config system interface
   edit HA-session-lag
      set type aggregate
      set lacp-mode static
      set member port13 port14 port15 port16
   end
```

You can only use a static mode LAG as the hardware session synchronization interface. When configuring the `hw-session-sync-dev` from the CLI, along with physical interfaces, only LAGs with `lacp-mode` set to `static` will appear in the list of interfaces that you can select.

The CLI blocks you from changing the `lacp-mode` of the `hw-session-sync-dev` LAG to `dynamic` or `passive`.

Use the following command to set the LAG as the FGCP HA hardware session synchronization interface.

```
config system ha
   set session-pickup enable
   set hw-session-sync-dev HA-session-lag
end
```

If you use a LAG as the hardware session synchronization interface, the LAG cannot be monitored by FGCP HA interface monitoring.

For the FortiGate 1800F and 1801F, you can only use the port25 to port40 interfaces as FGCP HA hardware session synchronization interfaces. Also on the FortiGate 1800F and 1801F, you can't use a LAG interface as the hardware session synchronization interface if the LAG includes interfaces that can't be used for hardware session synchronization.

Hardware session synchronization can use a lot of bandwidth so you should use a dedicated data interface or data interface LAG. Both FortiGates in the FGCP HA cluster must use the same data interface or data interface LAG for FGCP HA hardware session synchronization and these interfaces must be directly connected.

FGCP HA hardware session synchronization timers

You can use the following options to set timers associated with hardware session synchronization after an FGCP HA failover:

```
config system ha
   set hw-session-hold-time <seconds>
   set hw-session-sync-delay <seconds>
end
```

`hw-session-hold-time` the amount of time in seconds after a failover to hold hardware sessions before purging them from the new secondary FortiGate. The range is 0 to 180 seconds. The default is 10 seconds.

`hw-session-sync-delay` the amount of time to wait after a failover before the new primary FortiGate synchronizes hardware sessions to the new secondary FortiGate. The range is 0 - 3600 seconds. The default is 150 seconds.

After an HA failover, the new secondary FortiGate waits for the `hw-session-hold-time` and then purges all sessions and frees up all resources. Then, after the `hw-session-sync-delay`, the new primary FortiGate synchronizes all hardware sessions to the new secondary FortiGate. The `hw-session-sync-delay` gives the new secondary FortiGate enough time to finish purging sessions and freeing up resources before starting session synchronization.

The default configuration means that there is a 150 second delay before sessions are synchronized to the new secondary FortiGate. You can use the new options to adjust the timers depending on the requirements of your network conditions. For example, if you would rather not wait 150 seconds for hardware sessions to be synchronized to the new secondary FortiGate, you can adjust the `hw-session-sync-delay` timer.

Optimizing FGCP HA hardware session synchronization with data interface LAGs

The information in this section applies to FGCP HA hardware session synchronization only. FGSP HA hardware session synchronization packets are distributed by the internal switch fabric to the NP7 processors just like normal data traffic.

For optimal performance, the number of interfaces in the data interface LAG used for FGCP HA hardware session synchronization should divide evenly into the number of NP7 processors. This will distribute FGCP HA hardware session synchronization traffic evenly among the NP7 processors.

For example, the FortiGate-4200F has four NP7 processors. For optimum performance, the data interface LAG used for FGCP HA hardware session synchronization should include four or eight data interfaces. This configuration distributes the hardware session synchronization sessions evenly among the NP7 processors.

For a FortiGate-4400F with six NP7 processors, the optimal data interface LAG would include six or twelve data interfaces.

For a FortiGate-3500F with three NP7 processors, the optimal data interface LAG would include three or six data interfaces.

LAGs with fewer interfaces than the number of NP7 processors will also distribute sessions evenly among the NP7 processors as long as the number of data interfaces in the LAG divides evenly into the number of NP7 processors.

For best results, all of the data interfaces in the LAG should be the same type and configured to operate at the same speed. You can experiment with expected traffic levels when selecting the number and speed of the interfaces to add the LAG. For example, if you expect to have a large amount of hardware session synchronization interface traffic, you can add more data interfaces to the LAG or use 25G instead of 10G interfaces for the LAG.

Recommended interface use for an FGCP HA hyperscale firewall cluster

When setting up an FGCP HA cluster of two FortiGates operating as hyperscale firewalls, you need to select interfaces to use for some or all of the following features:

- Management.
- HA heartbeat (also called HA CPU heartbeat).
- HA session synchronization (also called HA CPU session synchronization).
- FGCP HA hardware session synchronization.
- Hardware logging.
- CPU logging.
- Logging to FortiAnalyzer

The following table contains Fortinet's recommendations for the FortiGate interfaces to use to support these features.

Interfaces	Recommended for
MGMT1 and MGMT2	Normal management communication with the FortiGates in the cluster.
HA1 and HA2	HA heartbeat (also called HA CPU heartbeat) between the FortiGates in the cluster.
AUX1 and AUX2	HA session synchronization (also called HA CPU session synchronization) or session pickup. The AUX1 and AUX2 interfaces are available only on the FortiGate 4200F/4201F and 4400F/4401F. For other FortiGate models, you can use any available interface or LAG for HA CPU session synchronization. For example, you may be able to use the HA1 and HA2 interfaces for both HA CPU heartbeat and HA CPU session synchronization. If you need to separate HA CPU heartbeat traffic from HA CPU session synchronization traffic, you can use a data interface or a data interface LAG for HA CPU session synchronization.
Data interface or data interface LAG	FGCP HA hardware session synchronization. If you use a data interface LAG as the FGCP HA hardware session synchronization interface, the LAG cannot be monitored by HA interface monitoring.
Data interface or data interface LAG	Hardware logging, CPU logging, and logging to a FortiAnalyzer. Depending on bandwidth use, you can use the same data interface or data interface LAG for all of these features.

FGSP HA hardware session synchronization

When configuring FortiGate Session Life Support Protocol (FGSP) clustering for two hyperscale firewall FortiGate peers, you can use FGSP HA hardware session synchronization to synchronize NP7 hyperscale firewall sessions between the FortiGate peers in the cluster. The FortiGate peers can be:

- Two FortiGates
- Two FGCP clusters
- One FortiGate and one FGCP cluster

Configuring the HA `hw-session-sync-dev` option is not required for FGSP HA hardware session synchronization. Instead, you set up a normal FGSP configuration for your hyperscale firewall VDOMs and use a data interface or data interface LAG as the FGSP session synchronization interface. The data interface can be a physical interface or a VLAN.

Select a data interface or create a data interface LAG for FGSP HA hardware session synchronization that can handle the expected traffic load. For example, from Fortinet's testing, hyperscale rates of 4,000,000 connections per second (CPS) can use 35Gbps of data for FGSP HA hardware session synchronization. If the CPS rate is higher, FGSP HA hardware session synchronization data use may spike above 50Gbps.

FGSP HA hardware session synchronization packets are distributed by the internal switch fabric to the NP7 processors just like normal data traffic. If you create a data interface LAG for FGSP HA hardware session synchronization, no special configuration of the data interface LAG is required for optimal performance.

FGSP HA hardware session synchronization does not support session filters (configured with the `config session-sync-filter option`).

For more information about FGSP, see FGSP.

Just like any FGSP configuration, the FortiGates must be the same model. The configurations of the hyperscale VDOMs on each FortiGate must also be the same. This includes VDOM names, interface names, and firewall policy configurations. You can use configuration synchronization to synchronize the configurations of the FortiGates in the FGSP cluster (see Standalone configuration synchronization). You can also configure the FortiGate separately or use FortiManager to keep key parts of the configuration, such as firewall policies, synchronized

Basic FGSP HA hardware session synchronization configuration example

The following steps describe how to set up a basic FGSP configuration to provide FGSP HA hardware session synchronization between one or more hyperscale firewall VDOMs in two FortiGate peers.

Use the following steps to configure FGSP on both of the peers in the FGSP cluster.

1. Enable FGSP for a hyperscale firewall VDOM, named MyCGN-hw12:

```
config system cluster-sync
  edit 1
    set peerip 1.1.1.1
    set syncvd MyCGN-hw12
  end
```

If your FortiGate has multiple hyperscale firewall VDOMs, you can add the names of the hyperscale VDOMs to be synchronized to the `syncvd` option. For example:

```
config system cluster-sync
   edit 1
      set peerip 1.1.1.1
      set syncvd MyCGN-hw12, MyCGN-hw22
   end
```

In most cases you should create only one cluster-sync instance. If you create multiple cluster-sync instances, all FGSP HA hardware session synchronization sessions will be sent to the interface used by each cluster-sync instance.

2. Configure FGSP session synchronization as required. All session synchronization options are supported. For example:

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

3. Configure networking on the FortiGate so that traffic to be forwarded to the peer IP address (in the example, 1.1.1.1) passes through a data interface or data interface LAG.

   This data interface or data interface LAG becomes the FGSP HA hardware session synchronization interface. If the data interface or data interface LAG is in the root VDOM, no additional configuration is required.

   If the data interface or data interface LAG is not in the root VDOM, you need to use the `peervd` option to specify the VDOM that the interface is in. For example, if the data interface or data interface LAG is in the MyCGN-hw12 VDOM:

```
config system cluster-sync
   edit 1
      set peerip 1.1.1.1
      set syncvd MyCGN-hw12, MyCGN-hw22
      set peervd MyCGN-hw12
   end
```

Operating a hyperscale firewall

This chapter is a collection of information that you can use when operating a FortiGate with hyperscale firewall features enabled.

Recommended NP7 traffic distribution for optimal CGNAT performance

On FortiGates with multiple NP7 processors, you can use the following command to configure how the internal switch fabric (ISF) distributes sessions to the NP7 processors.

```
config system global
   config system npu
      set hash-config {src-dst-ip | 5-tuple | src-ip}
   end
```

Changing the `hash-config` causes the FortiGate to restart.

A configuration change that causes a FortiGate to restart can disrupt the operation of an FGCP cluster. If possible, you should make this configuration change to the individual FortiGates before setting up the cluster. If the cluster is already operating, you should temporarily remove the secondary FortiGate(s) from the cluster, change the configuration of the individual FortiGates and then re-form the cluster. You can remove FortiGate(s) from a cluster using the **Remove Device from HA cluster** button on the **System > HA** GUI page. For more information, see Disconnecting a FortiGate.

`src-ip`, (the default) sessions are distributed by source IP address. All sessions from a source IP address are processed by the same NP7 processor. `src-ip` is the recommended setting for optimal CGNAT performance. Other `hash-config` settings can distribute client sessions from a single source address to multiple NP7 processors. This can result in CGNAT sessions not being established or timing out when expected. As well, using `src-ip` guarantees that all sessions from a given source IP address use the same public source IP address. This is not guaranteed if you select the other `hash-config` settings. For more information, see NPU traffic distribution in the Carrier-Grade NAT Architecture Guide.

`5-tuple`, use 5-tupple source and destination IP address, IP protocol, and source and destination TCP/UDP port hashing. This option is available on FortiGates with 2 to the power of x NP7 processors (Where x is 2, 3, 4). Currently 5-tuple is available on FortiGates with 2, 4, and 16 NP7 processors. Using `5-tuple` distribution can result in some CGNAT sessions not being established or timing out when expected. As well, using `5-tuple` may cause sessions from a single client source IP address to be assigned different public source IP addresses.

`src-dst-ip`, use 2-tupple source and destination IP address hashing. This option is available on FortiGates with a number of NP7 processors that don't add up to 2 to the power of x (for example, FortiGates with 3 or 6 NP7 processors). Using `src-dst-ip` distribution can result in some CGNAT sessions not being established or timing out when expected. As well, using `src-dst-ip` may cause sessions from a single client source IP address to be assigned different public source IP addresses.

Hyperscale firewall inter-VDOM link acceleration

If hyperscale firewall support is enabled, you apply NP7 acceleration to inter-VDOM link traffic by creating inter-VDOM links with the `type` set to `npupair`. For example:

```
config system vdom-link
  edit <name>
    set type npupair
  end
```

The command creates a pair of interfaces that are connected logically. For example, the following command:

```
config system vdom-link
  edit vdom-link0
    set type npupair
  end
```

Creates two interfaces, named `vdom-link00` and `vdom-link01`.

The default NPU VDOM inter-VDOM links (for example npu0_vlink0, npu0_vlink1, npu1_vlink0, and so on) are not supported for links to or from VDOMs with hyperscale firewall acceleration enabled.

Hyperscale firewall SNMP MIB and trap fields

This section describes hyperscale firewall SNMP MIB and trap fields.

IP pool MIB and trap fields

You can use the following MIB fields to get hyperscale firewall IP pool information:

```
FgFwIppStatsEntry ::= SEQUENCE {
    fgFwIppStatsName          DisplayString,
    fgFwIppStatsType          DisplayString,
    fgFwIppStatsStartIp       IpAddress,
    fgFwIppStatsEndIp         IpAddress,
    fgFwIppStatsTotalSessions Gauge32,
    fgFwIppStatsTcpSessions   Gauge32,
    fgFwIppStatsUdpSessions   Gauge32,
    fgFwIppStatsOtherSessions Gauge32,
    fgFwIppStatsTotalPBAs     Gauge32,
    fgFwIppStatsInusePBAs     Gauge32,
    fgFwIppStatsExpiringPBAs  Gauge32,
    fgFwIppStatsFreePBAs      Gauge32,
    fgFwIppStatsFlags         DisplayString,
    fgFwIppStatsGroupName     DisplayString,
    fgFwIppStatsBlockSize     Gauge32,
    fgFwIppStatsPortStart     InetPortNumber,
    fgFwIppStatsPortEnd       InetPortNumber,
    fgFwIppStatsStartClientIP IpAddress,
    fgFwIppStatsEndClientIP   IpAddress,
    fgFwIppStatsRscTCP        Gauge32,
    fgFwIppStatsRscUDP        Gauge32,
```

```
    fgFwIppStatsUsedRscTCP    Gauge32,
    fgFwIppStatsUsedRscUDP    Gauge32,
    fgFwIppStatsPercentageTCP Gauge32,
    fgFwIppStatsPercentageUDP Gauge32
}
```

The following 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 }
```

Hyperscale firewall policy MIB fields

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

OID: 1.3.6.1.4.1.12356.101.5.1.2

Index	MIB field	Description
.3	fgFwHsPolStatsTable	IPv4 hyperscale firewall policy statistics table.
.3.1	fgFwHsPolStatsEntry	IPv4 hyperscale firewall policy statistics entry.
.3.1.1	fgFwHsPolID	IPv4 hyperscale firewall policy ID.
.3.1.2	fgFwHsPolPktCount	IPv4 hyperscale firewall policy packet count.
.3.1.3	fgFwHsPolByteCount	IPv4 hyperscale firewall policy byte count.
.3.1.4	fgFwHsPolLastUsed	The last date and time the Ipv4 hyperscale firewall policy was used to start a session.
.4	fgFwHsPol6StatsTable	IPv6 hyperscale firewall policy stats table.
.4.1	fgFwHsPol6StatsEntry	IPv6 hyperscale firewall policy statistics entry.

Index	MIB field	Description
.4.1.1	fgFwHsPol6ID	IPv6 hyperscale firewall statisticsID.
.4.1.2	fgFwHsPol6PktCount	IPv6 hyperscale firewall policy packet count.
.4.1.3	fgFwHsPol6ByteCount	IPv6 hyperscale firewall policy byte count.
.4.1.4	fgFwHsPol6LastUsed	The last date and time the IPv6 hyperscale firewall policy was used to start a session.

Queries of these fields follow the convention .oid.<vdom-id>.<policy-id>

Example SNMP query for IPv4 hyperscale firewall policy statistics:

```
$ snmpwalk -v2c -c public <ip-address> 1.3.6.1.4.1.12356.101.5.1.2.3.1
```

Example SNMP query for IPv6 hyperscale firewall policy statistics:

```
$ snmpwalk -v2c -c public <ip-address> 1.3.6.1.4.1.12356.101.5.1.2.4.1
```

SNMP queries for hardware session counts

You can use the following MIB fields to send SNMP queries for NP7 IPv4 and IPv6 hardware session counts and session setup rates.

The fgSysNpuSesCount MIB field returns the total session count for both IPv4 and IPv6 sessions. The fgSysNpuSes6Count MIB field always returns 0.

Path:FORTINET-FORTIGATE-MIB:fortinet.fnFortiGateMib.fgSystem.fgSystemInfo

OID: 1.3.6.1.4.1.12356.101.4.1

Index	MIB field	Description
.24	fgSysNpuSesCount	NP7 IPv4 and IPv6 session count.
.25	fgSysNpuSesRate1	NP7 IPv4 session setup rate in the last 1 minute.
.26	fgSysNpuSesRate10	NP7 IPv4 session setup rate in the last 10 minutes.
.27	fgSysNpuSesRate30	NP7 IPv4 session setup rate in the last 30 minutes.

Index	MIB field	Description
.28	fgSysNpuSesRate60	NP7 IPv4 session setup rate in the last 60 minutes.
.29	fgSysNpuSes6Count	0
.30	fgSysNpuSes6Rate1	NP7 IPv6 session setup rate in the last 1 minute.
.31	fgSysNpuSes6Rate10	NP7 IPv6 session setup rate in the last 10 minutes.
.32	fgSysNpuSes6Rate30	NP7 IPv6 session setup rate in the last 30 minutes.
.33	fgSysNpuSes6Rate60	NP7 IPv6 session setup rate in the last 60 minutes.

SNMP queries for NAT46 and NAT64 policy statistics

You can use the following MIB fields to send SNMP queries for hyperscale firewall NAT46 and NAT64 policy statistics. These MIB fields are available from the latest FORTINET-FORTIGATE-MIB.mib.

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

OID: 1.3.6.1.4.1.12356.101.5.1.2

Index	MIB field	Description
.5	fgFwHsPol46StatsTable	NAT46 hyperscale firewall policy statistics table.
.5.1	fgFwHsPol46StatsEntry	NAT46 hyperscale firewall policy statistics entry.
.5.1.1	fgFwHsPol46ID	NAT46 hyperscale firewall policy ID.
.5.1.2	fgFwHsPol46PktCount	NAT46 hyperscale firewall policy packet count.
.5.1.3	fgFwHsPol46ByteCount	NAT46 hyperscale firewall policy byte count.
.5.1.4	fgFwHsPol46LastUsed	The last date and time the NAT46 hyperscale firewall policy was used to start a session.
.6	fgFwHsPol64StatsTable	NAT64 hyperscale firewall policy statistics table.

Index	MIB field	Description
.6.1	fgFwHsPol64StatsEntry	NAT64 hyperscale firewall policy statistics entry.
.6.1.1	fgFwHsPol64ID	NAT64 hyperscale firewall policy ID.
.6.1.2	fgFwHsPol64PktCount	NAT64 hyperscale firewall policy packet count.
.6.1.3	fgFwHsPol64ByteCount	NAT64 hyperscale firewall policy byte count.
.6.1.4	fgFwHsPol64LastUsed	The last date and time the NAT64 hyperscale firewall policy was used to start a session.

Queries of these fields follow the convention `.oid.<vdom-id>.<policy-id>`

Example SNMP query for NAT46 hyperscale firewall policy statistics:

```
$ snmpwalk -v2c -c public <ip-address> 1.3.6.1.4.1.12356.101.5.1.2.5.1
```

Example SNMP query for NAT64 hyperscale firewall policy statistics:

```
$ snmpwalk -v2c -c public <ip-address> 1.3.6.1.4.1.12356.101.5.1.2.6.1
```

SNMP queries of NP7 fgProcessor MIB fields

FortiGates with NP7 processors can now respond to SNMP queries for the following paths and OIDs:

- Path: FORTINET-FORTIGATE-MIB:fgProcessorCount
  OID: 1.3.6.1.4.1.12356.101.4.4.1
- Path: FORTINET-FORTIGATE-MIB:fgProcessorModuleCount
  OID: 1.3.6.1.4.1.12356.101.4.5

For example, for a FortiGate-4200F:

```
root@pc1:~# snmpwalk -v2c -c REGR-SYS 10.1.100.1 1.3.6.1.4.1.12356.101.4.4.1
FORTINET-FORTIGATE-MIB::fgProcessorCount.0 = INTEGER: 84
root@pc1:~# snmpwalk -v2c -c REGR-SYS 10.1.100.1 1.3.6.1.4.1.12356.101.4.5
FORTINET-FORTIGATE-MIB::fgProcessorModuleCount.0 = INTEGER: 5
FORTINET-FORTIGATE-MIB::fgProcModIndex.1 = INTEGER: 1
FORTINET-FORTIGATE-MIB::fgProcModIndex.2 = INTEGER: 2
FORTINET-FORTIGATE-MIB::fgProcModIndex.3 = INTEGER: 3
FORTINET-FORTIGATE-MIB::fgProcModIndex.4 = INTEGER: 4
FORTINET-FORTIGATE-MIB::fgProcModIndex.5 = INTEGER: 5
FORTINET-FORTIGATE-MIB::fgProcModType.1 = OID: FORTINET-FORTIGATE-MIB::fgProcModIntegrated
FORTINET-FORTIGATE-MIB::fgProcModType.2 = OID: FORTINET-FORTIGATE-MIB::fgProcModFnXE2
FORTINET-FORTIGATE-MIB::fgProcModType.3 = OID: FORTINET-FORTIGATE-MIB::fgProcModFnXE2
FORTINET-FORTIGATE-MIB::fgProcModType.4 = OID: FORTINET-FORTIGATE-MIB::fgProcModFnXE2
FORTINET-FORTIGATE-MIB::fgProcModType.5 = OID: FORTINET-FORTIGATE-MIB::fgProcModFnXE2
FORTINET-FORTIGATE-MIB::fgProcModName.1 = STRING: integrated_cpus
FORTINET-FORTIGATE-MIB::fgProcModName.2 = STRING: Integrated_NPU (np7_0)
FORTINET-FORTIGATE-MIB::fgProcModName.3 = STRING: Integrated_NPU (np7_1)
```

```
FORTINET-FORTIGATE-MIB::fgProcModName.4 = STRING: Integrated_NPU (np7_2)
FORTINET-FORTIGATE-MIB::fgProcModName.5 = STRING: Integrated_NPU (np7_3)
FORTINET-FORTIGATE-MIB::fgProcModDescr.1 = STRING: Fortinet integrated CPU module (main CPUs
built into device)
FORTINET-FORTIGATE-MIB::fgProcModDescr.2 = STRING: Fortinet integrated CPU module (NPUs
built into device)
FORTINET-FORTIGATE-MIB::fgProcModDescr.3 = STRING: Fortinet integrated CPU module (NPUs
built into device)
FORTINET-FORTIGATE-MIB::fgProcModDescr.4 = STRING: Fortinet integrated CPU module (NPUs
built into device)
FORTINET-FORTIGATE-MIB::fgProcModDescr.5 = STRING: Fortinet integrated CPU module (NPUs
built into device)
FORTINET-FORTIGATE-MIB::fgProcModProcessorCount.1 = INTEGER: 80
FORTINET-FORTIGATE-MIB::fgProcModProcessorCount.2 = INTEGER: 1
FORTINET-FORTIGATE-MIB::fgProcModProcessorCount.3 = INTEGER: 1
FORTINET-FORTIGATE-MIB::fgProcModProcessorCount.4 = INTEGER: 1
FORTINET-FORTIGATE-MIB::fgProcModProcessorCount.5 = INTEGER: 1
FORTINET-FORTIGATE-MIB::fgProcModMemCapacity.1 = Gauge32: 397046052
FORTINET-FORTIGATE-MIB::fgProcModMemCapacity.2 = Gauge32: 8388608
FORTINET-FORTIGATE-MIB::fgProcModMemCapacity.3 = Gauge32: 8388608
FORTINET-FORTIGATE-MIB::fgProcModMemCapacity.4 = Gauge32: 8388608
FORTINET-FORTIGATE-MIB::fgProcModMemCapacity.5 = Gauge32: 8388608
FORTINET-FORTIGATE-MIB::fgProcModMemUsage.1 = Gauge32: 4
FORTINET-FORTIGATE-MIB::fgProcModMemUsage.2 = Gauge32: 0
FORTINET-FORTIGATE-MIB::fgProcModMemUsage.3 = Gauge32: 0
FORTINET-FORTIGATE-MIB::fgProcModMemUsage.4 = Gauge32: 0
FORTINET-FORTIGATE-MIB::fgProcModMemUsage.5 = Gauge32: 0
FORTINET-FORTIGATE-MIB::fgProcModSessionCount.1 = Gauge32: 19
FORTINET-FORTIGATE-MIB::fgProcModSessionCount.2 = Gauge32: 0
FORTINET-FORTIGATE-MIB::fgProcModSessionCount.3 = Gauge32: 0
FORTINET-FORTIGATE-MIB::fgProcModSessionCount.4 = Gauge32: 0
FORTINET-FORTIGATE-MIB::fgProcModSessionCount.5 = Gauge32: 0
FORTINET-FORTIGATE-MIB::fgProcModSACount.1 = Gauge32: 0
FORTINET-FORTIGATE-MIB::fgProcModSACount.2 = Gauge32: 0
FORTINET-FORTIGATE-MIB::fgProcModSACount.3 = Gauge32: 0
FORTINET-FORTIGATE-MIB::fgProcModSACount.4 = Gauge32: 0
FORTINET-FORTIGATE-MIB::fgProcModSACount.5 = Gauge32: 0
```

BGP IPv6 conditional route advertisement

IPv6 BGP conditional route advertisement supports traffic failover for a FortiGate with hyperscale firewall features operating as a CGNAT translator connected to two ISPs over IPv6.

When the FortiGate can connect to the primary ISP, IPv6 BGP routes to the primary ISP are shared with the networks (LANs) behind the FortiGate. With BGP IPv6 conditional route advertisement enabled, if the FortiGate connection to the primary ISP fails, the FortiGate acquires IPv6 BGP routes to the secondary ISP and advertises these routes to the networks (LANs) behind the FortiGate.

Use the following configuration to enable IPv6 conditional route advertisement:

```
config router bgp
  config neighbor
    edit <name>
      config conditional-advertise6
```

```
            edit <name>
               set condition-routemap <name>
               set condition-type {exist | non-exist}
            end
```

`exist` true if condition route map is matched.

`non-exist` true if condition route map is not matched.

BGP IPv6 conditional route advertisement configuration example

The following configuration shows how to use the `condition-type` option to control how a FortiGate advertises routes when it is connected to two external routers.

When `condition-type` is set to `non-exist` the FortiGate advertises route2 (2003:172:22:1::/64) to Router2 when it learns route1 (2003:172:28:1::/64). When `condition-type` is set to `exist`, the FortiGate will not advertise route2 (2003:172:22:1::/64) to Router2 when it knows route1 (2003:172:28:1::/64).

```
config router prefix-list6
   edit adv-222
      config rule
         edit 1
            set prefix6 2003:172:22:1::/64
         end

config router prefix-list6
   edit list6-1
      config rule
         edit 1
            set prefix6 2003:172:28:1::/64
         end

config router route-map
   edit map-222
      config rule
         edit 1
            set match-ip6-address adv-222
         end
config router route-map
   edit "map-281"
      config rule
         edit 1
            set match-ip6-address list6-1
         end
config router bgp
   set as 65412
   set router-id 1.1.1.1
   set ibgp-multipath enable
   set network-import-check disable
   set graceful-restart enable
      config neighbor
         edit 2003::2:2:2:2
            set soft-reconfiguration6 enable
            set remote-as 65412
            set update-source loopback1
               config conditional-advertise6
               edit map-222
```

```
                set condition-routemap map-281
                set condition-type {exist | non-exist}
            end
        edit 2003::3:3:3:3
            set soft-reconfiguration6 enable
            set remote-as 65412
            set update-source loopback1
        end
```

Hyperscale firewall VDOM asymmetric routing with ECMP support

In most cases asymmetric routing with ECMP support works the same way in a hyperscale firewall VDOM as in a normal VDOM, with the following notes and exceptions:

- The `auxiliary-session` and `asymroute-icmp` options of the `config system settings` command do not have to be enabled for the hyperscale firewall VDOM for asymmetric routing to work.
- Make sure that original routes (O-routes) do not overlap with reverse routes (R-routes). If you have created overlapping O- and R-routes, all reply traffic uses the same O-route.
- If possible, create an even number of ECMP paths. Traffic distribution is uneven if you have an odd number of ECMP paths. For example, if your configuration includes one O-route and three R-routes, the reply traffic distribution will be approximately 2:1:1 among the three R-routes.

Hyperscale firewall VDOM session timeouts

Using the following command you can define session timeouts for a specific protocols and port ranges for a hyperscale firewall VDOM. These session timeouts apply to sessions processed by the current hyperscale firewall VDOM. You can set up different session timeouts for each hyperscale firewall VDOM.

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

`protocol <protocol-number>` a protocol number in the range 0 to 255. Default 0.

`timeout <timeout>` the time in seconds after which a matching idle session is terminated. Range 1 to 2764800. Default 300.

`refresh-direction {outgoing | incoming | both}` control whether idle outgoing or incoming or both outgoing and incoming sessions are terminated when the timeout is reached.

Global session timeouts apply to sessions in hyperscale firewall VDOMs that do not match `config system session-ttl` settings in individual hyperscale firewall VDOMs.

You can also override global and per-VDOM session timeouts by setting the `tcp-timeout-pid` and `udp-timeout-pid` options in individual hyperscale firewall policies. See Session timeouts for individual hyperscale policies on page 75.

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
```

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
```

Use the following command to apply a TCP and a UDP timeout profile to a hyperscale firewall policy:

```
config vdom
   edit <hyperscale-firewall-vdom-name>
      config firewall policy
         edit 1
            set action accept
            set policy-offload enable
            ...
            set tcp-timout-pid <tcp-profile-id>
            set udp-timout-pid <udp-profile-id>
            ...
         end
```

For more information about creating TCP timeout profiles, see Configuring hyperscale TCP timeout profiles.

For more information about creating UDP timeout profiles, see Configuring hyperscale UDP timeout profiles.

Modifying trap session behavior in hyperscale firewall VDOMs

Hyperscale VDOMs create trap sessions for all sessions that need to be handled by the CPU. Trap sessions make sure CPU sessions are successfully sent to the CPU. If CPU sessions are not trapped, they may be incorrectly converted to hardware sessions and dropped.

You can use the following command to modify trap session behavior in a hyperscale firewall VDOM

```
config system settings
   set trap-session-flag {udp-both | udp-reply | tcpudp-both | tcpudp-reply | trap-none}
end
```

`udp-both` trap UDP send and reply sessions.

`udp-reply` trap UDP reply sessions only.

`tcpudp-both` trap TCP and UDP send and reply sessions. This is the default setting.

`tcpudp-reply` trap TCP and UDP reply sessions only.

`trap-none` disable trapping sessions.

The default setting creates trap sessions for all TCP and UDP sessions to be handled by the CPU. You can change the trap session behavior depending on CPU sessions processed by the VDOM.

Enabling or disabling the NP7 VLAN lookup cache

You can use the following command to enable or disable VLAN lookup (SPV/TPV) caching for hyperscale firewall sessions.

```
config system npu
   set vlan-lookup-cache {disable | enable}
end
```

This option is enabled by default.

Due to a known issue, with hyperscale firewall enabled, this option should be disabled if you configure more than 256 VLANs. Enabling or disabling `vlan-lookup-cache` requires a system restart. So you should only change this setting during a maintenance window.

Setting the hyperscale firewall VDOM default policy action

You can use the following system settings option for each hyperscale firewall VDOM to set the default firewall policy action for that VDOM. The default action determines what NP7 processors do with TCP and UDP packets that are not accepted by any firewall policies.

```
config system settings
   set hyperscale-default-policy-action {drop-on-hardware | forward-to-host}
end
```

`drop-on-hardware` the default setting, NP7 processors drop TCP and UDP packets that don't match a firewall policy. In most cases you would not want to change this default setting since it means the CPU does not have to process TCP and UDP packets that don't match firewall policies. In most cases, this option should reduce the number of packets sent to the CPU. With this option enabled, all other packet types (for example, ICMP packets) that don't match a firewall policy are sent to the CPU. Packets accepted by session helpers are also sent to the CPU.

`forward-to-host` NP7 processors forward packets that don't match a firewall policy to the CPU. If the packet is forwarded to the CPU, the packet will be matched with the policy list and eventually be subject to the implicit deny policy and dropped by the CPU. This setting can affect performance because the CPU would be handling these packets.

Reassembling fragmented packets

FortiGates with NP7 processors that are licensed for hyperscale firewall features support reassembling fragmented packets in sessions offloaded to the NP7 processors.

To support reassembling fragmented packets, the NP7 processor `hash-config` can be set to `src-dst-ip`, `5-tuple`, or `src-ip`. As well, NP7 `ip-reassemly` must be enabled. You can also adjust the `ip-reassembly` minimum and maximum timeouts. The currently recommended configuration includes the following minimum and maximum timeouts. You can adjust these timeouts for your network configuration and traffic profile.

```
config system npu
   set hash-config {src-dst-ip | 5-tuple | src-ip}
      config ip-reassembly
         set status enable
         set min_timeout 64
         set max_timeout 200000
      end
```

For more information about the `hash-config` option, see hash-config {src-dst-ip | 5-tuple | src-ip}.

For more information on the `ip-reassembly` option, see Reassembling and offloading fragmented packets.

Hash table message queue mode

You can use the following commands to change the hyperscale firewall NP7 hash table message queue mode.

```
config system npu
   set htab-msg-queue {data | idle | dedicated}
   set htab-dedi-queue-nr <number-of-queues>
end
```

You can use the `htab-msg-queue` option to alleviate performance bottlenecks that may occur when hash table messages use up all of the available hyperscale NP7 data queues.

You can use the following commands to get the hash table message count and rate.

```
diagnose npu np7 msg htab-stats {all| chip-id}
diagnose npu np7 msg htab-rate {all| chip-id}
```

You can use the following command to show MSWM information:

```
diagnose npu np7 mswm
```

You can use the following command to show NP7 Session Search Engine (SSE) drop counters:

```
diagnose npu np7 dce-sse-drop 0 v
```

You can use the following command to show command counters:

```
diagnose npu np7 cmd
```

The following `htab-msg-queue` options are available:

- `data` (the default) use all available data queues.
- `idle` if you notice the data queues are all in use, you can select this option to use idle queues for hash table messages.
- `dedicated` use between 1 to 8 of the highest number data queues. Use the option `htab-dedi-queue-nr` to set the number of data queues to use.

If you are using dedicated queues for hash table messages for hyperscale firewall sessions, you can use the `htab-dedi-queue-nr` option to set the number of queues to use. The range is 1 to 8 queues. The default is 4 queues.

Message-related diagnose commands:

```
diagnose npu np7 msg
summary          Show summary of message counters. [Take 0-1 arg(s)]
msg-by-mod       Show/clear message counters by source module. [Take 0-2 arg(s)]
msg-by-code      Show/clear message counters by message code. [Take 0-2 arg(s)]
msg-by-que       Show/clear message counters by RX queue. [Take 0-2 arg(s)]
msg-by-cpu       Show/clear message counters by CPU. [Take 0-2 arg(s)]
htab-stats       Show/clear hash table message counters. [Take 0-2 arg(s)]
htab-rate        Show/clear hash table message rate. [Take 0-2 arg(s)]
ipsec-stats      Show/clear IPSec message counters. [Take 0-2 arg(s)]
ipsec-rate       Show/clear IPSec message rate. [Take 0-2 arg(s)]
ipt-stats        Show/clear IP tunnel message counters. [Take 0-2 arg(s)]
ipt-rate         Show/clear IP tunnel message rate. [Take 0-2 arg(s)]
mse-stats        Show/clear MSE message counters. [Take 0-2 arg(s)]
mse-rate         Show/clear MSE message rate. [Take 0-2 arg(s)]
spath-stats      Show/clear hyperscale message counters. [Take 0-2 arg(s)]
spath-rate       Show/clear hyperscale message rate. [Take 0-2 arg(s)]
tpe-tce-stats    Show/clear TPC/TCE message counters. [Take 0-2 arg(s)]
tpe-tce-rate     Show/clear TPE/TCE message rate. [Take 0-2 arg(s)]
```

MSWM diag commands.

```
diagnose npu np7 mswm
mswm-all          Show/clear all MSWM counters. [Take 0-2 arg(s)]
module-to-mswm    Show/clear module-to-MSWM counters. [Take 0-2 arg(s)]
mswm-to-module    Show/clear MSWM-to-module counters. [Take 0-2 arg(s)]
mswh-all          Show/clear all MSWH counters. [Take 0-2 arg(s)]
module-to-mswh    Show/clear module-to-MSWH counters. [Take 0-2 arg(s)]
mswh-to-hrx       Show/clear MSWH-to-HRX counter. [Take 0-2 arg(s)]
```

Diagnose command to show SSE drop counters:

```
diagnose npu np7 dce-sse-drop 0 v
```

Diagnose command to show command counters:

```
diagnose npu np7 cmd
all             Show/clear all command counters. [Take 0-2 arg(s)]
sse             Show/clear SSE command counters. [Take 0-2 arg(s)]
mse             Show/clear MSE command counters. [Take 0-2 arg(s)]
dse             Show/clear DSE command counters. [Take 0-2 arg(s)]
lpm-rlt         Show/clear LPM/RLT command counters. [Take 0-2 arg(s)]
rate            Show/clear command rate. [Take 0-2 arg(s)]
measure-rate    Enable/disable command rate measurement. [Take 0-1 arg(s)]
```

Optimizing NP7 network session setup (NSS) engine performance

On FortiGates with multiple NP7 processors operating with hyperscale enabled, you can use the following command to optimize NP7 network session setup (NSS) engine performance.

```
config system npu
   set nss-threads-option {4T-EIF | 4T-NOEIF | 2T}
end
```

Where:

`4T-EIF` the NSS is configured with four threads and the Endpoint Independent Filtering (EIF) feature is allowed (the default). NSS with four threads supports the maximum NP7 Connections Per Second (CPS) performance.

`4T-NOEIF` the NSS is configured with four threads and the EIF feature is not allowed. Also supports the maximum NP7 CPS performance.

`2T` the NSS is configured with two threads and the EIF feature is allowed. This setting reduces the maximum NP7 CPS performance.

Changing the `nss-threads-option` causes the FortiGate to restart.

A configuration change that causes a FortiGate to restart can disrupt the operation of an FGCP cluster. If possible, you should make this configuration change to the individual FortiGates before setting up the cluster. If the cluster is already operating, you should temporarily remove the secondary FortiGate(s) from the cluster, change the configuration of the individual FortiGates and then re-form the cluster. You can remove FortiGate(s) from a cluster using the **Remove Device from HA cluster** button on the **System > HA** GUI page. For more information, see Disconnecting a FortiGate.

If your system includes hyperscale firewall policies with EIF enabled, you can keep the default setting of `4T-EIF` for the best maximum CPS performance. However, in heavy traffic conditions, operating the NSS with four threads with hyperscale firewall policies with EIF enabled may cause errors and lead to sessions being lost. If your hyperscale system with EIF enabled experiences these issues under heavy traffic, you can select `2T` to operate the NSS with two threads. Operating the NSS with two threads reduces maximum CPS performance, but with two threads NP7 CPS performance is still much better than CPS performance on non-hyperscale systems.

If all of your hyperscale firewall policies disable EIF you can select `4T-NOEIF` for optimal CPS performance and stability.

Setting the NP7 TCP reset timeout

You can use the following command to adjust the NP7 TCP reset timeout

```
config system npu
   tcp-rst-timeout <timeout>
end
```

The NP7 TCP reset (RST) timeout in seconds. The range is 0-16777215. The default timeout is 5 seconds. The default timeout is optimal in most cases, especially when hyperscale firewall is enabled. A timeout of 0 means no time out.

Configuring background SSE scanning

To support reporting accurate UDP session statistics, normal UDP session synchronization is disabled for FortiGates with hyperscale firewall features enabled and background Session Search Engine (SSE) scanning is used to keep UDP sessions synchronized.

Background SSE scanning uses the CPU instead of the NP7 processors and can cause CPU spikes; however, these spikes should not usually affect overall performance. You can use the following command to adjust background SSE scanning behavior:

```
config system npu
   config background-sse-scan
      set scan {disable | enable}
      set stats-update-interval <interval>
      set udp-keepalive-interval <interval>
   end
```

`scan` enable or disable background SSE scanning. This option is enabled by default. If disabled, UDP O-session and R-session synchronization is enabled so UDP sessions will remain synchronized. However, the statistics reported by traffic logging for UDP O-sessions will be incorrect.

`stats-update-interval` statistics update interval in seconds. The range is 300 to 1073741823 seconds and the default update interval is 300 seconds. You can increase the statistics update interval to reduce how often the CPU is used for SSE background scanning.

`udp-keepalive-interval` UDP keepalive interval in seconds. The range is 90 to 1073741823 seconds and the default keepalive interval is 90 seconds. The 90 second keepalive interval is recommended because the default UDP session timeout is 180 seconds. If you increase the keepalive interval, some UDP sessions may be dropped prematurely.

Allowing packet fragments for NP7 NAT46 policies when the DF bit is set to 1

The packet size increase that occurs when a NAT46 hyperscale firewall policy converts an IPv4 packet into an IPv6 packet can cause the packet to be dropped if the larger packet exceeds the outgoing interface MTU and the DF bit is set to 1 (do not fragment). You can use the following command to cause NP7 processors to override the DF setting and fragment and forward the packet instead of dropping it. This is a global setting that affects all NAT64 traffic offloaded by NP7 processors.

```
config system npu
   set nat46-force-ipv4-packet-forwarding enable
end
```

When this option is disabled, packets with DF=1 that exceed the outgoing interface MTU are dropped and an ICMP error is returned to the sender.

Hyperscale firewall get and diagnose commands

This section describes some `get` and `diagnose` commands that you can use to display hyperscale firewall information.

Diagnose commands are intended for debug purposes only. Regular use of these commands can consume CPU and memory resources and cause other system related issues.

Displaying information about NP7 hyperscale firewall hardware sessions

Use the `diagnose sys npu-session` command to view NP7 hardware sessions as well as sessions that are not offloaded to NP7 processors. You can list and clear NP7 hardware sessions and create filters to control the sessions that are listed or cleared.

You can also use `diagnose sys session list` and `diagnose sys session6 list` to list sessions that have not been offloaded.

diagnose sys npu-session list [{44 | 46}]

List IPv4 NP7 hardware sessions or sessions not offloaded to NP7 processors. If you have set up an IPv4 filter, this command lists sessions that match the IPv4 filter.

This command displays the current session list stored in the logging buffer. If **Hyperscale SPU Offload Log Settings** is set to **Hardware** (or the `config log npu-server` option `log-processor` is set to `hardware`), the logging buffer includes all session details.

If **Hyperscale SPU Offload Log Settings** is set to **Host** (or the `config log npu-server` option `log-processor` is set to `host`), the command displays fewer details about the session list, because CPU or host logging only maintains a subset of all of the information available for each session in the session list. For example, the CPU or host logging session table includes the default session timeout information for each session. This session timeout value may not be accurate for some sessions. The actual session timeout is available from the session list maintained by the NP7 Session Search Engine (SSE) session table. To see the actual session timeout for NP7 sessions, you must use the `diagnose sys npu-session list-full` command.

`(no options)` list IPv4 and NAT46 NP7 sessions.

`44` list IPv4 NP7 sessions.

`46` list NAT46 NP7 sessions.

`host` list IPv4 sessions that have not been offloaded to NP7 processors.

diagnose sys npu-session list6 [{66 | 64}]

List IPv6 NP7 hardware sessions or sessions that have not been offloaded to NP7 processors. If you have set up an IPv6 filter, this command lists sessions that match the IPv6 filter.

This command displays the current session list stored in the logging buffer. If **Hyperscale SPU Offload Log Settings** is set to **Hardware** (or the `config log npu-server` option `log-processor` is set to `hardware`), the logging buffer includes all session details.

If **Hyperscale SPU Offload Log Settings** is set to **Host** (or the `config log npu-server` option `log-processor` is set to `host`), the command displays fewer details about the session list, because CPU or host logging only maintains a subset of all of the information available for each session in the session list. For example, the CPU or host logging session table includes the default session timeout information for each session. This session timeout value may not be accurate for some sessions. The actual session timeout is available from the session list maintained by the NP7 Session Search Engine (SSE) session table. To see the actual session timeout for NP7 sessions, you must use the `diagnose sys npu-session list-full` command.

(no options) list IPv6 and NAT64 NP7 sessions.

`66` list IPv6 NP7 sessions.

`64` list NAT64 NP7 sessions.

`host` list IPv6 sessions that have not been offloaded to NP7 processors.

diagnose sys npu-session list-full [{44 | 46}]

List IPv4 NP7 hardware sessions and include more information about each session than that provided by the `list` option. If you have set up an IPv4 filter, this command lists sessions that match the IPv4 filter.

This command displays the current IPv4 NP7 hyperscale firewall hardware session list by sending a query to the NP7 Session Search Engine (SSE). The output does not depend on the hardware logging configuration because the command queries the SSE. However, because the commands are querying the SSE, the response time will be longer.

(no options) list IPv4 and NAT46 NP7 sessions.

`44` list IPv4 NP7 sessions.

`46` list NAT46 NP7 sessions.

diagnose sys npu-session list-full6 [{66 | 64}]

List IPv6 NP7 hardware sessions and include more information about each session than that provided by the `list6` option. If you have set up an IPv6 filter, this command lists sessions that match the IPv4 filter.

This command displays the current IPv6 NP7 hyperscale firewall hardware session list by sending a query to the NP7 SSE. The output does not depend on the hardware logging configuration because the command queries the SSE. However, because the commands are querying the SSE, the response time will be longer.

(no options) list IPv6 and NAT64 NP7 sessions.

`66` list IPv6 NP7 sessions.

`64` list NAT64 NP7 sessions.

diagnose sys npu-session list-brief [{44 | 46}]

View summary information about IPv4 sessions offloaded to NP7 processors.

The command output includes lists of sessions organized by session type and a total number of sessions for each session type. Summary information for each session includes the protocol, expiry time, source and destination addresses, and source and destination NAT addresses.

diagnose sys npu-session list-brief6 [{66 | 64}]

View summary information about IPv6 sessions offloaded to NP7 processors.

The command output includes lists of sessions organized by session type and a total number of sessions for each session type. Summary information for each session includes the protocol, expiry time, source and destination addresses, and source and destination NAT addresses.

diagnose sys npu-session clear [{44 | 46}]

Clear (delete) IPv4 NP7 hardware sessions or sessions that have not been offloaded to NP7 processors. If you have set up an IPv4 filter, this command clears sessions that match the IPv4 filter.

(no options) clear IPv4 and NAT46 NP7 sessions.

`44` clear IPv4 NP7 sessions.

`46` clear NAT46 NP7 sessions.

`host` clear IPv4 sessions that have not been offloaded to NP7 processors.

diagnose sys npu-session clear6 [{66 | 64}]

Clear (delete) IPv6 hardware sessions or sessions that have not been offloaded to NP7 processors. If you have set up an IPv6 filter, this command clears sessions that match the IPv6 filter.

(no options) clear IPv6 and NAT64 NP7 sessions.

`66` clear IPv6 NP7 sessions.

`64` clear NAT64 NP7 sessions.

`host` clear IPv6 sessions that have not been offloaded to NP7 processors.

diagnose sys npu-session stat [verbose [{44 | 66 | 64 | 46}]]

View summary information about NP7 hardware sessions and hardware logging.

(no options) show the NP7 hardware session count, the hardware session setup rate, and some log rates.

`verbose [{44 | 66 | 64 | 46}]]` show more information about NP7 hardware sessions. Use the additional options to display more detailed information for a subset of the NP7 hardware sessions. Stats are also displayed for each session. If you have set up filters, information is displayed for sessions that match the filters.

Using the `verbose` option scans the SSEs of all available NP7 processors in the FortiGate and sends this data to the CPU. On a busy system processing a large number of hardware sessions, this process can send a very large number of messages that may overrun the messaging driver. As a result, the `verbose` output may show lower than expected session counts. This problem is expected to be addressed in future releases.

diagnose sys npu-session purge

Clear all NP7 hardware sessions.

diagnose sys npu-session filter {filter-options}

Filter the IPv4 sessions that you list or clear. You can use `filter-options` to display or clear sessions from specific VDOMs, display sessions for specific policy IDs, to specific source and destination addresses, and so on. Use the CLI help to list all of the options available. Use the `clear` option to clear the IPv4 filter. Use the `negate` option to create an inverse filter.

diagnose sys npu-session filter6 {filter-options}

Filter the IPv6 sessions that you list or clear. You can use `filter-options` to display or clear sessions from specific VDOMs, display sessions for specific policy IDs, to specific source and destination addresses, and so on. Use the CLI help to list all of the options available. Use the `clear` option to clear the IPv6 filter. Use the `negate` option to create an inverse filter.

Examples

To list IPv4 NP7 hardware sessions enter:

```
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
```

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 pmon for NP7 performance monitoring

You can use the `diagnose npu np7 pmon` command to get detailed NP7 performance information.

```
diagnose npu np7 pmon {<np7-id> | all} {0 | b | brief | 1 | v | verbose}
```

`np7-id` is the ID of the NP7 processor starting with 0.

`all` means show information for all NP7 processors.

`{0 | b | brief}` show information for all non-zero counters.

`{1 | v | verbose}` show all counters.

For more information about NP7 performance monitoring and general NP7 troubleshooting , see the Fortinet Community article Troubleshooting Tip: NP7 troubleshooting .

The command output shows load information for the NP7 EIF interfaces in the NP7 processor. The result is an overall picture of how busy the NP7 processor is. The following example shows the load on all NP7 EIF interfaces in one NP7 processor.

```
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
```

The following command output shows usage information for non-zero counters for NP7 processor with ID=3.

```
diagnose npu np7 pmon 3 b

[NP7_3]
         EIF0_IGR EIF1_IGR EIF0_EGR EIF1_EGR HRX      HTX      DFR
-------- -------- -------- -------- -------- -------- -------- --------
Usage%   0        0        0        0        0        0        0
-------- -------- -------- -------- -------- -------- -------- --------
```

```
         SSE0     SSE1     SSE2     SSE3
-------- -------- -------- -------- --------
Usage%   5        5        5        5
-------- -------- -------- -------- --------
         IPSEC    IPTI     IPTO     L2TI     L2TO     VEP      IVS
-------- -------- -------- -------- -------- -------- -------- --------
Usage%   0        0        0        0        0        0        0
-------- -------- -------- -------- -------- -------- -------- --------
         PLE      MSE      SYNK     DSE      NSS
-------- -------- -------- -------- -------- --------
Usage%   0        0        0        12       0
-------- -------- -------- -------- -------- --------
* EIFx_IGR: EIF ingress, EIFx_EGR: EIF egress
```

Hyperscale firewall license status

Use the `get system status` command to verify that your hyperscale firewall license is enabled:

```
get system status
...
Hyperscale license: Enabled
...
end
```

Displaying IP pool usage information

Use the following diagnose commands from a hyperscale firewall VDOM to display details about CGN IP pools 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 get-priv <public-ipv4> [<public-port>]
diagnose firewall ippool get-pub <private-ipv4>
diagnose firewall ippool get-pub6 <private-ipv6>
diagnose firewall ippool {list {pba | nat-ip | user} | stats}
```

`stats` list the total number of CGN IP pools that have been allocated, the number of currently active client IP addresses, NAT IP addresses, and PBA blocks.

`pba` list currently active source addresses of CGN clients and the PBA blocks assigned to them.

`user` list currently active source addresses of CGN clients and the number of PBA blocks assigned to them.

`nat-ip` list currently active public IP addresses and the number of PBA blocks and user sessions connected to each public IP.

`get-priv <public-ipv4> [<public-port>]` query private information of a public IPv4 address and optionally a port number.

`get-pub <private-ipv4>` query public information of a private IPv4 address.

`get-pub6 <private-ipv6>` query public information of a private IPv6 address.

diagnose firewall ippool list

Use `diagnose firewall ippool list` with no options to display 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 firewall usage information (basically placeholder information that doesn't represent actual CGN IP pool usage).
- NP7 hyperscale firewall policy engine (or PLE) usage information (actual CGN IP pool usage information).

If a CGN IP pool has not been added to a hyperscale firewall policy, then only the kernel firewall information is shown.

The following example includes a CGN IP pool named `test-cgn-pba-1` that has been added to a hyperscale firewall policy. The first 5 lines of output contain configuration and kernel firewall usage information. The final four lines of output, beginning with `grp=N/A` is NP7 hyperscale firewall policy engine (or PLE) usage information. These final four lines include the correct usage information for the CGN IP pool.

The IP pool in the example named `test-cgn-opba-1` has not been added to a hyperscale firewall policy and only contains configuration and kernel firewall usage information.

```
diagnose firewall ippool list
list ippool info:(vf=cgn-hw1)
ippool test-cgn-pba-1: id=1, block-sz=64, num-block=8, fixed-port=no, use=4
        ip-range=172.16.201.181-172.16.201.182 start-port=5117, num-pba-per-ip=944
        clients=1, inuse-NAT-IPs=1
        total-PBAs=1888, inuse-PBAs=1, expiring-PBAs=0, free-PBAs=99.95%
        allocate-PBA-times=1, reuse-PBA-times=0
        grp=N/A, start-port=8117, end-port=8629
        npu-clients=1, npu-inuse-NAT-IPs=1, total-NAT-IP=2
        npu-total-PBAs=16, npu-inuse-PBAs=4/0, npu-free-PBAs=75.00%/100.00%
        npu-tcp-sess-count=256, npu-udp-sess-count=0
ippool test-cgn-opba-1: id=2, block-sz=256, num-block=8, fixed-port=no, use=2
        ip-range=172.16.201.183-172.16.201.184 start-port=5117, num-pba-per-ip=236
        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
```

The following example shows two CGN IP pools named `cgn-pool1` and `cgn-pool2` that have been added to a CGN IP pool group named `cgn_pool_grp1`. The information displayed for the IP pools in the group is the same as is displayed for individual IP pools, except that the `grp` field includes an IP pool group name.

Also, 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. As a result, the usage information displayed for each IP pool is the same, since it is the information for the entire group.

```
F2K61F-TIGER-194-31 (global) # sudo cgn-hw1 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
```

diagnose firewall ippool list pba

This command lists the PBAs in the IP pools in the current VDOM. For each IP pool, the command lists the client IP, NAT IP, NAT port range, port block index, and a kernel reference counter. The final line of the command output shows the number of PBAs allocated by NP7 processors for this VDOM

```
diag firewall ippool list pba
user 10.1.100.200: 172.16.201.181 8117-8180, idx=0, use=1
user 10.1.100.200: 172.16.201.181 8181-8244, idx=1, use=1
user 10.1.100.200: 172.16.201.181 8245-8308, idx=2, use=1
user 10.1.100.200: 172.16.201.181 8309-8372, idx=3, use=1
Total pba in NP: 4
```

diagnose firewall ippool list nat-ip

This command lists the NAT IPs in use in the VDOM. For each NAT IP, the command shows the number of PBAs allocated for the NAT IP and the number of PBAs in use:

```
diag firewall ippool list nat-ip
NAT-IP 172.16.201.181: pba=8, use=4
Total nat-ip in NP: 1
```

diagnose firewall ippool list user

This command lists all of the user IP addresses allocated by NP7 processors for the current VDOM. For each user IP address, the command lists the number of PBAs assigned to the user IP and the number of PBAs being used. The final line of the command output shows the total number of user IPs in use for the current VDOM.

```
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
```

Session setup information

Use the `get sys performance status` command to show hardware session setup information:

```
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
```

HA hardware session synchronization status

Use the `get system ha status` command to show the status of the hardware HA session synchronization link.

```
get system ha status
...
HW SessionSync dev stats:
    FG421FTK19900013:
         port24: in-sync
...
```

Viewing and changing NP7 hyperscale firewall blackhole and loopback routing

You can use the following diagnose command to view the current LPM routing configuration. You can also use this command to add and remove routes. Because this is a diagnose command, any changes are reverted to defaults when the FortiGate restarts:

```
diagnose lpmd route {add | del | dump | query | stats | ktrie | debug}
```

`add` add a route to the NP7 policy engine routing table.

`del` delete a route from the NP7 policy engine routing table.

`dump` list the NP7 policy engine routing table.

`query` look up detailed information for LPM entries.

`stats` display LPM compiler statistics.

`ktrie {next_hop | stats | query | route | vdom}` display KTRIE routing database information.

debug {set | show | query} set debug flags, show current debug level, and query kernel route entries.

The syntax for the `add` and `del` command is:

```
diagnose lpmd route {add | del} <dst> <prefixlen> <gwy> <oif> <table> <scope> <type> <proto>
     <prio> <tos> <flags>
```

For blackhole and loopback routes, set `<flags>` to the following `nh_flags` values:

- For blackhole routes the `nh_flags` value is 0x80.
- For loopback routes, the `nh_flags` value is 0x100.

For example, use the following command to add a blackhole route to the NP7 policy engine routing table:

```
diagnose lpmd add 12.1.1.10 24 12.1.1.1 port24 254 253 1 2 0 1 1
```

The following command will delete this route from the NP7 policy engine routing table:

```
diagnose lpmd del 12.1.1.10 24 12.1.1.1 port24 254 253 1 2 0 1 1
```

NP7 hyperscale firewall packet sniffer

You can use the following command as a hyperscale firewall packet sniffer. This packet sniffer displays information about packets offloaded by NP7 processors. You can also use this command to mirror sniffed packets to a FortiGate interface.

```
diagnose npu sniffer {start | stop | filter}
```

Diagnose commands such as `diagnose npu sniffer` are intended for debug purposes only. Regular use of these commands can consume CPU and memory resources and cause other system related issues. Only use them when required and in the case of a packet sniffer, make sure to stop it when you are done using it. For example, to stop the NP7 packet sniffer, enter `diagnose npu sniffer stop`.

Use `start` and `stop` to start or stop displaying packets on the CLI. Before the sniffer will start you need to use the `filter` to specify the packets to display. Use the command `diagnose sniffer packet npudbg` to display sniffed packets on the CLI.

Use `filter` to create a definition of the types of packets to display. Filter options include:

`selector` you can create up to four filters (numbered 0 to 3). Use this command to create a new filter or select the stored filter to be used when you start the packet sniffer. You can also use this command to have multiple filters active at one time. See below for an example of sniffing using multiple active filters.

`intf <interface-name>` the name of an interface to display packets passing through that interface. You can monitor traffic on any interface except IPv4 or IPv6 IPsec VPN tunnel interfaces.

`dir {0 | 1 | 2}` the direction of the packets passing through the interface. `0` displays ingress packets, `1` displays egress packets, and `2` displays both ingress and egress packets.

`ethtype <type>` the ethertype of the packets to sniff if you want to see non-IP packets.

`protocol <number>` the IP protocol number of the packets to sniff in the range 0 to 255. The packet sniffer can only sniff protocols that can be offloaded by the NP7 processors.

`srcip <ipv4-ip-address>/<ipv4-mask>` an IPv4 IP address and netmask that matches the source address of the packets to be sniffed.

`dstip <ipv4-ip-address>/<ipv4-mask>` an IPv4 IP address and netmask that matches the destination address of the packets to be sniffed.

`ip <ipv4-ip-address>/<ipv4-mask>` an IPv4 IP address and netmask that matches a source or destination address in the packets to be sniffed.

`srcip6 <ipv6-ip-address>/<ipv6-mask>` an IPv6 IP address and netmask that matches the source address of the packets to be sniffed.

`dstip6 <ipv6-ip-address>/<ipv6-mask>` an IPv6 IP address and netmask that matches the destination address of the packets to be sniffed.

`ip6 <ipv6-ip-address>/<ipv6-mask>` an IPv6 IP address and netmask that can match source or destination addresses in the packets to be sniffed.

`sport <port-number>` layer 4 source port of the packets to be sniffed.

`dport <port-number>` layer 4 destination port of the packets to be sniffed.

`port <port-number>` layer 4 source or destination port of the packets to be sniffed.

`outgoing_intf <interface>` the name of the interface out of which to send mirrored traffic matched by the filter.

`outgoing_vlan <vlan-id>` the VLAN ID added to mirrored traffic matched by the filter and sent out the mirror interface.

`clear` clear all filters.

Packet sniffer examples

First, a basic example to sniff offloaded TCP packets received by the port23 interface. In the following example:

- The first line clears the filter.
- The second line sets the sniffer to look for packets on port23.
- The third line looks for packets exiting the interface.
- The fourth line looks for TCP packets.
- The fifth line starts the sniffer.
- The sixth line starts displaying the packets on the CLI.

```
diagnose npu sniffer filter
diagnose npu sniffer filter intf port23
diagnose npu sniffer filter dir 2
diagnose npu sniffer filter protocol 6
diagnose npu sniffer start

diagnose sniffer packet npudbg
```

Second, an example that uses the following two filters:

- The first filter, selector 0, looks for incoming and outgoing TCP packets on port1.
- The second filter, selector 1, looks for outgoing UDP packets on port2.
- The final line starts displaying packets for both filters on the CLI.

```
diagnose npu sniffer filter selector 0
diagnose npu sniffer filter intf port1
diagnose npu sniffer filter protocol 6
diagnose npu sniffer filter dir 2
diagnose npu sniffer start

diagnose npu sniffer filter selector 1
diagnose npu sniffer filter intf port2
diagnose npu sniffer filter protocol 17
diagnose npu sniffer filter dir 1
diagnose npu sniffer start
```

```
diagnose sniffer packet npudbg
```

www.fortinet.com