New Features

FortiADC 7.6.5

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

December 23, 2025
FortiADC 7.6.5 New Features

TABLE OF CONTENTS

Change Log **5**
Overview **6**
Index 7
7.6.0 8
7.6.1 10
7.6.2 12
7.6.3 13
7.6.4 13
7.6.5 14
Web Application Firewall **15**
Enhanced file type detection 7.6.1 16
Client IP support for attack source identification 7.6.1 25
WAF Adaptive Learning 28
Bot Detection enhancement 48
Security Fabric **55**
FortiGate Security Fabric Connector 56
FortiView **62**
OWASP Top 10 Compliance dashboard 63
System **70**
Settings 71
Send FortiADC Threat Telemetry to FortiGuard 7.6.1 72
NTP authentication 7.6.1 76
Administrator lockout controls in CLI 79
Direct VDOM Access for Administrators 80
Cloud Auto Scaling 83
Azure Autoscaling for FortiADC VMSS 7.6.1 84
High Availability 87
Enhance Status Reporting Accuracy in Active-Passive HA 7.6.2 88
HA cluster supports maximum 8 member nodes 89
HA management interface network options via CLI 95
Virtual MAC address option as interface in Active-Passive HA via CLI 97
New and enhanced CLI commands to force HA nodes into standby mode 98
Administrator 100
RADIUS and TACACS+ VDOM Override Support 7.6.5 101
RADIUS and TACACS+ Access Profile Override via CLI 7.6.4 102
SNMP 103
Enhanced SNMP Authentication and Encryption 104
Certificate 107
ACME TLS-ALPN-01 Enhancements 108
Network **113**
IPv6 Support for HA Management Interface 7.6.3 114
IPv6 Router Advertisement Support via CLI 7.6.3 115
IPsec-Based Authentication and Encryption for OSPFv3 via CLI 7.6.3 119

Link Layer Discovery Protocol (LLDP) Support 7.6.2 122
Support for OSPF Version 3 (OSPFv3) 7.6.1 129
Server Load Balance 135
L4 Virtual Server Support for ESP Packets (IPsec VPN without NAT-T) 7.6.3 137
Server-side support for HTTP/2 connections 7.6.1 139
Client address option enabled for HTTP/3 virtual server 7.6.1 149
Scripting groups for predefined HTTP scripts 7.6.1 151
New HTTP script for enhanced error handling 7.6.1 158
Waiting Room for virtual queuing through HTTP scripting 160
AWS autoscaling group discovery 166
New health check down options 169
HTTP3 support for HTTP to HTTPS Redirection 170
Link Load Balance 172
SLB local traffic support 172
Global Load Balance 173
Multiple Global DNS Policy support in FQDN zones 174
DNS forwarding support at zone level with no matching hostname 176
DNS forwarding log debug in CLI 179
User Authentication 180
User Group Match Conditions for Authentication Control 7.6.5 181
Extended maximum authentication timeout 7.6.1 182
Log & Report 185
Kafka Integration for Log Export 7.6.4 186
Enhanced Syslog encryption via CLI 7.6.1 189
Syslog support for IPv6 FQDNs 7.6.1 196
GUI 197
Improved firmware upgrade process with progress tracking 7.6.1 198
Platform 199
Expanded Local Certificate Group Member Limit 7.6.4 201
OpenSSL Upgrade to 3.1.8 7.6.4 202
OCI DRCC support 7.6.4 203
New FortiADC-VMUL License Model and Expanded VDOM Support for Virtual Appliances 7.6.3 204
Data Partition Expansion 7.6.2 205
Support FortiFlex Token in User Data for Public Cloud BYOL 7.6.2 207
Instance type support for AWS/Azure/GCP 7.6.1 208
FortiFlex support for cloud-init in Proxmox (KVM) 208
Troubleshooting 209
New CLI Commands for Virtual Server and Pool Statistics 7.6.4 210
Health Check debug log enhancement in CLI 212

Change Log

Date	Change Description
July 22, 2024	FortiADC 7.6 New Features initial release.
November 25, 2024	Added 7.6.1 new features.
April 25, 2025	Added 7.6.2 new features.
August 1, 2025	Added 7.6.3 new features.
October 8, 2025	Added 7.6.4 new features.
December 23, 2025	Added 7.6.5 new features.

Overview

This guide provides details of new features introduced in FortiADC 7.6. For each feature, the guide provides detailed information on configuration, requirements, and limitations, as applicable. Features are organized into the following sections:

- Web Application Firewall on page 15
- Security Fabric on page 55
- FortiView on page 62
- System on page 70
- Server Load Balance on page 135
- Link Load Balance on page 172
- Global Load Balance on page 173
- Platform on page 199
- Troubleshooting on page 209

For features introduced in 7.6.1 and later versions, the version number is appended to the end of the topic heading. For example, GUI enhancements for FortiGuard DLP service 7.4.1 was introduced in 7.4.1. If a topic heading has no version number at the end, the feature was introduced in 7.4.0.

For a list of features organized by version number, see Index on page 7.

Index

The following index provides a list of all new features added to FortiADC 7.6. The index allows you to quickly identify the version where the feature first became available in FortiADC.

Select a version number to navigate in the index to the new features available for that patch:

- 7.6.0 on page 8
- 7.6.1 on page 10
- 7.6.2 on page 12
- 7.6.3 on page 13
- 7.6.4 on page 13
- 7.6.5 on page 14

7.6.0

Web Application Firewall

- WAF Adaptive Learning on page 28
- Bot Detection enhancement on page 48

Security Fabric

- FortiGate Security Fabric Connector on page 56

FortiView

- OWASP Top 10 Compliance dashboard on page 63

System

Settings

- Administrator lockout controls in CLI on page 79
- Direct VDOM Access for Administrators on page 80

High Availability

- HA cluster supports maximum 8 member nodes on page 89
- HA management interface network options via CLI on page 95
- Virtual MAC address option as interface in Active-Passive HA via CLI on page 97
- New and enhanced CLI commands to force HA nodes into standby mode on page 98

Certificate

- ACME TLS-ALPN-01 Enhancements on page 108

Server Load Balance

- Waiting Room for virtual queuing through HTTP scripting on page 160
- AWS autoscaling group discovery on page 166
- New health check down options on page 169
- HTTP3 support for HTTP to HTTPS Redirection on page 170

Link Load Balance

- SLB local traffic support on page 172

Global Load Balance

- Multiple Global DNS Policy support in FQDN zones on page 174
- DNS forwarding support at zone level with no matching hostname on page 176
- DNS forwarding log debug in CLI on page 179

Platform

- FortiFlex support for cloud-init in Proxmox (KVM) on page 208

Troubleshooting

- Health Check debug log enhancement in CLI on page 212

7.6.1

Web Application Firewall

- Enhanced file type detection 7.6.1 on page 16
- Client IP support for attack source identification 7.6.1 on page 25

Network

- Support for OSPF Version 3 (OSPFv3) 7.6.1 on page 129

Server Load Balance

- Server-side support for HTTP/2 connections 7.6.1 on page 139
- Client address option enabled for HTTP/3 virtual server 7.6.1 on page 149
- Scripting groups for predefined HTTP scripts 7.6.1 on page 151
- New HTTP script for enhanced error handling 7.6.1 on page 158

System

Settings

- Send FortiADC Threat Telemetry to FortiGuard 7.6.1 on page 72
- NTP authentication 7.6.1 on page 76

Cloud Auto Scaling

- Azure Autoscaling for FortiADC VMSS 7.6.1 on page 84

SNMP

- Enhanced SNMP Authentication and Encryption on page 104

User Authentication

- Extended maximum authentication timeout 7.6.1 on page 182

Log & Report

- Enhanced Syslog encryption via CLI 7.6.1 on page 189
- Syslog support for IPv6 FQDNs 7.6.1 on page 196

GUI

- Improved firmware upgrade process with progress tracking 7.6.1 on page 198

Platform

- Instance type support for AWS/Azure/GCP 7.6.1 on page 208

7.6.2

System

High Availability

- Enhance Status Reporting Accuracy in Active-Passive HA 7.6.2 on page 88

Network

- Link Layer Discovery Protocol (LLDP) Support 7.6.2 on page 122

Platform

- Data Partition Expansion 7.6.2 on page 205
- Support FortiFlex Token in User Data for Public Cloud BYOL 7.6.2 on page 207

7.6.3

Network

- IPv6 Support for HA Management Interface 7.6.3 on page 114
- IPv6 Router Advertisement Support via CLI 7.6.3 on page 115
- IPsec-Based Authentication and Encryption for OSPFv3 via CLI 7.6.3 on page 119

Server Load Balance

- L4 Virtual Server Support for ESP Packets (IPsec VPN without NAT-T) 7.6.3 on page 137

Platform

- New FortiADC-VMUL License Model and Expanded VDOM Support for Virtual Appliances 7.6.3 on page 204

7.6.4

System

Administrator

- RADIUS and TACACS+ Access Profile Override via CLI 7.6.4 on page 102

Log & Report

- Kafka Integration for Log Export 7.6.4 on page 186

Platform

- Expanded Local Certificate Group Member Limit 7.6.4 on page 201
- OpenSSL Upgrade to 3.1.8 7.6.4 on page 202
- OCI DRCC support 7.6.4 on page 203

7.6.5

System

Administrator

- RADIUS and TACACS+ VDOM Override Support 7.6.5 on page 101

User Authentication

- User Group Match Conditions for Authentication Control 7.6.5 on page 181

Web Application Firewall

The FortiADC 7.6 release includes new features and enhancements in **Web Application Firewall**:

Enhanced file type detection 7.6.1 on page 16

FortiADC has enhanced its WAF file restriction module with file type identification based on extensions, expanding detection beyond the previously supported text, video, audio, and compressed files.

Client IP support for attack source identification 7.6.1 on page 25

FortiADC introduces the new **Use Original IP** option in the WAF Profile that leverages the "X-Forwarded-For" HTTP extension header to enable WAF modules to inspect the client's original IP address rather than the HTTP source IP for attack source identification.

WAF Adaptive Learning on page 28

FortiADC's new WAF Adaptive Learning feature leverages machine learning technology to autonomously identify and mitigate potential threats. By dynamically generating and refining protection policies, FortiADC significantly enhances the intelligence and automation of security defenses. The Adaptive Learning engine utilizes advanced machine learning algorithms to perform deep packet inspection and traffic analysis, constructing comprehensive datasets from incoming traffic patterns. Using these datasets, the Adaptive Learning engine can then provide actionable recommendations for optimizing WAF policies, ensuring they are meticulously aligned with the unique traffic characteristics of each application.

Note: The WAF Adaptive Learning feature requires the WAF Signature license or Application Security bundle license. If you do not already have a valid license, FortiADC offers a 30-day trial license to explore the WAF Adaptive Learning functionality. The trial license is activated automatically upon upgrading to FortiADC 7.6.0.

Bot Detection enhancement on page 48

FortiADC has enhanced its Bot Detection functionality to enable more granular classification of Malicious Bots, including detailed categorization. Within the Bot Detection policy, you can configure the enable/disable status of each bot within the Malicious Bot category, allowing precise inclusion or exclusion for detection.

Enhanced file type detection - 7.6.1

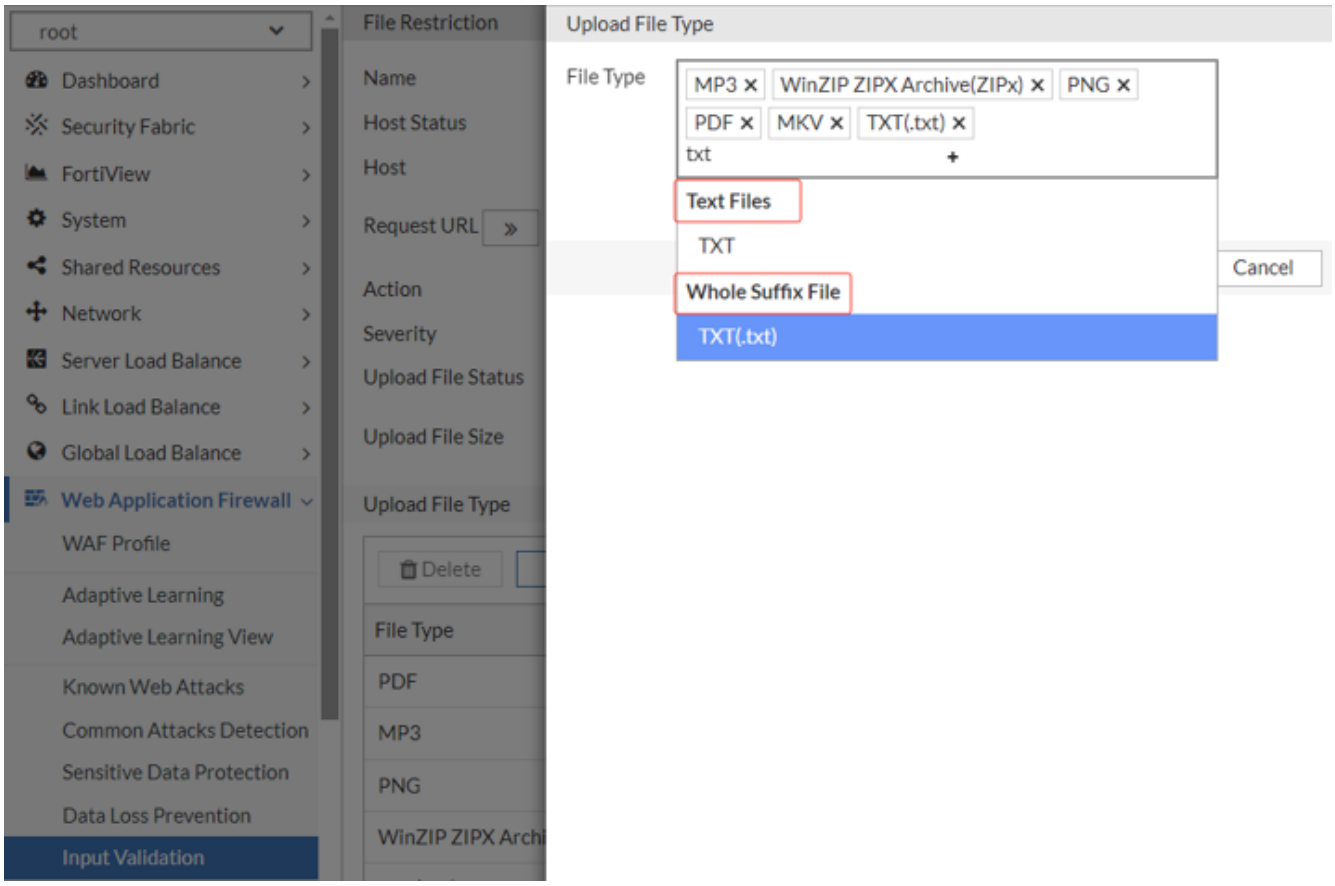
FortiADC has enhanced its WAF file restriction module with file type identification based on extensions, expanding detection beyond the previously supported text, video, audio, and compressed files. The update also streamlines configuration by clearly displaying file categories, offering more granular control over traffic handling and improving security enforcement through precise file type recognition.

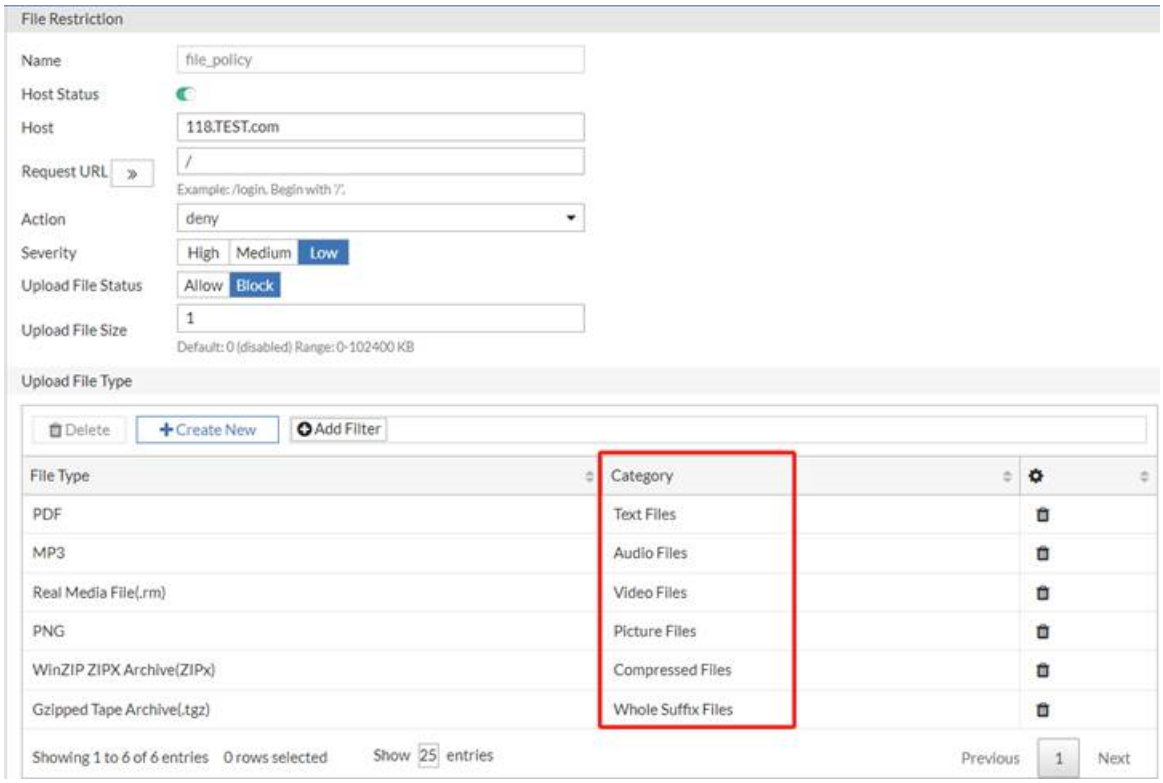
Other WAF modules, including Adaptive Learning, can leverage the enhanced file type detection mechanism, utilizing file signatures and suffix matching, for precise file type identification.

This information is also available in the FortiADC 7.6.1 Administration Guide:

- Configuring a File Restriction rule

Updates to the File Restriction configuration:





File Type Identification in FortiADC

FortiADC employs two methods for file type identification: file type signatures and suffix matching.

File Type Signatures:

FortiADC examines specific attributes of a file to determine its content type by detecting unique signatures, or magic codes, associated with predefined file types based on MIME types and magic numbers (file signatures). If the detected file type matches one specified in the file restriction rule, the system enforces the corresponding action. Supported file type categories include **Audio Files**, **Compressed Files**, **Picture Files**, **Text Files**, and **Video Files**.

Suffix Matching:

FortiADC can also identify files based on their suffix (extension). If the file suffix matches an entry under the **Whole Suffix Files** category in the file restriction rule, the associated action is triggered.

When both file type signature and suffix matching are configured, suffix matching takes precedence. If the file suffix matches, the file restriction rule is applied immediately. If the suffix does not match but the file signature does, the file restriction rule will still be enforced.

Additionally, if multiple files of different types are uploaded in a single HTTP transaction, and one file type violates the rule, the entire transaction will be rejected, resulting in all files being blocked.

For the full list of the supported file types, see Supported File Types on page 19.

To configure a File Restriction rule:

1. Go to **Web Application Firewall > Input Validation**.
2. Click the **File Restriction** tab.
3. Click **Create New** to display the configuration editor.
4. Configure the following File Restriction settings:

Setting	Description
Name	Enter a unique File Restriction policy name. Valid characters are `A-Z`, `a-z`, `0-9`, _, and -. No space is allowed. Note: Once saved, the name of a File Restriction policy cannot be changed.
Host Status	Enable to require that the Host: field of the HTTP request match a protected host name's entry in order to match the URL access rule. Also configure Host.
Host	The **Host** option is available if **Host Status** is **enabled**. Select which protected host name's entry (either a web host name or IP address) that the Host: field of the HTTP request must be in to match the URL access rule.
Request URL	The HTTP request URL must be start with /. eg./login. This item must be set when configuring the rule. FortiADC will match the other item (rule) when matching the request URL; if the match fails, FortiADC will not attempt to match others.
Action	Select the action profile that you want to apply. See Configuring WAF Action objects. The default value is Alert.
Severity	When FortiADC records violations of this rule in the attack log, each log message contains a **Severity Level** (severity_level) field. Select which severity level FortiADC uses when using Input Validation: • Low • Medium • High The default value is **Low**.
Upload File Status	Allow: Only allow the selected file type to upload. Block: Block any upload of the selected file type.
Upload File Size	The maximum size of the uploaded file.

5. Click **Save**.
   Once the File Restriction configuration is saved, the Upload File Type section can be configured.
6. Under the **Upload File Type** section, click **Create New** to display the configuration editor.
7. In the **File Type** field, select the supported file types for the uploaded file.
8. Click **Save** to update the File Restriction configuration.

After the File Restriction rule has been saved, you can include it in an Input Validation Policy.

Supported File Types

Category	File Type
Audio Files	MP3 MIDI WAVE AVI Apple CoreAudio (.caf) Microsoft Advanced Streaming (.asf) Real Audio File (.ra) Apple Lossless Audio (.m4a) Digital Speech Standard (.dss) Advanced Audio Coding (.aac)
Compressed Files	RAR ZIP TAR 7-ZIP Debian Package Microsoft Cabinet File Unix Archiver File (.ar) Installshield Cabinet Archive Data AIN Archive Data (.ain) BZIP2 Archive (.bz2) WinZIP ZIPX Archive (ZIPx) Gzipped Tape Archive (TGZ) Extensible Archive (XAR)
Picture Files	GIF JPG BMP PNG TIFF/TIF Windows Metafile Format (.wmf) Corel Draw Picture Windows Icon Microsoft Document Image (.mdi) Windows Enhanced Metafile (.emf) Photoshop Image File (.psd) JPEG-2000 Image File Format (.jp2) Multipage PCX Bitmap File (.dcx)
Text Files	PDF XML

Category	File Type
	CHM EXE RTF Windows Help File (.hlp) Windows Mobile Note (.pwi) Windows Registry Text (.reg) SQL Server 2000 Database (.mdf) Java Archive (.jar) Windows Printer Spool File (.shd) Windows Shortcut File (.lnk) Quark Express Document (.qxd) Windows MS Info File (.mof) Microsoft Access Database (.mdb) SPSS Data (.sav) XPS Word (.docx) Word Macro-Enabled (.docm) Word Template (.dotx) Word Macro-Enabled Template (.dotm) Excel (.xlsx) Excel Macro-Enabled (.xlsm) Excel Template (.xltx) Excel Macro-Enabled Template (.xltm) Excel Add-In (.xlam) PPT (.pptx) PPT Macro-Enabled (.pptm) PPT Template (.potx) PPT Macro-Enabled Template (.potm) PPT Add-In (.ppam) PPT Show (.ppsx) PPT Macro-Enabled Show (.ppsm) Visio Drawing (.vsdx) Visio Macro-Enabled Drawing (.vsdm) Visio Stencil (.vssx) Visio Macro-Enabled Stencil (.vssm) Visio Template (.vstx) Visio Macro-Enabled Template (.vstm) VMware Virtual Disk File (.vmdk) RedHat Package Manager file (.rpm) Lotus WordPro document (.lwp)

Category	File Type
	Adobe Encapsulated PostScript file (.eps) Lotus 1-2-3 spreadsheet (.wk) SkinCrafter skin file (.skf) Nero CD Compilation (.nri) TXT Microsoft Office Word (.doc) Microsoft Office Excel (.xls) Microsoft Office PowerPoint (.ppt) Hancom Office Hanword (.hwp) Electronic Publication (.epub) Dynamic link library (.dll) SYS File (.sys) COM File (.com) CMD File (.cmd) Binary File (.bin) Scalable Vector Graphics (.svg) PHP (.php) Perl (.pl) Python (.py) Ruby (.rb) Microsoft Software Installer (.msi) Batch File (.bat) Privacy Enhanced Mail (.pem) x509 certificate (.cer) x509 certificate (.crt)
Video Files	Real Media File (.rm) MPEG v4 3GPP Macromedia Flash Windows Animated Cursor DVD Video Movie File (.vob) MKV
Whole Suffix Files	TXT (.txt) ZIP (.zip) 7-ZIP (.7z) Debian Package (.pkg) Unix Archiver File (.ar) AIN Archive Data (.ain) BZIP2 Archive (.bz2)

Category	File Type
	Gzipped Tape Archive (.tgz) Word (.docx) Word Macro-Enabled (.docm) Word Template (.dotx) Word Macro-Enabled Template (.dotm) Excel (.xlsx) Excel Macro-Enabled (.xlsm) Excel Template (.xltx) Excel Macro-Enabled Template (.xltm) Excel Add-In (.xlam) PPT (.pptx) PPT Macro-Enabled (.pptm) PPT Template (.potx) PPT Macro-Enabled Template (.potm) PPT Add-In (.ppam) PPT Show (.ppsx) PPT Macro-Enabled Show (.ppsm) Visio Drawing (.vsdx) Visio Macro-Enabled Drawing (.vsdm) Visio Stencil (.vssx) Visio Macro-Enabled Stencil (.vssm) Visio Template (.vstx) Visio Macro-Enabled Template (.vstm) PDF (.pdf) XML (.xml) EXE (.exe) Rich Text Format (.rtf) Windows Help File (.hlp) Windows Mobile Note (.pwi) Windows Registry Text (.reg) SQL Server 2000 Database (.mdf) Java Archive (.jar) Windows Printer Spool File (.shd) Window Shortcut File (.lnk) Quark Express Document (.qxd) Windows MS Info File (.mof) Microsoft Access Database (.mdb) SPSS Data (.sav) RedHat Package Manager file (.rpm) VMware Virtual Disk File (.vmdk)

Category	File Type
	Lotus WordPro document (.lwp) Adobe Encapsulated PostScript file (.eps) Lotus 1-2-3 spreadsheet (.wk) SkinCrafter skin file (.skf) Nero CD Compilation (.nri) Microsoft Office Word (.doc) Microsoft Office Excel (.xls) Microsoft Office PowerPoint (.ppt) Hancom Office Hanword (.hwp) PHP (.php) JSP (.jsp) ASPX (.aspx) GIF (.gif) JPG (.jpg) BMP (.bmp) PNG (.png) Microsoft Metafile Format (.wmf) Windows Icon (.icon) Microsoft Document Image (.mdi) Windows Enhanced Metafile (.emf) Photoshop Image File (.psd) JPEG-2000 Image File Format (.jp2) Multipage PCX Bitmap File (.dcx) SQL (.sql) Cascading Style Sheets (.css) ASP (.asp) CSV (.csv) PHP3 (.php3) PHTML (.phtml) Workflow File (.workflow) Scalable Vector Graphics (.svg) MSG (.msg) OpenDocument Spreadsheet (.ods) OpenDocument Text (.odt) Privacy-Enhanced Mail (.pem) Electronic Publication (.epub) Advanced Audio Coding (.aac) Personal Information Exchange (.pfx) Personal Information Exchange (.p12) Microsoft Software Installer (.msi)

Category	File Type
	Batch File (.bat) Dynamic link library (.dll) SYS File (.sys) COM File (.com) CMD File (.cmd) Binary File (.bin) Tab-Separated Values (.tsv) Android Package Kit (.apk) Compressed package file (.xapk) APK set archive (.apks) APKMirror Bundle file (.apkm) Distinguished Encoding Rules (.der)

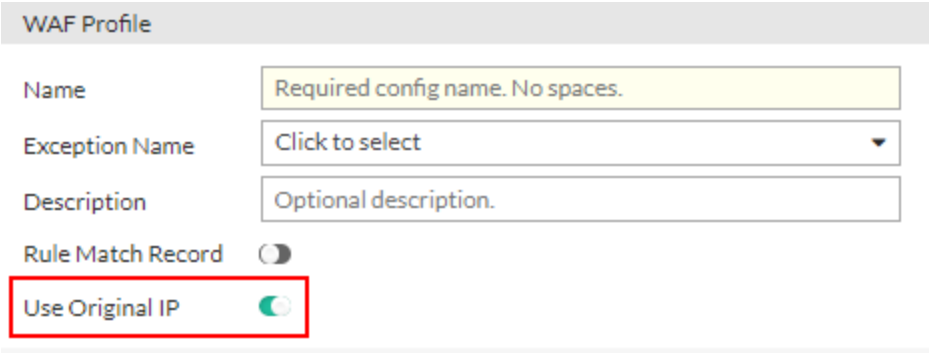
Client IP support for attack source identification - 7.6.1

FortiADC introduces the new **Use Original IP** option in the WAF Profile that leverages the "X-Forwarded-For" HTTP extension header to enable WAF modules to inspect the client's original IP address rather than the HTTP source IP for attack source identification. When enabled, this feature updates log records with the original client IP upon attack detection. It applies to both HTTP and HTTPS traffic and supports IPv4 and IPv6 addressing, ensuring comprehensive inspection across protocols.

This information is also available in the FortiADC 7.6.1 Administration Guide and CLI Reference:

- Configuring a WAF Profile
- config security waf profile

When FortiADC is deployed behind a proxy, all incoming requests appear to originate from the proxy's IP address. If FortiADC detects malicious requests, it may block the proxy's IP, resulting in service disruptions for all legitimate clients routed through it. The **X-Forwarded-For** header, appended by the proxy, contains the original client's IP address. FortiADC can utilize this header to accurately identify the client, enabling the application of relevant security policies and actions. This allows FortiADC to log attack incidents based on the original client IP, improving precision in threat detection and response mechanisms.



To enable Use Original IP in the WAF Profile:

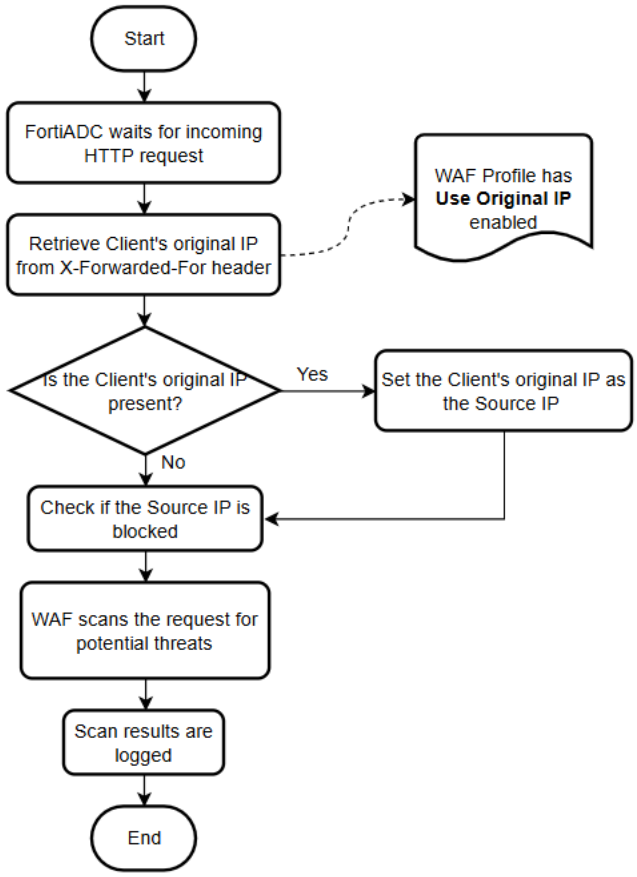
1. Navigate to Web Application Firewall > WAF Profile.
   The configuration page displays the WAF Profile tab.
2. Click **Create New** to display the configuration editor. Alternative, you can edit an existing user-defined WAF Profile.
3. Enable the **Use Original IP** option. This is disabled by default.
   When enabled, the client's original IP address from the "X-Forwarded-For" header is used in place of the standard source IP, impacting specific WAF functions. This setting affects WAF Blocked IP matching, Known Good Bots in Bot Detection, and source IP-based rules in the Exception feature. Log records are updated with the original client IP for detected attacks, ensuring accurate logging. This feature applies to both HTTP and HTTPS traffic and supports IPv4 and IPv6 addressing.
   Note: If the HTTP request packet does not contain the "X-Forwarded-For" information or contains invalid data, the system will default to using the HTTP source IP address instead.

CLI updates in `config security waf profile`:

```
config security waf profile
   edit <name>
      set use-original-ip {enable|disable}
   next
end
```

Process Flow

The following process outlines the steps taken by FortiADC when handling an incoming HTTP request with the **Use Original IP** option enabled in the WAF profile. This workflow ensures accurate client identification and effective threat detection while maintaining network security.



1. FortiADC awaits for an incoming HTTP request.
2. Retrieve the client's original IP:
   If **Use Original IP** is enabled in the WAF profile, retrieve the client's original IP address from the X-Forwarded-For header.
3. Check for the client's original IP in the X-Forwarded-For header:
   a. **Yes**: If the client's original IP is present, set it as the source IP.
   b. **No**: If absent, proceed to the next checkpoint.
4. Verify if the source IP is blocked:
   When **Use Original IP** is enabled, use the client's original IP from the X-Forwarded-For header instead of the

standard Source IP for this check.

5. Scan the request for potential threats:
   The WAF scans the request for potential threats. Certain rules, such as the **Source IP** in Exception lists and Known Good Bots in Bot Detection, will use the client's original IP if **Use Original IP** is enabled.
6. Log the scan results:
   The results are logged, and if **Use Original IP** is enabled, the client's original IP from the X-Forwarded-For header is recorded in place of the standard source IP.

WAF Adaptive Learning

The new FortiADC WAF Adaptive Learning feature dynamically generates tailored recommendations to refine protection policies, enabling rapid and effective WAF configuration to counter threats. By continuously performing deep packet inspection and traffic analysis, the Adaptive Learning engine constructs comprehensive datasets from traffic patterns, allowing it to autonomously identify and mitigate threats with precision. Using these insights, the Adaptive Learning engine generates actionable recommendations for WAF policy optimization, ensuring security measures align with the unique traffic characteristics of each web application. This enhances both security posture and operational performance, allowing for real-time adjustment and fine-tuning of security measures.

Note: The WAF Adaptive Learning feature requires the WAF Signature license or Application Security bundle license. If you do not already have a valid license, FortiADC offers a 30-day trial license to explore the WAF Adaptive Learning functionality. The trial license is activated automatically upon upgrading to FortiADC 7.6.0.

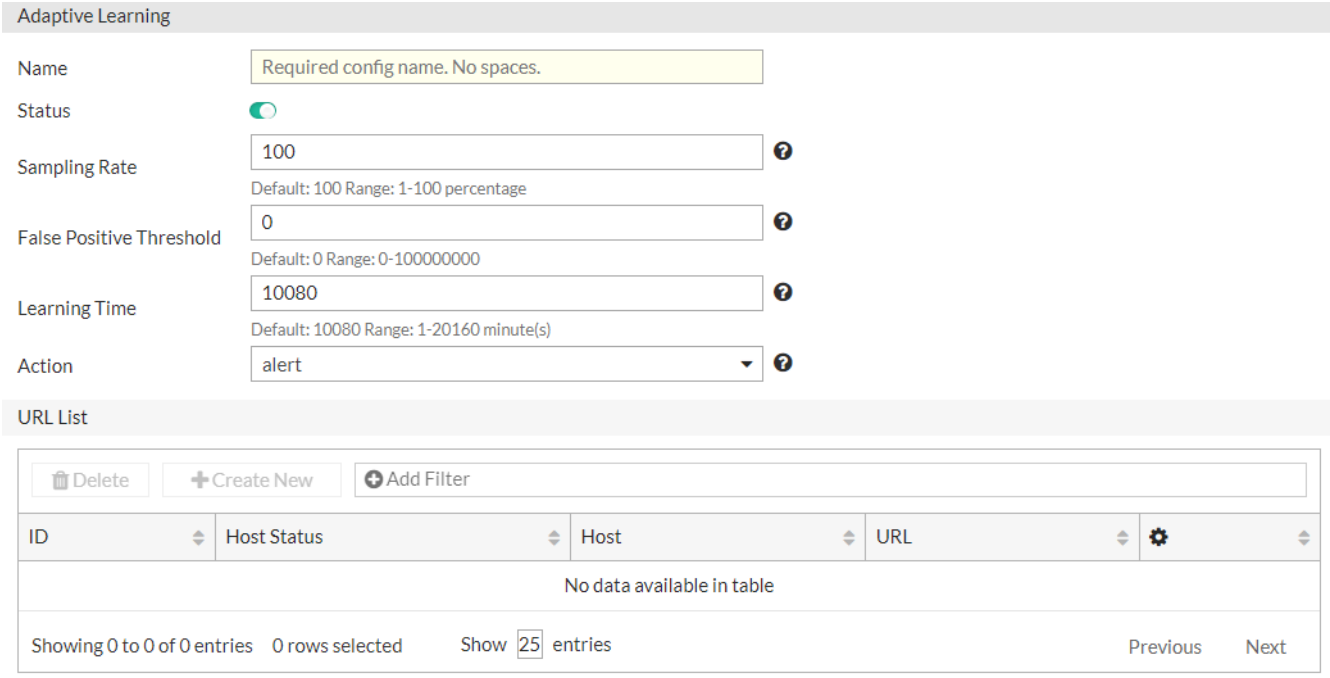
This information is also available in the FortiADC 7.6.0 Administration Guide and Script Reference Guide:

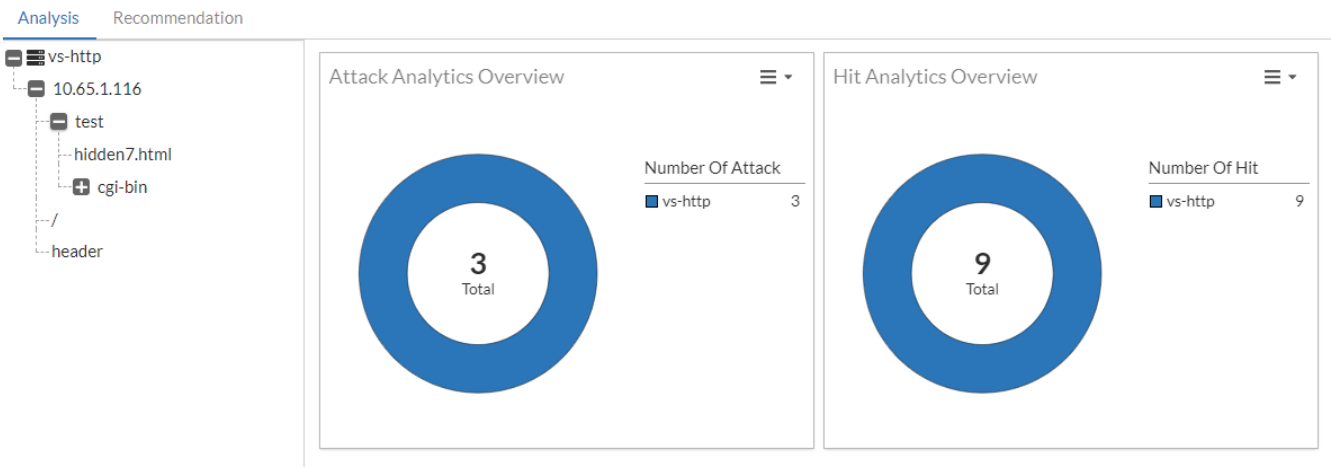
- WAF Adaptive Learning
- config security waf adaptive-learning
- diagnose debug module autolearn

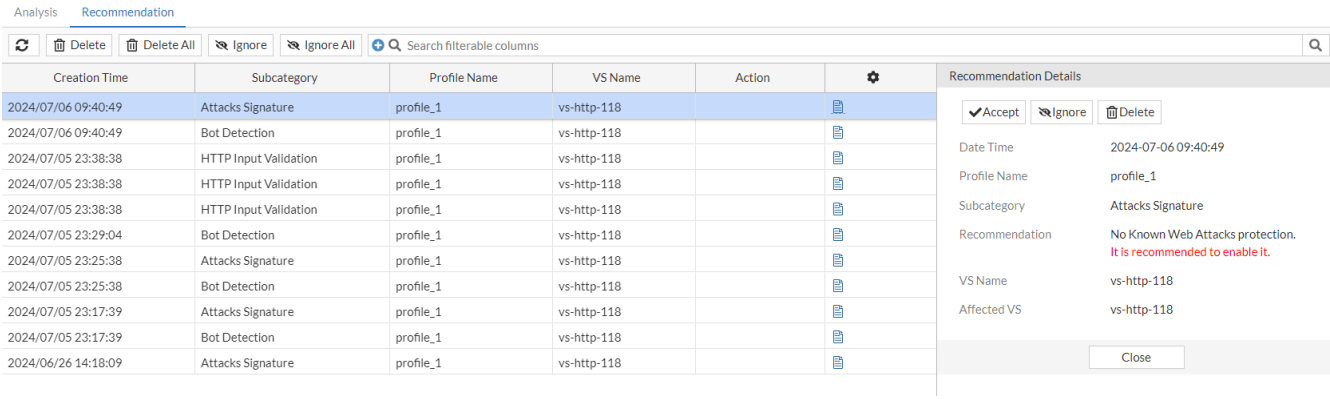
Here are some practical examples of how users can apply the WAF Adaptive Learning functionality:

- Create new WAF policies as part of a new virtual server setup — users can utilize the Adaptive Learning recommendation functionality to automatically generate and apply basic WAF policies for a new WAF Profile.
- Enhance the security of existing WAF policies — users can apply the Adaptive Learning recommendations that identify application traffic pattern changes to adjust policy settings accordingly.
- Identify false positive triggers that unnecessarily trigger WAF policy violations — users can leverage the Adaptive Learning engine's functionality to identify false positive triggers in WAF policies to adjust policy settings to exclude these false triggers.

The WAF Adaptive Learning engine features two key components: the Adaptive Learning policy that defines the dataset, and the Adaptive Learning statistics that include the analytical results and actionable recommendations. To learn more details about how to configure the Adaptive Learning policies and view the analysis results, see Adaptive Learning configuration overview on page 30 and Adaptive Learning analysis and recommendations on page 35.







In FortiADC 7.6.0, the Adaptive Learning feature supports the following WAF modules:

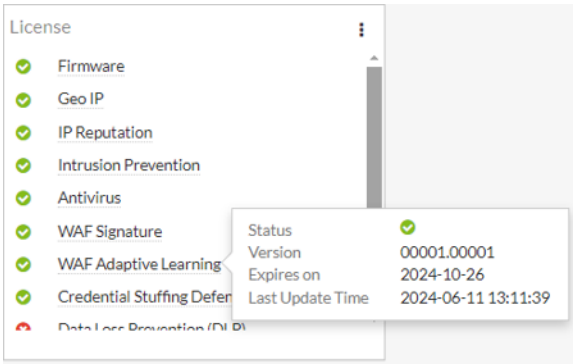
- Web Attack Signature
- Bot Detection
- Input Validation (Parameter validation and Hidden field validation)
- JSON Protection
- XML Protection

More WAF features will be supported in Adaptive Learning in future releases.

Licensing requirements

The WAF Signature license or Application Security bundle license (which covers Web Security Signatures and Adaptive Learning) is required to use the WAF Adaptive Learning feature.

If there is no valid WAF Signature license, a 30-day trial for the Adaptive Learning feature is available. This 30-day trial will be automatically activated upon the first boot up in FortiADC 7.6.0. Once the 30-day trial expires, all Adaptive Learning related configurations will become hidden in the GUI and the existing Adaptive Learning policy will stop working if no valid WAF Signature license is in place.



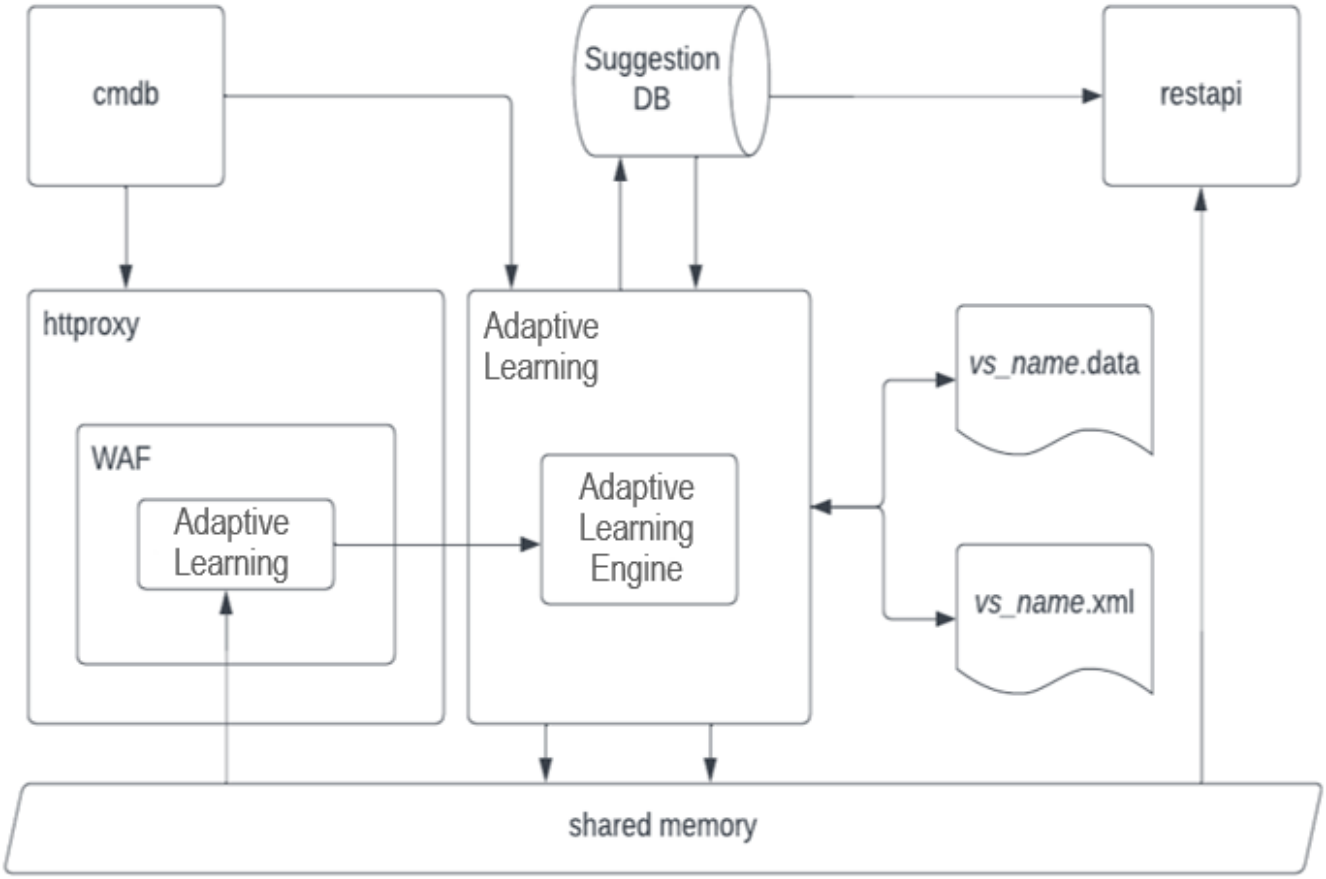
		Network Security	Application Security	AI Security
Network Security	IP Reputation and GeoIP	●	●	●
	Advanced Malware Protection	●	●	●
	Intrusion Prevention Service	●	●	●
Application Security	WAF Signatures and Adaptive Learning		●	●
	Credential Stuffing Defense		●	●
	Sandbox Cloud		●	●
	FortiGuard DLP		●	●
AI Security	Threat Analytics			●
	Advanced Bot Protection			●

Adaptive Learning configuration overview

To enable the FortiADC Adaptive Learning engine for continuous deep analysis of incoming traffic, you can configure an Adaptive Learning policy or use one of the three predefined configurations offered by FortiADC. By applying this to a WAF profile, the engine samples traffic at a defined rate based on the configured WAF policies. This traffic sampling

allows the engine to dynamically learn and adapt to various web application elements, such as hostnames, URLs, query parameters, hidden fields, cookies, and file types. This advanced learning capability ensures that the WAF policies are precisely tuned to specific traffic characteristics, enhancing both security and performance.

The Adaptive Learning workflow begins with the httproxy module, where it receives HTTP requests and responses and then forwards them to the WAF Adaptive Learning module for inspection. The module filters the traffic, extracting critical elements such as URLs, source IP addresses, HTTP methods, parameters, cookies, and status codes. This extracted data is then dispatched to the Adaptive Learning engine for further analysis. Traffic is filtered based on a predefined sampling rate, and URLs with specific suffixes (e.g., .jpg, .ico, .mp3, .css) are excluded from the analysis.



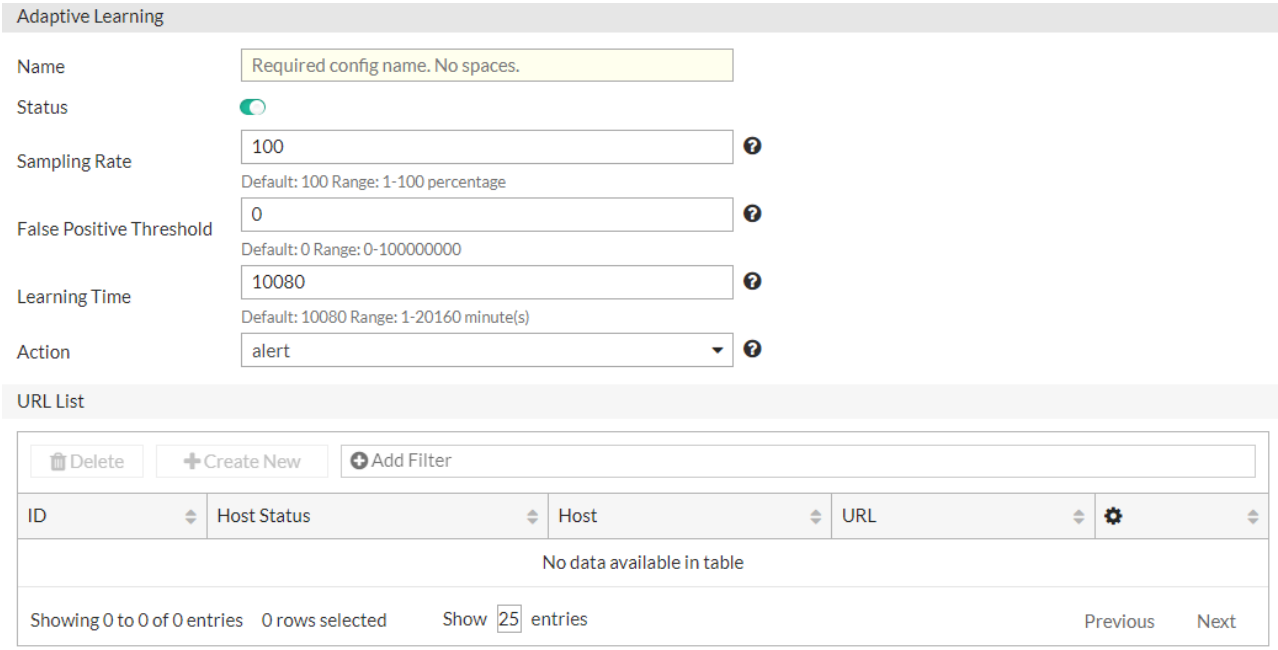
The Adaptive Learning engine performs continuous analysis on the filtered traffic received from httproxy, storing statistical information in its memory and periodically flushing it to local database files. If the learning results demonstrate sufficient stability within the defined Learning Time, the Adaptive Learning engine can generate and propose specific recommendations.

Adaptive Learning policies are configured per VDOM, and will take effect only after it is applied to a WAF Profile.

To configure an Adaptive Learning policy:

1. Go to **Web Application Firewall > Adaptive Learning**.
2. Click **Create New** to display the configuration editor.

3. Enable **Status** to view the Adaptive Learning configuration parameters.



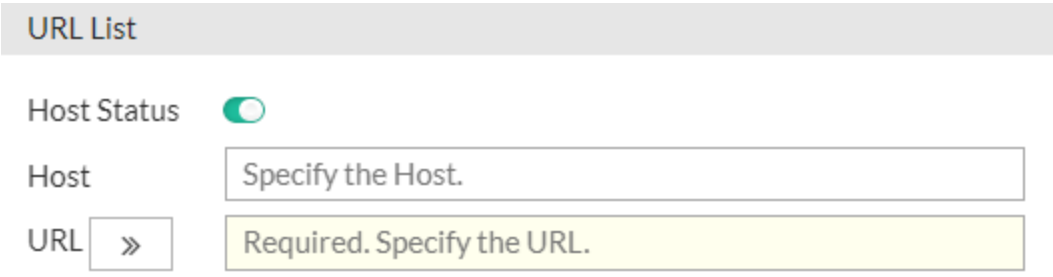
4. Configure the following Adaptive Learning policy settings:

Setting	Description
Name	Specify a unique name for the Adaptive Learning configuration object. Valid characters are `A-Z`, `a-z`, `0-9`, `_`, and `-`. No space is allowed. Once saved, the name of the Adaptive Learning configuration cannot be changed.
Sampling Rate	Specify the percentage of received requests and their responses that will be sampled. For example, if the sampling rate is 50%, then for every 100 requests, the first 50 requests will be sampled. The default is 100, and the acceptable range is 1-100.

False Positive Threshold	Specify the threshold at which triggered events should be considered a false positive.
	In scenarios when requests that trigger a WAF policy violation are received from multiple different sources within a certain time period, the False Positive Threshold can be set to allow the Adaptive Learning engine to identify these triggered events as false positives and recommend adjustments to the WAF policy.
	The default is 0, and the acceptable range is 0-100000000.
	For example:
	`False Positive Threshold - 2`
	`Learning Time - 10`
	`WAF policy - WAF Signature Profile`

Setting	Description
	When requests trigger a specific WAF signature ID violation are received from 2 different clients within the 10 minute Learning Time, then Adaptive Learning will generate a recommendation to disable the specific signature ID avoid triggering false positive results.
Learning Time	Specify the Learning Time period in minutes. The default is 10080 minutes, and the acceptable range is 1-20160 minute(s). Adaptive Learning will only generate recommendations if the analysis (or "learning") results are "stable" within the specified time period. For the learning results to be stable, the Adaptive Learning engine must not detect any drastic flux in request rates, parameter lengths or types, or longer JSON/XML element names of values, among other configurable limit checks that are configurable in the policies.
Action	Set the action to take after you accept the recommendation for the WAF policy from Adaptive Learning. • alert — WAF policies will allow the traffic to pass and log the event. • deny — WAF policies will the drop current attack session by HTTP 403 message, and log the event. • block — WAF policies will drop the current attack session by HTTP 403 message and block the attacker (according the attacker's IP address) for 1 hour, and log the event. • silent-deny — WAF policies will drop the current attack session by HTTP 403 message, without logging the event. • captcha — WAF policies will allow the traffic to pass if the client successfully fulfills the CAPTCHA request, and log the event. The default action is alert.

5. Click **Save** to save the Adaptive Learning settings.
   Once the configuration is saved, the **URL List** becomes configurable. The Adaptive Learning policy will be applied to the request URLs in the URL List.
6. Under the **URL List** section, click **Create New** to display the configuration editor.



7. Configure the following URL List settings:

Setting	Description
Host Status	If enabled, require authorization only for the specified host. If disabled, ignore hostname in the HTTP request header and require authorization for requests

Setting	Description
	with any Host header. Disabled by default.
Host	The **Host** option is available if **Host Status** is enabled. Specify the HTTP Host header. If Host Status is enabled, the policy matches only if the Host header matches this value. Complete, exact matching is required. For example, `www.example.com` matches `www.example.com` but not `www.example.com.hk`.
Request URL	The literal URL, such as `/index.php`, or a regular expression, such as `^/*.php` that the HTTP request must contain in order to match the rule. Multiple URLs are supported.

8. Click **Save**.
   Once the URL List configuration is saved, you are returned to the Adaptive Learning configuration editor.
9. Click **Save** again to apply the newly created URL List configuration to the Adaptive Learning configuration.

Once the Adaptive Learning policy is saved, you can apply it to the WAF profile.

Predefined Adaptive Learning policy configurations:

FortiADC offers three predefined Adaptive Learning policies you can apply directly in the WAF Profile or you can clone to use as a template to define your own policy. Please note that these predefined configurations are read-only and cannot be modified directly.

Predefined policy	Parameter	Setting
Fast_Learning	Sampling Rate	100
	False Positive Threshold	0
	Learning Time	1440
	Action	alert
	URL List	
	Host Status	disable
	URL	/
Medium_Learning	Sampling Rate	80
	False Positive Threshold	0
	Learning Time	10080
	Action	alert
	URL List	
	Host Status	disable
	URL	/

Predefined policy	Parameter	Setting
Slow_Learning	Sampling Rate	50
	False Positive Threshold	0
	Learning Time	20160
	Action	alert
	URL List	
	Host Status	disable
	URL	/

New CLI commands to support WAF Adaptive Learning:

```
config security waf adaptive-learning
   edit <name>
      set status {enable|disable}
      set sampling-rate <integer>
      set least-learning-time <integer>
      set false-positive-threshold <integer>
      set action <datasource>
      config  url-list
         edit <No.>
            set host-status {enable|disable}
            set host <string>
            set request-url <regex>
         next
      end
   next
end
```

```
config security waf profile
   edit <name>
      set adaptive-learning <datasource>
   next
end
```

Adaptive Learning analysis and recommendations

On the **Adaptive Learning View** page, you can access statistical outputs derived from the Adaptive Learning engine's continuous, deep analysis of comprehensive datasets constructed from incoming traffic samples. The Adaptive Learning statistics are organized into two tabs: Analysis on page 35 and Recommendation on page 41.

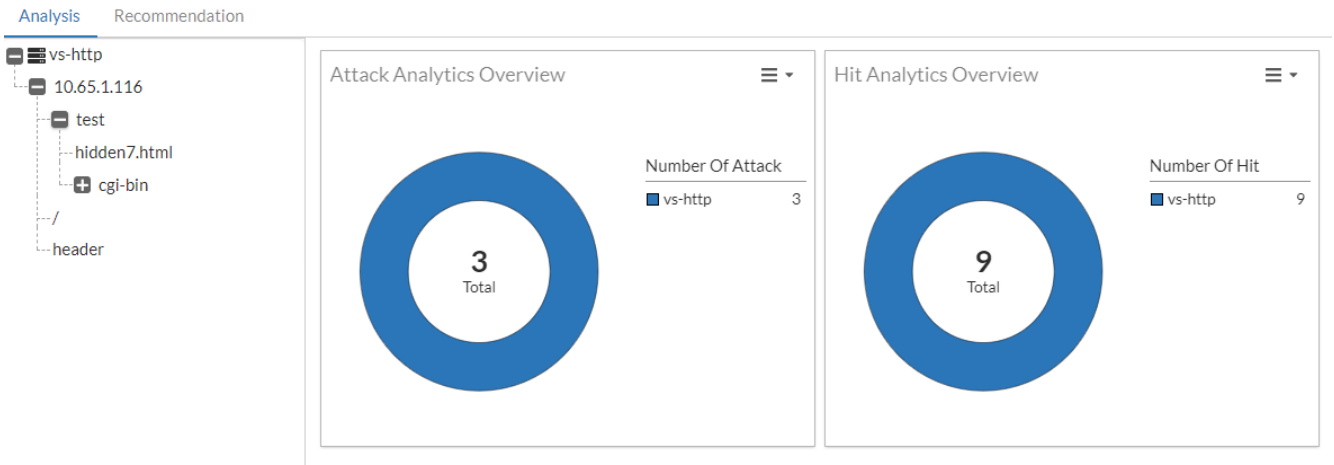
Analysis

The Analysis page provides graphical representations of the statistical outputs generated from the WAF Adaptive Learning engine. This page features a navigation tree that allows you to drill down to various levels of data. Each view presents a dashboard composed of various widgets that can be moved and resized. Statistics may be presented as tables or circle graphs that break-down the percentage against the total count.

Adaptive Learning statistics data are saved to the local database, ensuring persistence across reboots and upgrades. Additionally, if the interval between two requests received under the same virtual server exceeds one minute, the learned statistics will be written to the database file.

Overviews

The Overview dashboard serves as the landing page of the Analysis tab. Here, you can view the Attack Analytics Overview and Hit Analytics Overview graphs that provide a comprehensive overview of the statistics collected from all virtual servers with Adaptive Learning enabled.

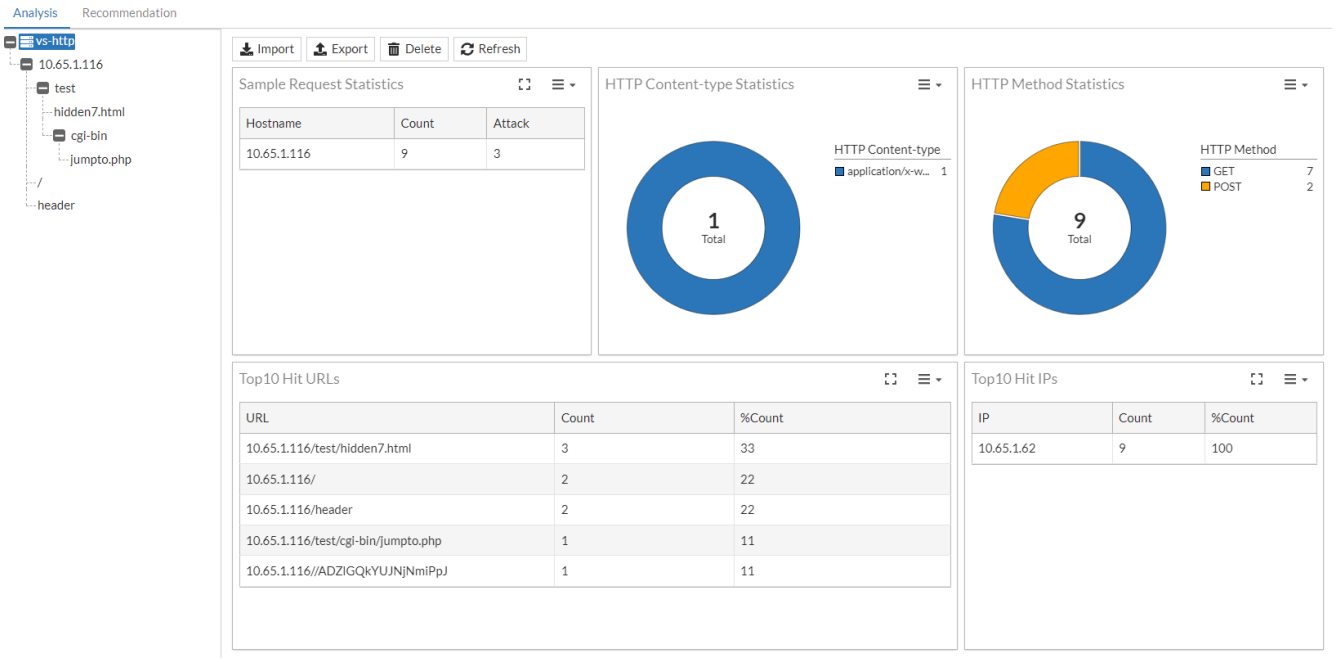


Virtual Server Statistics

At the virtual server level, various graphs and tables consolidate the Adaptive Learning analysis statistics from the virtual server traffic.

You can perform the following actions on the virtual server statistics:

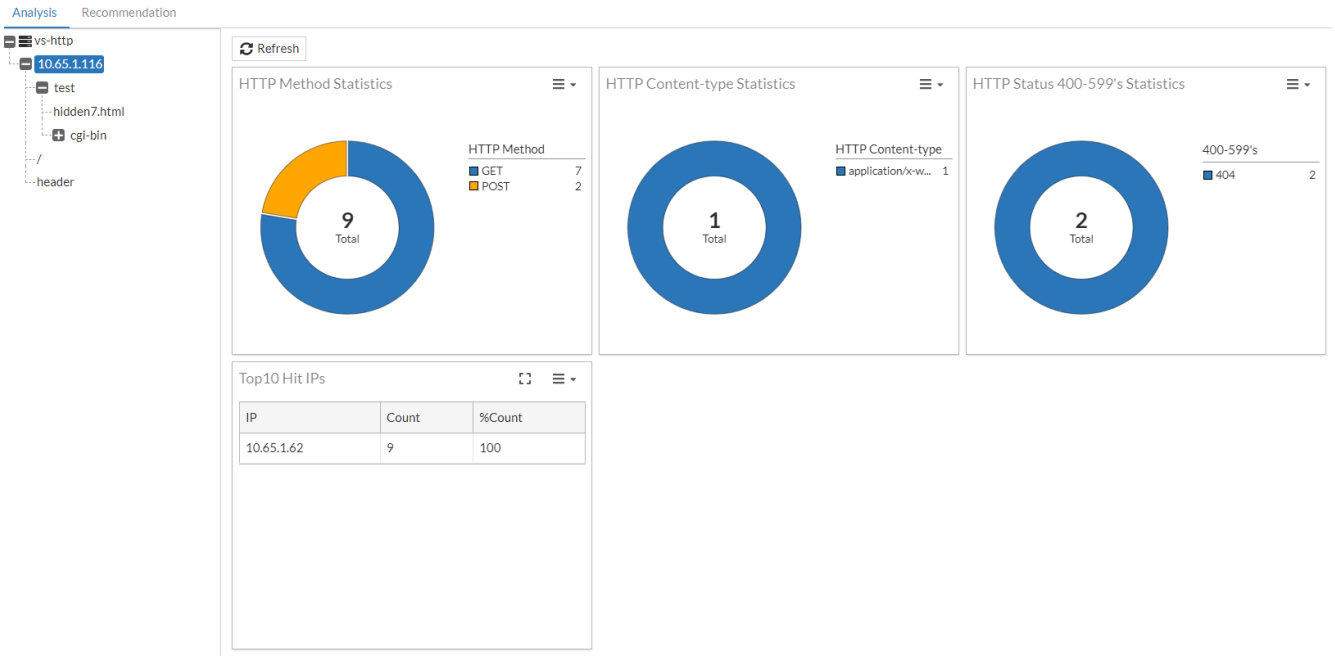
- Import — You may import a .zip file of statistics exported from another virtual server you wish to migrate.
- Export — The virtual server URL tree (.xml) and associated statistics (.bin) will be compressed into a .zip file and downloaded automatically. The .bin file statistics for each URL includes the parameters, hidden fields and other information that are used to determine if a recommendation should be generated.
- Delete — Both the URL trees and associated statistics will be cleared. All subsequent recommendations will be generated based on new traffic sampling.
- Refresh — Manually refresh the statistics for the current page.



Statistic	Description
Count	The number of requests sent to Adaptive Learning for processing.
Attack	The number of requests that are denied by WAF policies.
%Count	The percentage of the current requests relative to the total number of requests.

Host IP Statistics

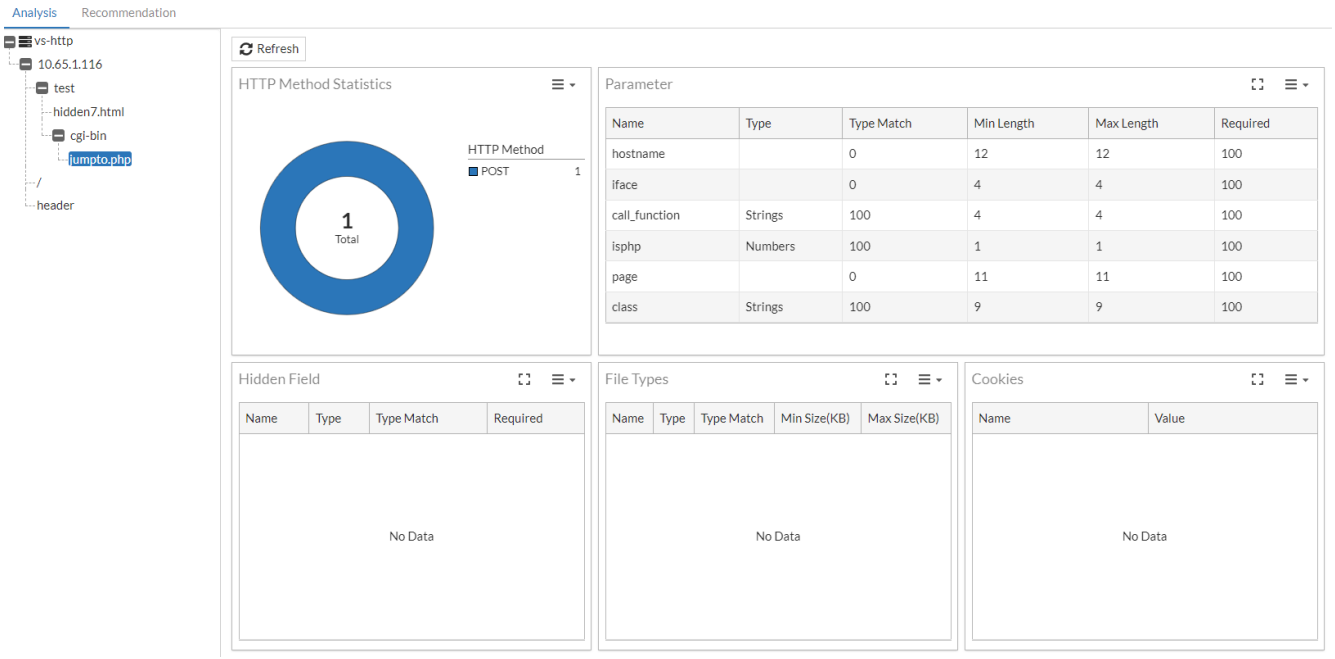
At the Host IP level, various graphs and tables consolidate the Adaptive Learning analysis statistics from each Host IP address.



Statistic	Description
Count	The number of requests sent to Adaptive Learning for processing.
%Count	The percentage of the current requests relative to the total number of requests.

URL Statistics

At the URL level, critical data about the URL is presented in a graph and several tables.



HTTP Method Statistics

The HTTP Method Statistics graph breaks down the types of methods that can be analyzed from this URL.

Parameter

The Parameter table captures the statistics used to generate the parameter validation policies.

Note: As HTML content is usually compressed in transferring, for most of the time, decompression should be enabled on the VS to correctly parse the HTML content.

Column	Description
Name	Parameters are parsed from the `<input>` fields within `<form>` elements or from query parameters in a URL in HTML. Parameters extracted from HTML forms are processed by the Adaptive Learning engine and used to generate recommendations. Parameters initially learned from query requests are displayed in the statistics table but do not generate recommendations. However, these parameters can be utilized to adjust those already learned from HTML forms.
Type	The type is parsed by the WAF's built-in DLP sensitive data-type database based on parameter values from the latest POST request, rather than the input type defined in the HTML source file. The input type parsed from the HTML form determines if a parameter should be learned or ignored, while the type displayed in the statistics table is parsed by the DLP database. • Parameter types ignored by Adaptive Learning: range, color, checkbox, radio, submit, reset, button, image. • Supported data types: Both predefined and customized data types are supported. If the type cannot be recognized by the Sensitive Data-Type (SDT) database, it will be shown as empty. Though the type is parsed from the POST request, a prior GET request is necessary to identify the parameter; otherwise, the parameter may not be recognized correctly.
Type Match	The percentage of the current/latest type among all received requests is calculated. Whenever the type changes, the count for the previous type is reset to 0.
Min Length	The minimum length of the value ever received for the current parameter.
Max Length	The maximum length of the value ever received for the current parameter.
Required	The percentage of times the parameter has a non-empty value out of the total occurrences. For example, if the parameter "fruit" is received 10 times and 3 times the value is empty, then the required percentage is 70%. This means that 70% of the times the parameter "fruit" is received, it has a non-empty value.

Hidden Field

The Hidden Field table captures the statistics used to generate recommendation for hidden field validation policies.

Column	Description
Name	Parameters with an input type "hidden" in the HTML form will be recognized as Hidden Field. Parameters are parsed from the `<input>` fields within `<form>` elements or from query parameters in a URL in HTML. Parameters extracted from HTML forms are processed by the Adaptive Learning engine and used to generate recommendations. Parameters initially learned from query requests are displayed in the statistics table but do not generate recommendations. However, these parameters can be utilized to adjust those already learned from HTML forms.
Type	The type is parsed by the WAF's built-in DLP sensitive data-type database based on parameter values from the latest POST request, rather than the input type defined in the HTML source file. The input type parsed from the HTML form determines if a parameter should be learned or ignored, while the type displayed in the statistics table is parsed by the DLP database. • Parameter types ignored by Adaptive Learning: range, color, checkbox, radio, submit, reset, button, image. • Supported data types: Both predefined and customized data types are supported. If the type cannot be recognized by the Sensitive Data-Type (SDT) database, it will be shown as empty. Though the type is parsed from the POST request, a prior GET request is necessary to identify the parameter; otherwise, the parameter may not be recognized correctly.
Type Match	The percentage of the current/latest type among all received requests is calculated. Whenever the type changes, the count for the previous type is reset to 0.
Required	The percentage of times the parameter has a non-empty value out of the total occurrences. For example, if the parameter "fruit" is received 10 times and 3 times the value is empty, then the required percentage is 70%. This means that 70% of the times the parameter "fruit" is received, it has a non-empty value.

File Types

The File Types table captures the statistics of the files uploaded through the requests.

Note: Currently, these statistics are not actively used to generate recommendations, however, these File Type statistics may be used to generate File Restriction policies in the future.

Column	Description
Name	Parsed from the multipart or form-data parts of the file.
Type	The type of the uploaded file is parsed by the WAF's built-in file-type database, which is used for file restriction policies. This is independent of the type defined in the multipart/form-data headers.
Type Match	The percentage of the current/latest type among all received requests is calculated. Whenever the type changes, the count for the previous type is reset to

Column	Description
	0.
Min Size (KB)	The minimum size ever uploaded of the same file name.
Max Size (KB)	The maximum size ever uploaded of the same file name.

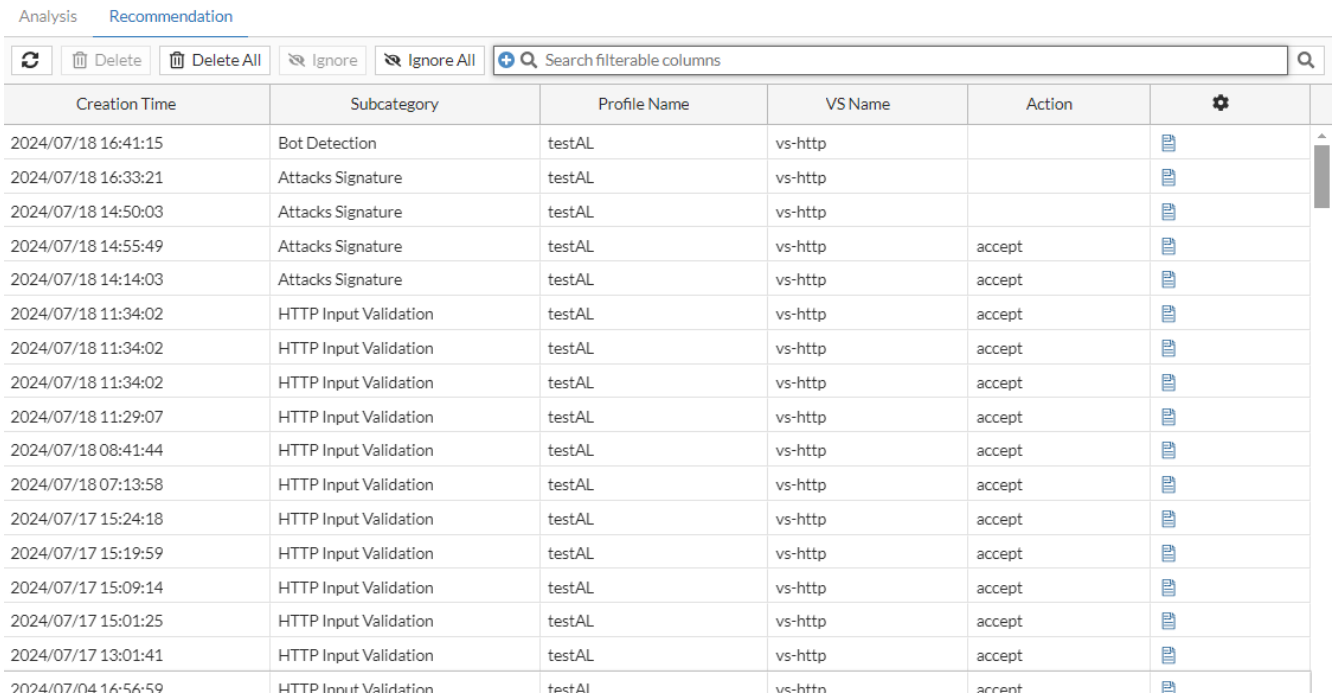
Cookies

The Cookies table captures the statistics of cookies through the requests. This statistic is not used to generate recommendations.

Column	Description
Name	Parsed from the cookie pairs from request the request header "Cookie: **name**=value".
Value	Parsed from the cookie pairs from request the request header "Cookie: name=**value**".

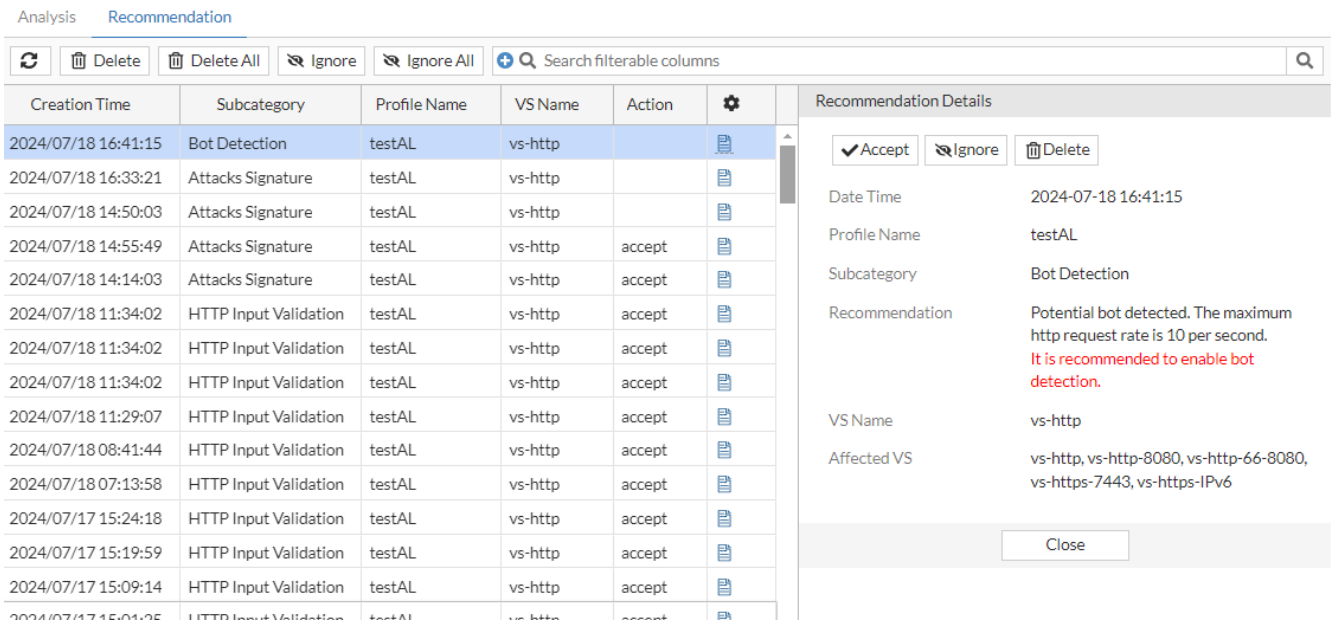
Recommendation

From the Recommendation page, you can view all the recommendations generated from the deep analysis performed by the Adaptive Learning engine.



Creation Time	Subcategory	Profile Name	VS Name	Action
2024/07/18 16:41:15	Bot Detection	testAL	vs-http	
2024/07/18 16:33:21	Attacks Signature	testAL	vs-http	
2024/07/18 14:50:03	Attacks Signature	testAL	vs-http	
2024/07/18 14:55:49	Attacks Signature	testAL	vs-http	accept
2024/07/18 14:14:03	Attacks Signature	testAL	vs-http	accept
2024/07/18 11:34:02	HTTP Input Validation	testAL	vs-http	accept
2024/07/18 11:34:02	HTTP Input Validation	testAL	vs-http	accept
2024/07/18 11:34:02	HTTP Input Validation	testAL	vs-http	accept
2024/07/18 11:29:07	HTTP Input Validation	testAL	vs-http	accept
2024/07/18 08:41:44	HTTP Input Validation	testAL	vs-http	accept
2024/07/18 07:13:58	HTTP Input Validation	testAL	vs-http	accept
2024/07/17 15:24:18	HTTP Input Validation	testAL	vs-http	accept
2024/07/17 15:19:59	HTTP Input Validation	testAL	vs-http	accept
2024/07/17 15:09:14	HTTP Input Validation	testAL	vs-http	accept
2024/07/17 15:01:25	HTTP Input Validation	testAL	vs-http	accept
2024/07/17 13:01:41	HTTP Input Validation	testAL	vs-http	accept
2024/07/04 16:56:59	HTTP Input Validation	testAL	vs-http	accept

To view the Recommendation Details, select the entry and click (page icon) to display the **Recommendation Details** dialog.



Date Time	Records the time when the recommendation was generated.
Profile Name	The WAF Profile name.
Subcategory	The WAF module to which the recommendation applies.
Recommendation	The rationale behind the recommendation and the specific action it advises.
VS Name	The name of the virtual server where the WAF Profile is applied.
Affected VS	Lists the virtual servers that will be affected by the action taken for this recommendation.

There are three actions you can take to address this recommendation:

- Accept — The corresponding policy will be automatically generated or adjusted. If the current policy is read-only (a predefined policy), it will be replaced with a cloned version and adjusted with the recommended settings.
- Ignore — This recommendation will be ignored, but will still be kept for reference. The corresponding notification will be considered "read", decreasing the notification count.
- Delete — This recommendation will be deleted and removed from the recommendation list. The corresponding notification will be considered "read", decreasing the notification count.

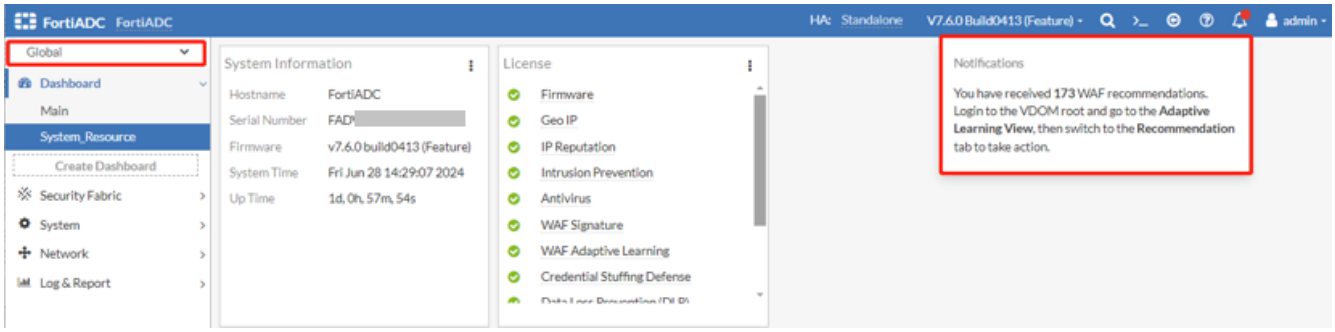
For practical demonstrations of how to use Adaptive Learning recommendations, refer to the following examples:

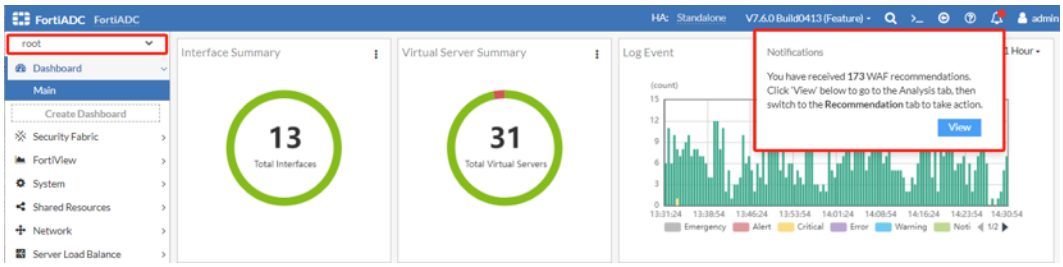
- Examples of Adaptive Learning applied to Web Attack Signature on page 43
- Examples of Adaptive Learning applied to Bot Detection on page 44
- Example of Adaptive Learning applied to HTML Parameter Validation on page 46

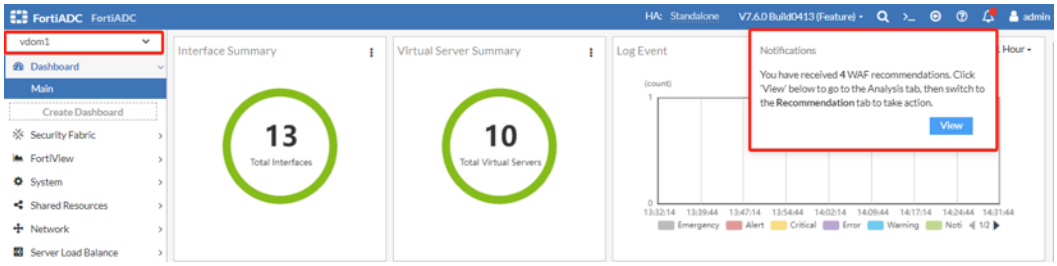
Recommendation Notifications

Adaptive Learning generates notifications for each unresolved recommendation. The displayed recommendation count differs based on the context in which the notification is viewed, either from the Global VDOM or specific VDOMs (root or non-root). In the Global VDOM, the recommendation count aggregates only those generated within the root VDOM.

Conversely, when viewed from individual VDOMs, the count reflects recommendations generated exclusively within the respective VDOM.







Examples of Adaptive Learning applied to Web Attack Signature

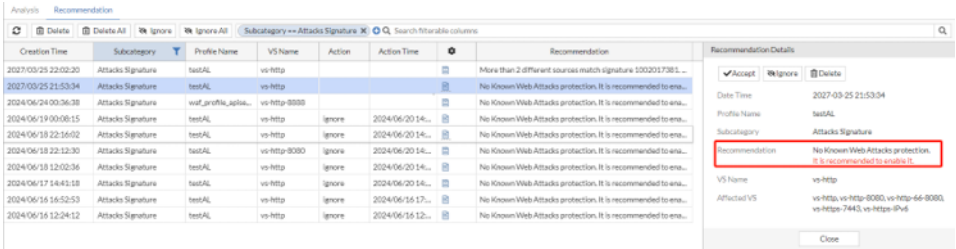
Triggering recommendations to set a new profile

If no WAF signature profile is configured in the WAF Profile, Adaptive Learning will recommend setting one upon receiving a request. Upon accepting the recommendation, a new WAF signature profile will be automatically created and attached to the WAF Profile.

This new WAF signature profile recommended by Adaptive Learning enables the following signatures:

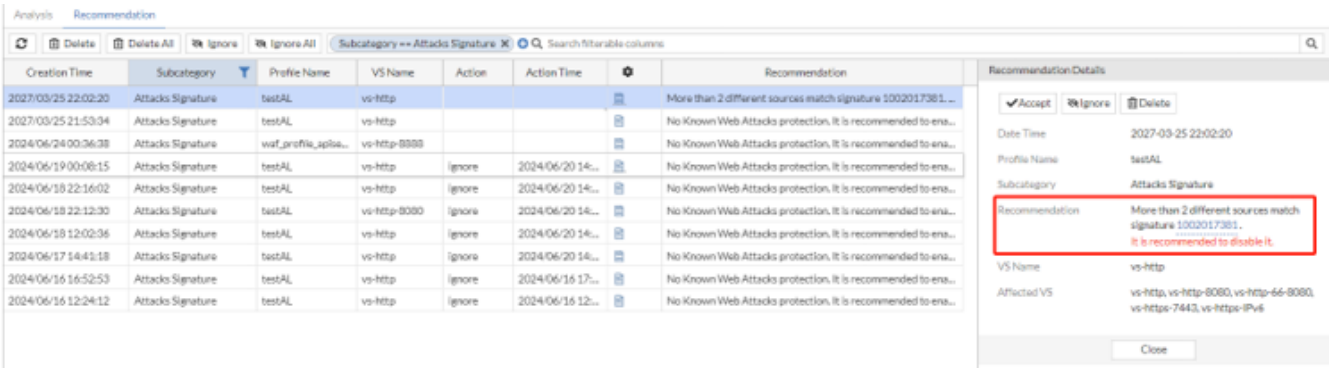
- Bad Robot
- Credit Card Detection
- Cross Site Scripting
- Generic Attacks
- Information Disclosure
- Known Exploits
- SQL Injection
- Trojans

The WAF action will be the same as the one set in the Adaptive Learning policy.



Triggering a False Positive recommendation to disable a signature ID

If requests triggering a specific signature ID violation exceed the client count set in the False Positive Threshold within the defined Learning Time, Adaptive Learning will recommend disabling that signature ID. Upon accepting the recommendation, the specific signature ID will be disabled in the WAF signature profile. If the signature profile in use is a predefined configuration, Adaptive Learning will automatically replace it with a cloned version where the affected signature ID is disabled.

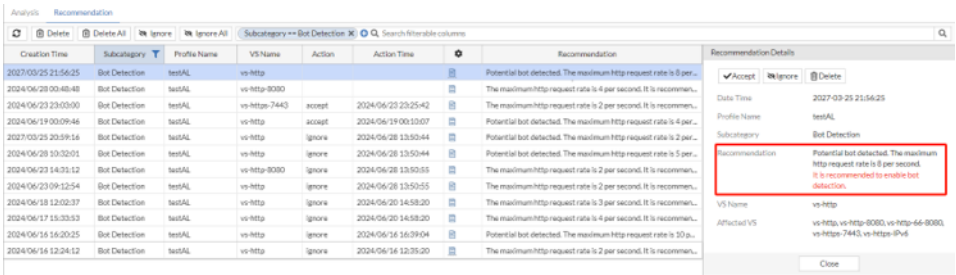


Examples of Adaptive Learning applied to Bot Detection

Triggering recommendations to set a new policy

If no Bot Detection policy is configured in the WAF profile, the system injects JavaScript into the HTML file before delivering it to the client. The system then monitors for a subsequent request indicating that the JavaScript has been executed, verifying the client's support for JavaScript. If the client fails to respond, it is flagged as a potential bot, and a recommendation to configure a Bot Detection Policy is generated.

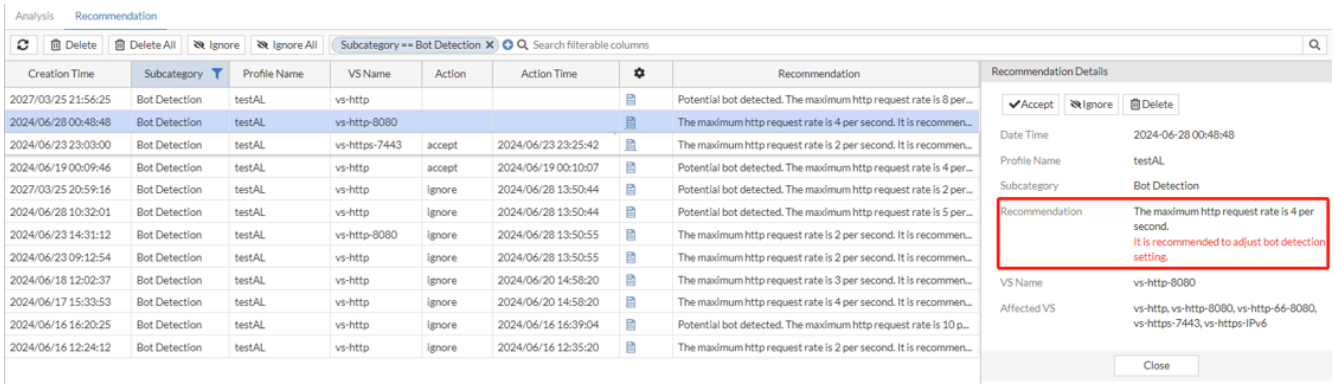
Upon accepting the recommendation, a Bot Detection policy will be created and attached to the WAF profile. This policy will enable detection for all Malicious Bots and Known Good Bots. The HTTP request rate will be set to match the current traffic sampling rate in the Adaptive Learning policy.



Triggering recommendations to adjust the request rate

When a Bot Detection policy is enabled, if the maximum HTTP request rate differs from the configured policy rate, Adaptive Learning will recommend to adjust the request rate.

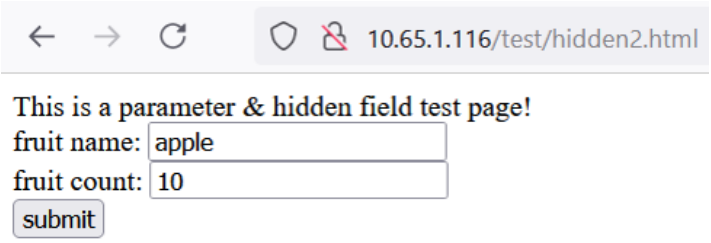
Upon accepting the recommendation, the HTTP request rate in the Bot Detection policy will be updated.



Triggering a False Positive recommendation to disable a malicious bot

When the Bad Robot Status option is enabled in the Bot Detection policy, if requests that trigger the bad bot detection come from more source IPs than the number specified in the False Positive Threshold, Adaptive Learning will recommend to disable the Bad Robot Status.

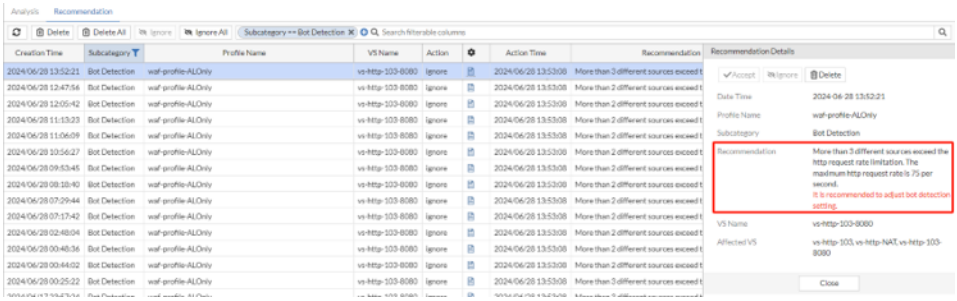
Upon accepting the recommendation, the Bad Robot Status in the Bot Detection policy will be disabled.



Triggering a False Positive recommendation to adjust the request rate

When the HTTP request rate is set, if the maximum request rate from more source IPs than the number specified in the False Positive Threshold exceeds the configured request rate within the defined Learning Time, Adaptive Learning will recommend to adjust the request rate.

Upon accepting the recommendation, the HTTP request rate in the Bot Detection policy will be updated.



Example of Adaptive Learning applied to HTML Parameter Validation

Triggering recommendations to set a new policy

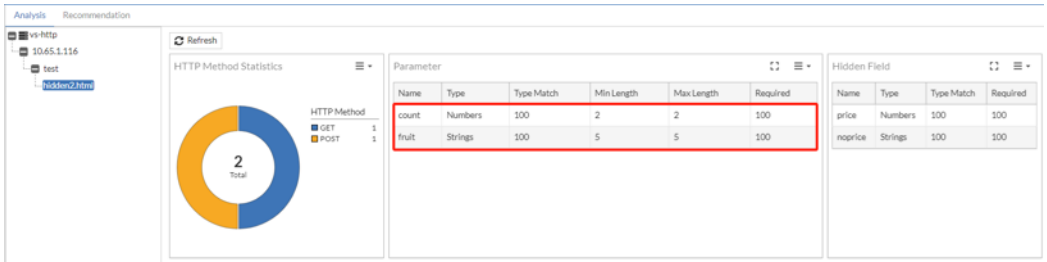
Adaptive Learning will recommend to set a Parameter Validation policy if a parameter is learned while there is no policy is set. The following conditions are considered:

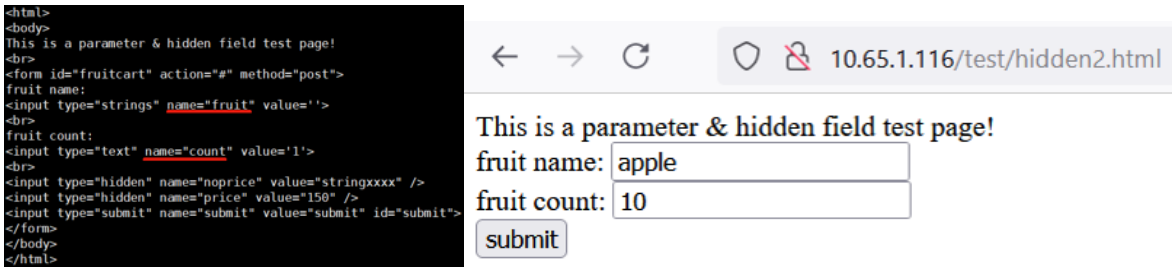
- No Input Validation policy is set in the WAF Profile.
- No Parameter Validation policy is included in the Input Validation policy in use.
- No Parameter Validation rule is set in the Parameter Validation policy in use.

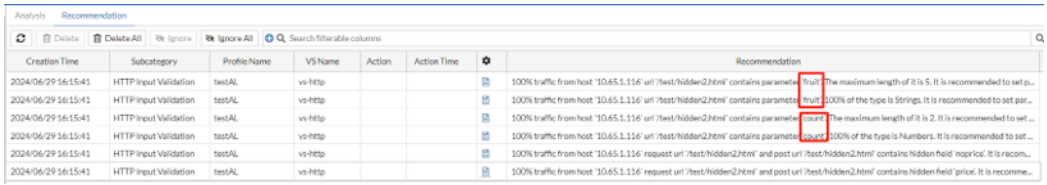
Typically, two recommendations will be generated for one parameter: one to set the Max Length, and another one to set the Data Type.

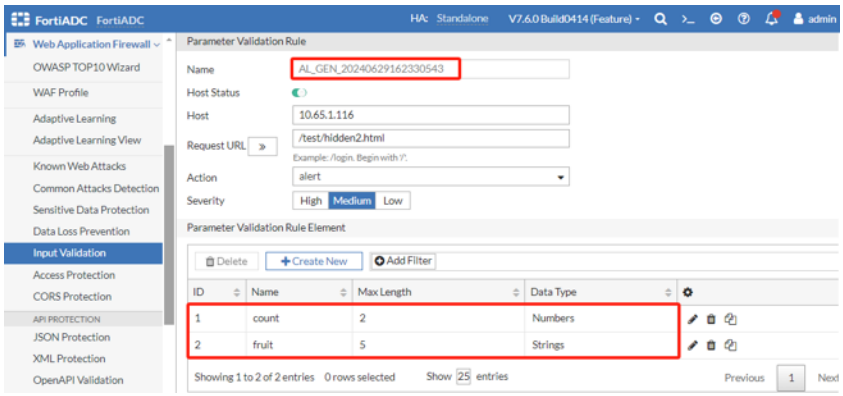
Upon accepting the recommendations, an Input Validation policy, a Parameter Validation policy or rule for the learned parameters will be created and attached to the WAF Profile. The policies will include the following specifications:

- The Request URL is set as the GET URL.
- The Max Length and Data Type are set according to the learned parameter.









Adaptive Learning troubleshooting and debugging

The following tools are available to troubleshoot and debug Adaptive Learning issues.

CLI commands to view debug logs relating to Adaptive Learning

Command	Guidelines
`diagnose debug module autolearn all | error | info | conf | event | stat | show` `diagnose debug enable`	To view the debug information for Adaptive Learning statistics collection, processing and recommendation generation.
`diagnose debug module waf adaptive-learning`	To view the debug information for traffic processed by httproxy, before delivering to Adaptive Learning.

Be aware of several limitations in the Adaptive Learning feature implementation in version 7.6.0:

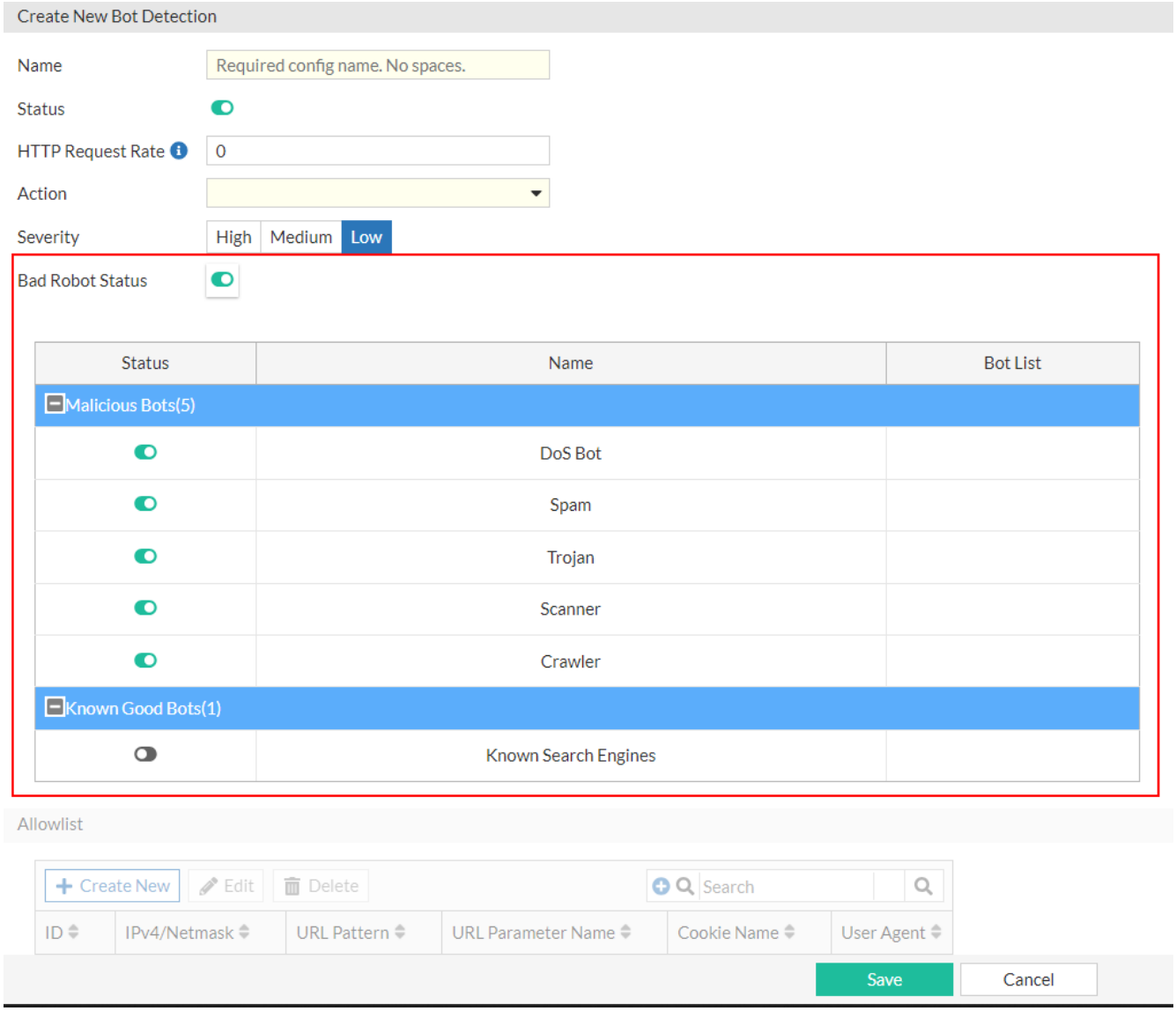
- Adaptive Learning statistics are not synchronized to HA peer nodes.
- Exceptions is currently not well supported — For example, when the incoming request matches the exception configured in the JSON/XML Protection policy or the allowlist of a Bot Detection policy, Adaptive Learning will still generate recommendations. However, if an Exception is configured in the WAF Profile, any traffic matched will be ignored by all WAF processing, including Adaptive Learning.
- For HTML Parameters or Hidden Field learning and recommendations, a GET request must be sent before sending a POST request; otherwise, the parameters may not be learned correctly. For the same virtual server, a specific parameter or hidden field will only support one URL, otherwise the rules sorted lower in the policy will not be updated correctly after accepting the recommendation.

Bot Detection enhancement

FortiADC has enhanced its Bot Detection functionality to enable more granular classification of Malicious Bots, including detailed categorization. Within the Bot Detection policy, you can configure the enable/disable status of each bot within the Malicious Bot category, allowing precise inclusion or exclusion for detection. Additionally, the Bot Detection policy configuration has improved Known Good Bots options, providing greater control over the inclusion or exclusion of specific bots.

This information is also available in the FortiADC 7.6.0 Administration Guide and CLI Reference Guide:

- Configuring a Bot Detection policy
- config security waf bot-detection



Bot Detection policies employ signature analysis and behavioral tracking to identify client traffic likely generated by automated bots rather than genuine human users. Legitimate bots, such as search engine crawlers, are classified as "good bots" because they perform essential search indexing operations, which can increase the visibility of your site to legitimate users.

Conversely, "bad bots" are known to generate malicious traffic that can compromise site availability and integrity. Examples of such activities include Distributed Denial of Service (DDoS) attacks and content scraping. To mitigate these threats, it is crucial to deploy effective bot mitigation strategies, such as IP reputation analysis, rate limiting, anomaly detection algorithms, and CAPTCHA challenges, to identify and block these harmful bots in real-time.

The FortiADC Bot Detection policy allows you to monitor and protect against suspected bot traffic using classifications from the Web Application Firewall (WAF) Signature database. Bots are categorized into five types of Malicious Bots and one type of Known Good Bots. You can customize the detection settings by selecting specific bots within these categories to include or exclude from monitoring and mitigation actions.

Edit Bot Detection

Name: test

Status

HTTP Request Rate: 0

Action: alert

Severity: High | Medium | Low

Bad Robot Status

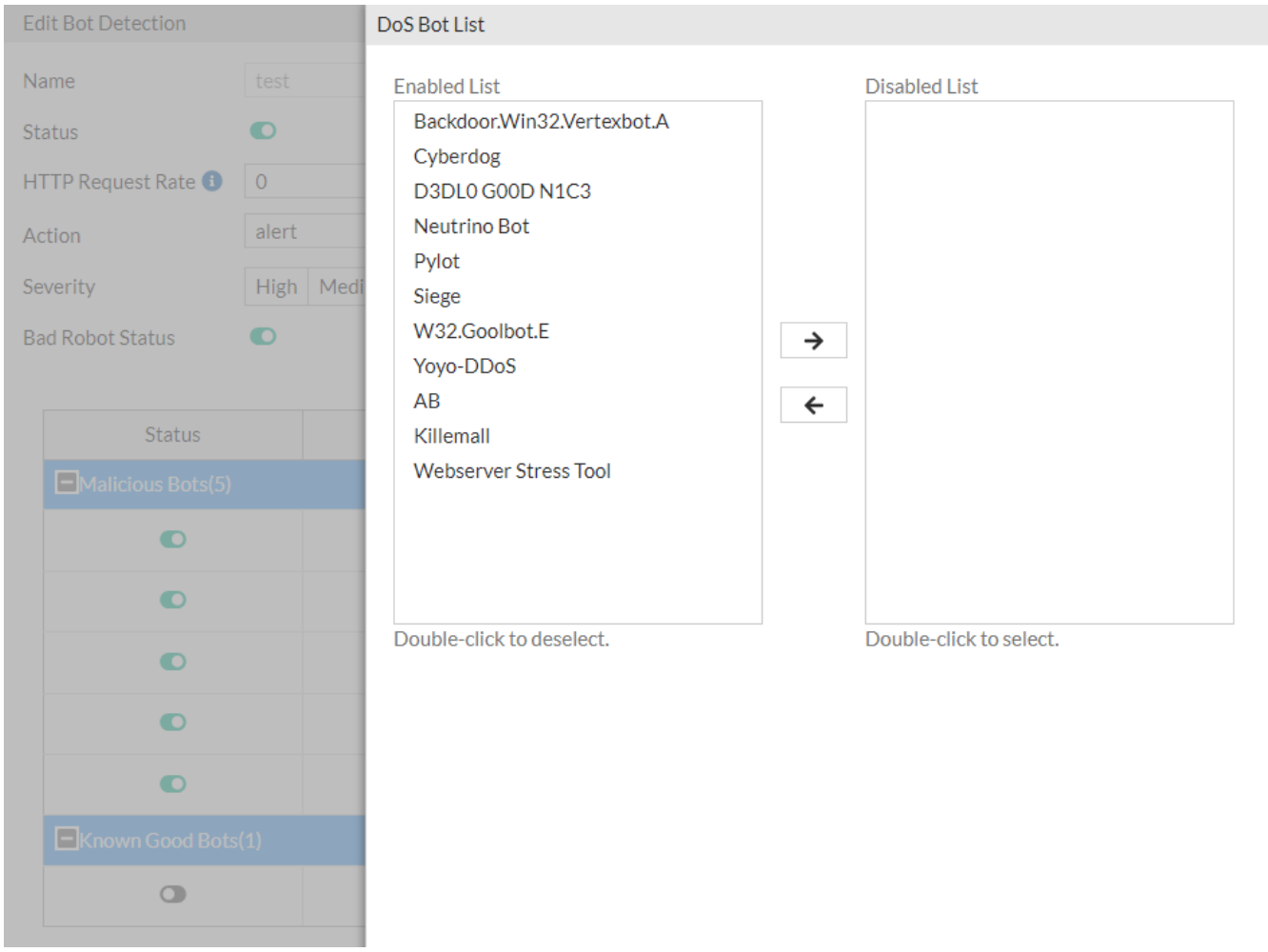
Status	Name	Bot List
Malicious Bots(5)		
	DoS Bot	
	Spam	
	Trojan	
	Scanner	
	Crawler	
Known Good Bots(1)		
	Known Search Engines	

Allowlist

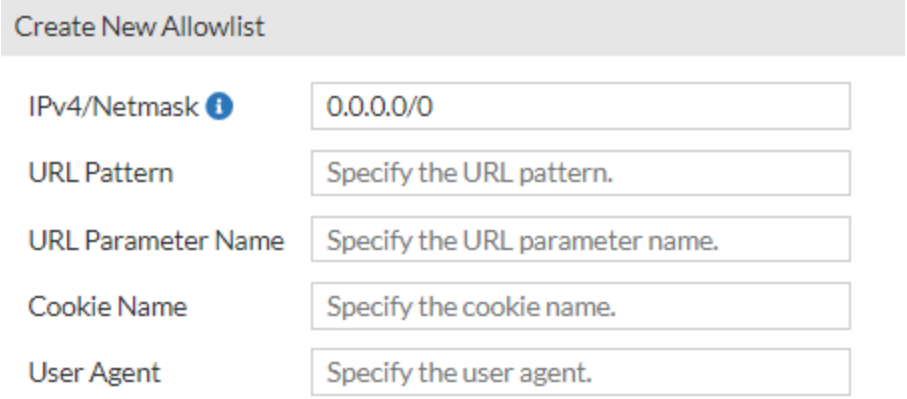
Create New | Edit | Delete | Search

ID	IPv4/Netmask	URL Pattern	URL Parameter Name	Cookie Name	User Agent

From the Bot List column, you can click the ☰ (collapsed menu icon) to enable or disable specific bots for detection.



For more granular control, you can configure an allowlist to specify IP addresses that should be exempt from bot detection, even if they match bot signatures. For instance, if a user utilizes a scanning tool such as DirBuster, which generates HTTP requests that would typically be classified as Malicious Bot activity, adding the user's source IP address to the allowlist ensures that this traffic is not flagged or blocked by the bot detection system. This allows for legitimate testing and scanning activities to proceed without interference from bot mitigation measures.



After you have configured Bot Detection policies, you can select them in the WAF Profile. Once you have attached the WAF Profile with Bot Detection to a virtual server, you can monitor the real-time data analytics from FortiView.

Before you begin:

- You must configure the connection to FortiGuard so the system can receive periodic WAF Signature Database updates, including Known Good Bot and Malicious Bot signatures and lists. See Configuring FortiGuard service settings.
- You must have Read-Write permission for Security settings.

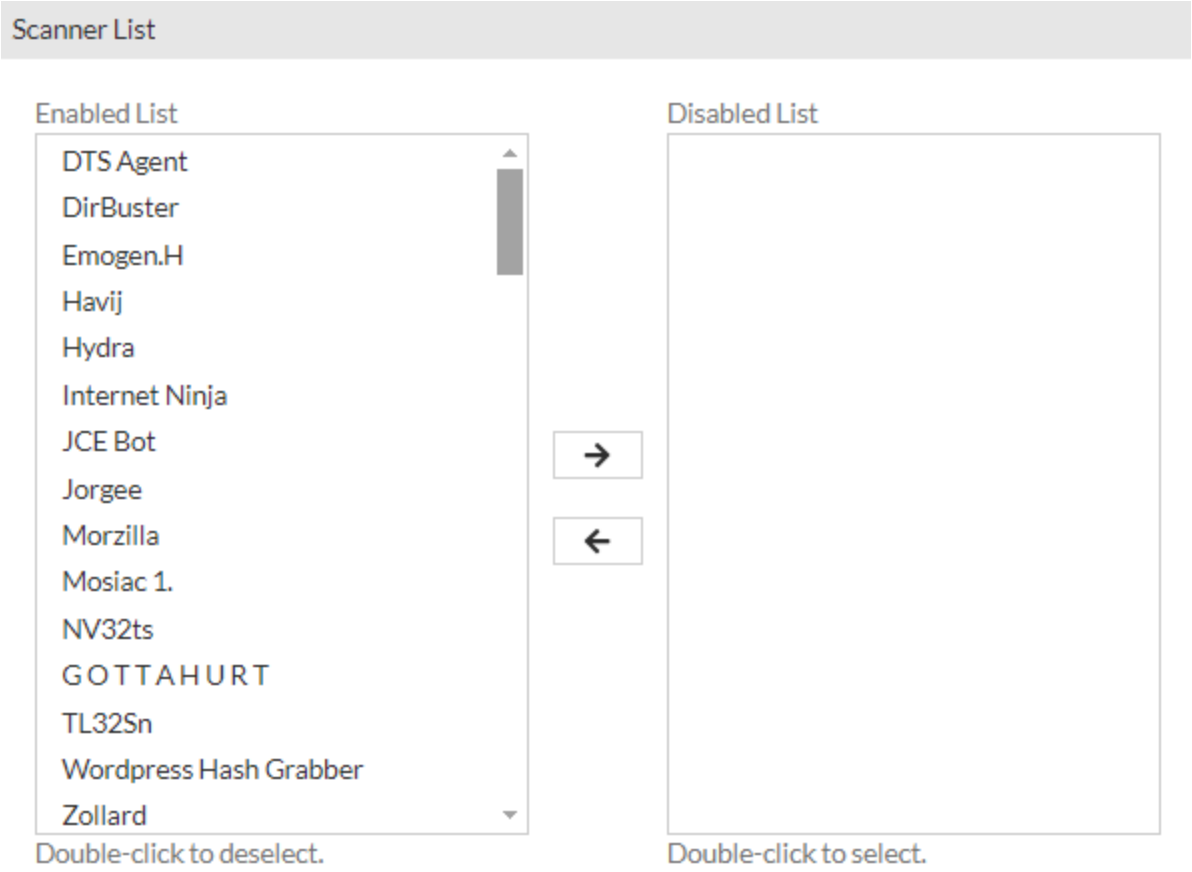
To configure a Bot Detection policy:

1. Go to **Web Application Firewall > Bot Detection**.
2. Click **Create New** to display the configuration editor.
3. Enable the **Status** to view the Bot Detection configuration options.
4. Once the Status is enabled, configure the following settings:

Setting	Description
Name	Configuration name. Valid characters are `A-Z`, `a-z`, `0-9`, `_`, and `-`. No spaces. After you initially save the configuration, you cannot edit the name.
HTTP Request Rate	Specify a threshold (HTTP requests/second/source) to trigger the action. Bots send HTTP request traffic at extraordinarily high rates. The source is tracked by source IP address and User-Agent. The default is 0 (off). The valid range is 0-100,000,000 requests per second.
Action	Select the action profile that you want to apply. See Configuring WAF Action objects. The default is alert.
Severity	• High—Log as high severity events. • Medium—Log as a medium severity events. • Low—Log as low severity events. The default is Low.
Bad Robot Status	Enable or disable detection for Malicious Bots. When enabled, a Malicious Bots blocklist is generated from WAF Signature updates from FortiGuard.
Malicious Bots	The **Malicious Bots** statuses are available if **Bad Robot Status** is **enabled**. You can enable or disable any of the five Malicious Bots categories: • DoS Bot • Spam • Trojan • Scanner • Crawler All Malicious Bots categories are enabled by default. **Note**: The Bot List can be configured once the initial Bot Detection policy configuration is saved.

Setting	Description
Known Good Bots	Enable or disable detection for Known Good Bots. When enabled, a Known Good Bots allowlist is generated from WAF Signature updates from FortiGuard. **Note**: The Bot List can be configured once the initial Bot Detection policy configuration is saved.

5. Click **Save**.
   Once the Bot Detection policy is saved initially, the Bot List for each enabled bot categories and the Allowlist becomes configurable.
6. Optionally, you can configure Bot Lists to include or exclude specific bots for detection.
   a. Under the Bot List column of each bot category, click the ☰ (collapsed menu icon) to display the configuration editor.
   b. By default, all bots are enabled. You can select and move each item to be in the **Enabled List** or **Disabled List**.
   The example below shows the Bot List for the Scanner category of Malicious Bots.



   c. Click **Save** to commit the changes and exit the dialog.
7. Optionally, you can configure an allowlist to specify IP addresses that should be exempt from bot detection, even if they match bot signatures.

a. Under the Allowlist section, click **Create New** to display the configuration editor.
b. Configure the following Bot Detection Allowlist settings:

Setting	Description
IPv4/Netmask	Matching subnet (CIDR format: 0.0.0.0/0).
URL Pattern	Matching string. Regular expressions are supported.
URL Parameter Name	Matching string. Regular expressions are supported.
Cookie Name	Matching string. Regular expressions are supported.
User Agent	Matching string. Regular expressions are supported.

c. Click **Save** to commit the changes and exit the dialog.

8. Click **Save** to commit all changes to the Bot Detection policy.
Once the Bot Detection policy is saved, it will be listed on the Bot Detection page. You can now reference this Bot Detection policy in a WAF Profile.

CLI updates:

```
config security waf bot-detection
   edit <name>
      set status enable
      set bad-robot enable
      config rule
         edit <ID>
            set status {enable|disable}
      config category
         edit <ID>
            set status {enable|disable}
```

It is recommended to configure Malicious Bot detection settings from the GUI, as the CLI can only identify each category and bot list item as internal IDs.

Security Fabric

The FortiADC 7.6 release includes new features and enhancements in **Security Fabric**:

FortiGate Security Fabric Connector on page 56

The new FortiGate Security Fabric Setup connector is introduced to support the second phase of Security Fabric Integrations. With this integration, FortiADC can be configured to join a Security Fabric through the root FortiGate. Once the FortiADC joins the Fabric, from the FortiGate, you will be able to view the FortiADC on topology pages, and create a dashboard Fabric Device widget to view FortiADC data.

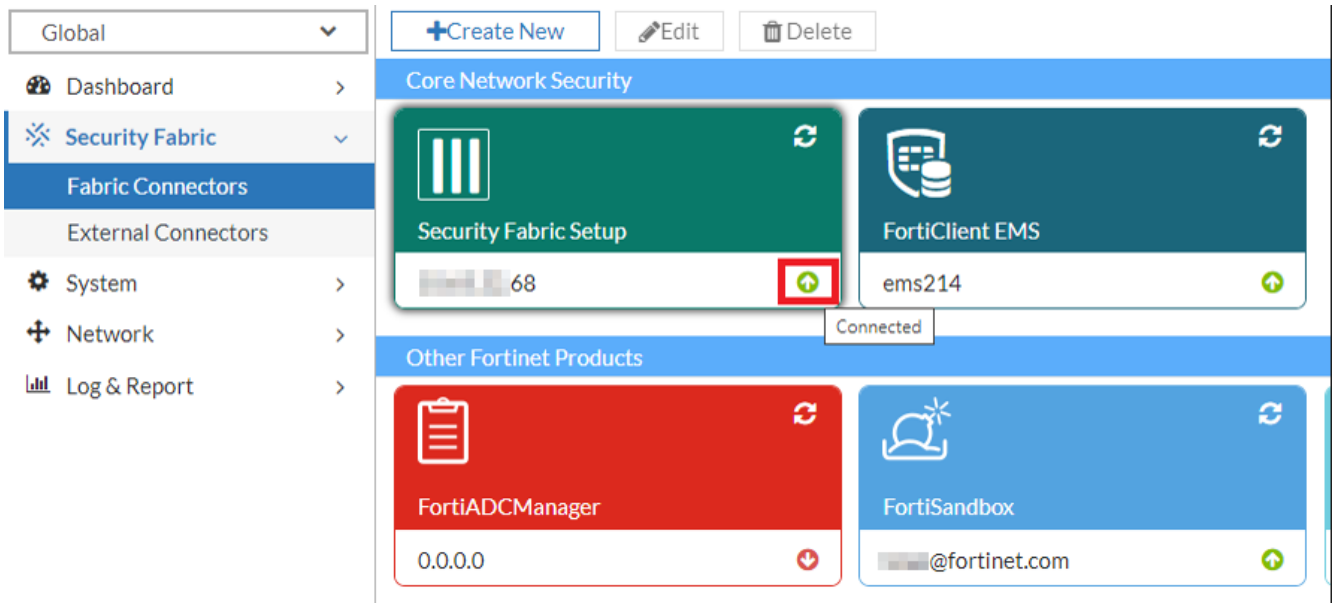
FortiGate Security Fabric Connector

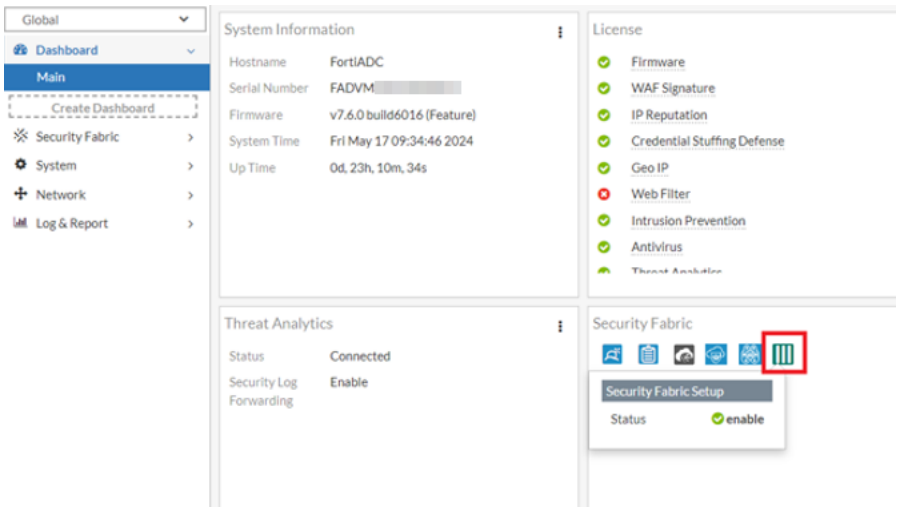
The FortiGate Security Fabric Setup connector facilitates the seamless integration of FortiADC into the FortiGate Security Fabric through the root FortiGate unit. Upon successful integration, the FortiADC device becomes a visible and interactive element within the Security Fabric topology views on the FortiGate interface. Furthermore, administrators can leverage the Fabric Device widget on the FortiGate dashboard to monitor FortiADC statuses, ensuring a comprehensive and unified security posture across the network.

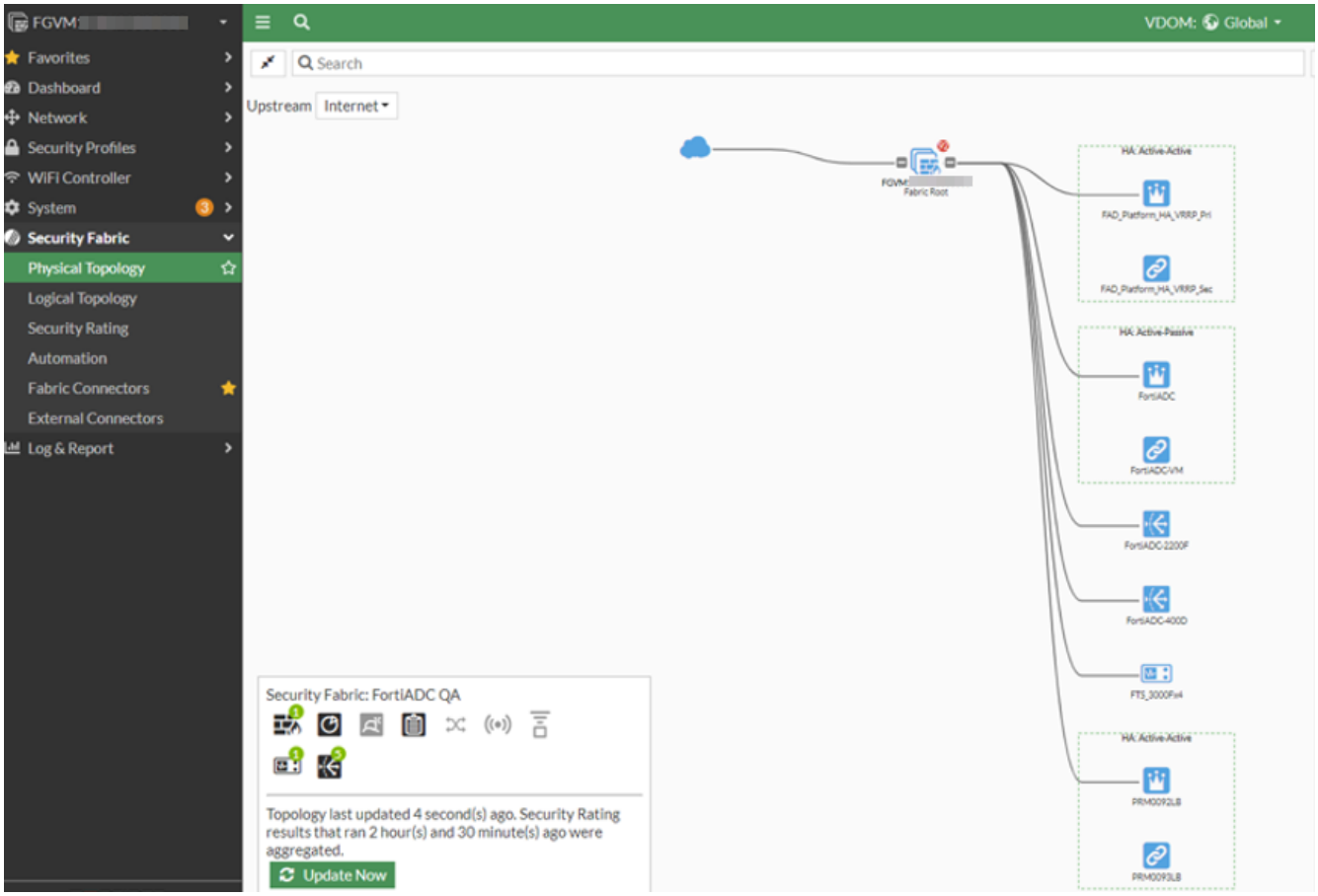
This information is also available in the FortiADC 7.6.0 Administration Guide and CLI Reference Guide:

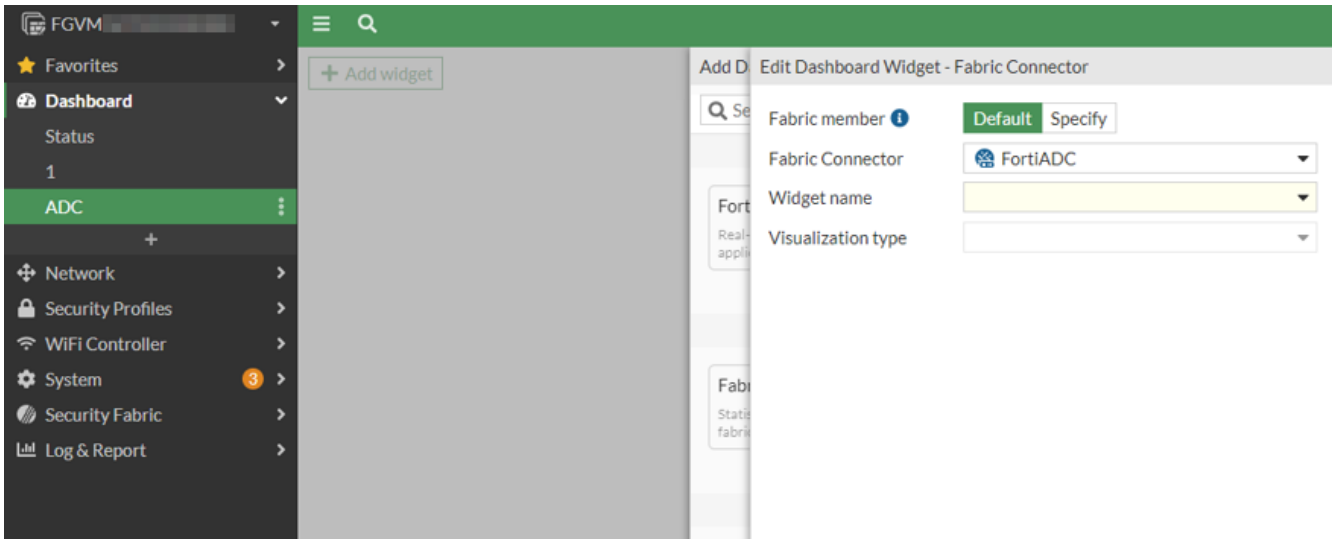
- FortiGate Security Fabric Connector
- config system csf
- diagnose debug module csfd

After configuring the Security Fabric Setup connector and successfully connecting to the FortiGate Security Fabric, from the FortiGate, you will be able to view the FortiADC on topology pages, and create a dashboard Fabric Device widget to view FortiADC data.




Global
Dashboard
Main
Create Dashboard
Security Fabric
System
Network
Log & Report
System Information
Hostname
FortiADC
Serial Number
FADVM
Firmware
v7.6.0 build6016 (Feature)
System Time
Fri May 17 09:34:46 2024
Up Time
0d, 23h, 10m, 34s
License
Firmware
WAF Signature
IP Reputation
Credential Stuffing Defense
Geo IP
Web Filter
Intrusion Prevention
Antivirus
Threat Analytics
Status
Connected
Security Log Forwarding
Enable
Security Fabric
Security Fabric Setup
Status
enable


FGVM
VDOM: Global
Favorites
Dashboard
Network
Security Profiles
WiFi Controller
System
Security Fabric
Physical Topology
Logical Topology
Security Rating
Automation
Fabric Connectors
External Connectors
Log & Report
Search
Upstream
Internet
Fabric Root
HA Active-Active
HA Active-Passive
FortiADC
FortiADC-VM
FortiADC-2200F
FortiADC-400D
Security Fabric: FortiADC QA
Topology last updated 4 second(s) ago. Security Rating results that ran 2 hour(s) and 30 minute(s) ago were aggregated.
Update Now



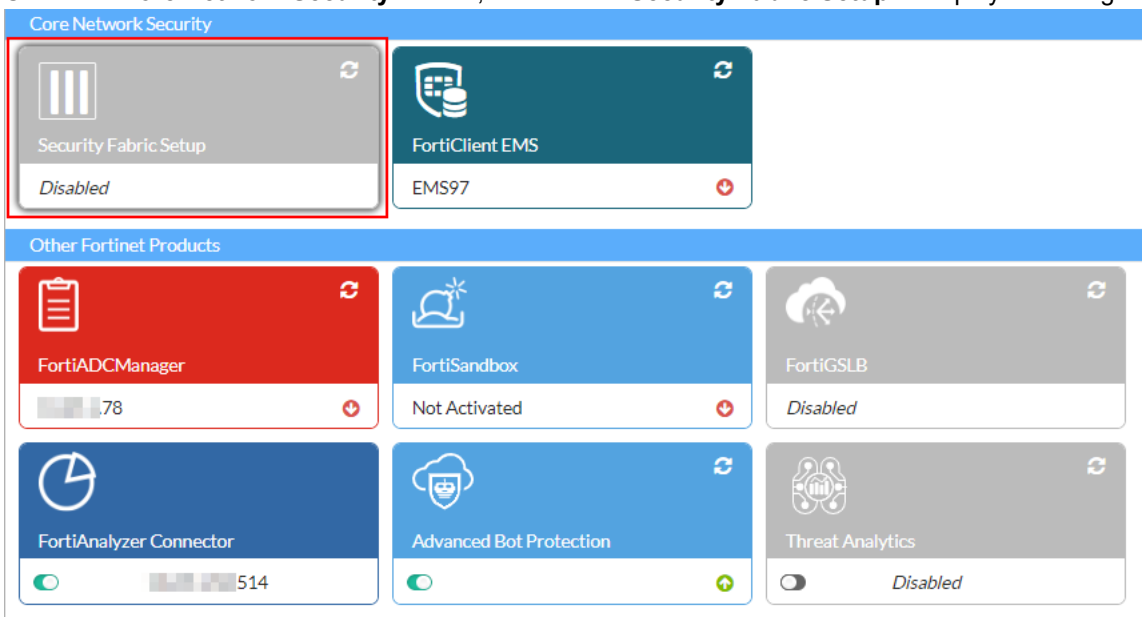
Before you begin:

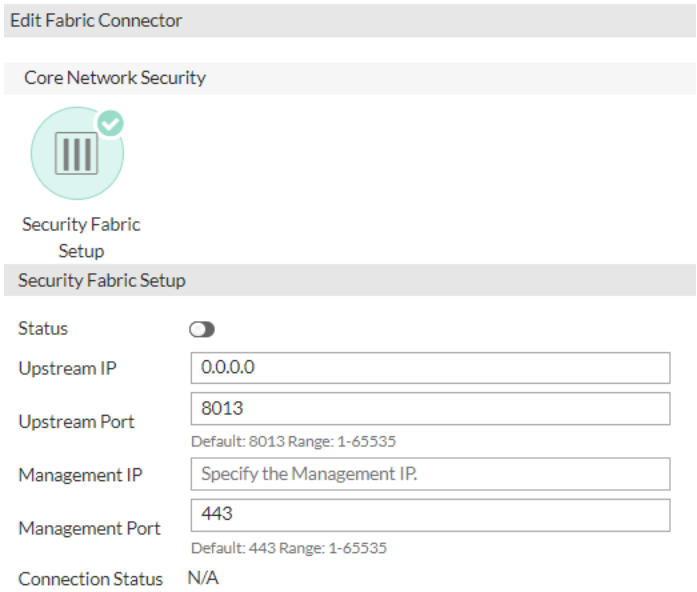
- You must have Read-Write access permission for FortiADC Systems settings.

To configure a FortiGate Security Fabric Setup connector:

1. Go to **Security Fabric > Fabric Connectors**.

2. Under the **Core Network Security** section, double-click **Security Fabric Setup** to display the configuration editor.





3. Configure the following settings:

Setting	Description
Status	Enable the FortiGate Security Fabric Setup connector. This is disabled by default.
Upstream IP	Specify the Upstream FortiGate IP address.
Upstream Port	Specify the Upstream port for the FortiGate. The default port is 8013. The valid range is 1-65535.
Management IP	Specify the Management IP address of this FortiADC appliance to join the Security Fabric.

Setting	Description
Management Port	Specify the Management port of this FortiADC appliance to join the Security Fabric. The default port is 443. The valid range is 1-65535.

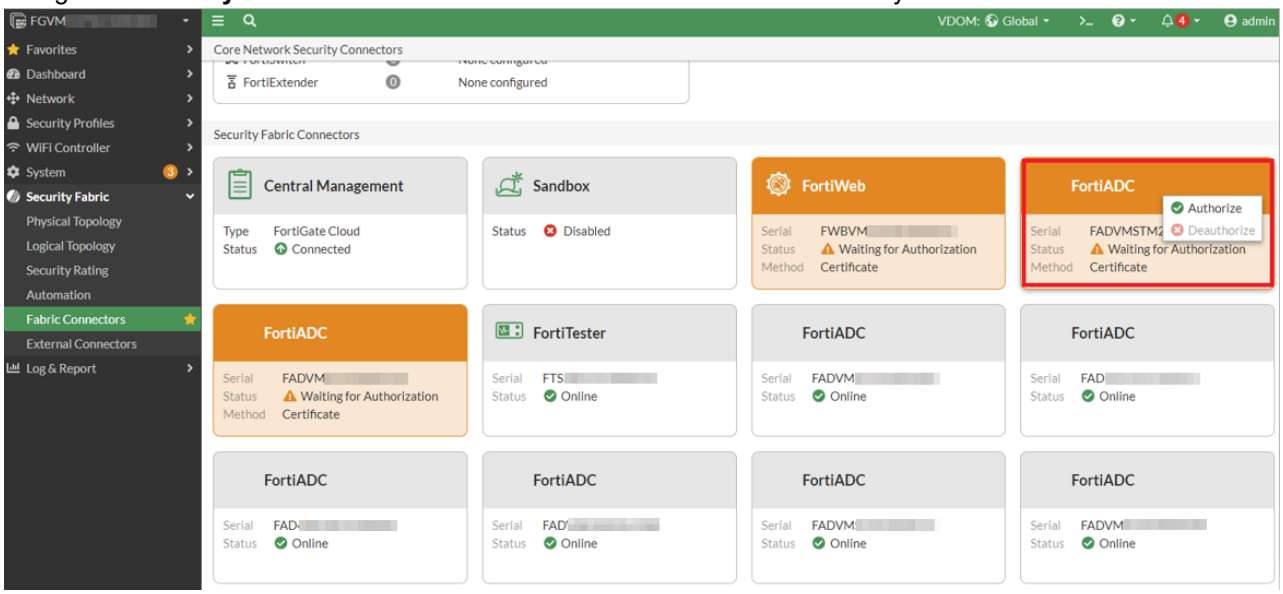
4. Click **Save**.
   The newly enabled Security Fabric Setup connector will display a Connection Status of **Pending Authorization**. The Security Fabric Setup connector will not be connected until the FortiADC has been authorized as a Fabric Device in FortiGate.

New CLI command:

```
config system csf
   set status {enable|disable}
   set upstream-ip <ip>
   set upstream-port <integer>
   set management-ip <ip>
   set management-port <integer>
end
```

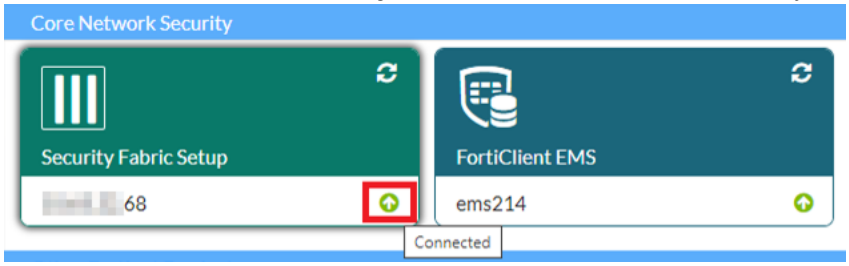
To authorize the FortiADC as a Fabric Device in FortiGate:

1. Login to FortiGate.
2. Navigate to **Security Fabric > Fabric Connectors** and select the Fabric device you want to authorize.



To check and troubleshoot the FortiGate Security Fabric Setup connector connection:

1. Go to **Security Fabric > Fabric Connectors**.
2. Under the **Core Network Security** section, locate the FortiGate Security Fabric Setup connector configuration.



3. The and icons indicate whether FortiGate has successfully authorized the FortiADC Fabric Device. Hover over the FortiGate Security Fabric Setup connector to see the status details. The table below lists the possible connection statuses for the FortiGate Security Fabric Setup connector.

Icon	EMS Status	Description
	Connected	The FortiADC has been successfully authorized as a Fabric Device through FortiGate.
	Pending Authorization	The FortiADC is waiting to be authorized as a Fabric Device in the FortiGate Security Fabric.
	Connecting	The FortiADC has successfully connected to the FortiGate server, but has not yet been authorized as a Fabric Device in the FortiGate Security Fabric.
	Authorization Rejected	The FortiADC has been Deauthorized as a Fabric Device in the FortiGate Security Fabric.
	Not Available	The FortiGate server was not reachable. Ensure the Upstream IP and system router is properly configured.
	Not Connected	The connection failed with unknown issue.

If the status is not Connected, edit the FortiGate Security Fabric Setup connector accordingly to troubleshoot the connection issue.

To further troubleshoot the FortiGate Security Fabric Setup connector issues, you can use the `diagnose debug module csfd all` CLI command to view debug logs.

FortiView

The FortiADC 7.6 release includes new features and enhancements in **FortiView**:

OWASP Top 10 Compliance dashboard on page 63

FortiADC introduces the new OWASP Top 10 Compliance dashboard evaluates your compliance rate against the OWASP Application Security Top 10 by assessing the security configuration of each application and categorizing them into the top 10 categories. Based on the assessment results, you can refine configurations to enhance the security posture of your virtual server.

OWASP Top 10 Compliance dashboard

FortiADC introduces the new OWASP Top 10 Compliance dashboard evaluates your compliance rate against the OWASP Application Security Top 10 by assessing the security configuration of each application and categorizing them into the top 10 categories. Based on the assessment results, you can refine configurations to enhance the security posture of your virtual server.

The OWASP Top 10 Compliance dashboard currently only supports HTTP, HTTPS, HTTP2, and Explicit-HTTP virtual servers; no other virtual server types will be included in the OWASP Top 10 Compliance rating.

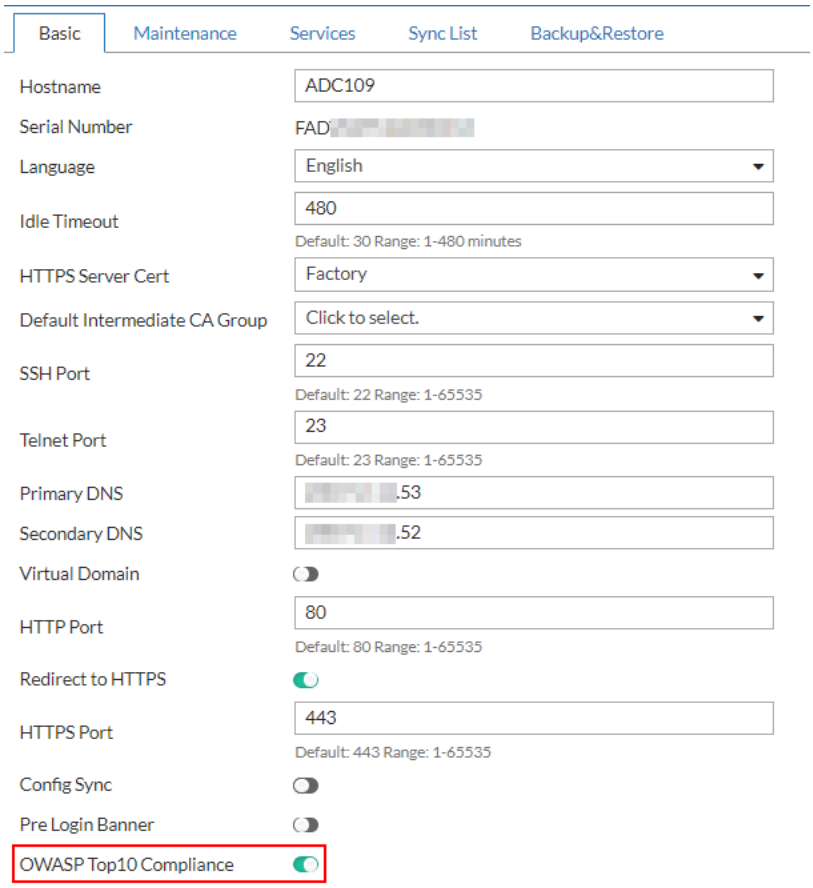
This information is also available in the FortiADC 7.6.0 Administration Guide:

- OWASP Top 10 Compliance

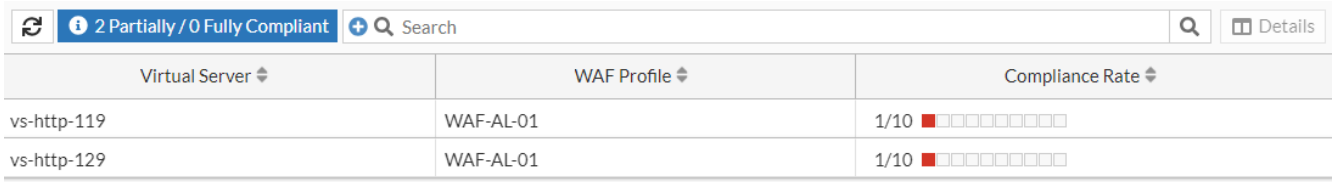
The OWASP Top 10 represents the most critical security risks to web applications as identified by OWASP (Open Web Application Security Project), an open community dedicated to enabling organizations to develop, purchase, and maintain applications and APIs that can be trusted. Organizations strive to comply with the OWASP Top 10 to ensure that the most critical security vulnerabilities in their web applications are addressed, allowing them to develop stronger defenses against cyber threats and reduce the risk of data breaches.

FortiADC's OWASP Top 10 Compliance dashboard empowers users to achieve full compliance by providing critical insights and evaluating the protection level of each server policy. The **Compliance Rate** of each virtual server is determined by analyzing their current security-related configurations—including WAF modules, Antivirus, FortiView, and Traffic log statuses—and assessing their level of protection against each OWASP Top 10 threat. This comprehensive analysis allows you to assess the effectiveness of your server policies and make the necessary adjustments to address OWASP Top 10 security risks effectively. To learn more about how the Compliance Rate is calculated, see Interpreting the Compliance Rate on page 65.

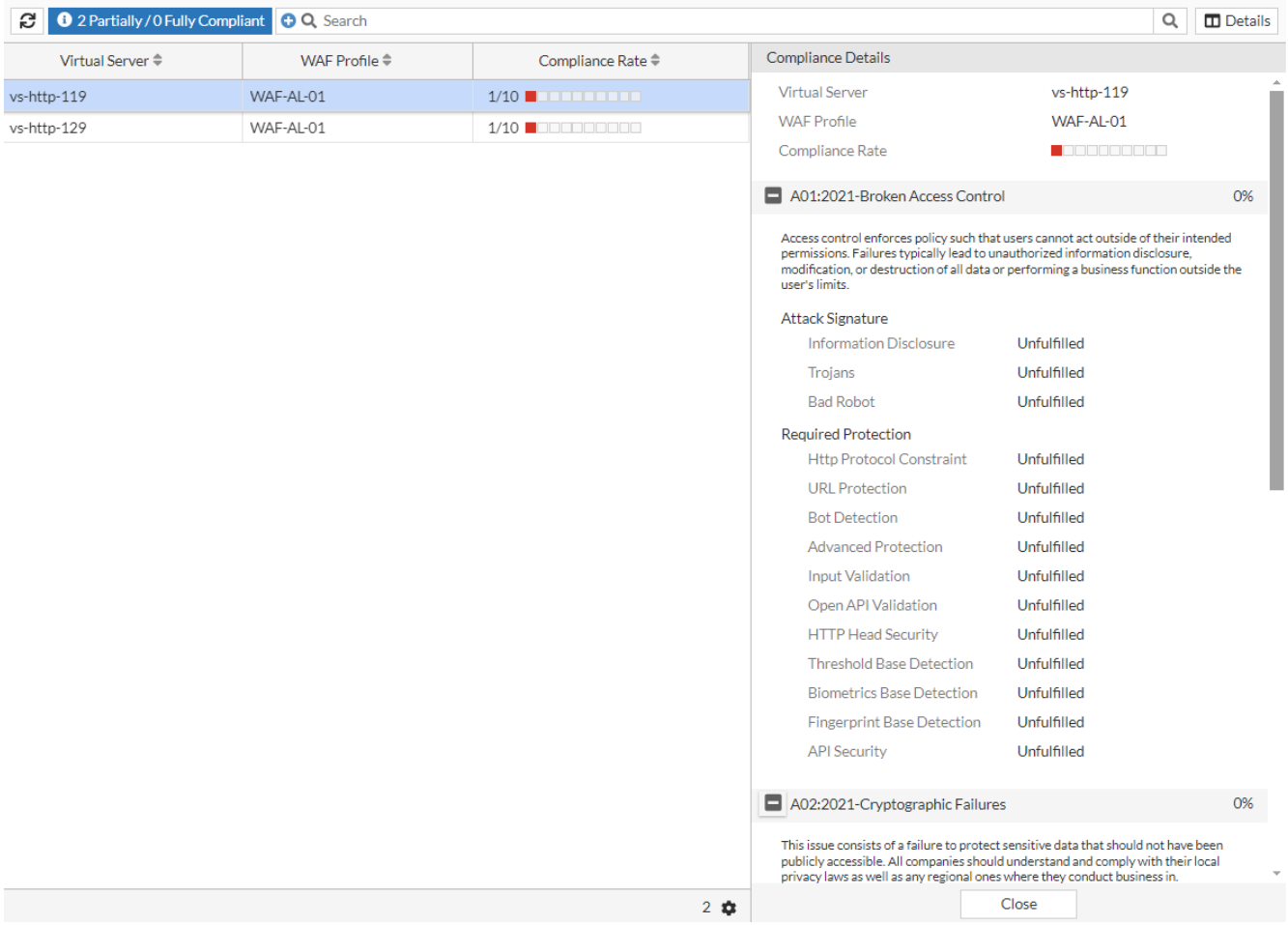
To view this FortiView dashboard, you must first enable the **OWASP Top10 Compliance** option in **Basic** settings from **System > Settings**.



After OWASP Top10 Compliance is enabled, you can view the Compliance Rate of your virtual servers from the **FortiView > OWASP Top 10 Compliance** page.



To view the **Compliance Details**, you can either select the Virtual Server and click **Details** or you can double-click the Virtual Server.



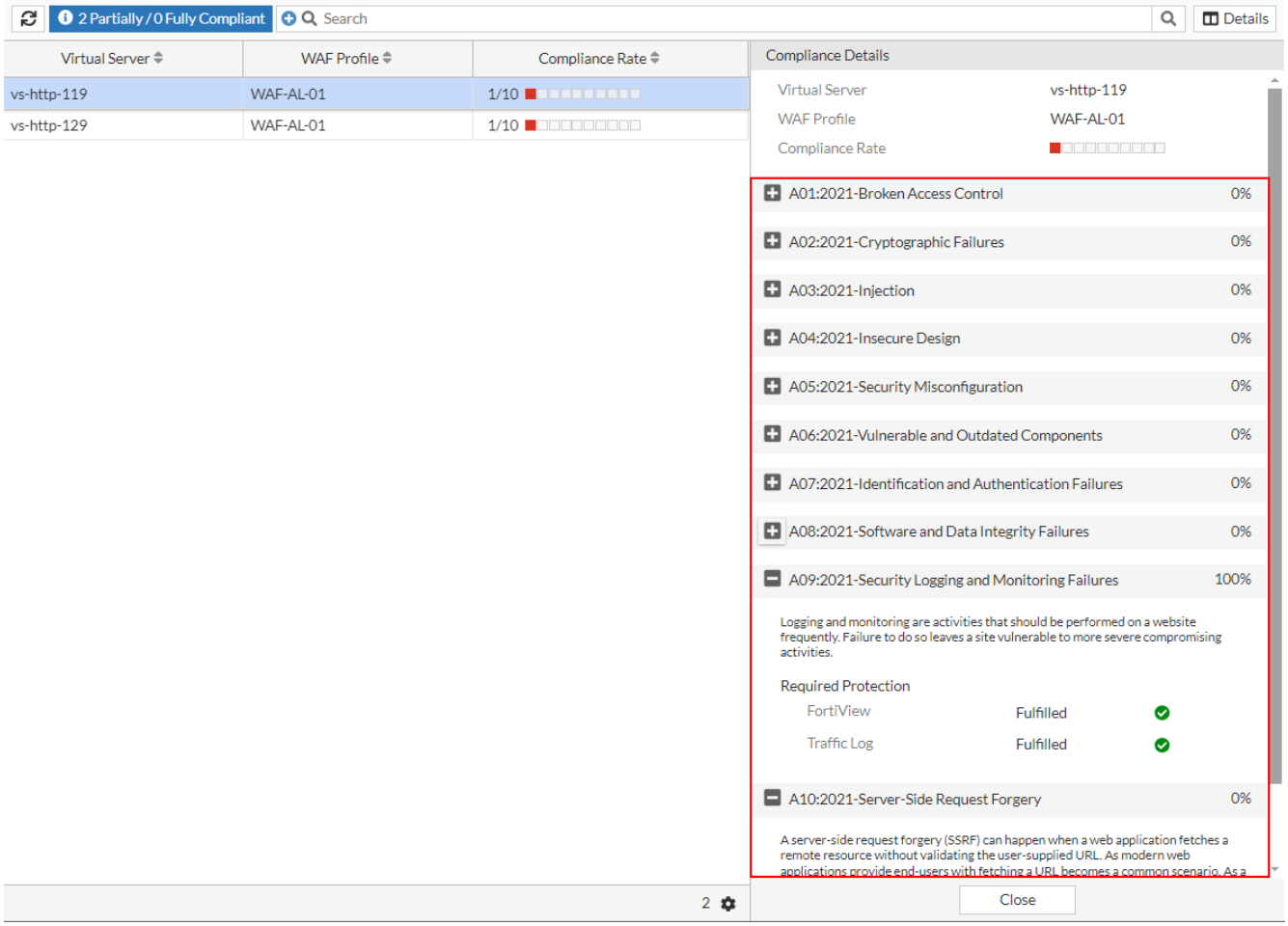
Interpreting the Compliance Rate

The **Compliance Rate** evaluates how well your virtual server policy aligns with the best practices recommended by OWASP for mitigating the Top 10 vulnerabilities. It assesses the configuration and rules in place to protect against these risks.

The Compliance Rate is calculated based on the percentage of compliance that has been reached for each OWASP Top 10 threat. For example, if the Compliance Rate is 1/10, then your virtual server configurations satisfies the protection requirements against 1 out of 10 OWASP Top 10 threats.

For each OWASP Top 10 threat, to reach 100% protection compliance, required protections must be fulfilled by completing specific security settings that target against the particular Top 10 vulnerability. The Compliance Rate is then calculated by aggregating the compliance percentage for all 10 threats.

In the example below, virtual server vs-http-119 has a Compliance Rate of 1/10, with only the **A09:2021-Security Logging and Monitoring Failures** vulnerability having fulfilled 100% of the protection requirements by enabling FortiView and Traffic Log.



The following lists the protection requirements for each OWASP Top 10 threat.

OWASP Top 10	Category	Configuration Setting
A01:2021-Broken Access Control Access control enforces policy to ensure that users cannot act outside of their intended permissions. Failures in access control typically lead to unauthorized information disclosure, modification or destruction of data, or performing business functions beyond the user's limits.	Attack Signature	Information Disclosure
		Trojans
		Bad Robot
	Required Protection	Http Protocol Constraint
		URL Protection
		Bot Detection
		Advanced Protection
		Input Validation
		Open API Validation
		HTTP Head Security
		Threshold Base Detection
		Biometrics Base Detection
		Fingerprint Base Detection
		API Security
A02:2021-Cryptographic Failures This issue involves a failure to protect sensitive data that should not be publicly accessible. Organizations should understand and comply with their local privacy laws as well as any regional regulations where they conduct business.	Attack Signature	Known Exploits
		Credit Card Detection
		Information Disclosure
	Required Protection	Cookie Security
		Data Loss Prevention
A03:2021-Injection An injection occurs when an attacker sends invalid data to a web application with the intention of making it perform actions it is not designed or programmed to do. Common types of injections include SQL, NoSQL, OS command, ORM, LDAP, and EL or OGNL injections.	Attack Signature	Generic Attacks
		Cross Site Scripting
		Cross Site Scripting(Extended)
		SQL Injection
		SQL Injection(Extended)
	Required Protection	SQL Injection Detection
		XSS Injection Detection
		XML Validation
		JSON Validation

OWASP Top 10	Category	Configuration Setting
A04:2021-Insecure Design Insecure design is a broad category representing various weaknesses, characterized by "missing or ineffective control design". An insecure design cannot be fixed by a perfect implementation because, by definition, the necessary security controls were never created to defend against specific attacks.	Attack Signature	Bad Robot
		Credit Card Detection
		Cross Site Scripting
		Generic Attacks
		Information Disclosure
		Known Exploits
		SQL Injection
		Trojans
	Required Protection	JSON Validation
		XML Validation
		Open API Validation
		API Gateway
		API Security
		Bot Detection
		Threshold Base Detection
		Biometrics Base Detection
		Fingerprint Base Detection
		Antivirus
A05:2021-Security Misconfiguration Security misconfigurations are design or configuration weaknesses resulting from configuration errors or shortcomings. Without a consistent and repeatable application security configuration process, systems are at a higher risk.	Attack Signature	Generic Attacks
	Required Protection	XML Validation
		JSON Validation
		CORS Protection
A06:2021-Vulnerable and Outdated Components Components with known vulnerabilities, such as CVEs, should be identified and patched, while outdated or malicious components should be evaluated for viability and the risks they may introduce. Attackers actively seek out websites using vulnerable components and aggressively exploit them.	Attack Signature	Generic Attacks
		Generic Attacks(Extended)
		Known Exploits

OWASP Top 10	Category	Configuration Setting
A07:2021-Identification and Authentication Failures Confirmation of the user's identity, authentication, and session management is critical to protect against authentication-related attacks. A broken authentication vulnerability can allow an attacker to use manual and/or automatic methods to try to gain control over any account they want in a system.	Attack Signature	Generic Attacks
	Required Protection	CSRF Protection
		Brute Force Attack Detection
		Credential Stuffing Defense
		API Gateway
A08:2021-Software and Data Integrity Failures Software and data integrity failures occur when code and infrastructure do not protect against integrity violations. These failures can take many forms, especially as the web evolves and the use of third-party code and services in web applications becomes increasingly common.	Attack Signature	Generic Attacks
A09:2021-Security Logging and Monitoring Failures Logging and monitoring should be performed frequently on a website. Failure to do so leaves the site vulnerable to more severe compromise.	Required Protection	FortiView
		Traffic Log
A10:2021-Server-Side Request Forgery Server-side request forgery (SSRF) occurs when a web application fetches a remote resource without validating the user-supplied URL. With modern web applications allowing end-users to fetch URLs, SSRF incidents are increasingly common.	Attack Signature	Generic Attacks

System

The FortiADC 7.6 release includes new features and enhancements in the following **System** modules:

- Settings on page 71
- Cloud Auto Scaling on page 83
- High Availability on page 87
- Administrator on page 100
- SNMP on page 103
- Certificate on page 107

Settings

Send FortiADC Threat Telemetry to FortiGuard 7.6.1 on page 72

FortiADC can now transmit threat telemetry data generated by its IPS and Antivirus modules to FortiGuard, enabling the continuous improvement of its threat detection and response capabilities. This integration enhances the overall efficacy of FortiGuard services by leveraging real-time data insights.

NTP authentication 7.6.1 on page 76

FortiADC now supports NTP server authentication to enhance the integrity and security of time synchronization. The available key algorithms include SHA1, SHA256, AES128, and AES256. This feature safeguards NTP traffic against vulnerabilities such as spoofing and tampering, ensuring secure communication between the FortiADC and NTP servers.

Administrator lockout controls in CLI on page 79

By default, the allowable number of password retry attempts is set to three, allowing the administrator a maximum of three attempts at logging into their account before they are locked out for a set amount of time (60 seconds by default). Now, using the new CLI options `admin-lockout-duration` and `admin-lockout-threshold` in `config system global`, you can now configure the number of attempts and the default wait time before the administrator can retry the password.

Direct VDOM Access for Administrators on page 80

FortiADC has enhanced the VDOM functionality to enable non-root VDOM administrators to log in through the root VDOM interface without being granted root VDOM privileges. The new **Direct VDOM Access** option allows a non-root VDOM administrator access to the root VDOM interface to modify settings strictly related to their allocated non-root VDOM.

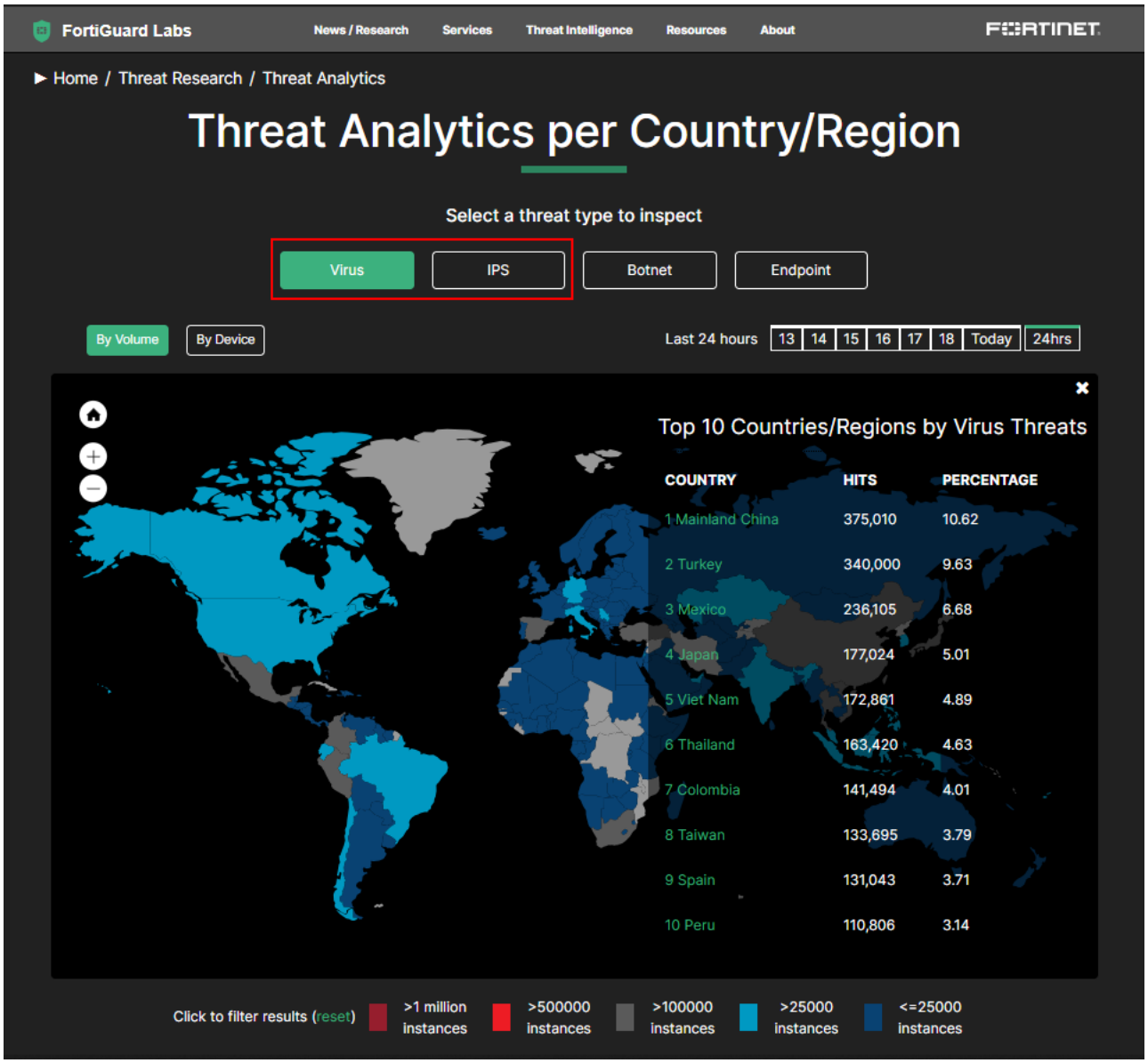
Send FortiADC Threat Telemetry to FortiGuard - 7.6.1

FortiADC can now transmit threat telemetry data generated by its IPS and Antivirus modules to FortiGuard, enabling the continuous improvement of its threat detection and response capabilities. This integration enhances the overall efficacy of FortiGuard services by leveraging real-time data insights. It is important to note that this data is not shared with external entities and is safeguarded in accordance with Fortinet's privacy policy.

This information is also available in the FortiADC 7.6.1 Administration Guide and CLI Reference Guide:

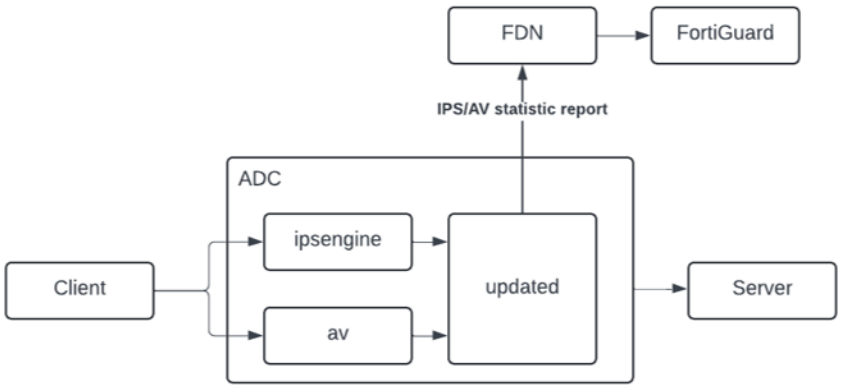
- Sending FortiADC Threat Telemetry to FortiGuard
- Configuring basic system settings
- config system global

FortiADC is designed to send threat telemetry data to FortiGuard to enhance threat intelligence and enable product-specific threat analysis. FortiGuard processes this data for analysis, including metrics on Antivirus, IPS, and other security statistics. The AV/IPS analysis has been standardized, allowing FortiADC to send IPS and Antivirus data to FortiGuard for comprehensive threat assessment.



FortiADC Threat Telemetry Workflow

The **FortiADC** Threat Telemetry Workflow outlines the systematic process through which the IPS and AV engines detect rule matches, collect relevant statistics, and transmit this data to FortiGuard for comprehensive analysis.

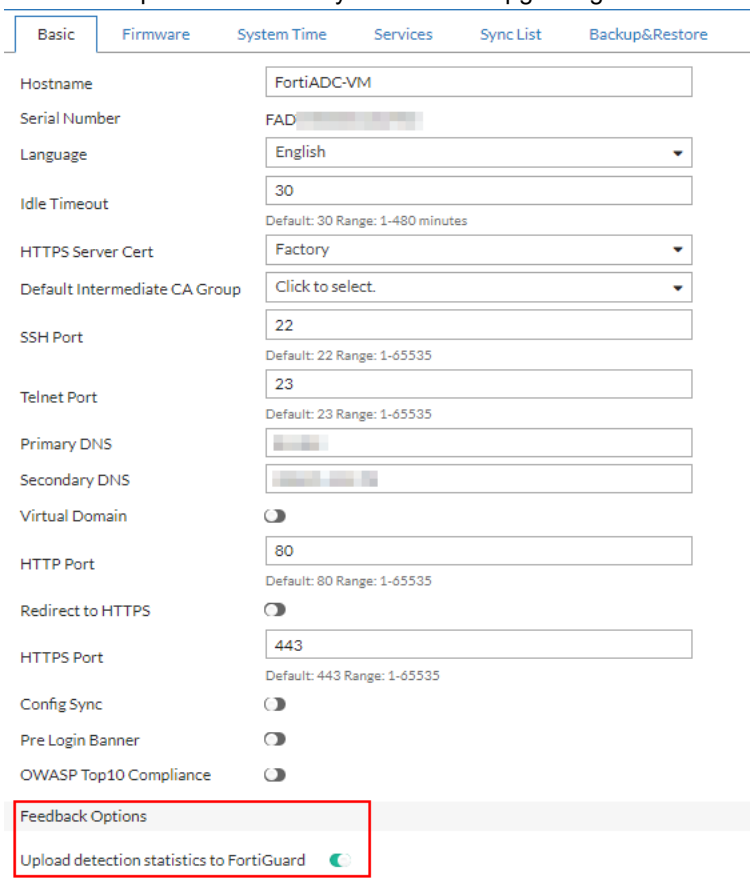


1. **Detection** — The IPS or AV engine detects rule matches based on the configured threat protection policies.
2. **Data Transmission to Daemon** — After detection, the IPS/AV engine sends the matched rule information to the updated daemon for further processing.
3. **Statistics Collection** — If the option is enabled to send statistics, the daemon collects IPS/AV statistics, including details of the detected rule matches.
4. **Data Transmission to FortiGuard** — The daemon sends the collected statistics to FortiGuard every 60 minutes by default. This reporting interval can be configured by the user through CLI.
5. **Data Processing** — Once FortiGuard receives the IPS/AV statistics, it processes the data for analysis and generates a report.
6. **Portal Display** — The processed data is then made available and displayed on the FortiGuard portal for user review.

To enable FortiADC to transmit threat telemetry data to FortiGuard:

1. Navigate to **System > Settings**.
2. From the **Basic** tab, scroll down to locate the **Feedback Options** section.
3. Enable the **Upload detection statistics to FortiGuard** option.

Note: This option is enabled by default after upgrading to FortiADC version 7.6.1.



CLI update in `config system global`:

```
config system global
   set fds-statistics {enable|disable}
   set fds-statistics-period <integer>
end
```

`fds-statistics`	Enable or disable FortiADC detection statistics upload to FortiGuard. This is enabled by default.
`fds-statistics-period`	Specify the FortiGuard statistics collection period in minutes. The default value is 60 minutes, and the valid range is 1-1440 minutes.

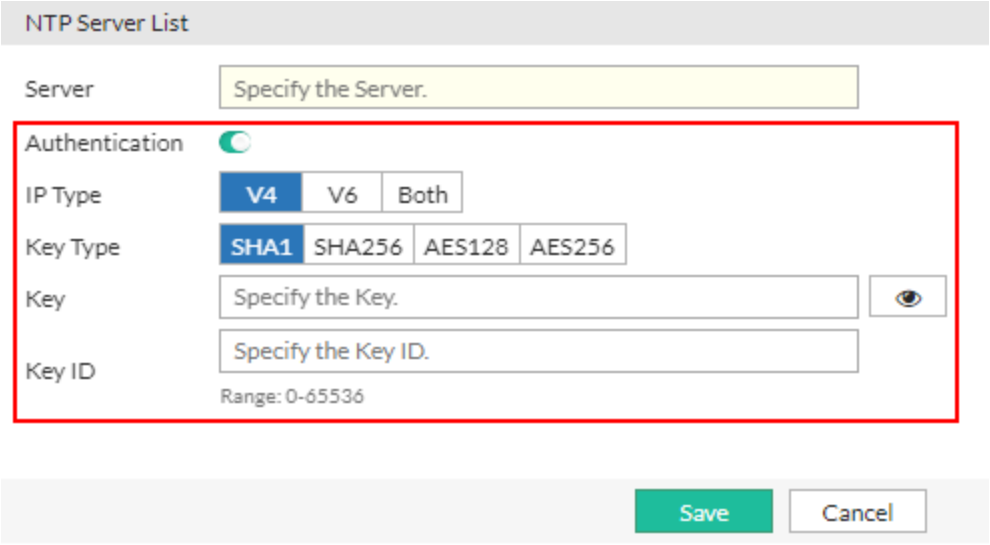
NTP authentication - 7.6.1

FortiADC now supports NTP server authentication to enhance the integrity and security of time synchronization. The available key algorithms include SHA1, SHA256, AES128, and AES256. This feature safeguards NTP traffic against vulnerabilities such as spoofing and tampering, ensuring secure communication between the FortiADC and NTP servers.

This information is also available in the FortiADC 7.6.1 Administration Guide and CLI Reference Guide:

- Configuring system time
- config system time ntp

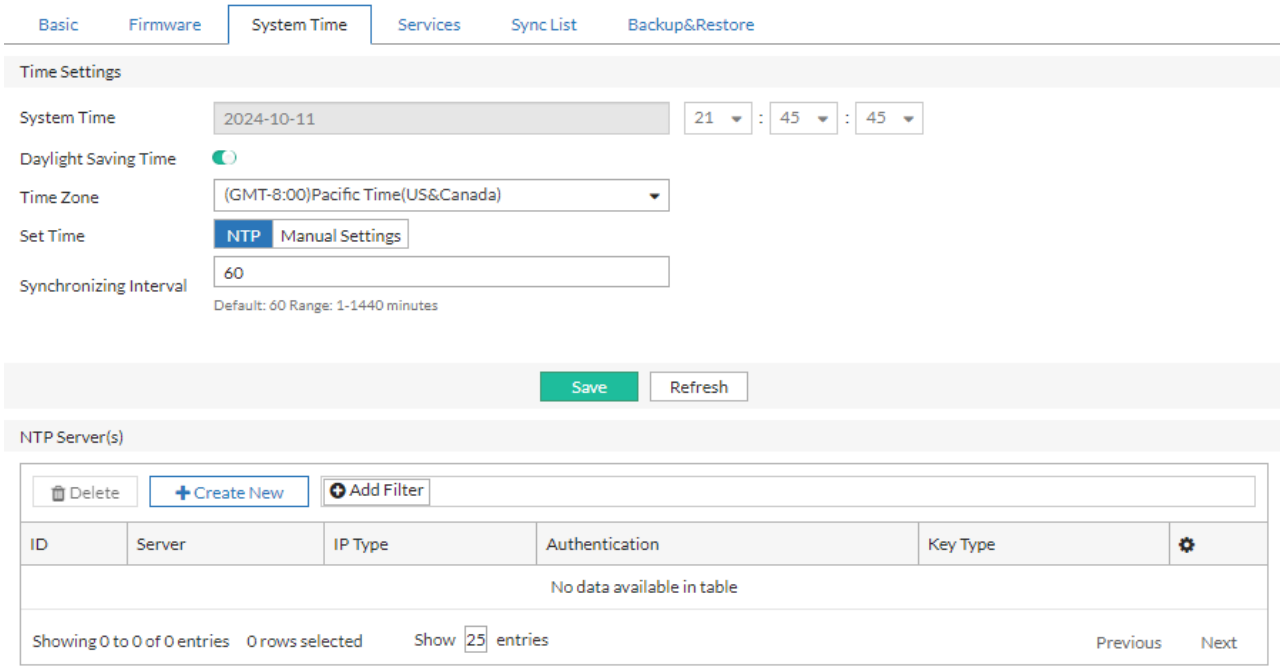
When configuring the **NTP Server List**, you can now enable the new **Authentication** option and select the preferred encryption method.



To configure system time settings with NTP:

1. Go to **System > Settings**.
2. Click the **System Time** tab.

3. Configure the Time Settings:



Setting	Guidelines
System Time	Displays the system time. You can use NTP to set the system time, or use the controls to set the system time manually. Specify time in HH:MM:SS format.
Daylight Saving Time	Enable if you want the system to adjust its own clock when its time zone changes between daylight saving time (DST) and standard time.
Time Zone	Select the time zone where the appliance is located.
Set Time	Select **NTP**.
Synchronizing Interval	The **Synchronizing Interval** option is available if **Set Time** is **NTP**. Specify how often the system synchronizes its time with the NTP server. The default is 60 minutes. The valid range is 1-1440.

4. Click **Save**.
   The **NTP Server(s)** section becomes configurable once the Time Settings configuration is successfully saved.
5. Under the NTP Server(s) section, click **Create New** to display the configuration editor.
6. Configure the NTP Server List settings:

Setting	Guidelines
Server	Specify a space-separated list of IP addresses or FQDNs for an NTP server or pool, such as `pool.ntp.org`. To find an NTP server, go to http://www.ntp.org. Ensure there are no duplicate entries. A server is deemed a duplicate if it shares the same IP address or hostname.

Setting	Guidelines
Authentication	Enable to apply authentication keys to secure the NTP server. This is disabled by default.
IP Type	The **IP Type** setting applies to the FQDNs used for the NTP server. FortiADC synchronizes time only with FQDN IP addresses that match the selected IP type. Select the IP type from the following: • V4 • V6 • Both The default option is V4.
Key Type	The **Key Type** option is available if **Authentication** is **enabled**. Select the key type from the following: • SHA1 • SHA256 • AES128 • AES256 The default option is SHA1.
Key	The **Key** option is available if **Authentication** is **enabled**. Specify the Key in hexadecimal format. The maximum length is 127 digits or characters.
Key ID	The **Key ID** option is available if **Authentication** is **enabled**. Specify the Key ID. The valid range is 0-65536

7. Click **Save**.

CLI updates in `config system time ntp`:

```
config system time ntp
   set ntpsync enable
   set syncinterval 1
   config  ntp-server
      edit 1
         set server <server_name>
         set authentication {enable|disable}
         set key-type {aes128|aes256|sha1|sha256}
         set key <key>
         set key-id <key_id>
         set ip-type {v4|v6|both}
      next
   end
end
```

Administrator lockout controls in CLI

By default, the allowable number of password retry attempts is set to three, allowing the administrator a maximum of three attempts at logging into their account before they are locked out for a set amount of time (60 seconds by default). Now, using the new CLI options `admin-lockout-duration` and `admin-lockout-threshold` in `config system global`, you can now configure the number of attempts and the default wait time before the administrator can retry the password.

This information is also available in the FortiADC 7.6.0 CLI Reference Guide:

- config system global

New CLI options:

```
config system global
   set admin-lockout-duration <integer>
   set admin-lockout-threshold <integer>
end
```

`admin-lockout-duration`	Amount of time in seconds that an administrator account is locked out after reaching the **admin-lockout-threshold** for repeated failed login attempts. The default is 60 seconds. The valid range is 1 to 2147483647 seconds.
`admin-lockout-threshold`	Number of failed login attempts before an administrator account is locked out for the **admin-lockout-duration**. The default is 3. The valid range is 1 to 10. It is recommended to keep the number of allowable attempts lower as increasing the number of retry attempts elevates the risk of unauthorized individuals successfully guessing the password.

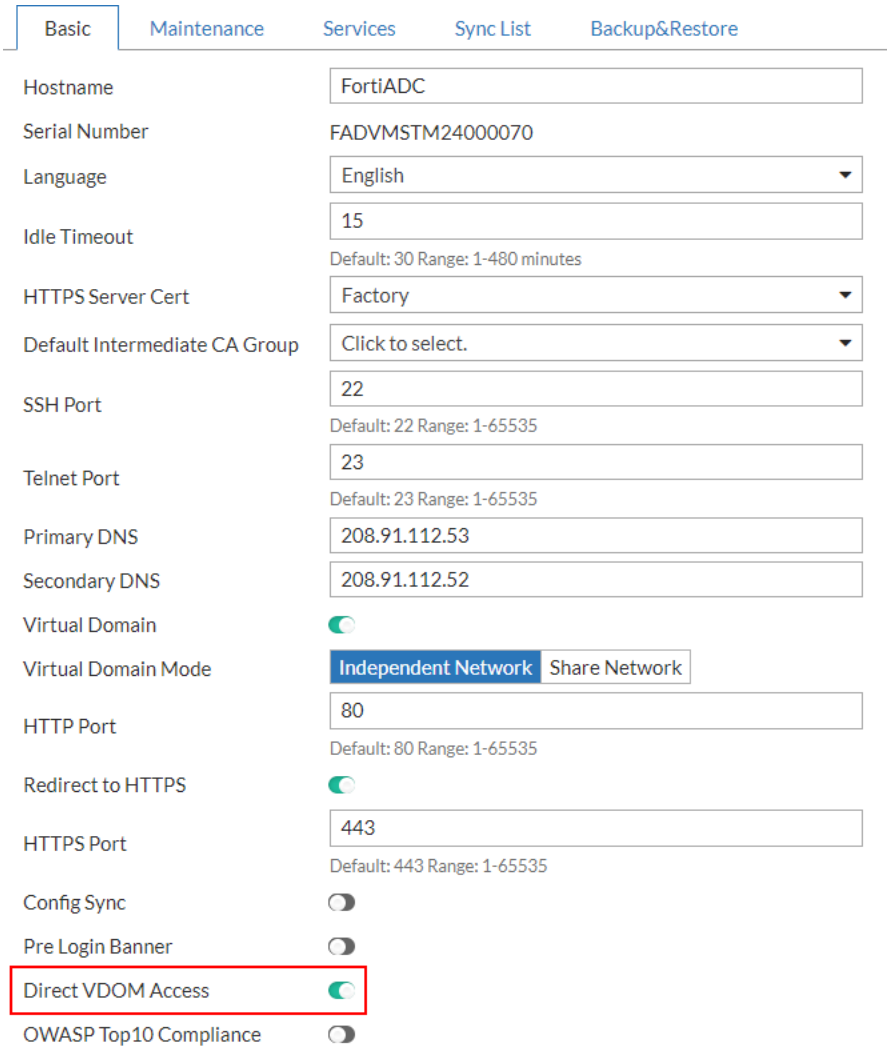
Direct VDOM Access for Administrators

FortiADC has enhanced the VDOM functionality to enable non-root VDOM administrators to log in through the root VDOM interface without being granted root VDOM privileges. The new **Direct VDOM Access** option allows a non-root VDOM administrator access to the root VDOM interface to modify settings strictly related to their allocated non-root VDOM.

This information is also available in the FortiADC 7.6.0 Administration Guide and CLI Reference Guide:

- Configuring basic system settings
- config system global

This new functionality provides further flexibility in scenarios where the root VDOM serves as the hub for network management activities of various applications deployed across non-root VDOMs by allowing non-root VDOM owners access to the root VDOM with restricted permissions to manage configurations for their assigned non-root VDOM. Whereas, previously, to allow application owners management access to configure aspects of their respective VDOMs (such as adding virtual servers or managing certificates) they would be granted management permissions for both their VDOM and the root VDOM, which provided unrestricted access to modify any root VDOM configurations. With the new **Direct VDOM Access** option, application owners can be granted restricted access to the root VDOM interface that only allows permissions to configure or modify settings to their designated VDOM only, ensuring they can only affect configurations within their defined scope of responsibility.



To enable Direct VDOM Access to allow a non-root VDOM administrator to log into the root VDOM interface:

1. From the non-root VDOM, navigate to **System > Settings**.
   The configuration page displays the **Basic** tab.

2. Configure the required settings:

Setting	Guidelines
Virtual Domain	The Virtual Domain option must be **enabled**.
Virtual Domain Mode	Select **Independent Network** as the Virtual Domain Mode.
Direct VDOM Access	When the **Virtual Domain Mode** is **Independent Network**, the **Direct VDOM Access** option becomes available. Enable Direct VDOM Access. This is disabled by default. Once enabled, all non-root VDOM administrators can login through the root VDOM interface without needing root VDOM privileges. From the root VDOM interface, the non-root VDOM administrator can access and modify the settings relating to their designated non-root VDOM.

3. Save the Basic settings changes.

With Direct VDOM Access enabled, you can now log in to the root VDOM interface using the same credentials for your assigned non-root VDOM.

CLI update in `config system global`:

```
config system global
   edit <name>
     set vdom-admin enable
     set vdom-mode independent-network
     set admin-bypass-vdom-check {enable|disable}
   next
end
```

Cloud Auto Scaling

Azure Autoscaling for FortiADC VMSS 7.6.1 on page 84

The Azure Autoscaling feature enables dynamic scaling of FortiADC Virtual Machine Scale Sets (VMSS) based on real-time CPU usage. A primary FortiADC and multiple secondaries distribute traffic from the Azure Load Balancer, optimizing resource allocation and enhancing system resilience by automatically adjusting instances to meet changing traffic demands, ensuring high availability and efficient load management.

Azure Autoscaling for FortiADC VMSS - 7.6.1

The Azure Autoscaling feature enables dynamic scaling of FortiADC Virtual Machine Scale Sets (VMSS) based on real-time CPU usage. A primary FortiADC and multiple secondaries distribute traffic from the Azure Load Balancer, optimizing resource allocation and enhancing system resilience by automatically adjusting instances to meet changing traffic demands, ensuring high availability and efficient load management.

This information is also available in the FortiADC 7.6.1 Administration Guide:

- Cloud Auto Scaling

The autoscaling feature, available in FortiADC 7.6.1 for On-demand (PAYG), can be deployed on the Azure platform using ARM templates. A server-less web application running on Azure App Service enables authorized access to resources within the auto-scaling cluster. This application selects the primary node in the FortiADC Virtual Machine Scale Set (VMSS) and communicates its IP address and VMID to the secondary nodes. Configuration synchronization occurs unidirectionally from the primary node to secondary nodes, ensuring consistent settings across the cluster.

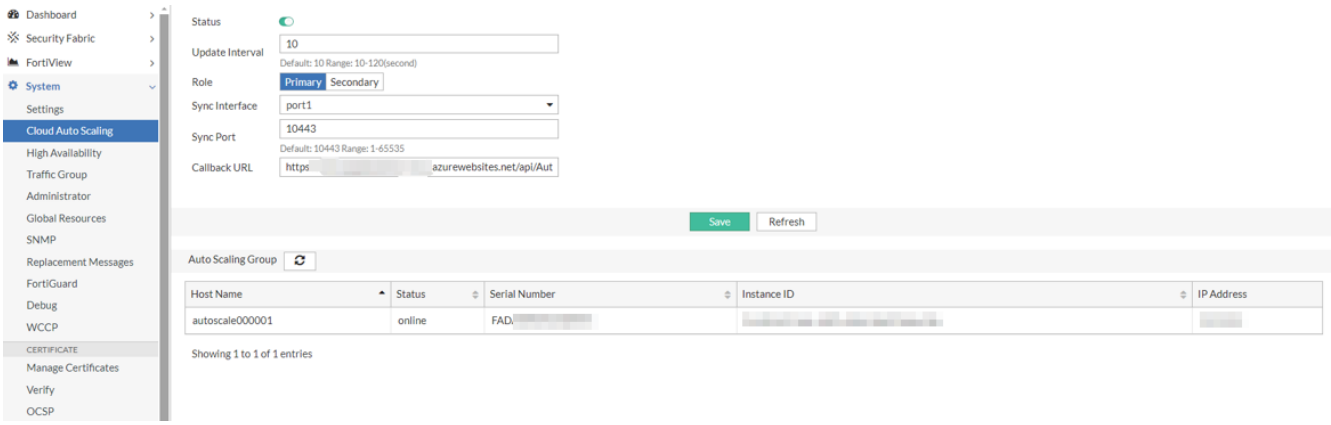
For detailed steps on deploying Azure autoscaling, refer to the FortiADC Azure Deployment Guide.

FortiADC Cloud Auto Scaling configuration

Once autoscaling is deployed on Azure, the status of FortiADC devices can be monitored through the GUI of the primary node. FortiADCs can be in one of two states:

- **init** — In this state, the FortiADC establishes a connection with the primary node and performs a full configuration synchronization.
- **online** — This state indicates the synchronization process has successfully completed.

Before connecting to any secondary node via GUI, console, or SSH, ensure its status has transitioned to "online."



Any updates to the Cloud Auto Scaling configuration in FortiADC are not automatically synchronized with the autoscaling group on the cloud platform. To prevent configuration discrepancies between nodes, the autoscaling group must be manually updated to maintain synchronization consistency.

The same information can be accessed via CLI:

Primary node:

```
config system auto-scale
   set status enable
   set role primary
   set callback-url https://example.com
   set hb-interval 10
   set config-sync-port 10443
end
```

Secondary node:

```
config system auto-scale
   set status enable
   set role secondary
   set primary-ip 10.10.0.6
   set callback-url https://example.com
   set hb-interval 10
   set config-sync-port 10443
end
```

Other considerations when Azure autoscaling is enabled include:

- The Azure Load Balancer (LB) Backend tab will be hidden from the user interface.
- The virtual server address will also be concealed, with the virtual server operating using the interface IP address and port instead.

Troubleshooting

Debug information can be accessed through the Azure Function App's AutoscaleHandler log or the FortiADC CLI.

In the AutoscaleHandler log, you can verify whether the FortiADC VMs in the VMSS are sending heartbeat callbacks in a timely manner and whether they are maintaining a healthy state. Check both the logs and DynamoDB records for the deployed resources.

On the FortiADC-VM, you can utilize CLI commands to monitor the status of the cloud autoscaling daemon. Use the following commands:

To review the heartbeat callback results and failover information in case a primary election is triggered:

```
# diagnose debug cloud-autoscale autoscaled
```

To check the synchronization between the primary and secondary FortiADCs, as well as view any crash logs if they exist:

```
# diagnose debug cloud-autoscale autoscale-tunnel
```

Limitations

Synchronization

The configuration synchronization for FortiADC is consistent with the settings applied when HA VRRP is enabled. Note that synchronization is not managed by the HA module.

When the FortiADC is operating as the primary node, it listens on the IP address `ip:10443`, where ip is the interface IP defined in the sync-interface under system auto-scale settings. When a secondary FortiADC connects to the primary, it initiates a full configuration synchronization to ensure consistency with the primary.

Configuration changes can only be initiated by the primary node. Any modifications made on a node assigned the secondary role will not be synchronized to other nodes within the VMSS.

High Availability

When auto-scaling is enabled, the HA mode must operate in standalone mode. Switching the HA mode to VRRP is prohibited while auto-scaling is active.

High Availability

Enhance Status Reporting Accuracy in Active-Passive HA 7.6.2 on page 88

This enhancement improves status reporting accuracy in Active-Passive HA mode by ensuring that only the active device reflects current health information, providing a clearer and more accurate view of the system's operational state.

HA cluster supports maximum 8 member nodes on page 89

FortiADC now supports a maximum of 8 nodes per HA cluster, whereas previously a maximum of 2 nodes were supported.

HA management interface network options via CLI on page 95

FortiADC introduces new CLI options in the `config system ha` command to allow you to customize your choice of network type as either IPVLAN or MACVLAN to create your HA management interface.

Virtual MAC address option as interface in Active-Passive HA via CLI on page 97

Through the `config system ha` CLI command, you now have the option to enable or disable the virtual MAC address as the interface for all nodes in an HA Active-Passive cluster.

New and enhanced CLI commands to force HA nodes into standby mode on page 98

For certain troubleshooting, maintenance, or testing scenarios, the ability to trigger HA failover manually can be useful. For this purpose, a new CLI command `execute ha force failover-standby` is introduced to apply to HA Active-Passive and Active-Active modes only. And the existing `execute ha force standby` command has been enhanced to apply the comparable functionality for HA Active-Active-VRRP.

Enhance Status Reporting Accuracy in Active-Passive HA - 7.6.2

This enhancement improves status reporting accuracy in Active-Passive HA mode by ensuring that only the active device reflects current health information, providing a clearer and more accurate view of the system's operational state.

- The standby device will no longer send health check results for virtual servers, pools, or pool members.
- On the standby device, the status of these objects will appear as "Unknown" and be displayed in gray in the GUI.
- When queried via the REST API, the status of virtual servers, pools, and pool members on the standby device will also return "Unknown."

HA cluster supports maximum 8 member nodes

FortiADC now supports a maximum of 8 nodes per HA cluster, whereas previously a maximum of 2 nodes were supported.

This information is also available in the FortiADC 7.6.0 Administration Guide:

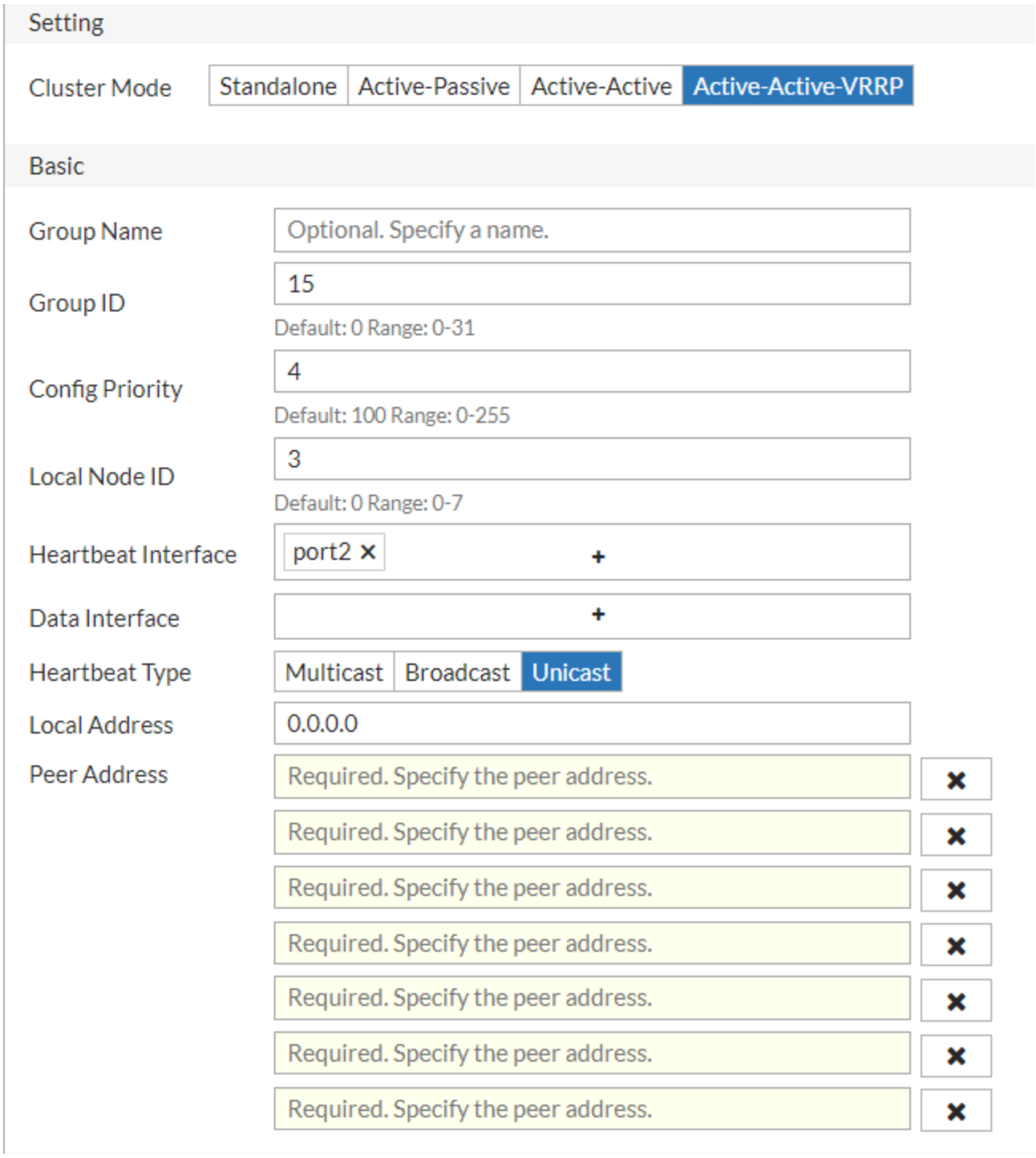
- Configuring HA settings
- Monitoring an HA cluster

HA configuration updates

The following configuration changes have been made to enable the support of 8 member nodes in one HA cluster.
Note: The configuration remains the same for Active-Passive and Active-Active HA modes where one Primary node is elected and all other nodes are designated as Secondary.

Active-Active-VRRP mode

When HA Active-Active-VRRP uses Unicast Heartbeat Type, you now must specify the Local Address, and at least one and up to seven Peer Addresses.



In CLI, each Peer IP address must be separated with a single space.

Example:

```
config system ha
  set mode active-active-vrrp
  set hbdev port2
```

```
  set group-id 19
  set local-node-id 1
  set hb-type unicast
  set local-address 192.168.100.141
  set peer-address 192.168.100.223 192.168.100.185 192.168.100.118 192.168.100.35
192.168.100.227 192.168.100.102 192.168.100.14
end
```

New Configuration page, Configuration Primary and Secondary designation

In the newly added **Configuration** page, each member node is listed. In particular, the **Config Source** column indicates whether a member node is the configuration source for synchronization.

Topology | Configuration

Host Name	State	Serial Number	Node ID	IP Address	Config Source	
FADC0	Secondary (Active)	FADV32TM24000029	0	169.254.255.106	Y	
FADC1	Secondary (Active)	FADV32TM23000208	1	169.254.47.63	N	
FADC3	Secondary (Active)	FADV32TM24000020	3	169.254.15.212	N	
FADC2	Secondary (Active)	FADV32TM24000019	2	169.254.82.86	N	
FADC6	Secondary (Active)	FADV32TM24000023	6	169.254.53.141	N	
FADC4	Secondary (Active)	FADV32TM24000021	4	169.254.10.116	N	
FADC7	Primary	FADV32TM24000024	7	169.254.8.125	N	
FADC5	Secondary (Active)	FADV32TM24000022	5	169.254.132.187	N	

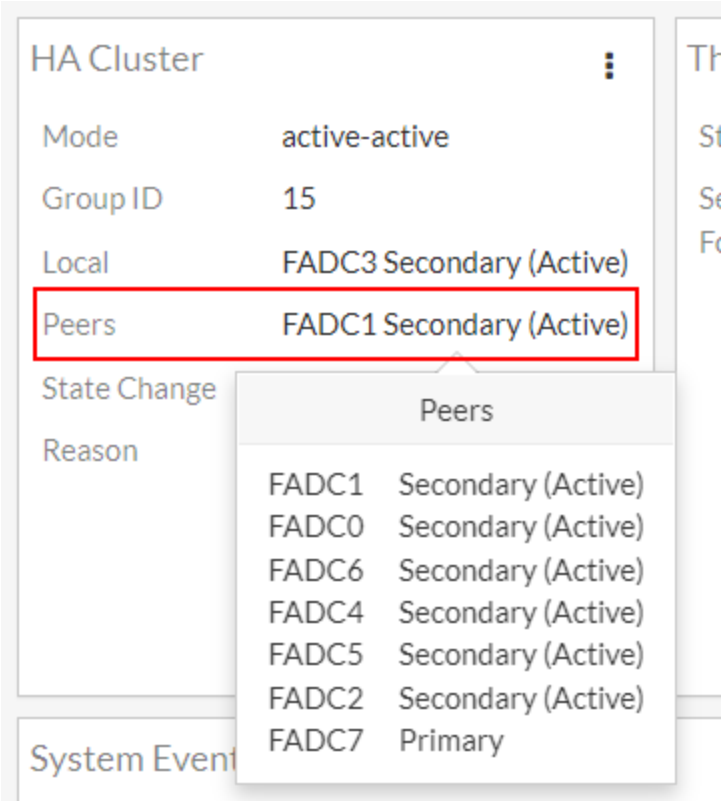
Column	Description
Host Name	The Host Name of the member node.
State	Indicates the HA status of the node and whether it is elected as the HA Primary or Secondary nodes.
Serial Number	The Serial Number of the FortiADC devices in the HA cluster.
Node ID	The Node ID of the member node.
IP Address	The IP address of the member node.
Config Source	Indicates whether the member node is the source of configurations for HA synchronization. This is determined by the **Config Priority** set in the HA setting. When the configuration priority values of the nodes are different, the configuration of the device with the lower configuration priority will prevail.

HA monitor enhancements to better display 8 member nodes

For this enhancement, several GUI improvements have been implemented to extend the HA monitoring capabilities to accommodate the increased cluster capacity.

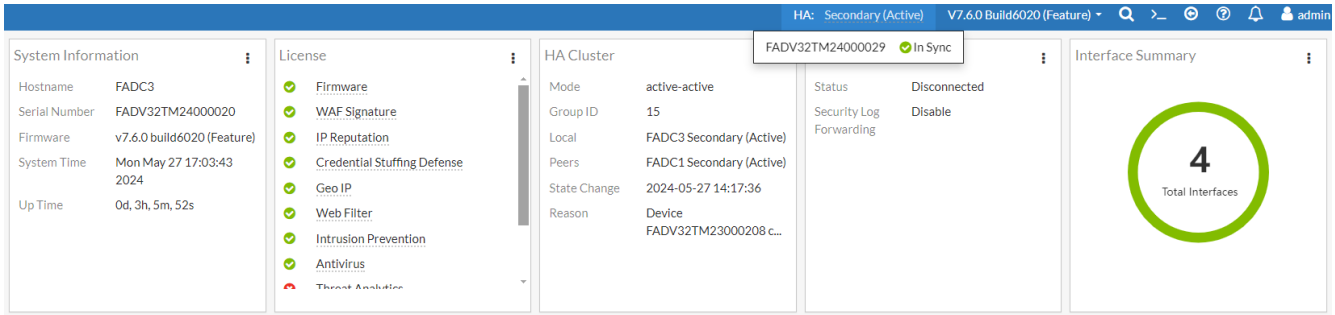
HA Cluster Dashboard widget pop-up displays the list of Peer nodes that are part of the HA cluster and their statuses

When you hover over the **Peers** field of the HA Cluster widget, it displays the list of Peer nodes that are part of the HA cluster and their statuses.

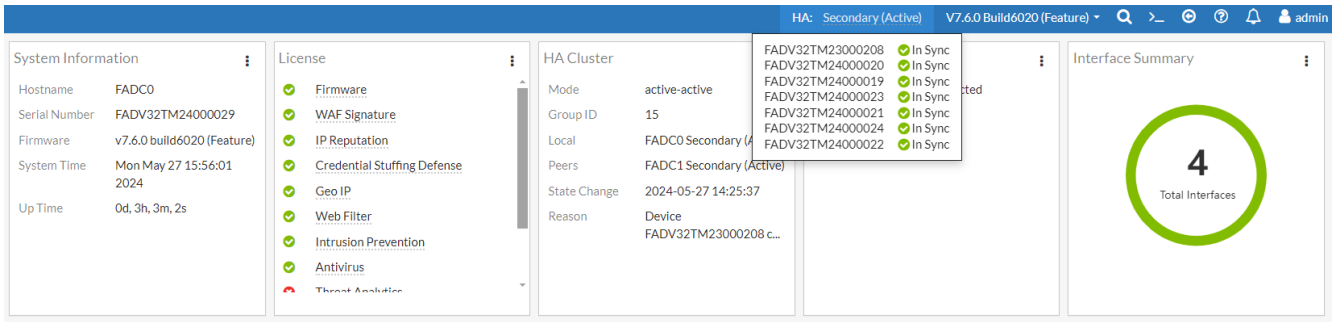


Top Navigation bar displays the HA status of the local node and the status of synchronized devices

In this example, you will see that the local node FADC3 is currently In Sync with another device with the serial number FADV32TM24000029. This means that the local node is not the configuration source, the device that it is currently synchronized to is the configuration source.

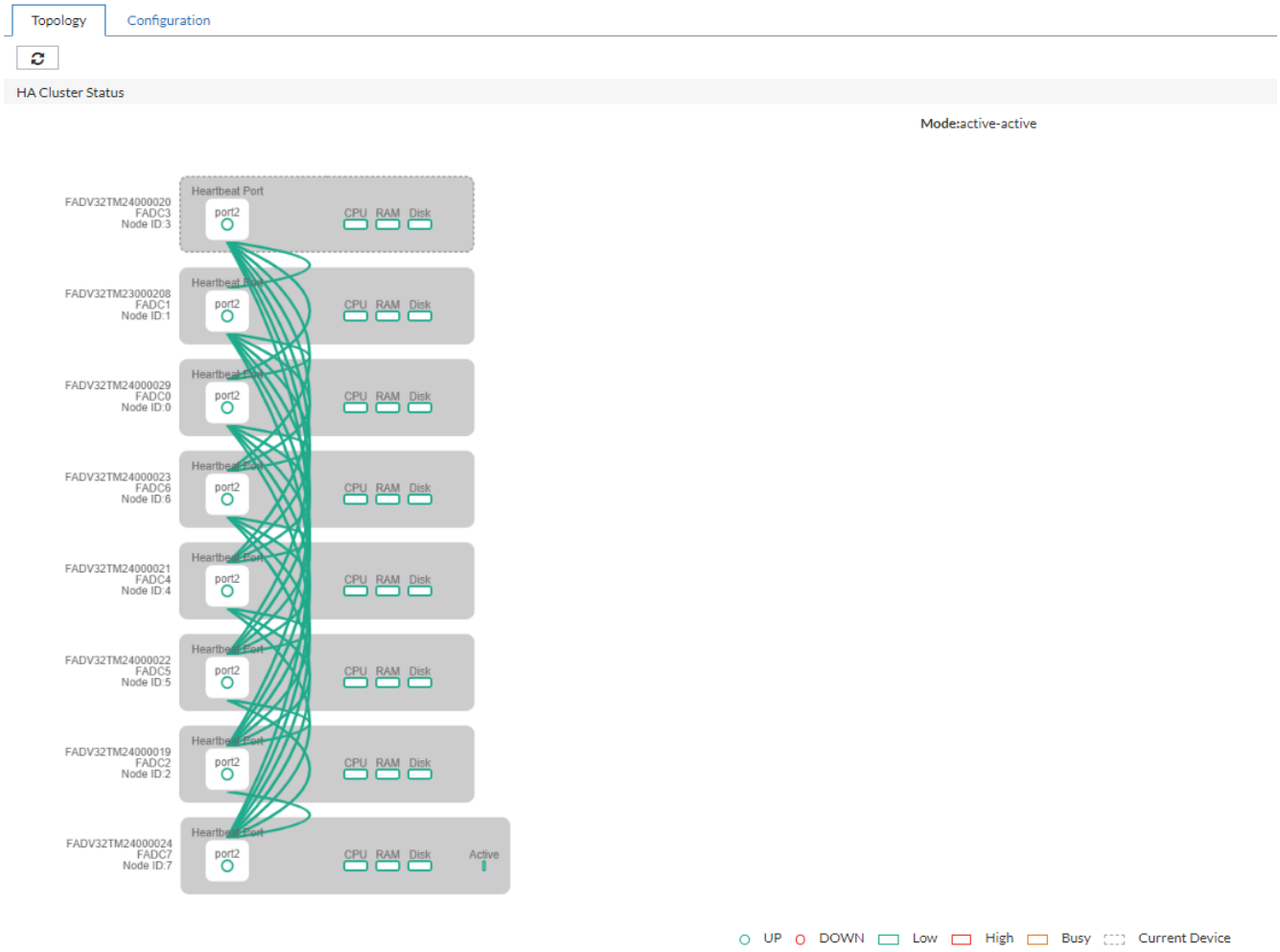


Below is what is displayed if your local node is the configuration source. You will see the list of devices that are currently synchronized to its HA configuration settings, including node FADC3 (you can verify this with the Serial Number from the System Information widget).

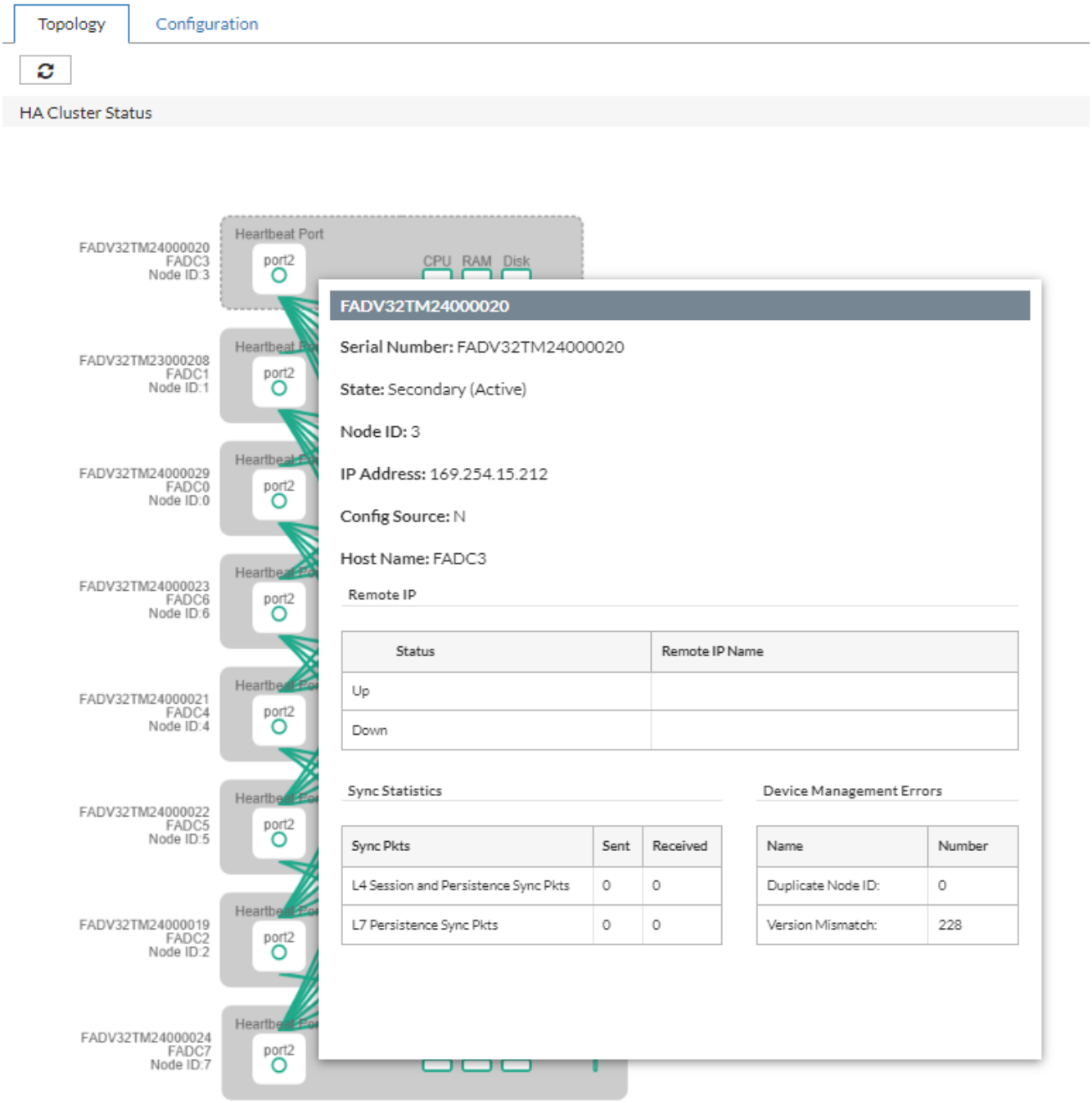


New Topology page that displays the HA cluster topology and detailed device information

To view the HA cluster topology, navigate to **System > High Availability > Topology**.



Hover over any device to drill down to the detailed information about the HA cluster member.


Topology
Configuration
HA Cluster Status
Heartbeat Port
port2
CPU
RAM
Disk
FADV32TM24000020
FADC3
Node ID:3
FADV32TM23000208
FADC1
Node ID:1
FADV32TM24000029
FADC0
Node ID:0
FADV32TM24000023
FADC6
Node ID:6
FADV32TM24000021
FADC4
Node ID:4
FADV32TM24000022
FADC5
Node ID:5
FADV32TM24000019
FADC2
Node ID:2
FADV32TM24000024
FADC7
Node ID:7
FADV32TM24000020
Serial Number: FADV32TM24000020
State: Secondary (Active)
Node ID: 3
IP Address: 169.254.15.212
Config Source: N
Host Name: FADC3
Remote IP
Status
Remote IP Name
Up
Down
Sync Statistics
Sync Pkts
Sent
Received
L4 Session and Persistence Sync Pkts
0
0
L7 Persistence Sync Pkts
0
0
Device Management Errors
Name
Number
Duplicate Node ID:
0
Version Mismatch:
228

HA management interface network options via CLI

FortiADC introduces new CLI options in the `config system ha` command to allow you to customize your choice of network type as either IPVLAN or MACVLAN to create your HA management interface.

This information is also available in the FortiADC 7.6.0 CLI Reference Guide:

- config system ha

Previously, the HA management interface is created only through IPVLAN, which references the user-specified Management Interface and makes the MAC address of the IPVLAN the actual HA management port; this results in the HA management interface to be identical to that of the underlying interface (port1). When the HA management interface shares the same MAC address as the underlying interface, it can result in a split-brain condition where HA members can no longer communicate with each other to exchange monitoring information because all nodes have presumed they are the primary due to using the same set of virtual MAC addresses for all ports. Now, with this new feature, you can use MACVLAN to create the HA management interface so that its MAC address can differ from its underlying interface.

New CLI options in `config system ha`

```
config system ha
   set mgmt-status enable
   set mgmt-network-type {ipvlan|macvlan}
   set mgmt-ip <IP address>
   set mgmt-mac-addr <MAC address>
end
```

`mgmt-network-type`	Select the HA management network to use: • ipvlan — IPVLAN is the default management network option if you are setting up the HA cluster for all virtualization platforms. The management network also defaults to IPVLAN if the FortiADC HA is upgraded from a previous version. • macvlan — Using the MACVLAN management network will allow the HA management interface to have its own MAC so there will be no MAC address conflicts. MACVLAN is the default management network option if you are setting up the HA cluster for hardware.
`mgmt-ip`	If you select IPVLAN as the management network, then specify a management IP address.
`mgmt-mac-addr`	If you select MACVLAN as the management network, then specify the management MAC address.

Example:

```
config system ha
  set mode active-passive
  set hbdev port2
  set group-id 27
  set mgmt-status enable
```

```
    set mgmt-interface port1
    set mgmt-ip 10.106.220.25/23
    set mgmt-ip-allowaccess https ping ssh http
    set mgmt-network-type macvlan
    set mgmt-mac-addr 5a:06:2e:e7:d4:ad
  end
```

Virtual MAC address option as interface in Active-Passive HA via CLI

Through the `config system ha` CLI command, you now have the option to enable or disable the virtual MAC address as the interface for all nodes in an HA Active-Passive cluster.

This information is also available in the FortiADC 7.6.0 CLI Reference Guide:

- config system ha

Previously, by default, the primary node in an HA Active-Passive cluster uses a virtual MAC address for all its network interfaces in order to smoothly transfer this virtual MAC address and traffic to the next selected primary node when failover occurs. With this new feature, you now have the option to enable or disable applying the virtual MAC address.

If the virtual MAC address option is disabled, the interfaces of the primary node in an HA Active-Passive cluster would use its real physical MAC addresses. And when failover occurs, the next primary node will use gratuitous ARP to announce the new real MAC addresses to process the traffic. Consequently, this allows the HA management interface of the secondary node to remain accessible when split-brain occurs due to interfaces being able to use their real MAC addresses instead sharing the same set of virtual MAC addresses for all interfaces.

Note: This option is not supported in Public Cloud and Hyper-V platforms.

New CLI options in `config system ha`

```
config system ha
   set mode active-passive
   set virtual-mac {enable|disable}
end
```

`virtual-mac`	Enable or disable using the virtual MAC addresses for the interfaces in HA Active-Passive mode. This option is enabled by default. When enabled, the primary node will use the virtual MAC addresses for its interfaces. When disabled, the primary node's interface will not use the virtual MAC addresses, and instead use its real physical MAC addresses. **Note**: Changing the status of the virtual-mac setting will trigger all nodes in the HA cluster to switch to INIT status and a new primary will be re-elected.

Example:

```
config system ha
  set mode active-passive
  set hbdev port2
  set group-id 27
  set virtual-mac disable
end
```

New and enhanced CLI commands to force HA nodes into standby mode

For certain troubleshooting, maintenance, or testing scenarios, the ability to trigger HA failover manually can be useful. For this purpose, a new CLI command `execute ha force failover-standby` is introduced to apply to HA Active-Passive and Active-Active modes only. And the existing `execute ha force standby` command has been enhanced to apply the comparable functionality for HA Active-Active-VRRP.

This information is also available in the FortiADC 7.6.0 CLI Reference Guide:

- execute ha force failover-standby
- execute ha force standby

New `execute ha force failover-standby`:

execute ha force failover-standby {set|unset}

execute ha force failover-standby status

`set`	Enables the failover-standby to force the local node of the HA Active-Passive or Active-Active cluster into standby status.
`unset`	Disables the failover-standby command to allow local node of the HA Active-Passive or Active-Active cluster to become active again.
`status`	Allows you to view whether the failover-standby has been set or unset.

Example:

```
(P) FADC # execute ha force failover-standby set
HA failover-standby is set.
(S) FADC # execute ha force failover-standby status
HA failover-standby is set.
(S) FADC # execute ha force failover-standby unset
HA failover-standby is unset.
(P) FADC # execute ha force failover-standby status
HA failover-standby is unset.
```

Enhanced `execute ha force standby`

```
execute ha force standby {traffic-group <traffic group name>|default} {set|unset}
execute ha force standby {traffic-group <traffic group name>|default} status
```

`traffic-group <traffic group name>`	If the **traffic-group** option is selected, then forced standby will apply to the specified traffic group.
`default`	If the **default** option is selected, then forced standby will apply to the local device traffic group.
`set`	Enables forced standby to apply to the selected traffic group option. Once enabled, all traffic in the traffic group will be taken over.

`unset`	Disables the forced standby command to allow the selected traffic group to become active again.
`status`	Allows you to view whether the forced standby command has been set or unset for the selected traffic group.

Example:

```
FADC # execute ha force standby traffic-group default set
This operation will make traffic group on this device force to standby, all traffic in this
traffic group will be taken over!
Do you want to continue? (y/n)y
HA force standby traffic-group of default is set.
FADC # execute ha force standby traffic-group default status
HA force standby traffic-group of default is set.
FADC # execute ha force standby traffic-group default unset
HA force standby traffic-group of default is unset.
```

Administrator

RADIUS and TACACS+ VDOM Override Support 7.6.5 on page 101

FortiADC 7.6.5 introduces support for dynamically determining an administrator's **VDOM scope** during **RADIUS** or **TACACS+** authentication, based on attributes returned by the remote authentication server. This capability is enabled through a new CLI option, `vdom-override`, under `config system admin`.

RADIUS and TACACS+ VDOM Override Support - 7.6.5

FortiADC 7.6.5 introduces support for dynamically determining an administrator's **VDOM scope** during **RADIUS** or **TACACS+** authentication, based on attributes returned by the remote authentication server. This capability is enabled through a new CLI option, `vdom-override`, under `config system admin`.

When configured, FortiADC evaluates VDOM information returned by the RADIUS or TACACS+ server and restricts the administrator's session access to the specified VDOMs. This allows administrators to centrally manage VDOM access on the authentication server rather than relying solely on locally configured administrator settings.

Configuration updates

FortiADC 7.6.5 introduces a new CLI-only option, `vdom-override`, under `config system admin` that applies **only to administrator accounts configured to authenticate using RADIUS or TACACS+**.

```
config system admin
    edit <admin_name>
        set auth-type {radius|tacacs+}
        set vdom-override {enable|disable}
    next
end
```

When enabled, `vdom-override` causes FortiADC to evaluate VDOM information returned by the remote authentication server during login and restrict the administrator's session access accordingly.

This option is not applicable to locally authenticated administrator accounts and must be explicitly enabled for each eligible administrator account.

Behavior and fallback handling

This feature applies only to administrators authenticated through **RADIUS** or **TACACS+**.

If the authentication server returns invalid, malformed, or non-existent VDOM values:

- The administrator login succeeds using the locally configured settings
- A system log entry is generated to record the unsuccessful override attempt

Limitations

- **VDOM override is supported only for non-global administrator accounts.**
  Global administrators retain access to all VDOMs and cannot have their VDOM scope restricted by attributes returned from a RADIUS or TACACS+ server.
- **VDOM override requires VDOM mode to be enabled on the system.**
  If VDOM mode is disabled, any VDOM information returned by the authentication server is ignored.
- **Two-factor authentication support depends on the authentication method.**
  Two-factor authentication is supported for RADIUS-based administrator authentication but is not supported for TACACS+ authentication.

RADIUS and TACACS+ Access Profile Override via CLI - 7.6.4

You can now override the local access profile of an administrator account during login when authenticating with **RADIUS** or **TACACS+** with the new CLI option `set accprofile-override` under `config system admin`. When enabled, FortiADC applies the access profile returned by the remote server instead of the one configured locally. The Access Profile Override feature is available only in the global context; it does not apply at the VDOM level.

This allows centralized AAA systems (Authentication, Authorization, and Accounting) to dynamically assign administrator roles such as *read-only* or *read-write*. It simplifies role management across multiple FortiADC devices, ensures consistent enforcement of access policies, and reduces administrative overhead.

This information is also available in the FortiADC 7.6.4 CLI Reference:

- config system admin

Key capabilities:

- **RADIUS support** – Applies the access profile specified in the Fortinet vendor-specific attribute (VSA) in the RADIUS server response.
- **TACACS+ support** – Applies the access profile specified in the admin_prof field in the TACACS+ server response.
- **Wildcard administrator support** – Wildcard admins inherit VDOM assignments from the base wildcard admin while applying the session profile returned from the remote server. This is supported for both RADIUS and TACACS+.
- **2FA integration** – Supported for RADIUS; the override is applied after successful two-factor authentication. (Not supported for TACACS+.)
- **Fallback behavior** – If the returned access profile is missing, invalid, or not defined on FortiADC, the administrator continues with the locally configured profile and a log entry is generated.

CLI configuration:

```
config system admin
    edit <name>
        set accprofile-override {enable|disable}
    next
end
```

By default, `accprofile-override` is disabled.

SNMP

Enhanced SNMP Authentication and Encryption on page 104

FortiADC has upgraded its SNMP security by introducing support for the stronger SHA-2 hashing algorithm, combined with AES-256 encryption. This enhancement ensures compliance with modern security standards and significantly reduces the risk of vulnerabilities associated with older protocols like MD5 and SHA-1.

Previously, FortiADC offered SNMP authentication using MD5 or SHA-1, alongside encryption with AES or DES. The addition of SHA-2 and AES-256 now provides more robust protection, aligning with current best practices for secure network management.

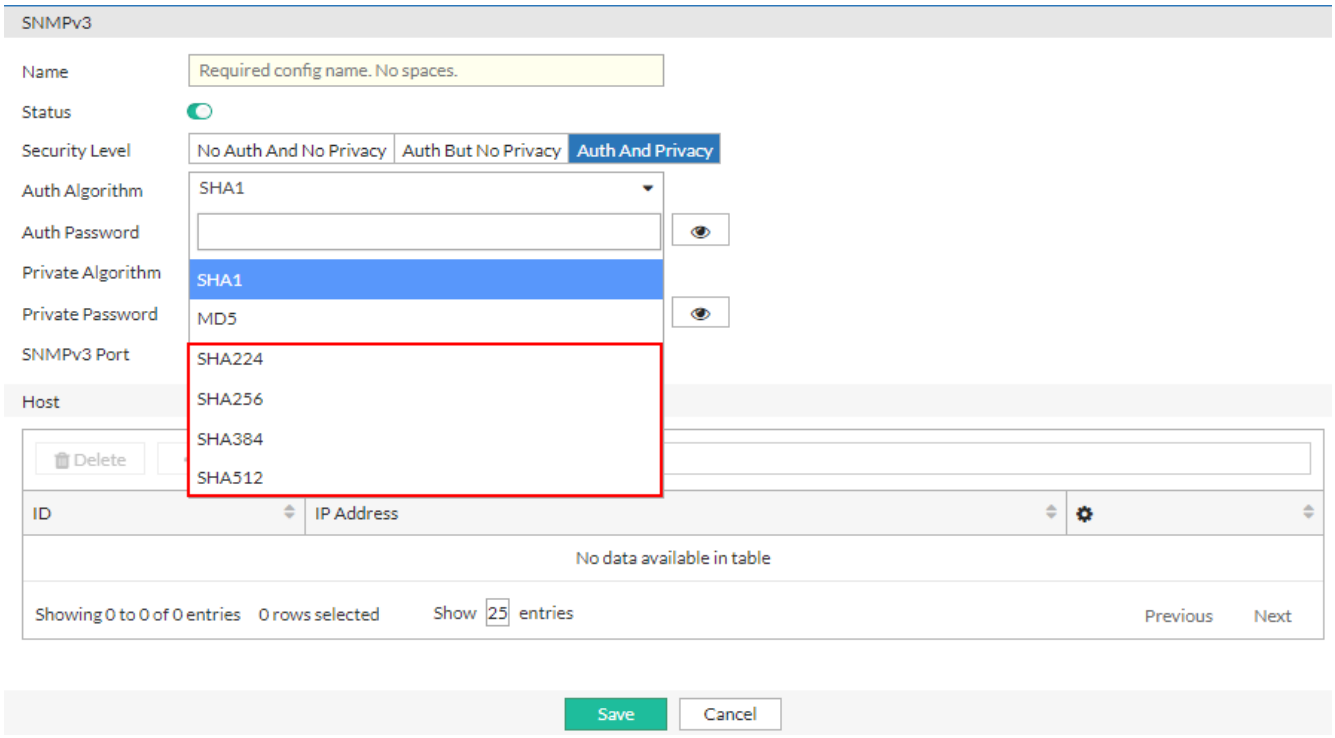
Enhanced SNMP Authentication and Encryption

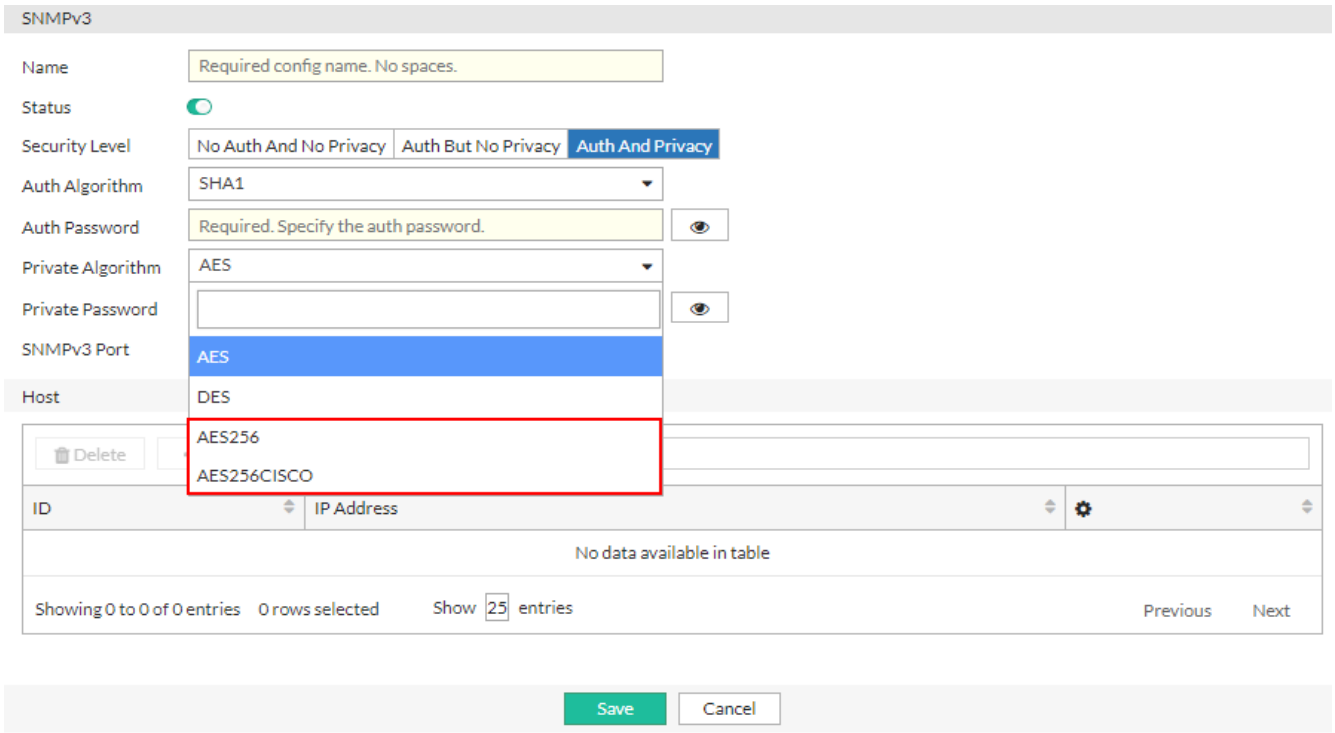
FortiADC has upgraded its SNMP security by introducing support for the stronger SHA-2 hashing algorithm, combined with AES-256 encryption. This enhancement ensures compliance with modern security standards and significantly reduces the risk of vulnerabilities associated with older protocols like MD5 and SHA-1.

Previously, FortiADC offered SNMP authentication using MD5 or SHA-1, alongside encryption with AES or DES. The addition of SHA-2 and AES-256 now provides more robust protection, aligning with current best practices for secure network management.

This information is also available in the FortiADC 7.6 Administration Guide and CLI Reference:

- WAF Adaptive Learning
- config security waf adaptive-learning
- diagnose debug module autolearn





CLI updates in `config system snmp user`:

```
config system snmp user
   edit <name>
      set auth-proto {sha1|md5|sha224|sha256|sha384|sha512}
      set priv-proto {aes|des|aes256|aes256cisco}
```

```
    next
  end
```

Certificate

ACME TLS-ALPN-01 Enhancements on page 108

FortiADC has made the following enhancements in ACME certificate generation:

• **Placeholder certificate generation and removal** — Required to fulfill the ACME TLS-ALPN-01 challenge, FortiADC has streamlined this process through the new **Certificate Group** parameter, allowing FortiADC to manage this process on the backend.

• **Multi-domain support for certificates using SAN** — The **Domain** parameter now allow multiple domains to support certificates generated with Subject Alternative Names (SAN).

ACME TLS-ALPN-01 Enhancements

FortiADC has made the following enhancements in ACME certificate generation:

• **Placeholder certificate generation and removal** — Required to fulfill the ACME TLS-ALPN-01 challenge, FortiADC has streamlined this process through the new **Certificate Group** parameter, allowing FortiADC to manage this process on the backend. For details, see Placeholder certificate generation and removal on page 108.

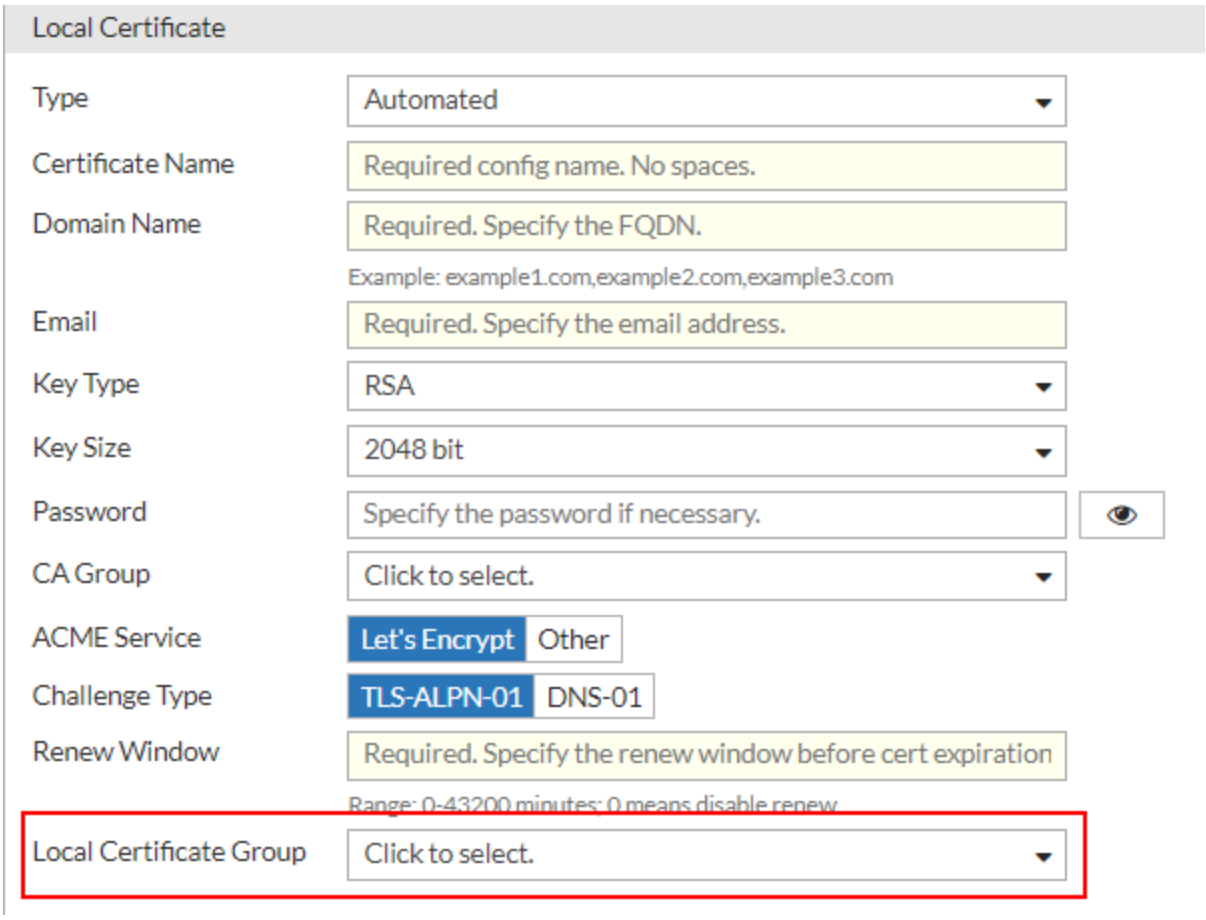
• **Multi-domain support for certificates using SAN** — The **Domain** parameter now allow multiple domains to support certificates generated with Subject Alternative Names (SAN). For details, see Multi-domain support for certificates using SAN on page 109.

This information is also available in the FortiADC 7.6.0 Administration Guide and CLI Reference Guide:

- Importing a local certificate
- execute certificate local import automated

Placeholder certificate generation and removal

Previously, to generate certificates using the TLS-ALPN-01 protocol requires users to setup a placeholder certificate for the VS, which is then directly replaced once the authentication certificate is generated. FortiADC now offers a way to streamline this process with the new **Local Certificate Group** parameter that allow FortiADC to manage the placeholder certificate generation and removal process on the backend. The option to use a placeholder certificate to fulfill the TLS-ALPN-01 challenge is still supported.



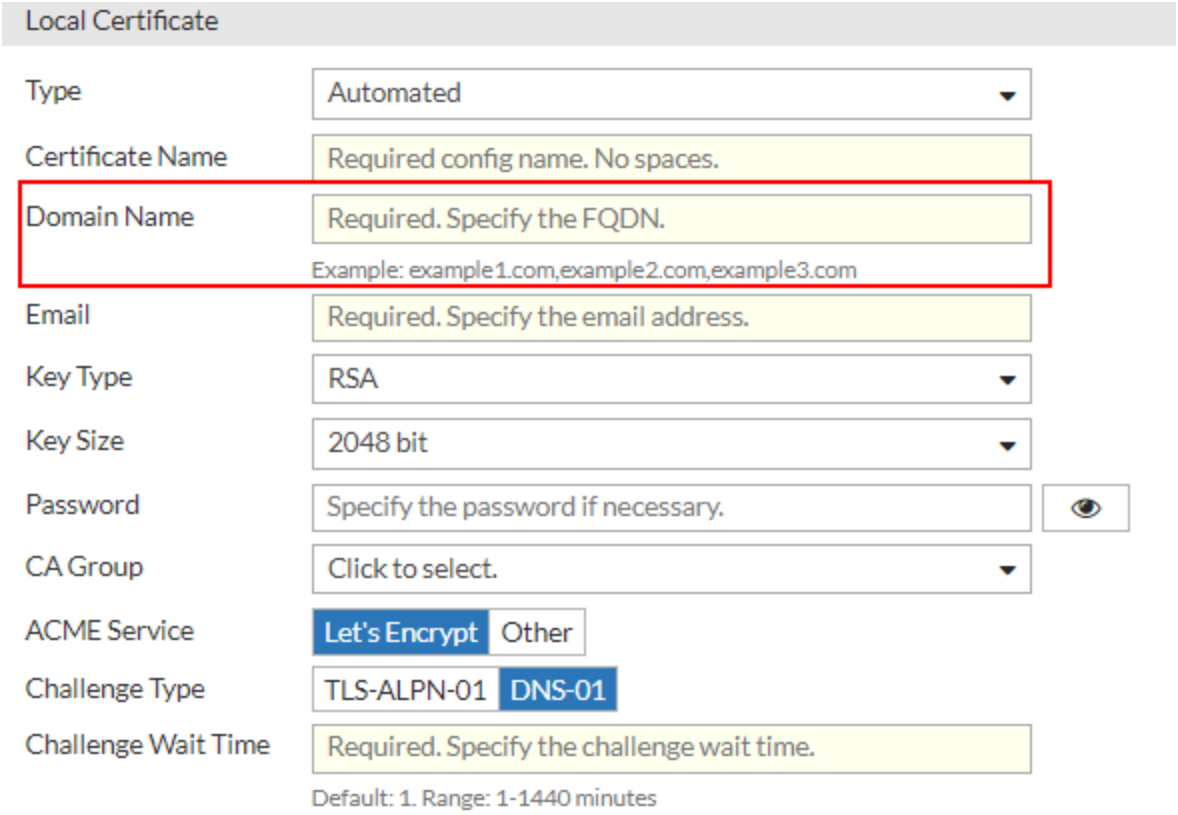
CLI update in `execute certificate local import automated`:

```
execute certificate local import automated <cert_name> <domain> <email> <key_type>
     {RSA|ECDSA} <key_size> {<key_size>|<curve_name>} <password> <server_url> <ca_group>
     <challenge_type> {tls-alpn-01|dns-01} <renew_win> <challenge_wait> <cert_group>
```

`<cert_group>`	Specify the **cert_group** if the **challenge_type** is **tls-alpn-01**. Select a local certificate group to allow FortiADC to manage the placeholder certificate generation and removal process required for the TLS-ALPN-01 protocol on the backend. This is optional. Alternatively, you can prepare a placeholder certificate to fulfill the TLS-ALPN-01 challenge instead.

Multi-domain support for certificates using SAN

To align with standard practice for certificates generated with Subject Alternative Names (SAN), FortiADC now supports multiple domains in the Domain parameter. In a multi-domain configuration, the first domain will be set as the Common Name (CN) and subsequent domains as SANs. Note that for the TLS-ALPN-01 authentication type, all domains should be resolved to the IP address set by the VS.



CLI update in `execute certificate local import automated`:

```
execute certificate local import automated <cert_name> <domain> <email> <key_type>
     {RSA|ECDSA} <key_size> {<key_size>|<curve_name>} <password> <server_url> <ca_group>
     <challenge_type> {tls-alpn-01|dns-01} <renew_win> <challenge_wait> <cert_group>
```

`<domain>`	Specify the web server domain(s) to be protected by the certificate. When inputting multiple domains, separate each domain using a comma with no additional spaces. **Note**: If the **challenge_type** is **tls-alpn-01**, all domains should be resolved to the IP address set by the VS.

To support the multi-domain enhancement, the DNS-01 challenge fulfillment process has also updated to handle the DNS-01 challenge information for multiple domains. The updated steps are outlined in the following.

Fulfilling the ACME DNS-01 challenge

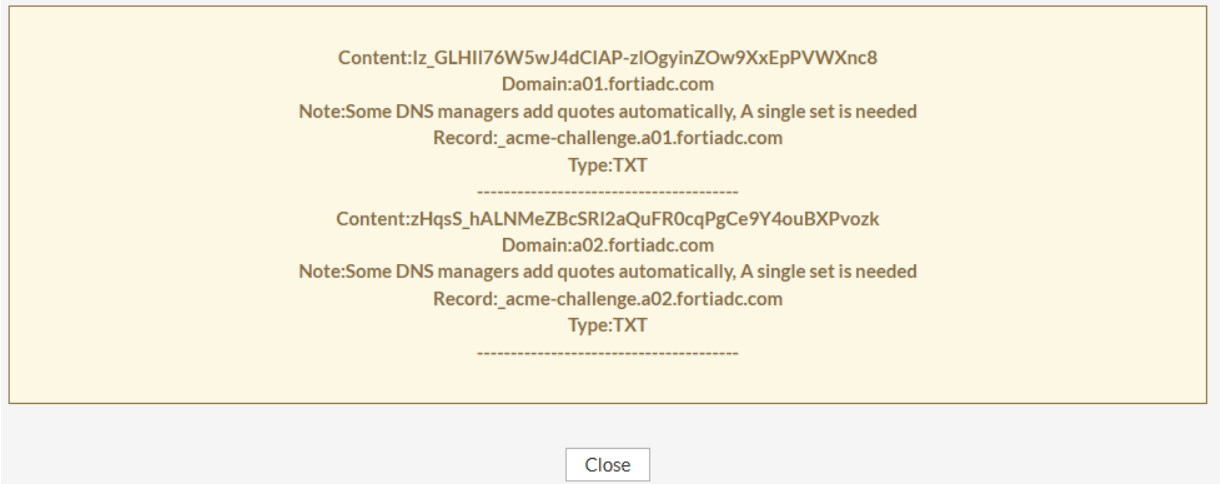
The DNS-01 challenge asks you to prove that you control the DNS for your domain name(s) by putting a specific value in a TXT record under that domain name.

After you have saved your automated local certificate configuration, the ACME DNS challenge information is generated. With this information, you will configure your Public DNS Service to create the TXT record.

Certificates generated by the ACME DNS-01 challenge cannot be renewed automatically. Please manually renew the certificate before it expires.

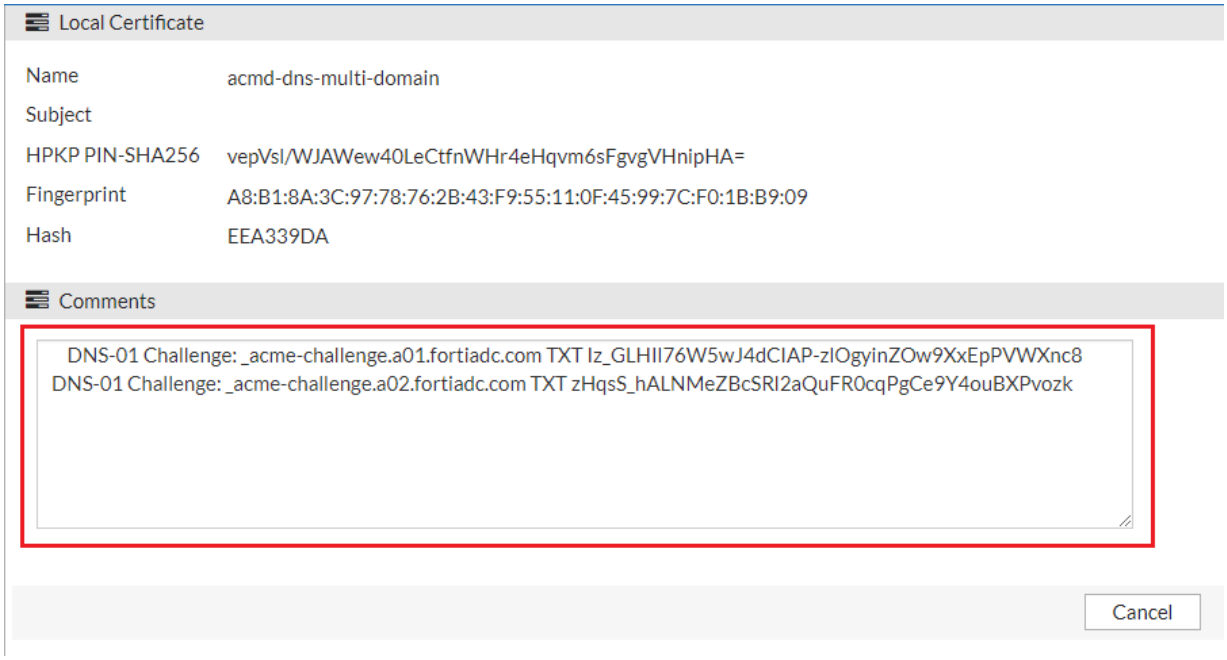
To add the record the DNS challenge information to the Public DNS Service:

1. Obtain the ACME DNS challenge information using either of the following methods.
   - After you save your automated local certificate configuration, you will be shown the challenge information for each domain. Save this information for use later.



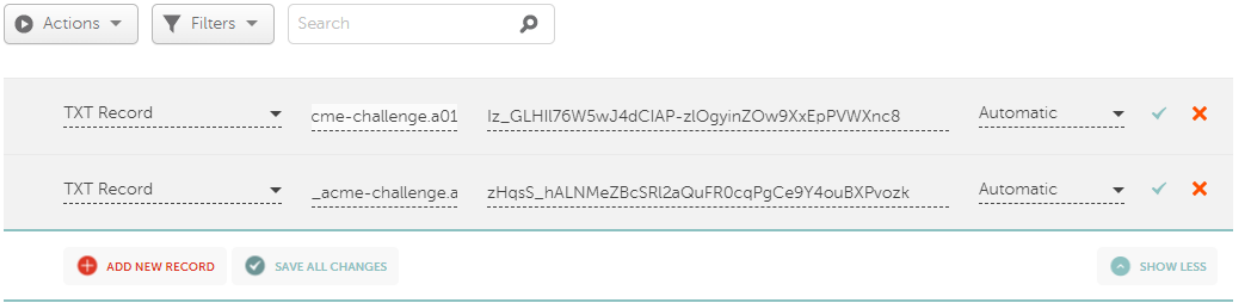
   Note: If using multiple domains, it is recommended you do not exceed 10 domains, as the excessive number of records cannot be displayed in the **Local Certificate** page. Which means that the DNS challenge information will only be available in this pop-up dialog, immediately after saving the certificate configuration.

   - In the **Local Certificate** page, locate the local certificate record and click the (View icon) to see the details.



Note: If the DNS challenge information exceeds the 520 character limit, then no records will be displayed in the Comments box. If there are no records in the Comments box, you can view it in the event log.

2. Login to your DNS service provider and go to your DNS Domain management page.
3. Add a record for each domain and input the challenge information into the corresponding fields.



4. Save the changes.
   The DNS configuration changes may take several minutes to take effect.

The ACME provider will then query the DNS system for that record to find a match. If there is a match, the ACME certificate passes validation (certificate status will progress from Pending → OK). However, if the record is not found within the specified challenge wait time then the certificate validation fails (certificate status is Fail).

If the certificate validation fails, then you will need to delete the record and import a new automated local certificate to try again.

It is recommended to set a longer challenge wait time to allow enough time for the DNS configuration changes to take effect. If the DNS configuration changes has not taken effect at the time the ACME provider queries the DNS system for the TXT record, then the validation will fail. Various factors may influence the speed of the DNS (such as the DNS service provider, network speed, network traffic), so the DNS configuration changes may take as long as 20 minutes to take effect.

Network

The FortiADC 7.6 release includes new features and enhancements to support the **Network**:

IPv6 Support for HA Management Interface 7.6.3 on page 114

FortiADC now supports configuration of an IPv6 address on the HA management interface, allowing administrators to access the system using either IPv4 or IPv6. This enhancement expands flexibility for deployments in dual-stack or IPv6-only environments.

IPv6 Router Advertisement Support via CLI 7.6.3 on page 115

This enhancement adds the new CLI command `config router ipv6-ra` to support the Router Advertisement (RA) functionality, allowing FortiADC to act as an IPv6 router and broadcast configuration information to hosts on the network. This feature helps fulfill RIPE-772 compliance requirements and supports automatic IPv6 address configuration (SLAAC) as well as DNS configuration via RA options.

IPsec-Based Authentication and Encryption for OSPFv3 via CLI 7.6.3 on page 119

FortiADC 7.6.3 introduces CLI-based support for IPsec authentication and encryption in OSPFv3, enabling secure exchange of dynamic routing information over IPv6. This enhancement helps meet regulatory and security requirements in environments where OSPFv3 packet confidentiality and integrity are mandatory.

Link Layer Discovery Protocol (LLDP) Support 7.6.2 on page 122

FortiADC now supports the Link Layer Discovery Protocol (LLDP), a vendor-neutral Layer 2 protocol standardized by IEEE 802.1AB. LLDP is widely used in enterprise and data center networks for automated topology discovery, link validation, and inventory management. By exchanging device identity, interface capabilities, and system-level information with directly connected peers, LLDP enables accurate network visibility and simplifies the troubleshooting of physical connectivity issues.

Support for OSPF Version 3 (OSPFv3) 7.6.1 on page 129

FortiADC now supports OSPFv3 (Open Shortest Path First version 3), a link-state routing protocol for IPv6 networks that extends OSPFv2's core functionality, including shortest path calculation using the Dijkstra algorithm and maintaining a Link-State Database (LSDB) for network topology management.

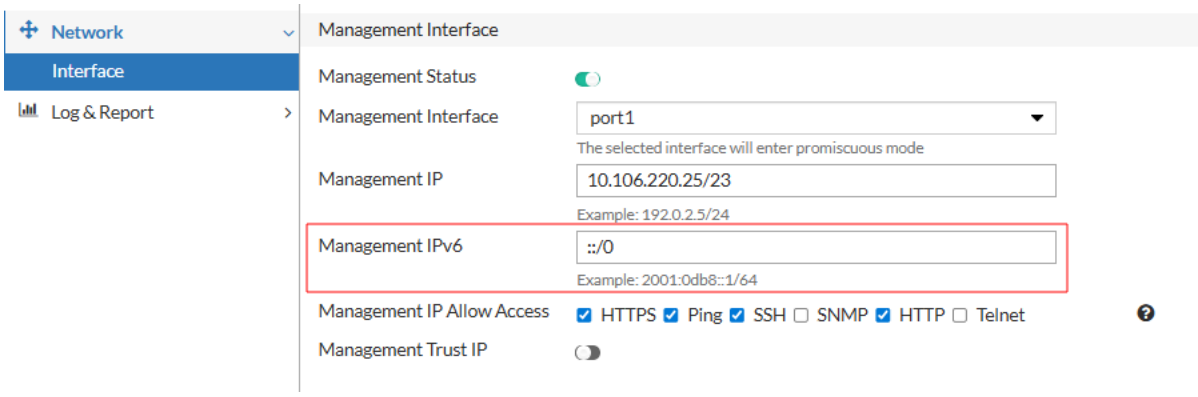
IPv6 Support for HA Management Interface - 7.6.3

FortiADC now supports configuration of an IPv6 address on the HA management interface, allowing administrators to access the system using either IPv4 or IPv6. This enhancement expands flexibility for deployments in dual-stack or IPv6-only environments.

This information is also available in the FortiADC 7.6.3 Administration Guide and CLI Reference:

- Configuring the management interface
- config system ha

Previously, only IPv4 was supported for the HA management interface. In FortiADC 7.6.3, a new **Management IPv6** option has been introduced in both the CLI and GUI, enabling IPv6 access for services such as HTTPS, SSH, SNMP, and PING, as permitted under the **Management IP Allow Access** configuration.



Key behaviors and constraints:

- The Management IPv6 address is **not synchronized** across the HA cluster.
- The IPv6 address must not conflict with any other system configuration, including interfaces, virtual servers, floating IPs, or IP pools.
- When **Management Trust IP** is enabled, you can now add IPv6 addresses to the **Management Trust IP Address List** to control which IPv6 sources are permitted to access the FortiADC through the HA management interface. This is also not synchronized across the HA cluster.

IPv6 Router Advertisement Support via CLI - 7.6.3

This enhancement adds the new CLI command `config router ipv6-ra` to support the Router Advertisement (RA) functionality, allowing FortiADC to act as an IPv6 router and broadcast configuration information to hosts on the network. This feature helps fulfill RIPE-772 compliance requirements and supports automatic IPv6 address configuration (SLAAC) as well as DNS configuration via RA options.

This information is also available in the FortiADC 7.6.3 CLI Reference:

- config router ipv6-ra

FortiADC can now send Router Advertisement messages that include:

- Prefix Information (RFC 4861)
- Recursive DNS Server (RDNSS) options (RFC 8106)
- DNS Search List (DNSSL) options
- Route Information Options (RIO)

RA is configured per interface and is handled by the `radvd` daemon. All configuration is CLI-based. When FortiADC is operating as an HA secondary device, RA advertisement is disabled.

New CLI command: config router ipv6-ra

```
config router ipv6-ra
   config ra-interface
      edit <interface-name>
         set interface <interface-name>
         set send-adv {enable | disable}
         set max-interval <seconds>
         set min-interval <seconds>
         set hop-limit <integer>
         set manage-flag {enable | disable}
         set other-flag {enable | disable}
         set route-pref {high | medium | low}
         set default-life <seconds>
         set reachable-time <milliseconds>
         set retrains-time <milliseconds>
         set link-mtu <integer>
         set adv-rio {enable | disable}
         config prefix-list
            edit <index>
               set prefix6 <prefix/length>
               set onlink-flag {enable | disable}
               set autonomous-flag {enable | disable}
               set preferred-life-time <seconds>
               set valid-life-time <seconds>
            next
         end
         config rdnss
```

```
            edit <index>
               set dns-server <IPv6 address>
               set life-time <seconds>
            next
         end
         config dnssl
            edit <index>
               set domain-name <domain>
               set life-time <seconds>
            next
         end
         config route-list
            edit <index>
               set route <prefix/length>
               set route-pref {high | medium | low}
               set route-life-time <seconds>
            next
         end
      next
   end
end
```

Parameter Descriptions

`config ra-interface` block:

This block defines the global RA parameters for a specific FortiADC interface. These settings control whether RA messages are sent, how often they are transmitted, and what router-level options are included in the advertisement.

Parameter	Description
`interface <interface-name>`	Specifies the physical or logical interface that will send RA messages.
`send-adv {enable | disable}`	Enables or disables RA message transmission and solicitation responses.
`max-interval <seconds>`	Maximum interval between unsolicited multicast RA transmissions.
`min-interval <seconds>`	Minimum interval between unsolicited multicast RA transmissions.
`hop-limit <integer>`	Default Hop Limit to be placed in the IP header of outbound packets.
`manage-flag {enable | disable}`	Indicates whether hosts should use DHCPv6 for address configuration (ManagedAddressConfiguration flag).
`other-flag {enable | disable}`	Indicates whether hosts should use DHCPv6 for other configuration (OtherConfiguration flag).
`route-pref {high | medium | low}`	Preference value for the advertising router.
`default-life <seconds>`	Lifetime associated with the default router.
`reachable-time <milliseconds>`	Time a node assumes a neighbor is reachable after a confirmation.

Parameter	Description
`retrains-time <milliseconds>`	Time between retransmitted Neighbor Solicitation messages.
`link-mtu <integer>`	MTU value to be advertised to hosts on the link.
`adv-rio {enable | disable}`	Enables or disables the Route Information Option (RIO) section.

`config prefix-list` block:

This block allows you to define one or more IPv6 prefixes that will be advertised in RA messages. These prefixes can be used by clients for stateless address autoconfiguration (SLAAC) or for on-link determination.

Parameter	Description
`prefix6 <prefix/length>`	IPv6 prefix to be advertised (e.g., `2001:db8::/64`).
`onlink-flag {enable | disable}`	When enabled, indicates the prefix is on-link.
`autonomous-flag {enable | disable}`	When enabled, allows SLAAC address configuration using this prefix.
`preferred-life-time <seconds>`	Duration the address remains preferred.
`valid-life-time <seconds>`	Duration the prefix is considered valid for on-link determination.

`config rdnss` block:

Use this block to configure Recursive DNS Server (RDNSS) options that advertise one or more DNS servers to hosts via RA messages, as defined in RFC 8106.

Parameter	Description
`dns-server <IPv6 address>`	One or more IPv6 addresses of recursive DNS servers.
`life-time <seconds>`	Time the RDNSS entries are valid for name resolution.

`config dnssl` block:

This block defines DNS Search List (DNSSL) options that inform clients of domain suffixes to use for resolving unqualified domain names.

Parameter	Description
`domain-name <domain>`	One or more domain name suffixes to use for DNS search.
`life-time <seconds>`	Time the DNSSL entries are valid.

`config route-list` block (RIO):

This block defines Route Information Options (RIO), which allow more specific routes to be advertised to hosts beyond the default route.

Parameter	Description
`route <prefix/length>`	IPv6 route prefix to advertise to hosts (e.g., `2001:db8::/96`).
`route-pref {high | medium | low}`	Preference value for the advertised route.
`route-life-time <seconds>`	Lifetime of the advertised route.

IPsec-Based Authentication and Encryption for OSPFv3 via CLI - 7.6.3

FortiADC 7.6.3 introduces CLI-based support for IPsec authentication and encryption in OSPFv3, enabling secure exchange of dynamic routing information over IPv6. This enhancement helps meet regulatory and security requirements in environments where OSPFv3 packet confidentiality and integrity are mandatory.

Administrators can now configure authentication and encryption parameters for OSPFv3 interfaces using the CLI, with support for the following algorithms:

- **Authentication**: md5, sha1, sha256, sha384, sha512
- **Encryption**: null, des, 3des, aes128, aes192, aes256

These parameters are applied on a per-interface basis and ensure that OSPFv3 neighbors authenticate and optionally encrypt routing protocol exchanges. All peers must use matching parameters to establish a valid neighbor relationship.

This information is also available in the FortiADC 7.6.3 CLI Reference:

- config router ospf6
- diagnose debug module ospf6d

CLI update in config router ospf6

```
config router ospf6
   config ipsec-keys
      edit <spi-number>
         set auth-key <string>
         set enc-key <string>
      next
   end
   config ospf6-interface
      edit <interface-name>
         set area-id <area-id>
         set interface <interface-name>
         set authentication {ah | esp}
         set ipsec-auth-alg {md5 | sha1 | sha256 | sha384 | sha512}
         set ipsec-enc-alg {null | des | 3des | aes128 | aes192 | aes256}
      next
   end
end
```

`config ipsec-keys`:

Defines the shared keys used for OSPFv3 IPsec security associations.

Parameter	Description
`edit <spi-number>`	Specifies a unique Security Parameter Index (SPI) used to identify the IPsec SA.
`auth-key <string>`	Specifies the ASCII-encoded authentication key. Length must match the selected

Parameter	Description
	algorithm.
`enc-key <string>`	Specifies the ASCII-encoded encryption key. Length must match the selected algorithm.

`config ospf6-interface`:

Enables IPsec protection for OSPFv3 packets on the specified interface.

Parameter	Description
`authentication {ah | esp}`	Specifies the type of IPsec protection to apply to OSPFv3 packets. Use `ah` (Authentication Header) to enable authentication only, or `esp` (Encapsulating Security Payload) to enable both authentication and encryption.
`ipsec-auth-alg {md5 | sha1 | sha256 | sha384 | sha512}`	Defines the algorithm used for authenticating OSPFv3 packets. The selected algorithm must match the key length and type configured under `ipsec-keys`. All OSPFv3 peers must use the same authentication algorithm to form neighbor relationships.
`ipsec-enc-alg {null | des | 3des | aes128 | aes192 | aes256}`	Defines the algorithm used for encrypting OSPFv3 packets. Use `null` to disable encryption (authentication-only mode). The selected algorithm must match the key configured under `ipsec-keys` and be compatible with all OSPFv3 peers.

Debugging Support

To assist with configuration validation and troubleshooting, FortiADC provides granular debug controls for OSPFv3. You can enable debug output for specific OSPFv3 modules using the following command:

```
diagnose debug module ospf6d <option>
```

Available options include:

- `abr` – ABR behavior
- `asbr` – ASBR functionality
- `border-routers` – Border router processing
- `flooding` – LSA flooding behavior
- `interface` – OSPFv3 interface interactions
- `lsa` – Link State Advertisements
- `message` – OSPFv3 message processing
- `neighbor` – Neighbor relationship formation
- `route` – Routing table calculations
- `spf` – Shortest Path First algorithm behavior
- `zebra` – Communication with the Zebra routing manager

Additional Notes

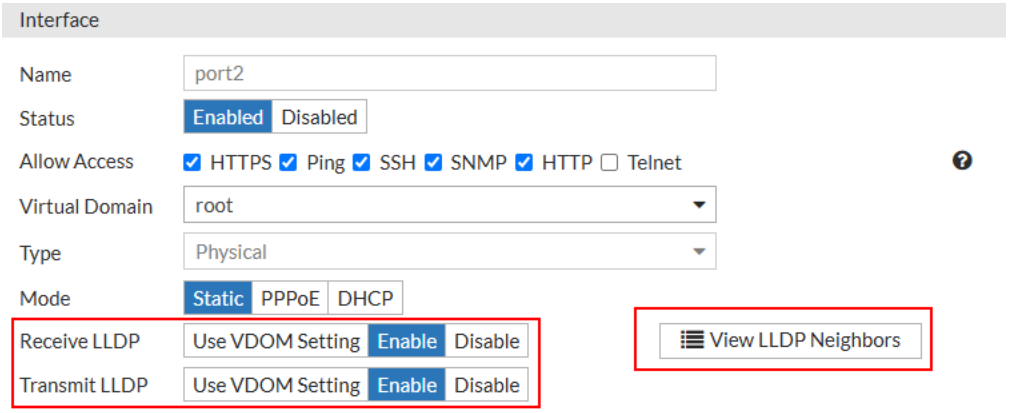
- Supported in VDOM environments. Each VDOM runs a separate OSPFv3 process.
- Compatible with Active-Active and Active-Passive HA. When a device becomes secondary, the OSPFv3 process will stop.

Link Layer Discovery Protocol (LLDP) Support - 7.6.2

FortiADC now supports the Link Layer Discovery Protocol (LLDP), a vendor-neutral Layer 2 protocol standardized by IEEE 802.1AB. LLDP is widely used in enterprise and data center networks for automated topology discovery, link validation, and inventory management. By exchanging device identity, interface capabilities, and system-level information with directly connected peers, LLDP enables accurate network visibility and simplifies the troubleshooting of physical connectivity issues.

This information is also available in the FortiADC 7.6.2 Administration Guide and CLI Reference:

- Configuring network interfaces
- Configuring Link Layer Discovery Protocol (LLDP) on FortiADC
- config system global
- config system setting
- config system interface
- diagnose lldprx neighbor



This enhancement introduces a flexible, three-tier configuration model—**Global**, **VDOM**, and **Interface** levels—allowing network administrators to selectively enable or disable LLDP transmission and reception based on operational, performance, or security requirements.

LLDP is supported on the following interface types:

- Physical interfaces
- Aggregate interfaces (LAG)
- Softswitch interfaces

The LLDP settings apply only to existing interfaces. These settings cannot be configured at the time of creating new interfaces but must be applied after the interface has been created and is available in the system.

Once LLDP is enabled, users can view neighbor information from the FortiADC GUI by clicking the **View LLDP Neighbors** button. For details, see Viewing LLDP Neighbor Information on page 125.

LLDP Configuration Model

FortiADC implements a hierarchical LLDP configuration model to provide flexible control across the system. LLDP settings can be defined at the global level, overridden at the VDOM level, and further refined per interface. This layered approach allows administrators to tailor LLDP behavior based on deployment architecture and operational policy.

Default Behavior

- **Global level**: LLDP is disabled by default.
- **VDOM level**: Defaults to `global`, inheriting the global setting.
- **Interface level**: Defaults to `vdom`, inheriting from the interface's assigned VDOM.

Global-Level Configuration (CLI only)

Sets the default LLDP behavior for all VDOMs.

```
config system global
    set lldp-reception {enable | disable}
    set lldp-transmission {enable | disable}
end
```

`lldp-reception`	• enable — Explicitly enables LLDP packet reception globally, allowing FortiADC to process LLDP frames from all directly connected devices across all VDOMs. • disable — Explicitly disables LLDP packet reception globally, preventing FortiADC from processing any incoming LLDP frames, regardless of VDOM configurations. Global LLDP packet reception is disabled by default.
`lldp-transmission`	• enable — Explicitly enables LLDP packet transmission globally, allowing FortiADC to send LLDP packets containing identity and capability information to all directly connected devices across all VDOMs. • disable — Explicitly disables LLDP packet transmission globally, preventing FortiADC from transmitting any LLDP packets, regardless of VDOM configurations. Global LLDP packet transmission is disabled by default.

VDOM-Level Configuration (CLI only)

Overrides the global setting for a specific VDOM.

```
config system settings
    set lldp-reception {enable | disable | global}
    set lldp-transmission {enable | disable | global}
end
```

`lldp-reception`	• enable — Explicitly enables LLDP packet reception for this VDOM, allowing it to process incoming LLDP frames from directly connected devices. • disable — Explicitly disables LLDP packet reception for this VDOM, preventing it from processing any incoming LLDP frames.

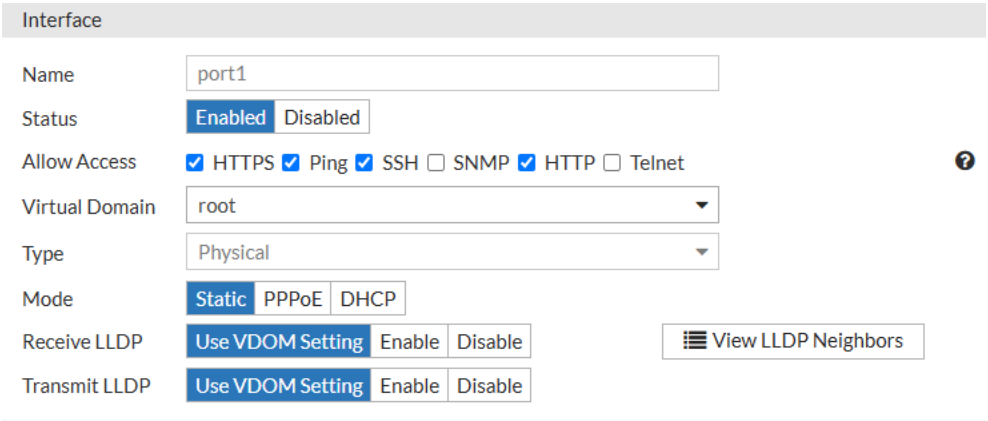
	• global — Inherits the global LLDP reception setting for this VDOM. The reception behavior follows the global configuration. VDOM LLDP packet reception inherits the global configuration by default.
`lldp-transmission`	• enable — Explicitly enables LLDP packet transmission for this VDOM, allowing it to send LLDP packets with identity and capability information to directly connected devices. • disable — Explicitly disables LLDP packet transmission for this VDOM, preventing it from sending any LLDP packets. • global — Inherits the global LLDP transmission setting for this VDOM. The transmission behavior follows the global configuration. VDOM LLDP packet transmission inherits the global configuration by default.

Interface-Level Configuration (GUI and CLI)

Defines LLDP behavior per physical or logical interface.

From the GUI:

1. Navigate to **Network > Interface**.
2. Edit an interface. The LLDP settings are configurable only in existing interfaces.



3. Configure the following LLDP settings:

Setting	Description
Receive LLDP	Controls whether the interface will process incoming LLDP frames from directly connected devices. Select from the following options: • Use VDOM Setting — Inherits the LLDP reception setting from the assigned VDOM configuration. If the VDOM is set to `global`, the interface indirectly inherits the global LLDP reception setting. This is the default option. • Enable — Explicitly enables LLDP packet reception on this interface, regardless of the VDOM or global configuration. • Disable — Explicitly disables LLDP packet reception on this interface. LLDP frames will be ignored and not processed.
Transmit LLDP	Controls whether the interface will transmit LLDP advertisements to directly connected neighbors. Select from the following options: • Use VDOM Setting — Inherits the LLDP transmission setting from the assigned VDOM configuration. If the VDOM is set to `global`, the interface indirectly inherits the global LLDP transmission setting. This is the default option. • Enable — Explicitly enables LLDP transmission on this interface, allowing it to send out LLDP packets containing identity and capability information. • Disable — Explicitly disables LLDP transmission on this interface. No LLDP packets will be generated or sent.

4. Click **Save** to commit the changes.

From the CLI:

```
config system interface
    edit <interface-name>
        set lldp-reception {enable | disable | vdom}
        set lldp-transmission {enable | disable | vdom}
    next
end
```

Viewing LLDP Neighbor Information

Once LLDP reception is enabled on an interface, FortiADC collects and displays LLDP neighbor data received from directly connected network devices. This information is critical for verifying Layer 2 connectivity, diagnosing topology-related issues, and gaining visibility into adjacent network infrastructure.

Viewing LLDP Neighbors per Interface in the GUI

To access LLDP neighbor information from the FortiADC GUI:

1. Navigate to **Network > Interface**.
2. Select an interface that has LLDP reception effectively enabled.

3. Click **View LLDP Neighbors** to view the list of detected LLDP neighbors for that interface.

LLDP Neighbors

Search

Port	MAC Address	Chassis ID	Neighbor Port ID	System Name	System Description	IP Addresses
port1	00:50:56:81:74:39	00:50:56:81:74:39	port1	FGT-TEST	FortiGate-VM64 v7.4.3,build2573,240201 (GA.F)	10.106.212.100
port1	00:50:56:81:77:54	00:50:56:81:77:54	ens160	ubuntu-20	Ubuntu 16.04.3 LTS Linux 4.4.0-210-generic #242-Ubuntu SMP Fri Apr 16 09:57:56 UTC 2021 x86_64	10.106.212.20
port1	00:50:56:81:ff:a9	00:50:56:81:FF:A9	port1	ADC-212-8	FortiADC-VM v7.6.2,build0576,250402 (Interim.M)	
port1	00:50:56:81:07:fe	00:50:56:81:07:FE	port1	ADC-212-1	FortiADC-VM v7.6.2,build0576,250402 (Interim.M)	
port1	00:50:56:81:ff:7a	00:50:56:81:FF:7A	port1	ADC-122-5	FortiADC-VM v7.6.2,build0576,250402 (Interim.M)	

Showing 1 to 5 of 5 entries Show 10 entries Previous 1 Next

Close

Column	Description
Port	The FortiADC interface receiving the LLDP packet.
MAC Address	MAC address of the remote LLDP neighbor device.
Chassis ID	Unique identifier of the neighbor's chassis.
Neighbor Port ID	Identifier of the neighbor's transmitting port.
System Name	Advertised hostname of the neighbor.
System Description	Vendor-specific description or device type of the neighbor.
IP Addresses	IP management addresses advertised by the LLDP neighbor.

Viewing LLDP Neighbors via CLI

FortiADC offers LLDP inspection via CLI in both global and VDOM contexts.

Global Context:

```
diagnose lldprx neighbor summary
```

Displays a summary of all LLDP neighbors discovered system-wide (across all VDOMs).

```
(P) ADC-212-8 # diagnose lldprx neighbor summary
1 port 'port1' 4  mac 00:50:56:81:77:54  chassis 00:50:56:81:77:54 port 00:50:56:81:77:54 system 'ubuntu-20'
2 port 'port1' 4  mac 00:0C:29:1E:DD:72  chassis 00:0C:29:1E:DD:72 port 00:0C:29:1E:DD:72 system 'ubuntu-120'
3 port 'port1' 4  mac 00:50:56:81:87:54  chassis 00:50:56:81:87:54 port 'port1' system 'ADC-212-7'
4 port 'port1' 4  mac 00:50:56:81:FF:7A  chassis 00:50:56:81:FF:7A port 'port1' system 'ADC-122-5'
5 port 'port2' 7  mac 00:50:56:81:A1:20  chassis 00:50:56:81:A6:C2 port 'port2' system 'FGT-LLDP'
6 port 'port2' 7  mac 00:50:56:81:5D:A1  chassis 00:50:56:81:87:54 port 'port2' system 'ADC-212-7'
7 port 'port2' 7  mac 00:50:56:81:5D:FB  chassis 00:50:56:81:77:54 port 00:50:56:81:5D:FB system 'ubuntu-20'
8 port 'port2' 7  mac 00:50:56:81:B9:0F  chassis 00:50:56:81:FF:7A port 'port2' system 'ADC-122-5'
9 port 'port6' 8  mac 00:50:56:81:F4:20  chassis 00:50:56:81:FF:7A port 'port6' system 'ADC-122-5'
10 port 'port10' 9  mac 00:50:56:81:93:4B  chassis 00:50:56:81:87:54 port 'port10' system 'ADC-212-7'
11 port 'port3' 10  mac 00:50:56:81:82:B9  chassis 00:50:56:81:A6:C2 port 'port6' system 'FGT-LLDP'
12 port 'port3' 10  mac 00:50:56:81:0F:96  chassis 00:50:56:81:A6:C2 port 'port3' system 'FGT-LLDP'
13 port 'port3' 10  mac 00:50:56:81:52:CE  chassis 00:50:56:81:87:54 port 'port3' system 'ADC-212-7'
```

VDOM Context:

`diagnose lldprx neighbor summary`

When run from within a VDOM context, this command only lists LLDP neighbors received on interfaces assigned to the current VDOM.

```
(P) ADC-212-8 # config vdom

(P) ADC-212-8 (vdom) # edit test

(P) ADC-212-8 (test) # diagnose lldprx neighbor summary
1 port 'port6' 8  mac 00:50:56:81:F4:20  chassis 00:50:56:81:FF:7A port 'port6' system 'ADC-122-5'
```

Detailed View:

`diagnose lldprx neighbor detail`

Available in both global and VDOM contexts, this command displays additional metadata for each neighbor entry, that includes the following:

```
(P) ADC-212-8 (test) # diagnose lldprx neighbor detail
1 port.index: 8
1 port.txt: port6
1 mac: 00:50:56:81:F4:20
1 created.ticks: 8000
1 created.ago: 27651
```

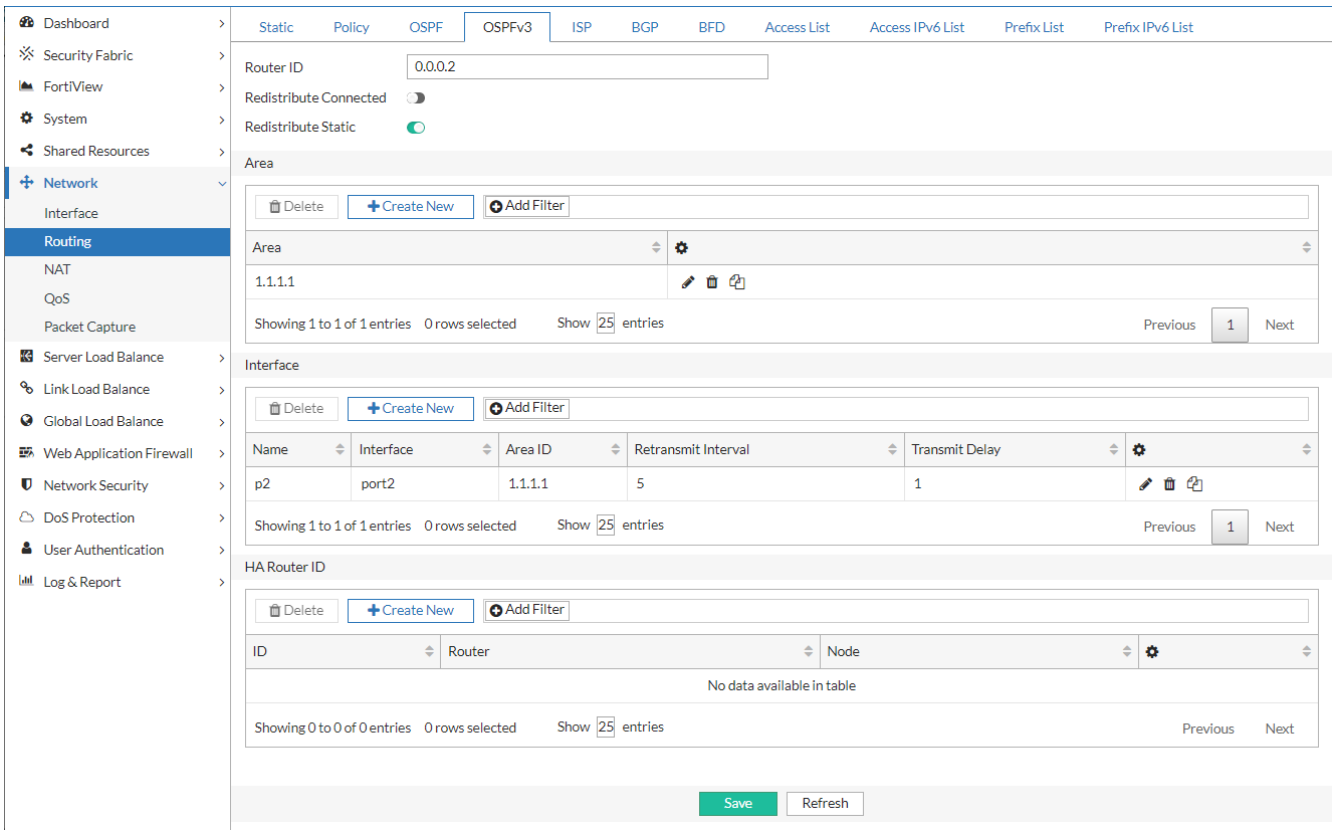
port.index	Internal interface index.
port.txt	Logical interface name.
mac	MAC address of the LLDP neighbor.
created.ticks	System tick count when the entry was created.
created.ago	Ticks elapsed since the neighbor was first learned (age of the entry).

Support for OSPF Version 3 (OSPFv3) - 7.6.1

FortiADC now supports OSPFv3 (Open Shortest Path First version 3), a link-state routing protocol for IPv6 networks that extends OSPFv2's core functionality, including shortest path calculation using the Dijkstra algorithm and maintaining a Link-State Database (LSDB) for network topology management. This initial implementation enables FortiADC to advertise IPv6 virtual server IP addresses, as well as "connected" and "static" IPv6 routes, improving routing efficiency in IPv6 environments.

This information is also available in the FortiADC 7.6.1 Administration Guide and CLI Reference:

- Configuring OSPFv3 routes
- config router ospf6



To configure OSPFv3:

1. Navigate to **Network > Routing**.
2. Click the **OSPFv3** tab.
3. Configure the Router settings.

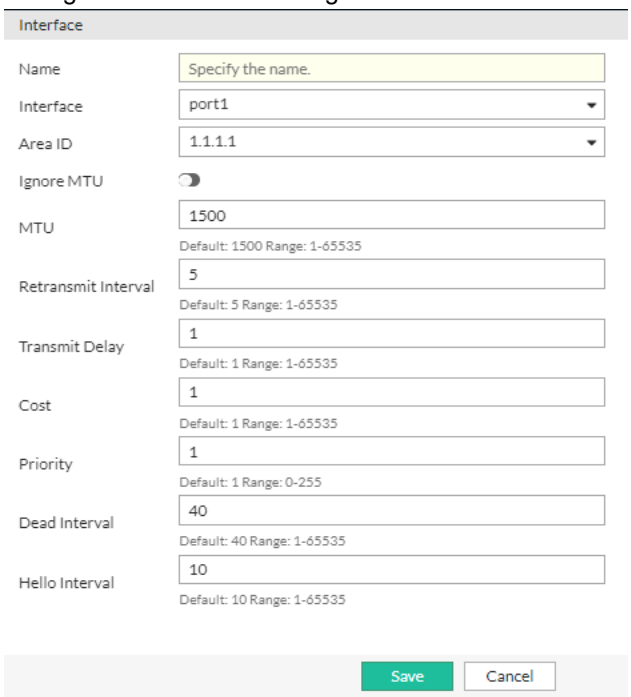
Setting	Description
Router ID	32-bit number that sets the router ID of the OSPF process. The router ID uses dotted decimal notation. The router ID must be an IP address of the router, and it must be unique within the entire OSPFv3 domain to the OSPFv3 speaker.
Redistribute Connected	Enable/disable to redistribute connected routes to OSPFv3, with the metric type and metric set if specified. Redistributed routes are distributed into OSPFv3 as Type-5 External LSAs into links to areas. This is enabled by default.
Redistribute Static	Enable/disable to redistribute static routes to OSPFv3, with the metric type and metric set if specified. Redistributed routes are distributed to OSPFv3 as Type-5 External LSAs into links to areas. This is disabled by default. When enabled, the static routing will be advertised to the OSPFv3 neighbors.

4. Configure the **Area**.
   a. Under the **Area** section, click **Create New** to display the configuration editor.
   The maximum number of supported Area configurations is 1.
   b. Configure the Area Authentication settings:

Setting	Description
Area	32-bit number that identifies the OSPFv3 area. An OSPFv3 area is a smaller part of the larger OSPF network. Areas are used to limit the link-state updates that are sent out. The flooding used for these updates would overwhelm a large network, so it is divided into these smaller areas for manageability.

   c. Click **Save** to commit the changes and exit the dialog.
   The new entry will appear under the Area section.
5. Configure the **Interface**.
   The IPv6 virtual server IP address will be advertised to OSPFv3 neighbors by putting the virtual server interface in the OSPFv3 interface.
   a. Under the Interface section, click **Create New** to display the configuration editor.
   The maximum number of supported Interface configurations is 128.

b. Configure the Interface settings:

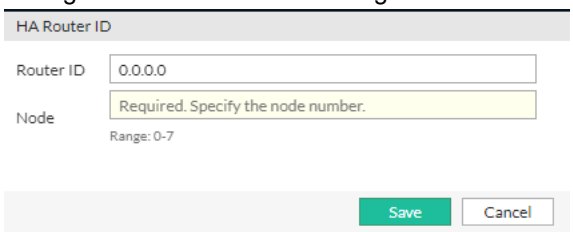


Setting	Description
Name	Configuration name. Valid characters are `A-Z`, `a-z`, `0-9`, `_`, and `-`. No spaces. After you initially save the configuration, you cannot edit the name.
Interface	Select the interface on which to enable OSPFv3.
Area ID	Select the Area ID.
Ignore MTU	Ignores MTU mismatches between OSPFv3 neighbors. This is disabled by default.
MTU	Specify the maximum packet size for transmission. The default value is 1500, and the valid range is 1-65535.
Retransmit Interval	Specify the interval between LSA retransmissions when acknowledgements are not received. The default value is 5 seconds, and the valid range is 1-65535.
Transmit Delay	Additional time added to the LSA age during transmission. The default value is 1 second, and the valid range is 1-65535.
Cost	Set the link cost for the specified interface, with lower metric values being preferred for path selection. This cost value is applied to the router-LSA's metric field and used in the SPF (Shortest Path First) calculation. The default value is 1, and the valid range is 1-65535.

Setting	Description
Priority	The router's priority setting determines its eligibility for DR/BDR (Designated Router/Backup Designated Router) election, with higher values increasing the likelihood of becoming the DR. The router with the highest priority will have a greater chance of being elected as the DR, while setting the priority to 0 makes the router ineligible for DR election. The default value is 1, and the valid range is 0-255.
Dead Interval	The Dead Interval defines the time, in seconds, after which a neighbor is declared down if no Hello packet is received. This interval must be consistent across all routers on a shared network. The default value is 40 seconds, and the valid range is 1-65535.
Hello Interval	The Hello Interval specifies the time between Hello packets sent on a configured interface to maintain neighbor relationships, while the Dead Interval determines the time after which a neighbor is declared down if no Hello packet is received. Both intervals must be consistent across all routers on the same network to ensure proper OSPF operation. The default value is 10 seconds, and the valid range is 1-65535.

c. Click **Save** to commit the changes and exit the dialog.
The new entry will appear under the Interface section.

6. Configure the **HA Router ID**.
   a. Under the HA Router ID section, click **Create New** to display the configuration editor.
   The maximum number of supported HA Router ID configurations is 128.
   b. Configure the HA Router ID settings:



Setting	Description
Router ID	You use the HA Router list configuration in an HA Active-Active deployment. On each HA cluster node, add an HA Router configuration that includes an entry for each cluster node. When the appliance is in standalone mode, it uses the primary OSPFv3 Router ID; when it is in HA mode, it uses the HA Router list ID. Specify a unique 32-bit router ID for the OSPFv3 process in dotted decimal notation. This router ID must be an IP address assigned to the router and must remain unique across the entire OSPFv3 domain to ensure proper identification of the OSPFv3 speaker. In an HA Active-Active-VRRP deployment, each Router ID must be configured independently for each node.

Setting	Description
	In an HA Active-Passive deployment, the router ID defaults to 0.0.0.0 in the secondary node.
Node	Specify the HA Node ID. The valid range is 0-7.

c. Click **Save** to commit the changes and exit the dialog.
The new entry will appear under the HA Router ID section.

7. Click **Save** again to commit the changes to each section of the OSPFv3 configuration.

Repeat the configuration steps for OSPFv3 on the secondary node. OSPFv3 requires at least two OSPFv3-capable devices to form a neighbor relationship and exchange routing information.

New CLI command `config router ospf6`:

```
config router ospf6
   set router-id <router-id>
   set redistribute-connected {enable|disable}
   set redistribute-static {enable|disable}
   config area
      edit <area id>
      next
   end
   config ospf6-interface
      edit <no.>
         set area-id <datasource>
         set interface <datasource>
         set cost <integer>
         set retransmit-interval <integer>
         set transmit-delay <integer>
         set priority <integer>
         set dead-interval <integer>
         set hello-interval <integer>
         set mtu-ignore {enable|disable}
         set mtu <integer>
      next
   end
   config ha-router-id-list
      edit <no.>
         set router-id <router-id>
         set node <integer>
      next
   end
end
```

Troubleshooting

The following CLI commands have been introduced to retrieve information about the OSPFv3 configuration.

Command	Guidelines
`get router ospf6`	Use this command to retrieve basic information of the OSPFv3 configuration.

Command	Guidelines

`get router info6 ospf6 database`
`get router info6 ospf6 interface`
`get router info6 ospf6 neighbor`
`get router info6 ospf6 route`
`get router info6 ospf6 status`
	Use these commands to retrieve detailed information of the OSPFv3 configuration.

`diagnose debug module ospf6d`	Use this command to retrieve detailed debug information for the OSPFv3.

Server Load Balance

The FortiADC 7.6 release includes new features and enhancements in **Server Load Balance**:

L4 Virtual Server Support for ESP Packets (IPsec VPN without NAT-T) 7.6.3 on page 137

FortiADC 7.6.3 adds support for handling ESP (Encapsulating Security Payload) protocol traffic in Layer 4 virtual servers. This enhancement enables FortiADC to support IPsec VPN deployments that operate without **NAT Traversal (NAT-T)**.

Previously, ESP packets were dropped when received by L4 virtual servers because the ESP protocol was not supported in Layer 4 SLB mode. This limitation prevented FortiADC from serving as a load balancer for IPsec VPN endpoints in non-NAT-T environments.

With this enhancement, FortiADC introduces session association logic that maps ESP packets to the IKE (Internet Key Exchange) session established during the initial VPN handshake.

Server-side support for HTTP/2 connections 7.6.1 on page 139

FortiADC now supports backend HTTP/2 connections and multi-connection modes for frontend HTTP/2 and HTTP/3 protocols. The new "backend" option in HTTP2 application profiles enables backend HTTP/2 connections and, when used with the HTTP/3 profile, allows multi-connection handling on the frontend for both HTTP/2 and HTTP/3.

Client address option enabled for HTTP/3 virtual server 7.6.1 on page 149

FortiADC has extended client address support to the HTTP3 Application Profile, enabling the HTTP/3 Virtual Server to connect to backend real servers using the client's IP address.

Scripting groups for predefined HTTP scripts 7.6.1 on page 151

FortiADC has introduced Scripting Groups to systematically categorize and organize its 50 plus predefined HTTP scripts into distinct categories: Authentication, Cookie, Feature, HTTP, IP, Optimization, Routing, SSL, TCP, Utility, and WAF.

New HTTP script for enhanced error handling 7.6.1 on page 158

FortiADC introduces the **HTTP:respond_errorfile()** API, enabling HTTP to respond with predefined error files directly. This API enhances functions such as the Waiting Room by allowing predefined error pages to be served directly without requiring custom content generation.

Waiting Room for virtual queuing through HTTP scripting on page 160

The new FortiADC Waiting Room feature allows you to place visitors in a virtual queue instead of being denied service directly when the server side reaches its configured capacity limit during high-demand situations. In this virtual Waiting Room, visitors can see their position in line and when their turn arrives, they are redirected to the requested page. The Waiting Room feature is supported in Layer 7 HTTP/HTTPS virtual servers through Lua scripting.

AWS autoscaling group discovery on page 166

FortiADC now supports AWS autoscaling group discovery to create dynamic real server pools. In the Real Server Pool configuration, you can now select the **AutoScaleGroup=xxx** tag in the **Service** and **Service Port** fields to automatically trigger the AWS SDN connector to add instances with the autoscaling group tag into the pool. When a scale-in or scale-out event occurs on the AWS side, the SDN connector will update the real server and pool based on the scale-in/scale-out result.

New health check down options on page 169

FortiADC introduces new **Action on Health Check Down** options for the real server pool configuration to enable various actions to be taken on existing connections in the event that a pool member fails a health check.

Note: This feature is only supported in Layer 4 Server Load Balancing.

HTTP3 support for HTTP to HTTPS Redirection on page 170

FortiADC now supports HTTP to HTTPS redirection in HTTP3 virtual servers. Now, in the Virtual Server configuration, you can enable the HTTP Redirect to HTTPS option when an HTTP3 server load-balance profile is selected. Previously, this option was only available for HTTPS and HTTP2 profiles.

L4 Virtual Server Support for ESP Packets (IPsec VPN without NAT-T) - 7.6.3

FortiADC 7.6.3 adds support for handling ESP (Encapsulating Security Payload) protocol traffic in Layer 4 virtual servers. This enhancement enables FortiADC to support IPSec VPN deployments that operate without **NAT Traversal (NAT-T)**.

Previously, ESP packets were dropped when received by L4 virtual servers because the ESP protocol was not supported in Layer 4 SLB mode. This limitation prevented FortiADC from serving as a load balancer for IPsec VPN endpoints in non-NAT-T environments.

With this enhancement, FortiADC introduces session association logic that maps ESP packets to the IKE (Internet Key Exchange) session established during the initial VPN handshake.

This information is also available in the FortiADC 7.6.3 Administration Guide:

- Layer 4 Server Load Balancing for IPsec VPN Hubs in a Hub-and-Spoke Topology

Session Mapping Logic

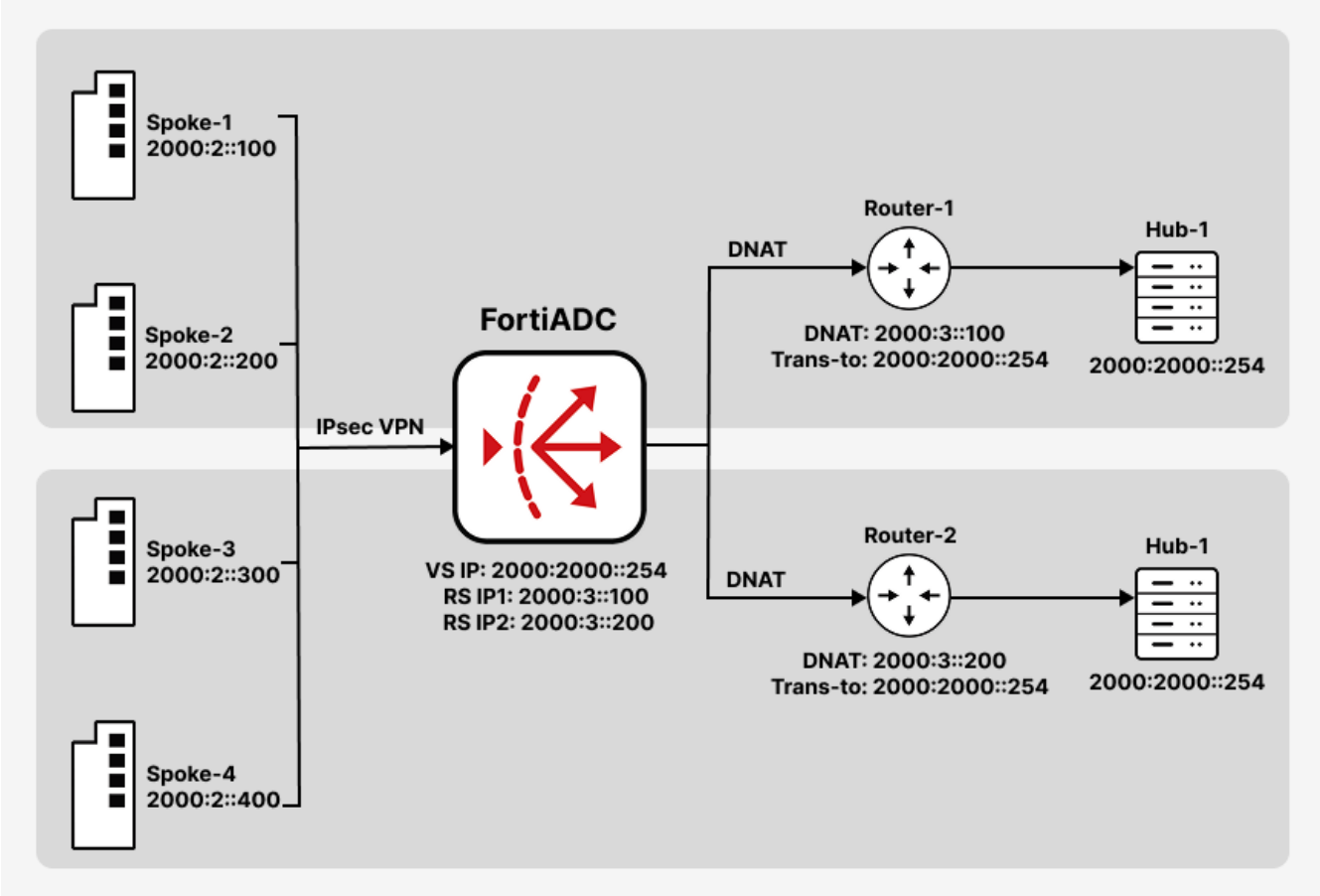
- During IKE negotiation, FortiADC creates a Layer 4 UDP session:

  ```
  <client_ip>:500 → <server_ip>:500 (UDP)
  ```

- When ESP traffic (protocol 50) is received, it lacks port numbers and does not match the existing session using the traditional 3-tuple lookup.
- FortiADC now performs a 5-tuple match by associating incoming ESP packets with the corresponding UDP session (source IP, destination IP, source port 500, destination port 500, protocol UDP).
- ESP traffic is then forwarded to the same real server that was selected during the IKE session.
- The same mechanism applies to both IPv4 and IPv6 traffic.

Use Case

This enhancement allows FortiADC to function as a Layer 4 load balancer for IPsec VPN hubs in **hub-and-spoke topologies**, supporting secure site-to-site VPN tunnels that use ESP directly without relying on NAT-T encapsulation.


Spoke-1
2000:2::100
Spoke-2
2000:2::200
Spoke-3
2000:2::300
Spoke-4
2000:2::400
IPsec VPN
FortiADC
VS IP: 2000:2000::254
RS IP1: 2000:3::100
RS IP2: 2000:3::200
DNAT
Router-1
DNAT: 2000:3::100
Trans-to: 2000:2000::254
Hub-1
2000:2000::254
DNAT
Router-2
DNAT: 2000:3::200
Trans-to: 2000:2000::254
Hub-1
2000:2000::254

Server-side support for HTTP/2 connections - 7.6.1

FortiADC now supports backend HTTP/2 connections and multi-connection modes for frontend HTTP/2 and HTTP/3 protocols. The new "backend" option in HTTP2 application profiles enables backend HTTP/2 connections and, when used with the HTTP/3 profile, allows multi-connection handling on the frontend for both HTTP/2 and HTTP/3. This setup mitigates head-of-line blocking and enables a virtual server to deliver HTTP/3 service with an HTTP/2 backend connection, ensuring true multi-stream concurrent access to backend servers. Note that certain limitations apply to this HTTP/3-based implementation; refer to the Administration Guide for details.

This information is also available in the FortiADC 7.6.1 Administration Guide and CLI Reference:

- Configuring HTTP2 profiles
- Configuring HTTP3 profiles
- Configuring Application profiles
- config load-balance http2-profile
- config load-balance http3-profile
- config load-balance profile

Implementing HTTP/2 Server-Side Connectivity

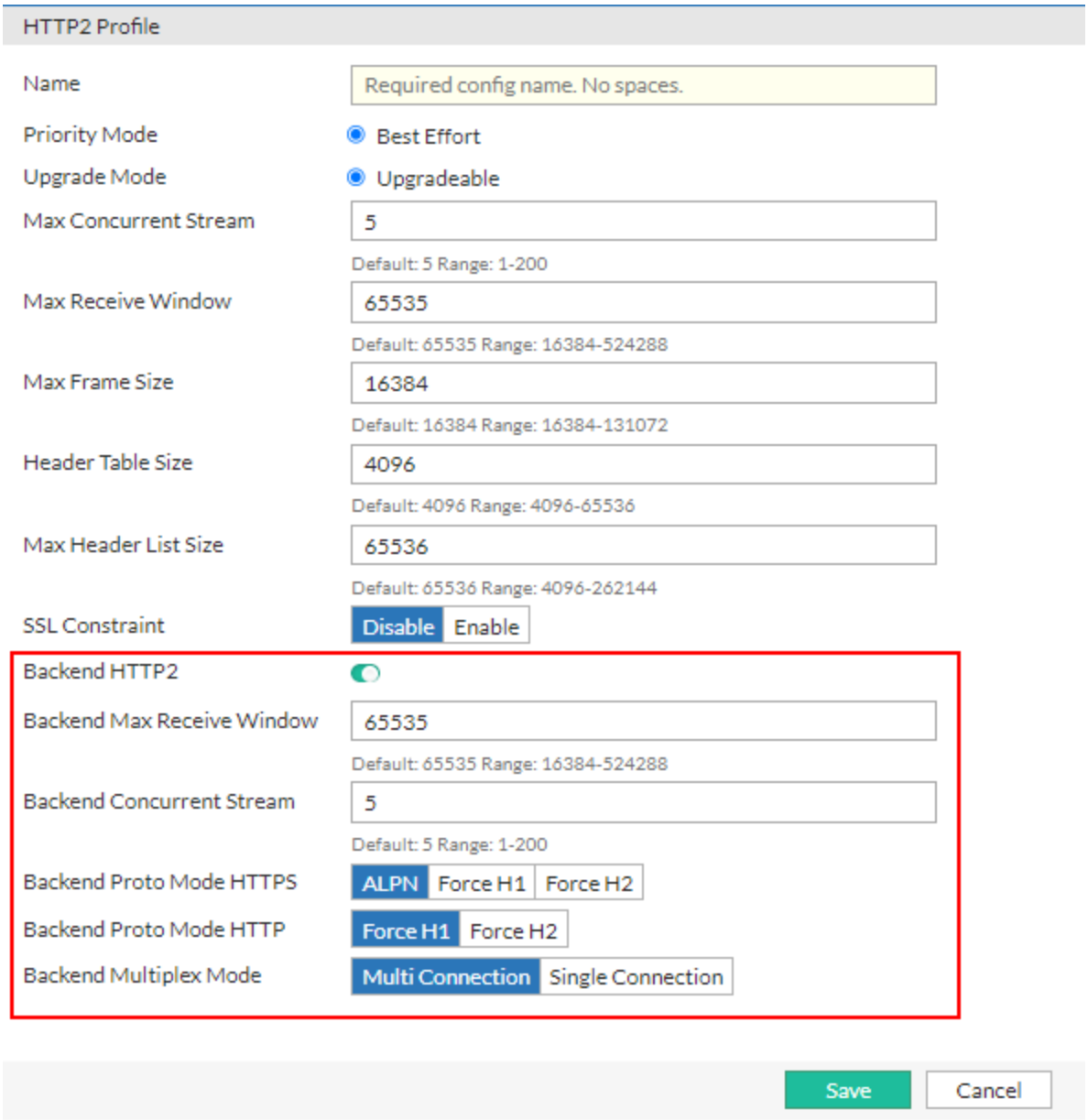
FortiADC has introduced two new predefined Application Profiles to facilitate HTTP/2 server-side connectivity within Layer 7 HTTP or HTTPS virtual servers. The profiles are **LB_PROF_HTTP2_END2END_H2** for HTTPS and **LB_PROF_HTTP2_END2END_H2C** for HTTP.

These predefined Application Profiles leverage the newly established backend HTTP/2 settings from the HTTP/2 profile, **LB_HTTP2_PROFILE_END2END_H2**. This configuration enables the HTTP/2 service to operate on the httproxy3 daemon.

Refer to the following sections for detailed updates on each configuration:

- HTTP2 Profile updates on page 140
- HTTP3 Profile updates on page 144
- Application Profile updates on page 147

HTTP2 Profile updates



To configure HTTP2 profiles:

1. Go to **Server Load Balance > Application Resources**.
2. Click the **HTTP2 Profile** tab.
3. Click **Create New** to display the configuration editor.
4. Configure the following settings:

Type	Profile Configuration Guidelines
Name	Specify a unique name for the HTTP2 profile.

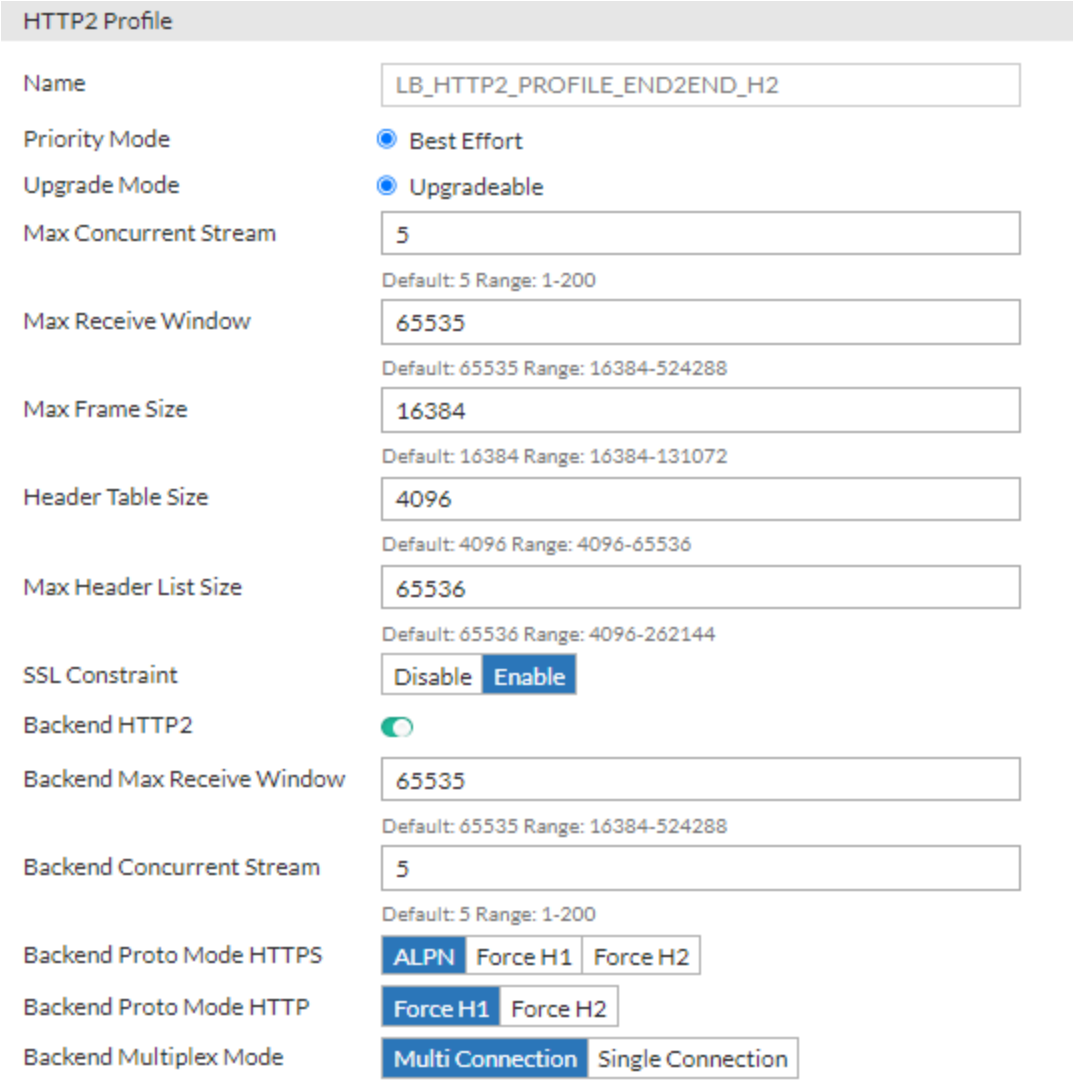
Type	Profile Configuration Guidelines
Priority Mode	Set to Best Effort. Not configurable.
Upgrade Mode	Set to Upgradeable. Not configurable.
Max Concurrent Stream	Specify the maximum number of concurrent streams available at one time. The default number is 5, and the valid range is 1-200.
Max Receive Window	Specify the maximum number of bytes that can be received without sending an acknowledgment response. The default value is 65535 bytes, and the valid range is 16384-524288.
Max Frame Size	Specify the max size of the data frames, in bytes that the HTTP2 protocol sends to the client. Setting a large frame size improves network utilization, but it can also affect concurrency. The default value is 16384 bytes, and the valid range is 16384-131072. **Note**: When **Backend HTTP2** is **enabled**, the **Max Frame Size** is not supported, as this cannot be set independently for the frontend and backend. Instead, the **HTTP2 Profile Max Frame Size** will override the **Tune Buffer Size** in the **Application Profile**.
Header Table Size	Specify the size of the header table, in KB. A larger table size allows for better HTTP header compression, but it requires more memory. The default value is 4096, and the valid range is 4096-65536.
Header List Limitation	Specify the size of the name value length , in bytes, that the HTTP2 protocol sends in a single header frame. The default value is 65536, and the valid range is 4096-262144.
SSL Constraint	Enable or disable SSL constraint. If enabled, the following conditions must be met: • The TLS implementation supports Server Name Indication. • The TLS implementation disables compression. • The TLS implementation disables renegotiation. • Renegotiation takes place before the connection preface is sent. • HTTP/2 uses cipher suites with ephemeral key exchange. • Ephemeral key exchange has a size of at least 2048 bits (for DHE) or a security level of at least 128 bits (for ECDHE). • Clients accept DHE no smaller than 4096 bits. • Stream or block ciphers are not used with HTTP.
Backend HTTP2	Enable/disable support for the backend HTTP/2 functionality. When enabled, the related virtual server will switch to httproxy3 for support. This is disabled by default. **Note**: The backend HTTP/2 implementation is built on HTTP/3, which introduces specific limitations. For details, see Configuring HTTP3 profiles.
Backend Max Receive Window	The **Backend Max Receive Window** option is available if **Backend HTTP2** is **enabled**. Specify the init-windows-size configuration for the backend HTTP/2 connection. The default value is 65535, and the valid range is 16384-524288.

Type	Profile Configuration Guidelines
Backend Concurrent Stream	The **Backend Concurrent Stream** option is available if **Backend HTTP2** is **enabled**. Specify the maximum limit for concurrent streams that the backend server can handle to ensure optimal performance and prevent overloading. The default value is 5, and the valid range is 1-200.
Backend Proto Mode HTTPS	The **Backend Proto Mode HTTPS** option is available if **Backend HTTP2** is **enabled**. Select the HTTPS server backend HTTP/2 protocol mode. • ALPN — Use Application-Layer Protocol Negotiation (ALPN). • Force H1 — Enforce HTTP/1. • Force H2 — Enforce HTTP/2. The default is ALPN.
Backend Proto Mode HTTP	The **Backend Proto Mode HTTP** option is available if **Backend HTTP2** is **enabled**. Select the HTTP server backend HTTP/2 protocol mode. • Force H1 — Enforce HTTP/1. • Force H2 — Enforce HTTP/2. The default is Force H1.
Backend Multiplex Mode	The **Backend Multiplex Mode** option is available if **Backend HTTP2** is **enabled**. Select the backend multiplexing mode. • Multi Connection — Multiple streams from the frontend are mapped to multiple backend connections. • Single Connection — All requests from multiple frontend connections are sent through a single backend connection. The default is Multi Connection.

5. Click **Save**.

Once the HTTP2 Profile configuration is saved, it can be referenced in an HTTP/HTTPS Application Profile configuration.

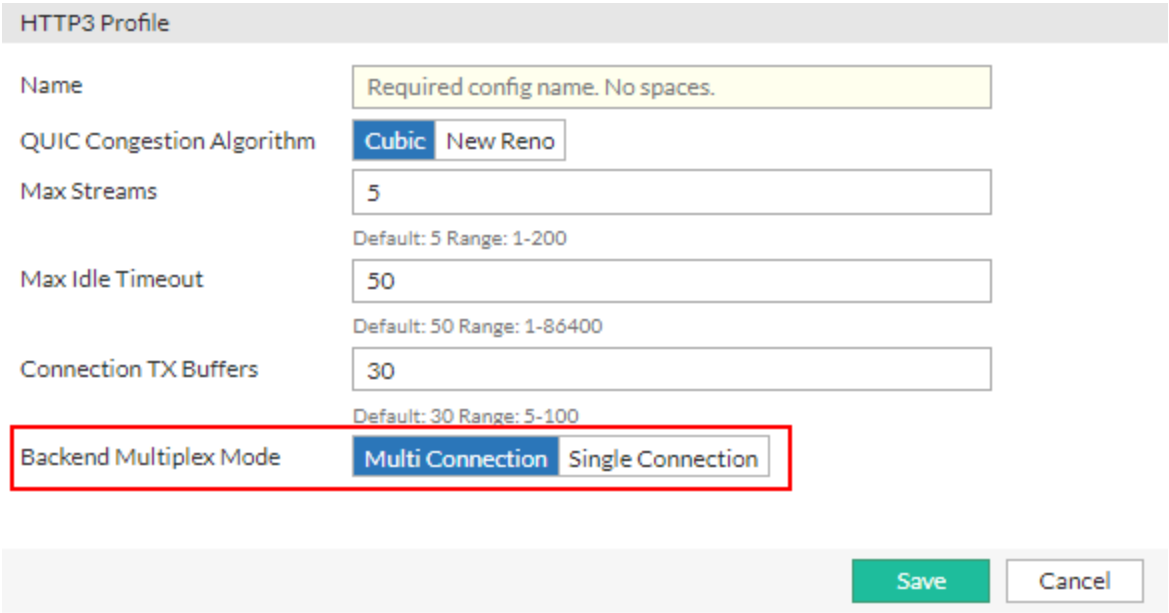
New predefined profile LB_HTTP2_PROFILE_END2END_H2:



CLI update in `config load-balance http2-profile`:

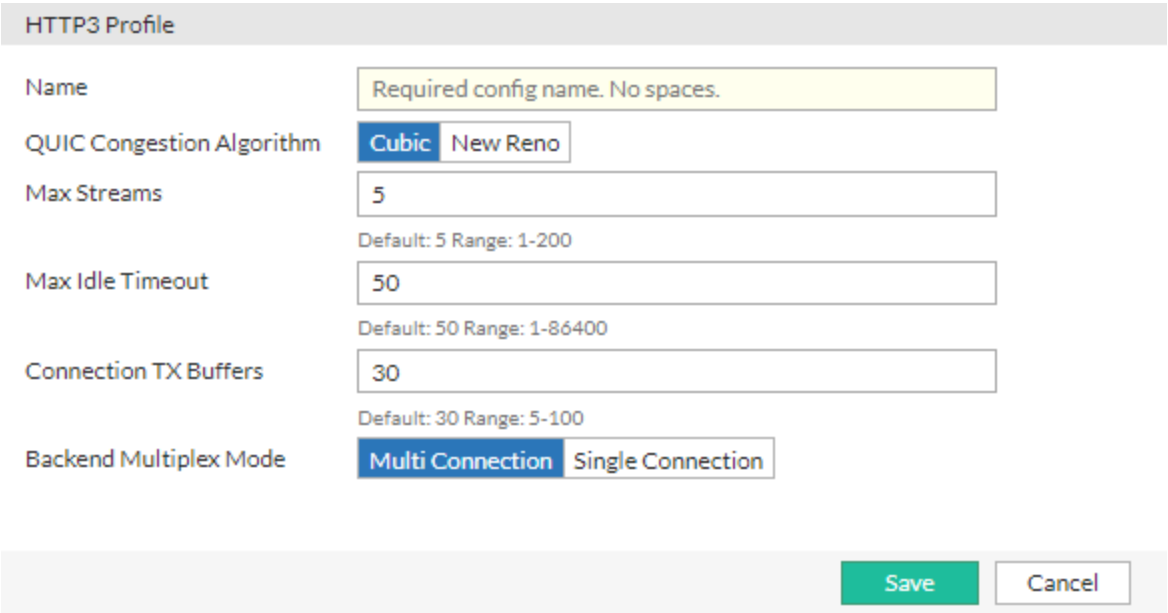
```
config load-balance http2-profile
   edit 1
      ...
      set backend-http2 enable
      set backend-http2-max-receive-window  <integer>
      set backend-http2-max-concurrent-stream  <integer>
      set backend-http2-protocol-mode-https {alpn|force-h1|force-h2}
      set backend-http2-protocol-mode-http {force-h1|force-h2}
      set backend-multiplexing-mode {single-connection|multi-connection}
   next
end
```

HTTP3 Profile updates



To configure an HTTP3 Profile:

1. Go to **Server Load Balance > Application Resources**.
2. Click the **HTTP3 Profile** tab.
3. Click **Create New** to display the configuration editor.



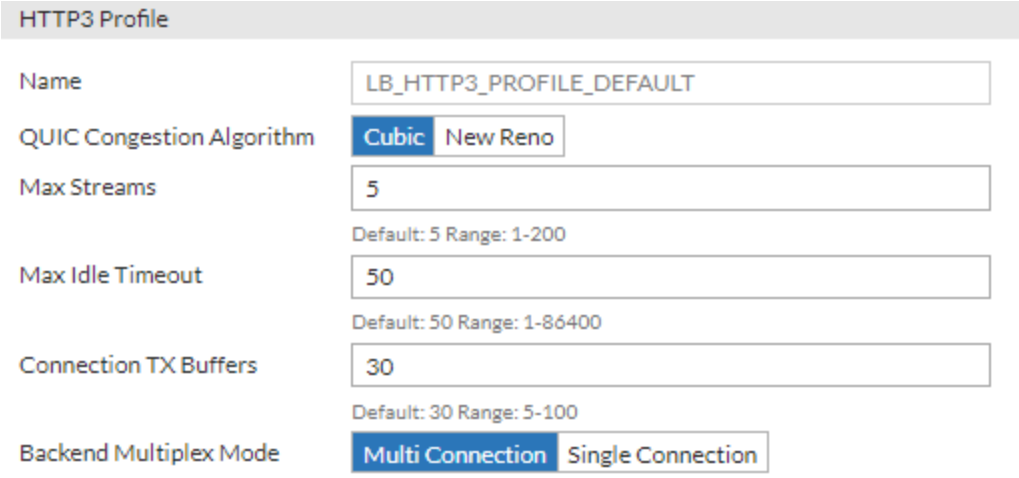
4. Configure the following settings:

Setting	Description
Name	Specify a unique name for the HTTP3 profile. Valid characters are `A-Z`, `a-z`, `0-9`, `_`, and `-`. No spaces. After you initially save the configuration, you cannot edit the name.
QUIC Congestion Algorithm	FortiADC supports Cubic and New Reno loss-based congestion control for QUIC, where the congestion control responds to packet loss events. Select the QUIC congestion algorithm to use: • Cubic • New Reno Cubic is the default congestion control algorithm.
Max Streams	Specify the maximum allowable number of HTTP/3 QUIC streams. The default value is 5, and the range is 1-200.
Max Idle Timeout	Specify the HTTP/3 QUIC connection idle timeout in seconds. When no data is transmitted over the HTTP/3 connection after the specified time has elapsed, the HTTP/3 connection will timeout. The HTTP/3 connection is tracked using a unique connection-ID instead of a UDP session. The default value is 50 seconds, and the range is 1-86400 seconds.
Connection TX Buffers	Specify the number of buffers to send on the HTTP/3 QUIC connection. This parameter significantly affects the performance of the HTTP/3 response direction. The higher the number of buffers are sent, the higher the performance will be. However, the memory usage increases. The default value is 30, and the range is 5-100.
Backend Multiplex Mode	Select the backend multiplexing mode. • Multi Connection — Multiple streams from the frontend are mapped to multiple backend connections. • Single Connection — All requests from multiple frontend connections are sent through a single backend connection. The default is Multi Connection.

5. Click **Save**.

Once the HTTP3 Profile configuration is saved, it can be referenced in an HTTPS Application Profile configuration.

Update to predefined profile LB_HTTP3_PROFILE_DEFAULT:



CLI update in `config load-balance http3-profile`:

```
config load-balance http3-profile
   edit 1
      ...
      set backend-multiplexing-mode {single-connection|multi-connection}
   next
end
```

Application Profile updates

New predefined profile LB_PROF_HTTP2_END2END_H2C:

Application Profile

Name: LB_PROF_HTTP2_END2END_H2C

Type: HTTP

Specifics

Client Timeout: 50
Default: 50 Range: 1-86400 seconds

Server Timeout: 50
Default: 50 Range: 1-86400 seconds

Connect Timeout: 5
Default: 5 Range: 1-86400 seconds

Queue Timeout: 5
Default: 5 Range: 1-86400 seconds

HTTP Request Timeout: 50
Default: 50 Range: 1-86400 seconds

HTTP Keepalive Timeout: 50
Default: 50 Range: 1-86400 seconds

X-Forwarded-For

HTTP Mode: Server Close | Once Only | Keep Alive

HTTP2: LB_HTTP2_PROFILE_END2END_H2

Advanced

Tune Buffer Size: 16384
Default: 8030 Range: 128-1048576

Max HTTP Headers: 200
Default: 100 Range: 10-10000

New predefined profile LB_PROF_HTTP2_END2END_H2:

Application Profile

Name: LB_PROF_HTTP2_END2END_H2

Type: HTTPS

Specifics

Client Timeout: 50
Default: 50 Range: 1-86400 seconds

Server Timeout: 50
Default: 50 Range: 1-86400 seconds

Connect Timeout: 5
Default: 5 Range: 1-86400 seconds

Queue Timeout: 5
Default: 5 Range: 1-86400 seconds

HTTP Request Timeout: 50
Default: 50 Range: 1-86400 seconds

HTTP Keepalive Timeout: 50
Default: 50 Range: 1-86400 seconds

Client Address
Use Client Address to connect to pool

X-Forwarded-For

HTTP Mode: Server Close | Once Only | Keep Alive

HTTP2: LB_HTTP2_PROFILE_END2END_H2

HTTP3: Click to select

Advanced

Tune Buffer Size: 16384
Default: 8030 Range: 128-1048576

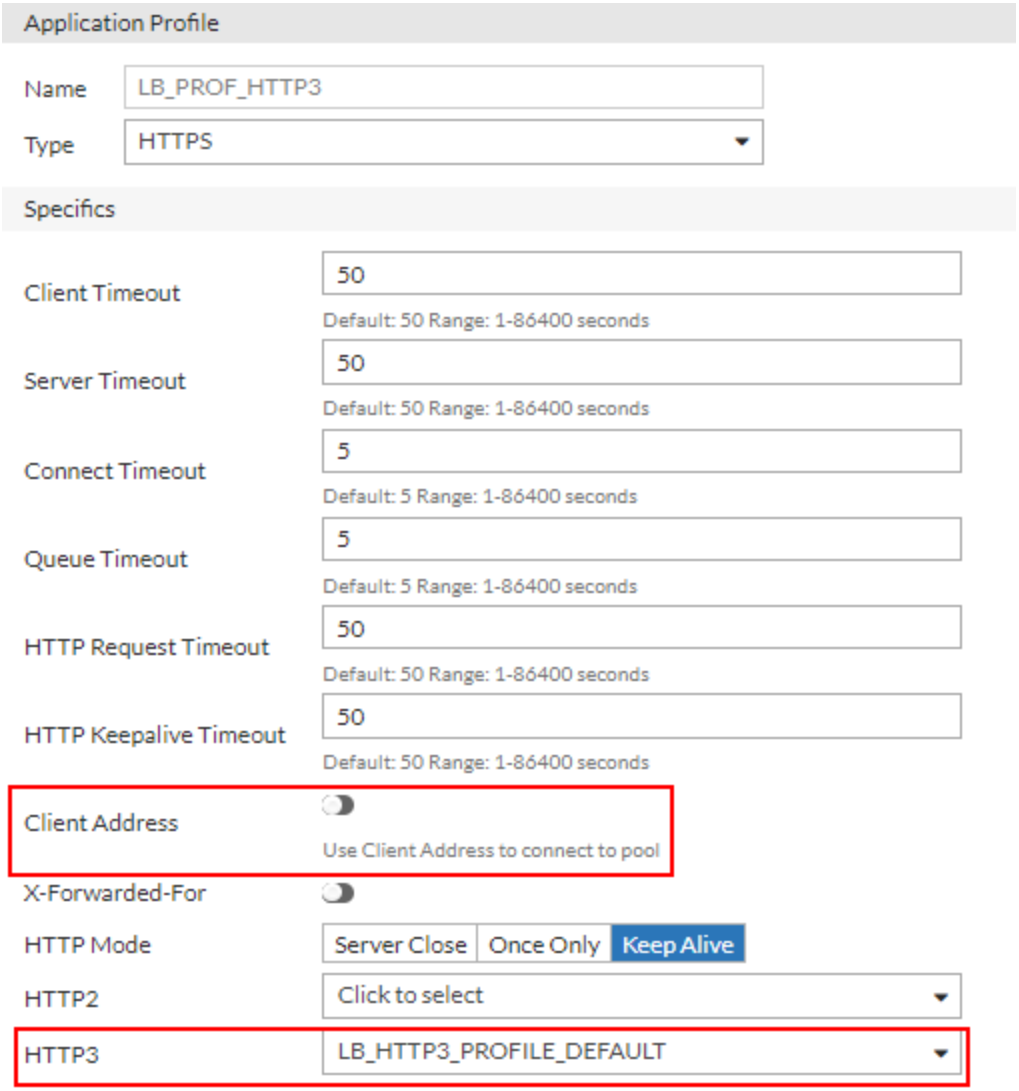
Max HTTP Headers: 200
Default: 100 Range: 10-10000

Client address option enabled for HTTP/3 virtual server - 7.6.1

FortiADC has extended client address support to the HTTP3 Application Profile, enabling the HTTP/3 Virtual Server to connect to backend real servers using the client's IP address. This feature is especially useful in complex environments where it's important to present the client's IP address directly to the backend servers.

This information is also available in the FortiADC 7.6.1 Administration Guide and CLI Reference:

- Configuring Application profiles
- config load-balance profile



Note: IPv6 address types cannot be used as the client address to connect to the backend real servers in HTTPS virtual servers.

CLI update in `config load-balance profile`:

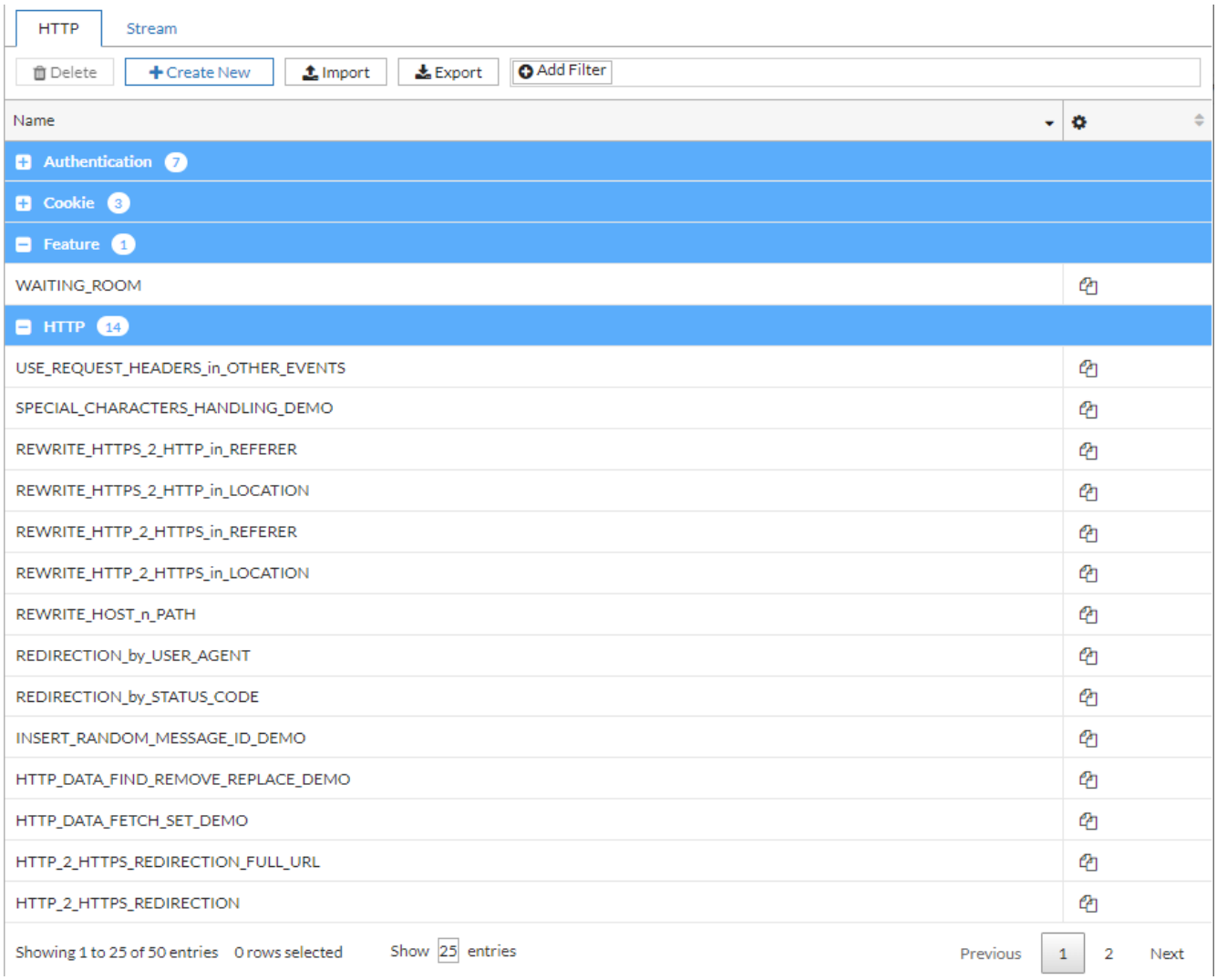
```
config load-balance profile
   edit <name>
      set type https
      ...
      set client-address disable/enable
      ...
     set http3-profile ...
   next
end
```

Scripting groups for predefined HTTP scripts - 7.6.1

FortiADC has introduced Scripting Groups to systematically categorize and organize its 50 plus predefined HTTP scripts into distinct categories: Authentication, Cookie, Feature, HTTP, IP, Optimization, Routing, SSL, TCP, Utility, and WAF. This enhancement improves operational efficiency and simplifies management, allowing administrators to quickly locate and deploy scripts within configurations, thereby optimizing performance and flexibility in handling HTTP traffic.

This information is also available in the FortiADC 7.6.1 Administration Guide and Script Reference Guide:

- Using HTTP scripting
- Predefined HTTP scripts



Group	Predefined script	Usage
Authentication	AUTH_COOKIE_BAKE	Allows you to retrieve the baked cookie and edit the cookie content.
	AUTH_EVENTS_n_COMMANDS	Lists the authentication event and commands.
	CUSTOMIZE_AUTH_KEY	Demonstrates how to customize the crypto key for authentication cookie.
	TWO_STEP_VERIFICATION	Demonstrates how to perform 2-Step Verification using FortiToken. One needs have authentication policy configured and selected in a virtual-server.
	TWO_STEP_VERIFICATION_2_NEW	Demonstrates how to perform 2-Step Verification using FortiToken for the second authentication group.
	TWO_STEP_VERIFICATION_2_SAME	Demonstrates how to perform 2-Step Verification for the second authentication group using the same token group.
	TWO_STEP_VERIFICATION_ CHANGE_KEY	Demonstrates how to change the AES key and its size for stored token group.
Cookie	COOKIE_COMMANDS	Lists the two cookie commands and shows how to use them.
	COOKIE_COMMANDS_USAGE	Demonstrates the sub-function to handle the cookie attribute "SameSite" and others.
	COOKIE_CRYPTO_COMMANDS	Used to perform cookie encryption/decryption on behalf of the real server.
Feature	WAITING_ROOM	The sample Waiting Room script demonstrates how you can place visitors in a virtual queue instead of denying them service directly when the server side reaches its configured capacity limit during high-demand situations. In this virtual Waiting Room, visitors can see their position in line and when their turn arrives, they are redirected to the requested page. Configuration parameters include the waiting room name, total resource limit threshold (default is 1000), and the Resource URL applicable to the waiting room. You can also customize the message displayed to users when they are placed in the waiting room by editing the HTML page section of the script.

Group	Predefined script	Usage
		Required data structures such as atomic counters and shared tables are already built into the script; however, you have the option to apply user-defined atomic counters and shared tables to customize the script.
HTTP	GENERAL_REDIRECT_DEMO	Redirects requests to a URL with user-defined code and cookie. **Note**: Do **not** use this script "as is". Instead, copy and customize the code, URL, and cookie.
	HTTP_2_HTTPS_REDIRECTION	Redirects requests to the HTTPS site. **Note**: This script can be used directly without making any changes.
	HTTP_2_HTTPS_REDIRECTION_ FULL_URL	Redirects requests to the specified HTTPS URL. **Note**: This script can be used directly without making any changes.
	HTTP_DATA_FETCH_SET_DEMO	Collects data in HTTP request body or HTTP response body. In `HTTP_REQUEST` or `HTTP_RESPONSE`, you could collect specified size data with `"size"` in `collect()`.In `HTTP_ DATA_REQUEST` or `HTTP_DATA_RESPONSE`. You could print the data use `"content"`, calculate data length with `"size"`, and rewrite the data with `"set"`. **Note**: Do **not** use this script "as is". Instead, copy it and manipulate the collected data.
	HTTP_DATA_FIND_REMOVE_ REPLACE_DEMO	Finds a specified string, removes a specified string, or replaces a specified string to new content in HTTP data. **Note**: Do **not** use this script "as is". Instead, copy it and manipulate the collected data.
	INSERT_RANDOM_MESSAGE_ID_ DEMO	Inserts a 32-bit hex string into the HTTP header with a parameter “Message-ID”. **Note**: This script can be used directly without making any changes.
	REDIRECTION_by_STATUS_CODE	Redirects requests based on the status code of server HTTP response (for example, a redirect to the mobile version of a site). **Note**: Do **not** use this script "as is". Instead, copy it and customize the condition in the server HTTP response status code and the URL values.

Group	Predefined script	Usage
	REDIRECTION_by_USER_AGENT	Redirects requests based on User Agent (for example, a redirect to the mobile version of a site). **Note**: You should not use this script "as is". Instead, copy it and customize the User Agent and URL values.
	REWRITE_HOST_n_PATH	Rewrites the host and path in the HTTP request, for example, if the site is reorganized. You should not use this script as is. Instead, copy it and customize the "old" and "new" hostnames and paths.
	REWRITE_HTTP_2_HTTPS_in_LOCATION	Rewrites HTTP location to HTTPS, for example, `rewrite "Location:http://www.example.com" to "Location:https://www.example.com".` **Note**: This script can be used directly without making any changes.
	REWRITE_HTTP_2_HTTPS_in_REFERER	Rewrites HTTP referer to HTTPS, for example, `rewrite "Referer: http://www.example.com" to "Referer: https://www.example.com".` **Note**: This script can be used directly without making any changes.
	REWRITE_HTTPS_2_HTTP_in_LOCATION	Rewrites HTTPS location to HTTP, for example, `rewrite "Location:https://www.example.com" to "Location:http://www.example.com".` **Note**: This script can be used directly without making any changes.
	REWRITE_HTTPS_2_HTTP_in_REFERER	Rewrites HTTPS referer to HTTP, for example, `rewrite "Referer: https://www.example.com" to "Referer: http://www.example.com".` **Note**: This script can be used directly without making any changes.
	SPECIAL_CHARACTERS_HANDLING_DEMO	Shows how to use those "magic characters" which have special meanings when used in a certain pattern. The magic characters are ( ) . % + - * ? [ ] ^ $
	USE_REQUEST_HEADERS_in_OTHER_EVENTS	Stores a request header value in an event and uses it in other events. For example, you can store a URL in a request event, and use it in a response event.

Group	Predefined script	Usage
		Note: Do **not** use this script "as is". Instead, copy it and customize the content you want to store, use `collect() in HTTP_REQUEST` to trigger `HTTP_DATA_REQUEST`,or use `collect() in HTTP_RESPONSE` to trigger `HTTP_DATA_ RESPONSE`.
IP	IP_COMMANDS	Used to get various types IP Address and port number between client and server side.
Optimization	MULTIPLE_SCRIPT_CONTROL_DEMO_1	Uses `demo_1` and `demo_2` script to show how multiple scripts work. `Demo_1` with priority 12 has a higher priority. Note: You could enable or disable other events. Do NOT use this script "as is". Instead, copy it and customize the operation.
	MULTIPLE_SCRIPT_CONTROL_DEMO_2	Uses `demo_1` and `demo_2` script to show how multiple scripts work. `Demo_2` with priority 24 has a lower priority. **Note**: You can enable or disable other events. Do **not** use this script "as is". Instead, copy it and customize the operation.
	RAM_CACHING_COMMANDS	Lists the RAM caching event and commands.
	RAM_CACHING_DYNAMIC	Demonstrates how to use script to do dynamic RAM caching. **Note**: Dynamic caching is identified by a configured ID. Ensure the RAM caching configuration is selected in the HTTP or HTTPS profile.
	RAM_CACHING_GROUPING	Demonstrates how to create multiple variations based on client IP address. The sort of grouping applies to both regular caching and dynamic caching. **Note**: Ensure the RAM caching configuration is selected in HTTP or HTTPS profile.
Routing	CONTENT_ROUTING_by_URI	Routes to a pool member based on URI string matches. **Note**: You should not use this script as is. Instead, copy it and customize the URI string matches and pool member names.
	CONTENT_ROUTING_by_X_FORWARDED_FOR	Routes to a pool member based on IP address in the X-Forwarded-For header.

Group	Predefined script	Usage
		Note: You should not use this script as is. Instead, copy it and customize the X-Fowarded-For header values and pool member names.
	PERSIST_COMMANDS	Demonstrates how to use persistence commands and event. The PERSISTENCE event is triggered when FortiADC receives the HTTP REQ and is ready to dispatch to the real server. You can set the entry in PERSISTENCE, then look up it in POST_PERSIST. FortiADC will dispatch to the dedicated server according to your entry set in PERSISTENCE if this session has not been assigned to the real server before.
SSL	OPTIONAL_CLIENT_ AUTHENTICATION	Performs optional client authentication. **Note**: Before using this script, you must have the following four parameters configured in the client-ssl-profile: • client-certificate-verify—Set to the verify you'd like to use to verify the client certificate. • client-certificate-verify-option—Set to optional • ssl-session-cache-flag—Disable. • use-tls-tickets—Disable.
	SSL_EVENTS_n_COMMANDS	Demonstrates how to fetch the SSL certificate information and some of the SSL connection parameters between server and client side.
TCP	SNAT_COMMANDS	Allows you to overwrite client source address to a specific IP for certain clients, also support IPv4toIPv6 or IPv6toIPv4 type. **Note**: Make sure the flag SOURCE ADDRESS is selected in the HTTP or HTTPS type of profile.
	SOCKOPT_COMMAND_USAGE	Allows user to customize the TCP_send buffer and TCP_receive buffer size.
	SOCKOPT_COMMANDS	Demonstrates how to the TCP:sockopt with usage examples.
	TCP_EVENTS_n_COMMANDS	Demonstrates how to reject a TCP connection from a client in TCP_ACCEPTED event.

Group	Predefined script	Usage
Utility	AES_DIGEST_SIGN_2F_COMMANDS	Demonstrates how to use AES to encryption/decryption data and some tools to generate the digest.
	ATOMIC_COUNTER_COMMANDS	Allows you to create and configure shared atomic counters that are accessible by multiple httproxy processes within one VS. The stored data is located in shared memories. In the Waiting Room setup, the atomic counters track variables at running time, including the current resource count, the current position in line, and the current total number of users in the waiting queue.
	CLASS_SEARCH_n_MATCH	Demonstrates how to use the `class_match` and `class_search` utility function.
	COMPARE_IP_ADDR_2_ADDR_GROUP_DEMO	Compares an IP address to an address group to determine if the IP address is included in the specified IP group. For example ,192.168.1.2 is included in 192.168.1.0/24. **Note**: Do **not** use this script "as is". Instead, copy it and customize the IP address and the IP address group.
	GEOIP_UTILITY	Used to fetch the GEO information country and possible province name of an IP address.
	MANAGEMENT_COMMANDS	Allow you to disable/enable rest of the events from executing.
	SHARED_TABLE_COMMANDS	Allows you to create and configure shared hash tables that are accessible by multiple httproxy processes within one VS. Both the table and stored data are located in shared memories. In the Waiting Room setup, the shared table is used to track current active resource occupiers such as active sessions.
	URL_UTILITY_COMMANDS	Demonstrates how to use those URL tools to encode/decode/parser/compare.
	UTILITY_FUNCTIONS_DEMO	Demonstrates how to use the basic string operations and random number/alphabet, time, MD5, SHA1, SHA2, BASE64, BASE32, table to string conversion, network to host conversion utility function
WAF	WAF_COMMANDS	Demonstrates how to use WAF related functions and events.

New HTTP script for enhanced error handling - 7.6.1

FortiADC introduces the **HTTP:respond_errorfile()** API, enabling HTTP to respond with predefined error files directly. This API enhances functions such as the Waiting Room by allowing predefined error pages to be served directly without requiring custom content generation. While similar functionality can be achieved with the existing HTTP:respond() API by manually constructing code and response content, the key advantage of HTTP:respond_errorfile() is the ability to reference and serve predefined error pages, optimizing efficiency in error handling and response management.

This information is also available in the FortiADC 7.6.1 Script Reference Guide:

- HTTP:respond_errorfile

HTTP:respond_errorfile

Allows the HTTP to respond with a specified error file. This function returns Boolean true if successful otherwise, returns Boolean false.

Syntax

HTTP:respond_errorfile();

Arguments

Parameter	Description
filename	A Lua string as the file name of the error file. The maximum length of a file name in Linux is 255 characters, including the file extension. Ensure the file name is valid, such as “403.html” or “path1/index.html”. If the file name is invalid, the API will still return as **true**, but the response will be **404 "Not Found"**. This parameter is mandatory.

Events

Applicable in the HTTP_REQUEST event.

Example

```
when HTTP_REQUEST {
    uri = HTTP:uri_get()
    filename = nil
    if uri=="/home/root" then
        filename = "403.html"
    …
     if filename ~= nil then
        HTTP:respond_errorfile(filename)
```

```
    end
}
```

Waiting Room for virtual queuing through HTTP scripting

The new FortiADC Waiting Room feature allows you to place visitors in a virtual queue instead of denying them service directly when the server side reaches its configured capacity limit during high-demand situations. In this virtual Waiting Room, visitors can see their position in line and when their turn arrives, they are redirected to the requested page. The queuing mechanism uses the FIFO (First In, First Out) servicing order to allow visitors who first enter the waiting room to be the first to leave the queue.

The Waiting Room feature is supported in Layer 7 HTTP/HTTPS virtual servers through Lua scripting.

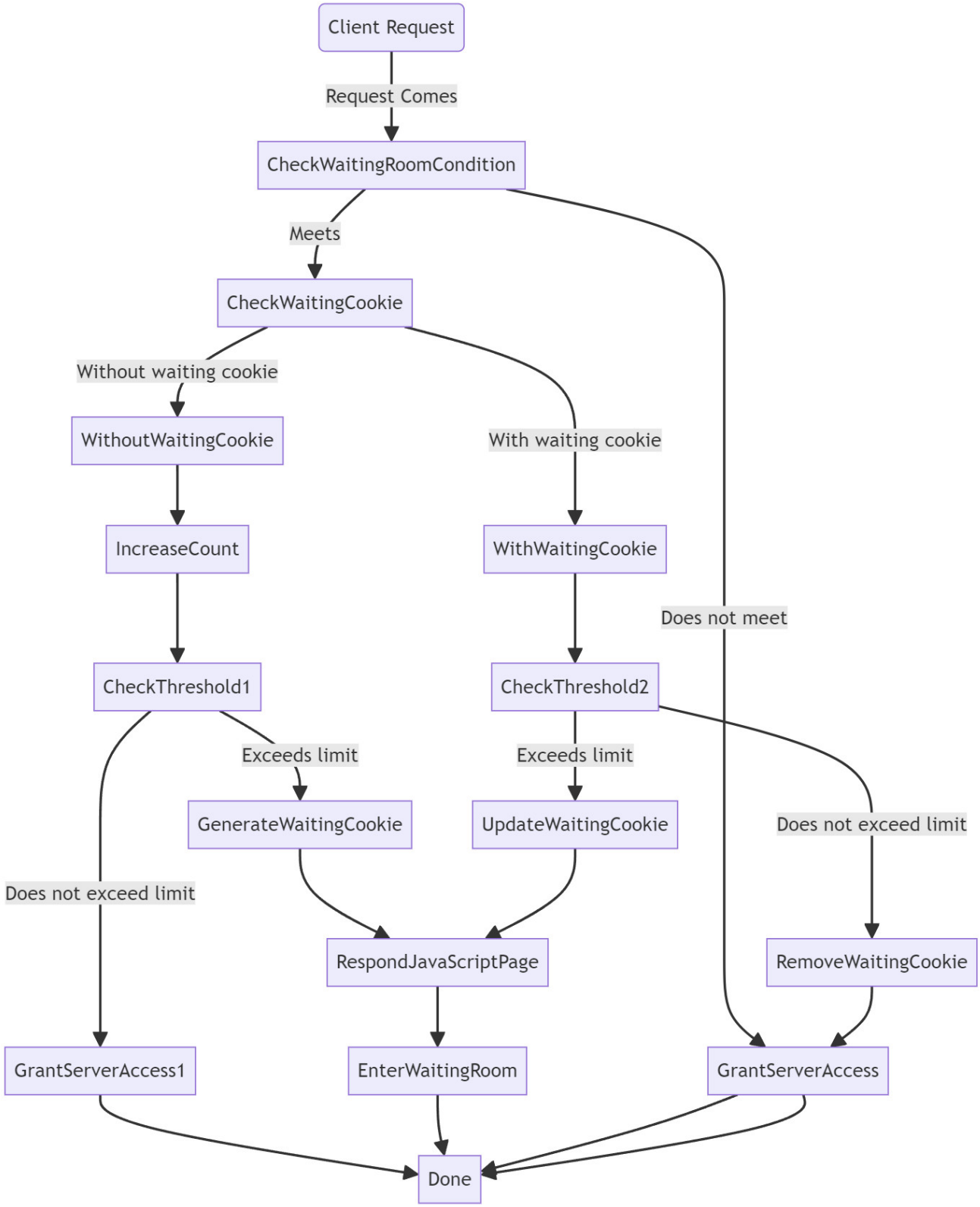
This information is also available in the FortiADC 7.6.0 Administration Guide and Script Reference Guide:

- Waiting Room for virtual queuing
- Predefined HTTP scripts

Overview

The Waiting Room page displays the visitor's queue position, and a JavaScript refreshes the page automatically on a regular interval as specified in the Waiting Room script (such as every 10 seconds). Upon each refresh cycle, a waiting room cookie is returned, at which point FortiADC will examine the cookie to determine whether the visitor will stay in the queue. As such, the Waiting Room page must remain open.

The Waiting Room flow chart below illustrates how FortiADC decides whether visitors remain in the queue, advance their queue position, or exit from the waiting room.



When a client request is made, FortiADC considers the following factors:

- Waiting room condition — if the request matches the waiting room conditions specified in the script, such as the resource URI.
- Resource threshold limit — if the admission of the visitor will exceed the resource threshold limit.
- Waiting cookie — If a waiting cookie is present.

Based on these factors, FortiADC determines the path to take visitors, of which there are five possible paths that lead to either the waiting room page or the intended destination. First, the request must meet the conditions to activate the waiting room, such as matching the resource URI. If the conditions are not met, then visitors will be directed to their intended destination instead of the waiting room. If the waiting room conditions are met, then visitors will be set onto the next stage to either queue in the waiting room or be directed to the requested page based on the resource threshold limit. In this second stage, FortiADC will check for the presence of a waiting cookie — visitors without a waiting cookie will be assigned one to enter the waiting room queue if their admission will exceed the resource threshold limit; however, if the limit is not exceeded then the visitor will be directed to the requested page instead. When FortiADC detects the presence of a waiting cookie, this indicates that the visitor is already in the waiting room queue. FortiADC will then evaluate the threshold limit and either advance the visitor's queue position or direct them to the requested page.

New predefined HTTP scripts:

To implement the Waiting Room feature, FortiADC has introduced three new predefined HTTP scripts.

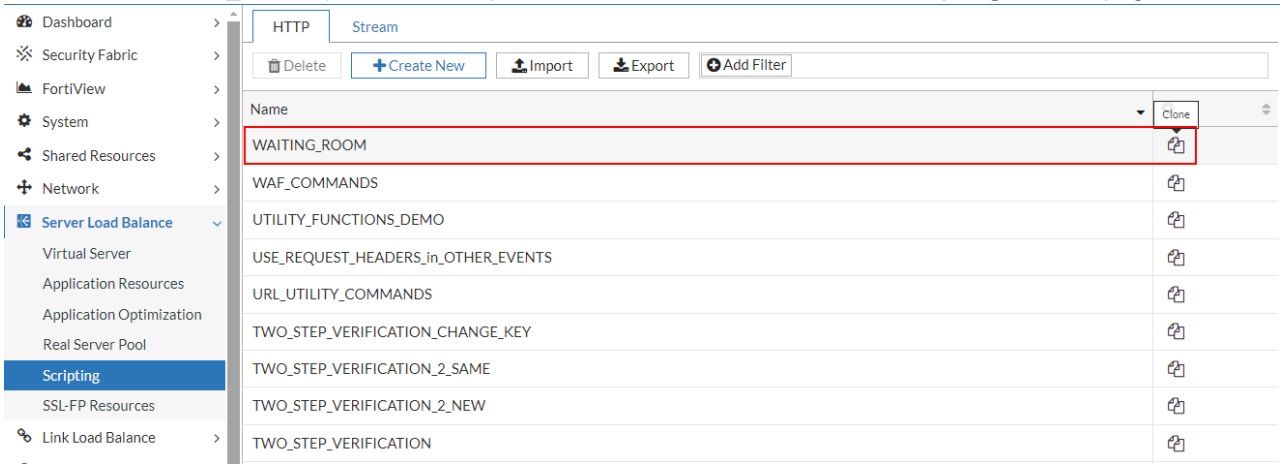
Predefined script	Description
WAITING_ROOM	The sample Waiting Room script demonstrates how you can place visitors in a virtual queue instead of denying them service directly when the server side reaches its configured capacity limit during high-demand situations. In this virtual Waiting Room, visitors can see their position in line and when their turn arrives, they are redirected to the requested page. Configuration parameters include the waiting room name, total resource limit threshold (maximum is 1000), and the Resource URL applicable to the waiting room. You can also customize the message displayed to users when they are placed in the waiting room by editing the HTML page section of the script. Required data structures such as atomic counters and shared tables are already built into the script; however, you have the option to apply user-defined atomic counters and shared tables to customize the script.
ATOMIC_COUNTER_ COMMANDS	Includes commands that allow you to create and configure shared atomic counters that are accessible by multiple httproxy processes within one VS. The stored data is located in shared memories. In the Waiting Room setup, the atomic counters track variables at running time, including the current resource count, the current position in line, and the current total number of users in the waiting queue.
SHARED_TABLE_COMMANDS	Includes commands that allow you to create and configure shared hash tables that are accessible by multiple httproxy processes within one VS. Both the table and stored data are located in shared memories. In the Waiting Room setup, the shared table is used to track current active resource occupiers such as active sessions.

Setting up the Waiting Room

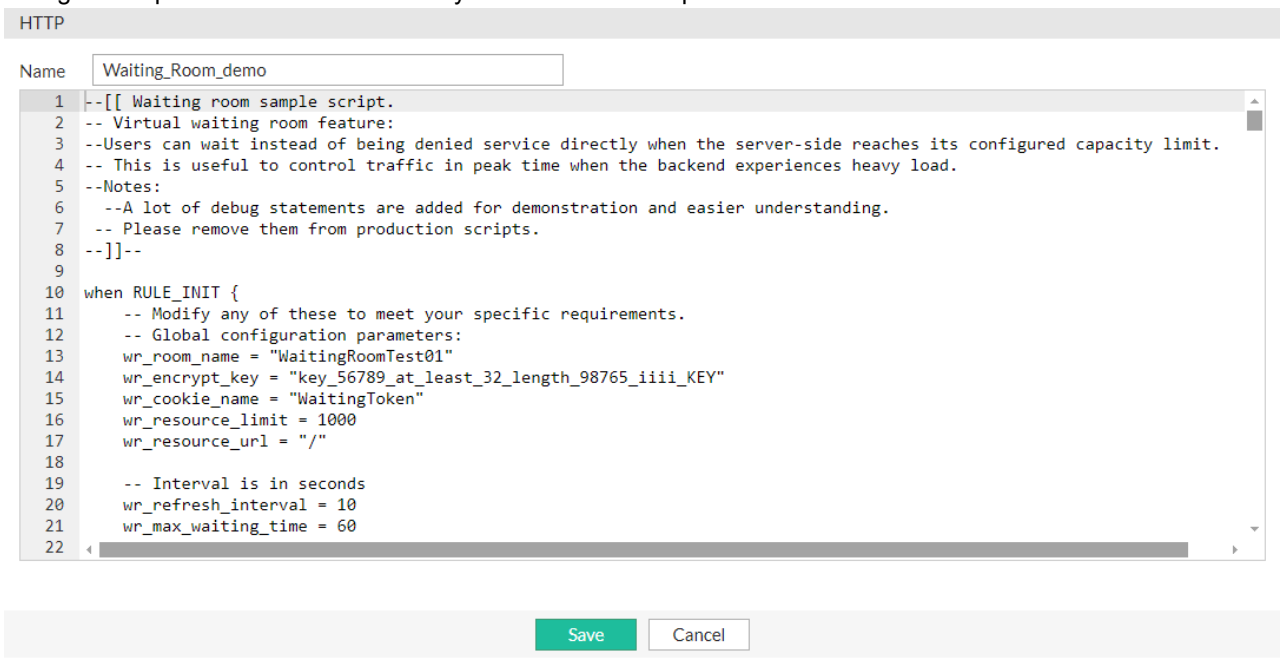
The WAITING_ROOM predefined HTTP script is designed to be used as a template for users to clone and modify as needed. Built-in instructions indicate which configuration parameters can be modified in the WAITING_ROOM script. All required data structures are already included in the template, such as atomic counters and shared tables; however, you have the option to create user-defined objects to use in your custom waiting room. The following steps describes the basic workflow to setting up a virtual waiting room to direct visitor overflow.

To set up a virtual waiting room:

1. Clone the **WAITING_ROOM** predefined script from the **Server Load Balance > Scripting > HTTP** page.



2. Name the cloned waiting room script and customize the parameters as needed, such as those under "Global configuration parameters" section. Save your customized script.

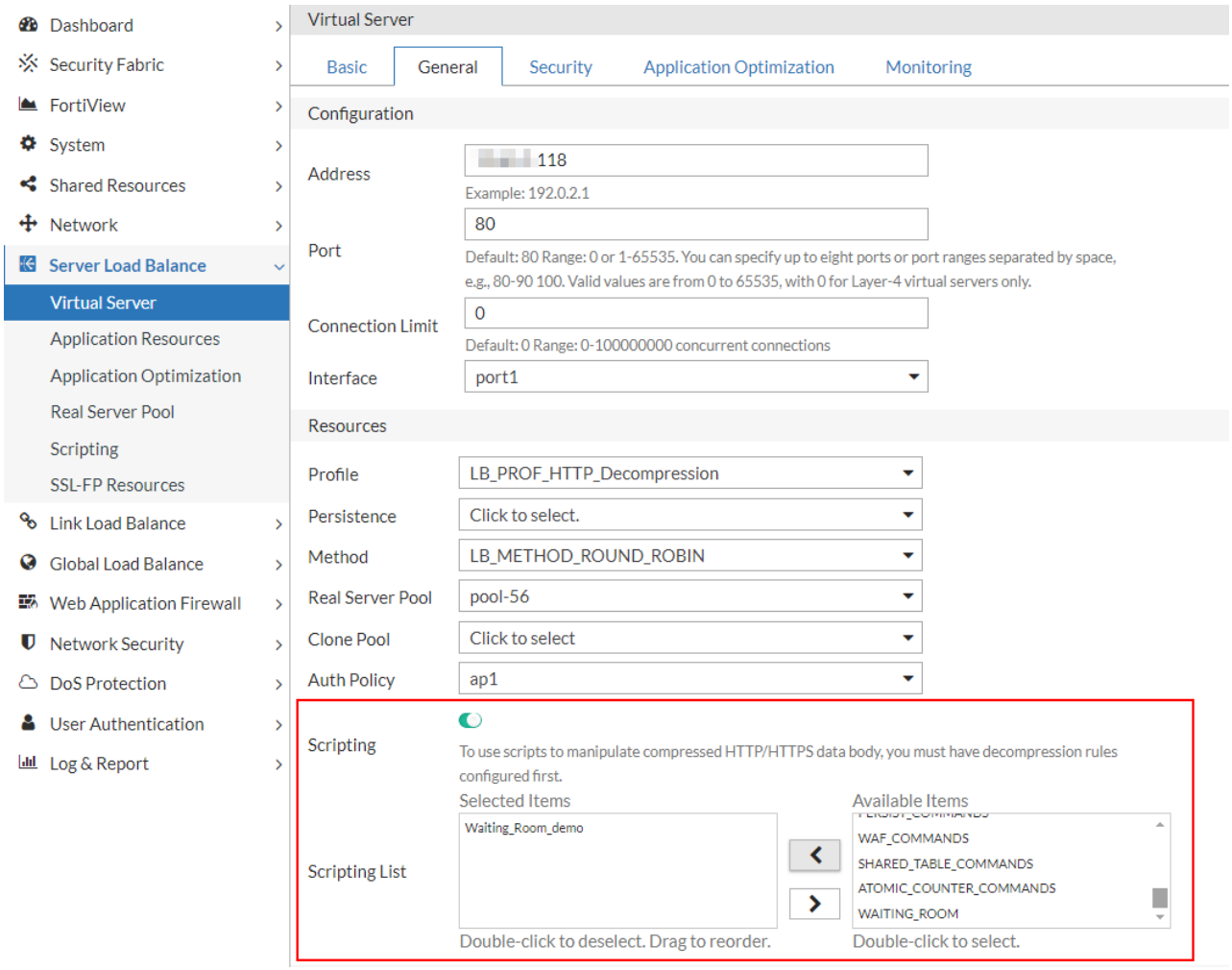


The resource limit, "wr_resource_limit" is often modified for testing purposes. A smaller threshold can make testing easier. For example, if the "wr_resource_limit = 2", then the threshold is three client sessions to trigger the waiting room. Otherwise, if the resource limit is 1000 instead, then it would require 1001 sessions to trigger the waiting room.

a. Optionally, you can customize and edit the HTML response page within the "waiting room HTML page" section of the script (line 39-71).

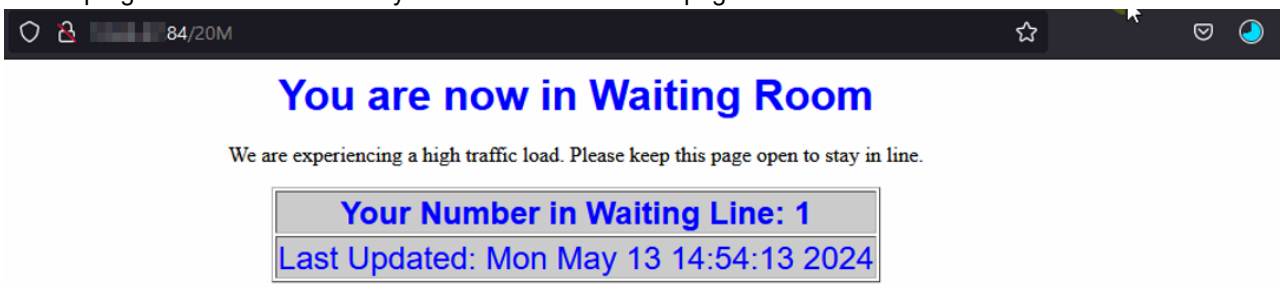
```
35  -- We reset the line num when all the current waiting clients get serviced,
36  -- Similar to repeat waiting numbers only till next day in real world.
37  wr_waiting_total_cnt_name = "WaitingTotalCounterDemo1"
38
39  -- This is the waiting room HTML page, the script will fill a waiting number and a last updated timestamp.
40  -- The embedded javascript is to refresh automatically. Other contents are customizable.
41  wr_payload_template =
42      [[
43      <html>
44      <head>
45      <meta http-equiv="Content-Language" content="en-us">
46      <meta http-equiv="Content-Type" content="text/html; charset=us-ascii">
47      <title>Waiting Room Test Page</title>
48          <style>
49              table {
50                  margin-left: auto;
51                  margin-right: auto;
52              }
53          </style>
54      </head>
55      <script type="text/javascript">
56
```

3. Apply your waiting room script in the virtual server. Ensure this virtual server is a Layer 7 HTTP or HTTPS VS and is enabled.



4. Use a browser on the client side to test the waiting room setup by making requests to the configured VS to ensure the waiting room behaves as intended.
You should be directed to your virtual waiting room and be placed in the queue if you have exceeded the resource

limit. After some time has passed, when some of the HTTP connections has been closed, your queue position should progress and then eventually be directed to the server page.



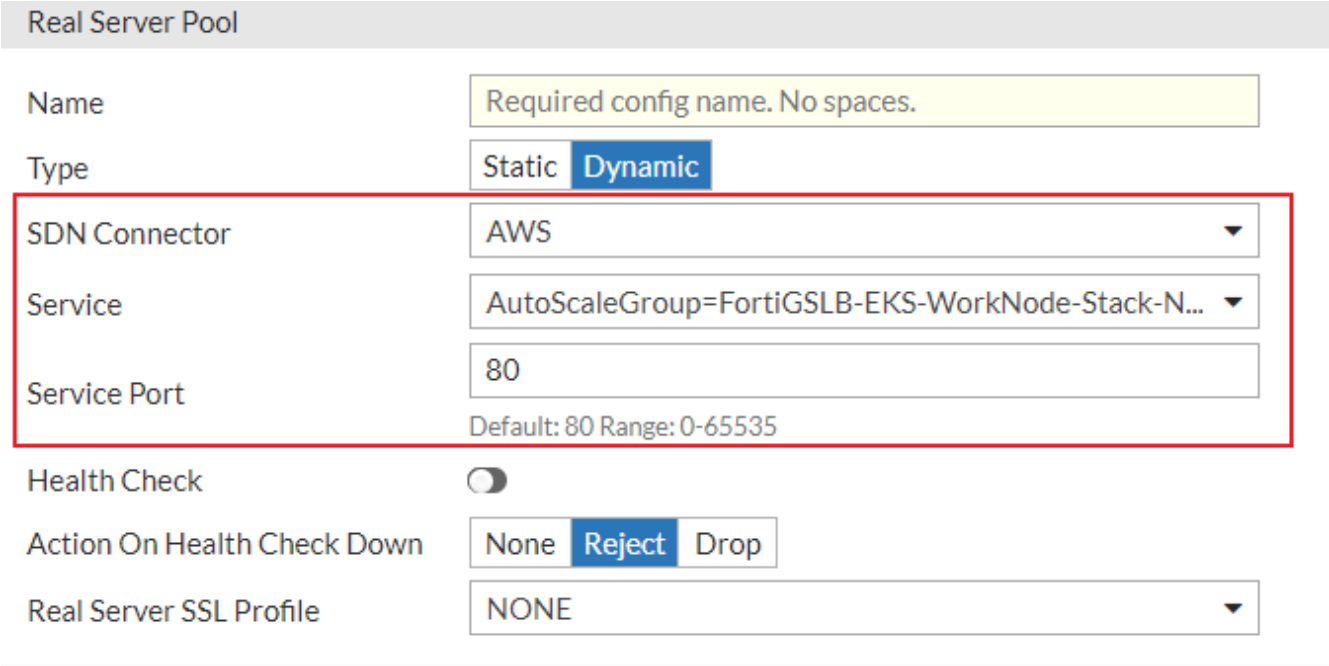
AWS autoscaling group discovery

FortiADC now supports AWS autoscaling group discovery to create dynamic real server pools. In the Real Server Pool configuration, you can now select the **AutoScaleGroup=xxx** tag in the **Service** and **Service Port** fields to automatically trigger the AWS SDN connector to add instances with the autoscaling group tag into the pool. When a scale-in or scale-out event occurs on the AWS side, the SDN connector will update the real server and pool based on the scale-in/scale-out result.

FortiADC have also enhanced the AWS SDN Connector configuration to enable the Region Name to be selected from a drop-down list instead of by manual input.

This information is also available in the FortiADC 7.6.0 Administration Guide and CLI Reference Guide:

- Using real server pools
- config load-balance pool



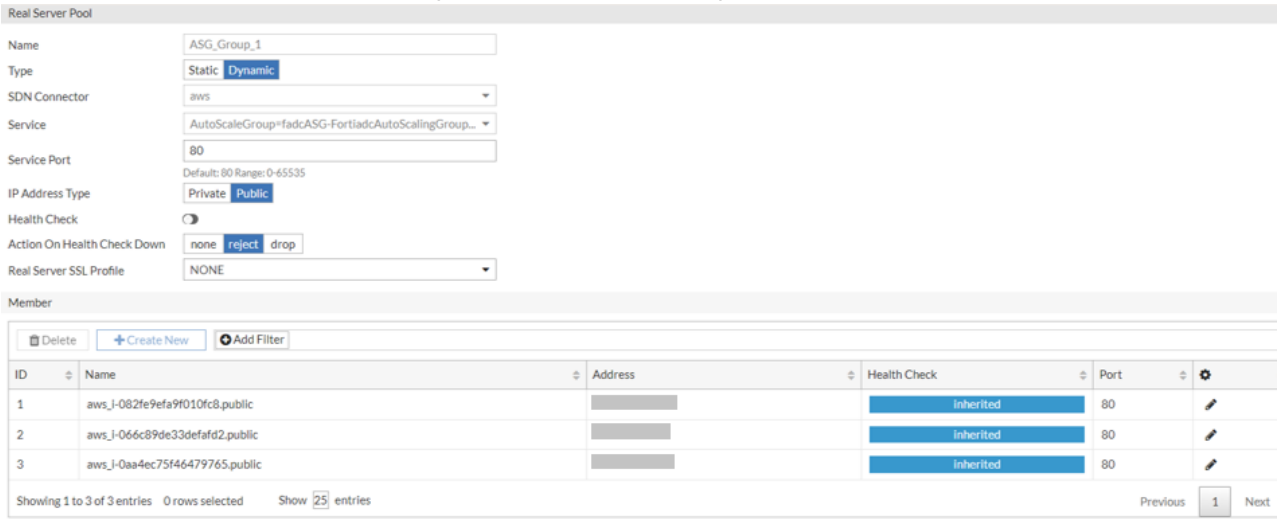
To create a dynamic real server pool using AWS autoscale services:

1. Go to **Server Load Balance > Real Server Pool**.
   The configuration page displays the **Real Server Pool** tab.
2. Click **Create New** to display the configuration editor.

3. Configure the required Real Server Pool settings:

Setting	Description
Name	Configuration name. Valid characters are `A-Z`, `a-z`, `0-9`, `_`, and `-`. No spaces. You reference this name in the virtual server configuration. **Note**: After you initially save the configuration, you cannot edit the name.
Type	**Dynamic**
SDN Connector	**AWS**
Service	Select a service with the **AutoScaleGroup=xxx** tag.
Service Port	Specify the service port. Default: 80 Range: 0-65535.

4. Click **Save** to confirm the Real Server Pool settings.
Once the Real Server Pool configuration is saved, the AWS SDN connector will add instances with the autoscaling group tag as pool members, appearing under the **Member** section. When a scale-in or scale-out event occurs on the AWS side, the SDN connector will update the real server and pool based on the scale-in/scale-out result.



Once a dynamic real server pool is created, you can no longer modify its Type, SDN Connector, or Service fields. You also cannot manually add or delete any of its pool members.

These real servers that were automatically populated by the dynamic pool are Read-Only and cannot be modified or deleted. However, if their dynamic real server pool is deleted or the SDN connector becomes invalid, then the real servers will be deleted automatically.

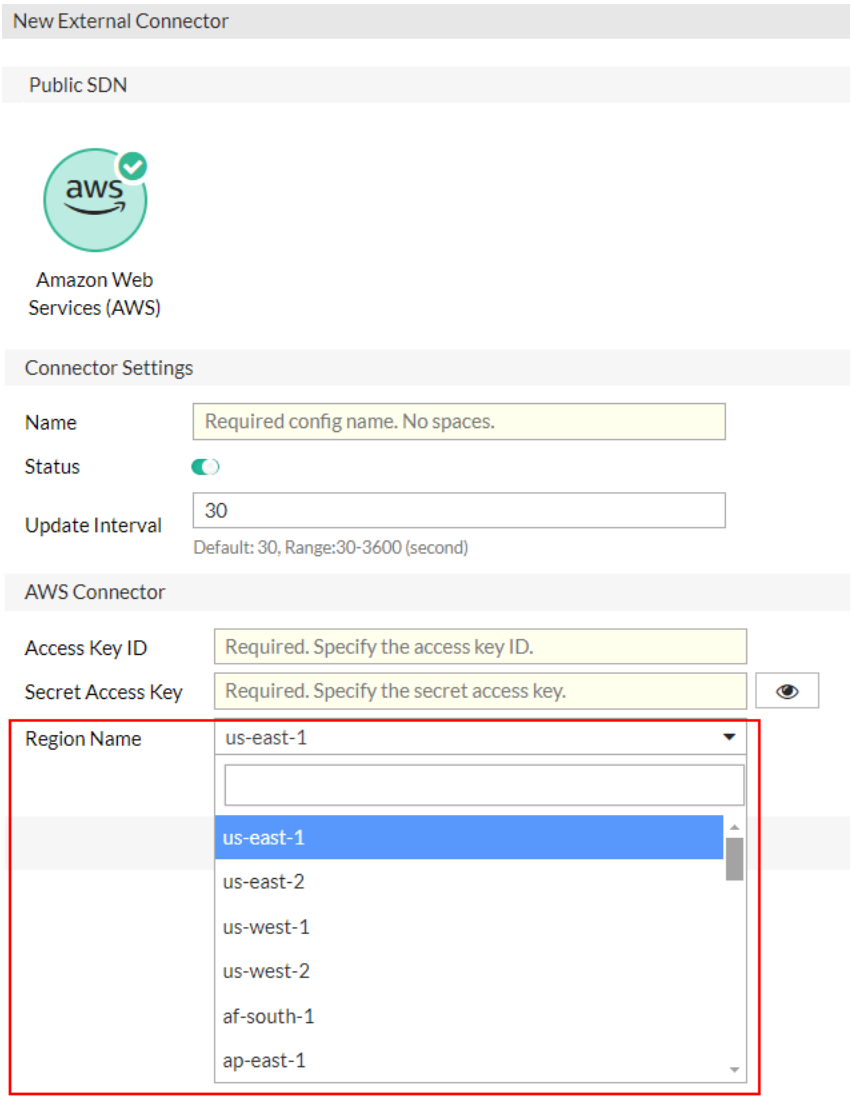
CLI update in `config load-balance pool`:

```
config load-balance pool
   edit <name>
     set type dynamic
     set sdn-connector aws
     set service AutoScaleGroup=FortiGSLB-EKS-WorkNode-Stack-NodeGroup-1E9GIIPMRV483
     set service-port 80
     set health-check-ctrl disable
     unset health-check-list
```

```
    set real-server-ssl-profile NONE
  next
end
```

GUI update in AWS connector configuration:

The **Region Name** field can now be selected from a drop-down menu.



New health check down options

FortiADC introduces new **Action on Health Check Down** options for the real server pool configuration to enable various actions to be taken on existing connections in the event that a pool member fails a health check.

Note: This feature is only supported in Layer 4 Server Load Balancing.

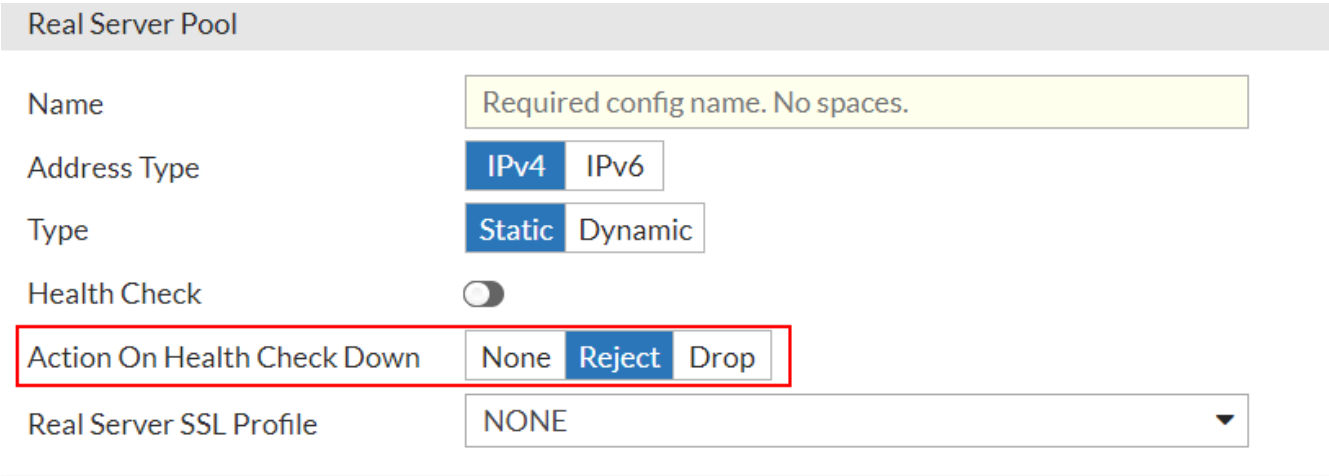
This information is also available in the FortiADC 7.6.0 Administration Guide and CLI Reference Guide:

- Using real server pools
- config load-balance pool

Previously, whenever a real server pool member fails a health check, all existing sessions to the pool member are automatically dropped, which then requires the client to re-authenticate and start the process again. The new Action on Health Check Down options offer three response actions to when a pool member fails the health check: None, Reject, and Drop.

Action on Health Check Down option	Description
None	New connections are scheduled to other available pool members. For existing active connections, they will expire after the session times out. In which case FortiADC will send TCP RST or ICMP unreachable message to clients. If a pool member passes the health check before the existing active connections expire, those connections will remain and will not be reset.
Reject	This is the default setting. FortiADC removes any current connections that associate with the pool member that failed the health check. For existing active connections, FortiADC will send TCP RST or ICMP unreachable message to clients. If there are other available pool members, FortiADC resets existing active connections and sends new arriving connections to the available pool member.
Drop	FortiADC silently removes the connections that associate with the pool member that failed the health check. All new connections are scheduled to other healthy pool members.

GUI configuration update in Real Server Pool > Real Server Pool

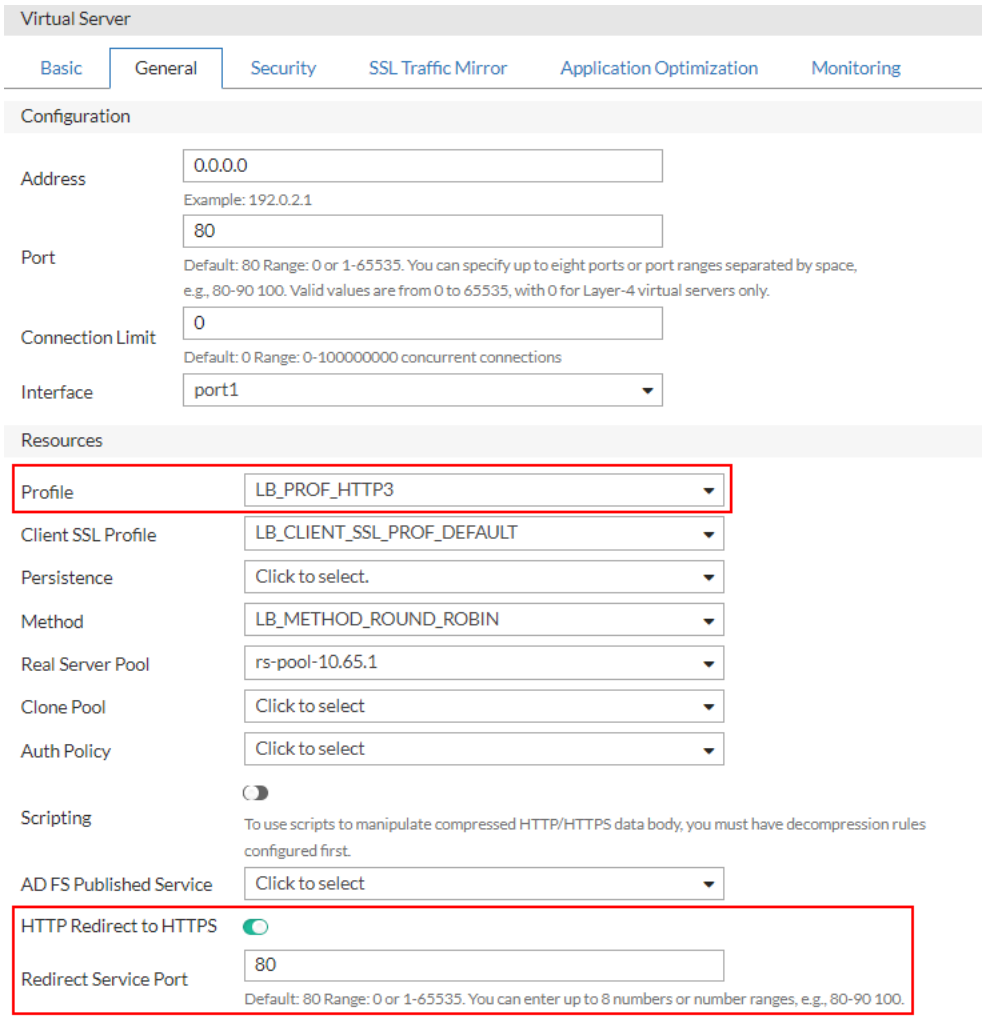


CLI update in `config load-balance pool`:

```
config load-balance pool
   edit <name>
     set health-check-ctrl enable
     set health-check-down-action {drop|none|reject}
   next
end
```

HTTP3 support for HTTP to HTTPS Redirection

FortiADC now supports HTTP to HTTPS redirection in HTTP3 virtual servers. Now, in the Virtual Server configuration, you can enable the HTTP Redirect to HTTPS option when an HTTP3 server load-balance profile is selected. Previously, this option was only available for HTTPS and HTTP2 profiles.



HTTP to HTTPS redirection is used to automatically direct users from an insecure HTTP URL to a secure HTTPS URL to ensure that the communication between the user's browser and the server is encrypted to protect data from being intercepted or tampered with by attackers. When incoming HTTP traffic reaches the virtual server on a configured port (such as port 80), the virtual server identifies it and generates an HTTP 302 Found response with a Location header pointing to the HTTPS URL (such as https://example.com). The client's browser then follows this redirect, initiating a new, secure connection to the virtual server over HTTPS (typically on port 443). This ensures that all HTTP requests are automatically redirected to HTTPS, enhancing connection security through encryption.

Link Load Balance

The FortiADC 7.6 release includes new features and enhancements in **Link Load Balance**:

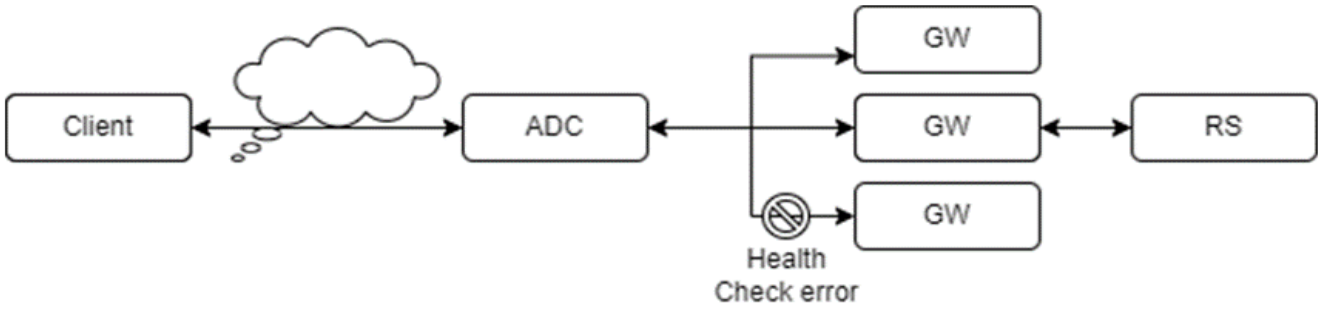
SLB local traffic support on page 172

FortiADC has optimized its Link Load Balancing (LLB) mechanism to ensure continuous Server Load Balancing (SLB) traffic flow, even in the event of a gateway failure. This enhancement allows client traffic to traverse through the FortiADC to the backend servers without interruption, maintaining seamless connectivity via the LLB routing path.

SLB local traffic support

FortiADC has optimized its Link Load Balancing (LLB) mechanism to ensure continuous Server Load Balancing (SLB) traffic flow, even in the event of a gateway failure. This enhancement allows client traffic to traverse through the FortiADC to the backend servers without interruption, maintaining seamless connectivity via the LLB routing path.

Previously, when client traffic was routed through the FortiADC to the real server via LLB, a failure of the selected gateway within the LLB group (such as due to a health check failure) could lead to stale gateway information being retained in the SLB system. This latency in refreshing the gateway status could result in traffic interruptions.



To mitigate this issue, FortiADC has enhanced its LLB routing mechanism to dynamically reallocate sessions to available gateways when a failure is detected in one of the gateways. This enhancement ensures continuous traffic flow by promptly redirecting sessions to healthy gateways within the LLB group, thereby maintaining seamless connectivity and preventing traffic interruptions.

Global Load Balance

The FortiADC 7.6 release includes new features and enhancements in **Global Load Balance**:

Multiple Global DNS Policy support in FQDN zones on page 174

FortiADC can now allow FQDN zones to be associated with multiple Global DNS Policies. This functionality enables you further granularity in global load balancing by employing multiple DNS policies to concurrently manage different traffic routing needs.

DNS forwarding support at zone level with no matching hostname on page 176

FortiADC introduces the new **Forward Host** option in GLB Zones that enable DNS queries to be forwarded to the remote server at the zone level with no requirement for a matching hostname. Previously, DNS queries forwarded from the GLB zone could not be resolved without a matching hostname. When the Forward Host option is enabled, the GLB zone forwarding will then only require the zone to match, and not the hostname as well.

DNS forwarding log debug in CLI on page 179

FortiADC has enhanced the `diagnose debug module named` CLI command to include debug information for DNS forwarding. Logs can now be generated for when DNS queries are forwarded to the remote DNS server to diagnose forwarding failures.

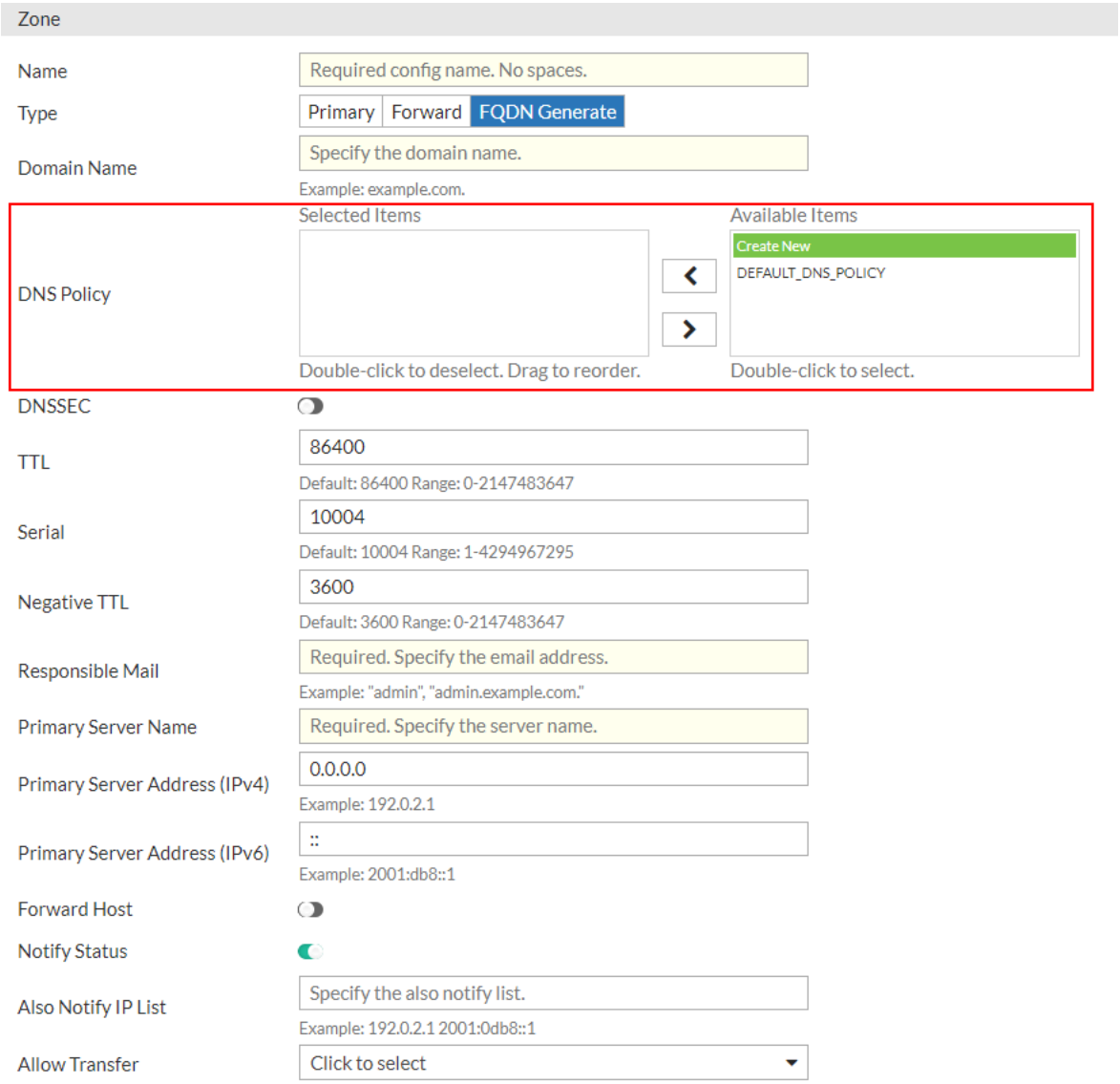
Multiple Global DNS Policy support in FQDN zones

FortiADC can now allow FQDN zones to be associated with multiple Global DNS Policies. This functionality enables you further granularity in global load balancing by employing multiple DNS policies to concurrently manage different traffic routing needs.

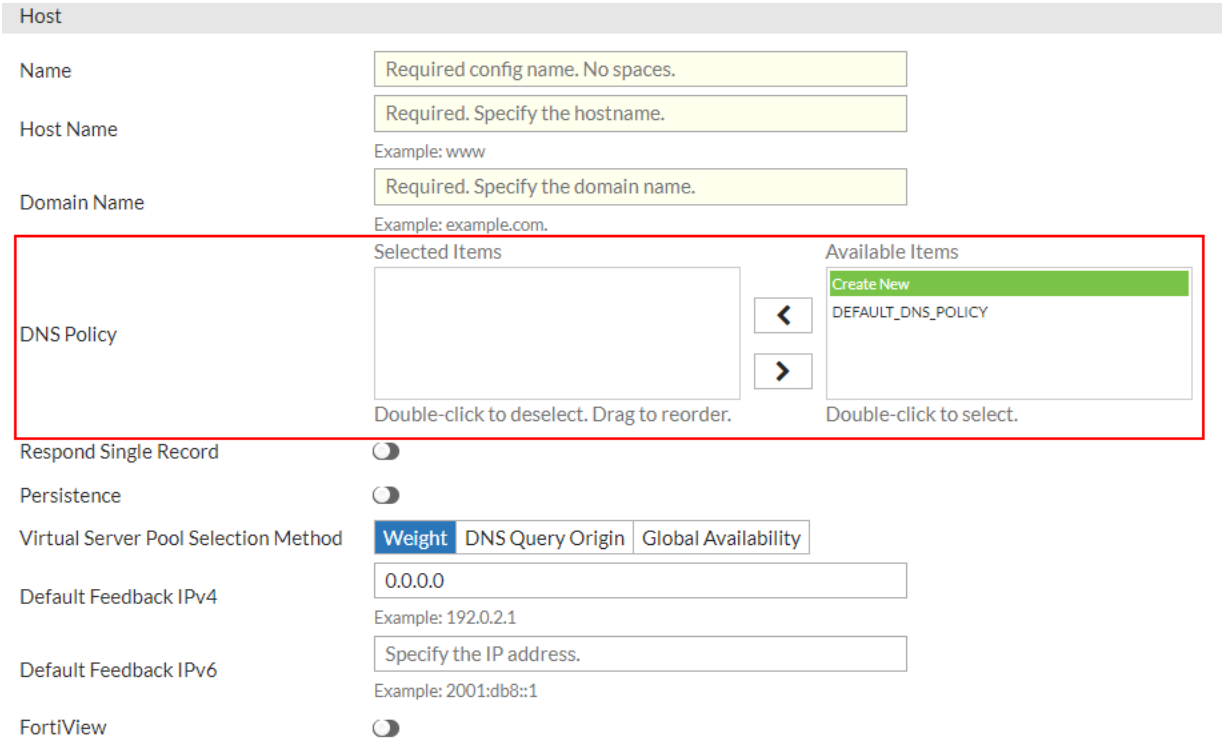
This information is also available in the FortiADC 7.6.0 Administration Guide:

- Configuring DNS zones
- Configuring hosts

Configuration update in Zone Tools > Zone:



Configuration update in FQDN > Host:



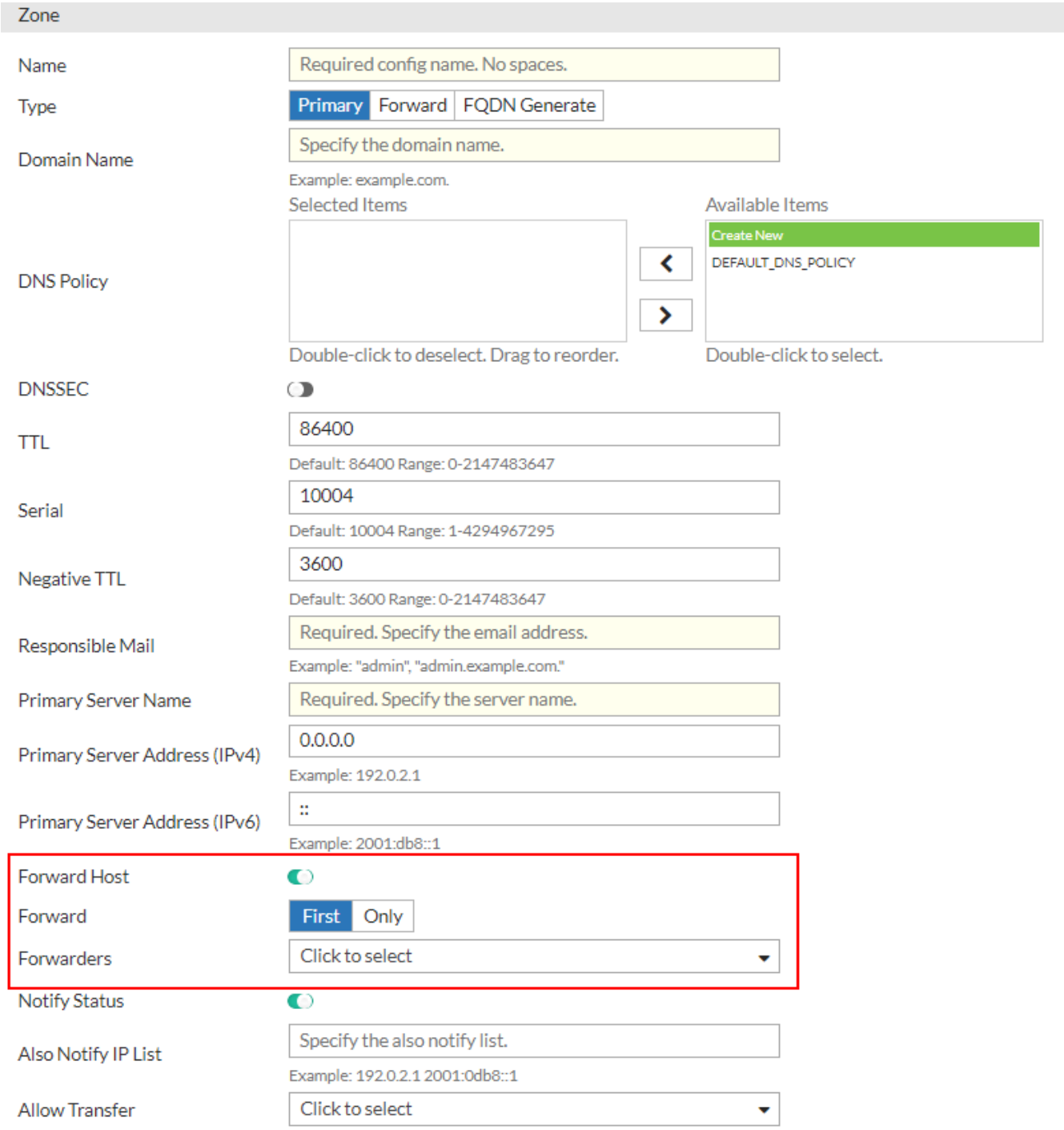
DNS forwarding support at zone level with no matching hostname

FortiADC introduces the new **Forward Host** option in GLB Zones that enable DNS queries to be forwarded to the remote server at the zone level with no requirement for a matching hostname. Previously, DNS queries forwarded from the GLB zone could not be resolved without a matching hostname. When the Forward Host option is enabled, the GLB zone forwarding will then only require the zone to match, and not the hostname as well.

This information is also available in the FortiADC 7.6.0 Administration Guide:

- Configuring DNS zones

GUI configuration update in Zone Tools > Zone



Setting	Description
Forward Host (new option)	Enable Forward Host to allow DNS queries to be forwarded to remote servers at the zone level. This is disabled by default.

Setting	Description
	This only requires the forwarded DNS query to match the zone and no other information is required to match such as the hostname.
Forward	The **Forward** option is now only available if **Forward Host** is **enabled**. • First—The DNS server queries the forwarder before doing its own DNS lookup. This is the default option. • Only—Only query the forwarder. Do not perform a DNS lookup. **Note**: The internal server caches the results it learns from the forwarders, which optimizes subsequent lookups.
Forwarders	The **Forwarders** option is now only available if **Forward Host** is **enabled**. Select a remote server configuration object.

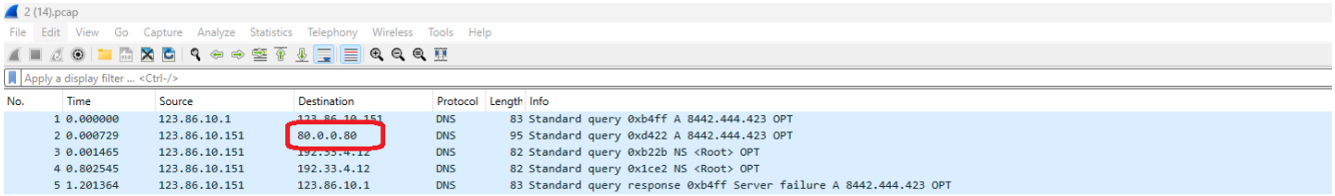
CLI update in `config global-dns-server zone`

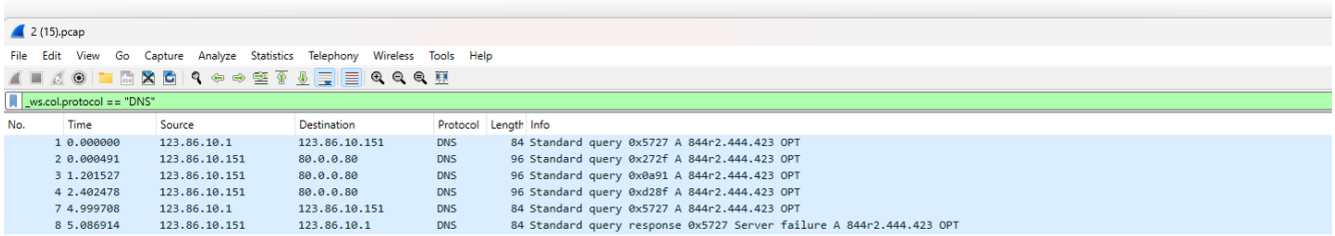
```
config global-dns-server zone
   edit <name>
      set type primary
      set forward-host {enable|disable}
      set forward {first|only}
      set forwarders <DNS server name>
   next
end
```

Verify DNS forwarding to remote server

To verify if the DNS queries are being successfully forwarded to the remote server at the zone level, you can send a test query and check its packet data. Using tools such as "nslookup" or "dig", you can send a DNS query, and then collect and view its packet data using PCAP tools such as "wireshark".

The example below, we will analyze the packet data to verify whether the DNS query was sent to the forwarder.





From this packet data, you can extract the following information:

1. Client 123.86.10.1 sent the DNS query (8442.444.423 A record) to 123.86.10.151.
2. Request for GLB zone domain matches record; no match for hostname.

3. The DNS query was sent to the forwarder (80.0.0.80).

In this example, it can be confirmed that the DNS query was successfully sent to the forwarder after the DNS server confirms the GLB zone domain match without requiring the hostname to match as well.

DNS forwarding log debug in CLI

FortiADC has enhanced the `diagnose debug module named` CLI command to improve troubleshooting and diagnostics for DNS forwarding failures, which will better support the DNS forwarding functionality available in global DNS policy, zone, and general settings. This enhancement enables the generation of detailed logs when DNS queries are forwarded to remote DNS servers, providing deeper insights into the forwarding process and aiding in the identification and resolution of potential issues.

User Authentication

The FortiADC 7.6 release includes new features and enhancements in **User Authentication**:

User Group Match Conditions for Authentication Control 7.6.5 on page 181

FortiADC 7.6.5 adds support for match conditions in user group member configurations, enabling administrators to define specific criteria that must be met before a member is used for authentication. This capability is available for all user group member types: Local, LDAP, RADIUS, NTLM, and TACACS+, and conditions are evaluated prior to credential verification.

RADIUS and TACACS+ Access Profile Override via CLI 7.6.4 on page 102

You can now override the local access profile of an administrator account during login when authenticating with **RADIUS** or **TACACS+** with the new CLI option `set accprofile-override` under `config system admin`. When enabled, FortiADC applies the access profile returned by the remote server instead of the one configured locally.

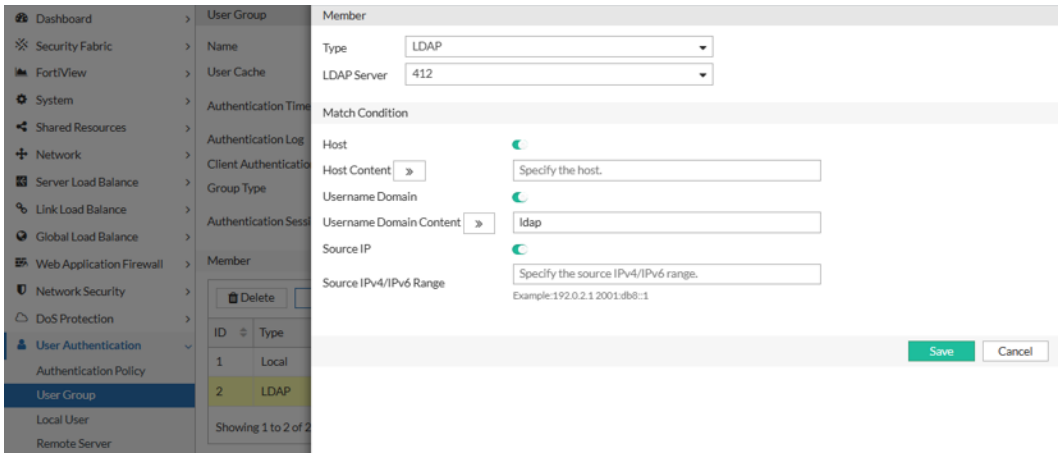
Extended maximum authentication timeout 7.6.1 on page 182

FortiADC has increased the maximum authentication timeout value from 60,000 to 120,000 milliseconds in the Authentication User Group settings.

User Group Match Conditions for Authentication Control - 7.6.5

FortiADC 7.6.5 adds support for match conditions in user group member configurations, enabling administrators to define specific criteria that must be met before a member is used for authentication. This capability is available for all user group member types: Local, LDAP, RADIUS, NTLM, and TACACS+, and conditions are evaluated prior to credential verification.

This enhancement improves how FortiADC handles authentication in multi-domain and distributed environments. In earlier versions, all configured user group members were queried sequentially, which could lead to unnecessary login prompts, slower authentication, or failed attempts in certain SSO scenarios. With match conditions, FortiADC can immediately route authentication requests to the correct backend server, improving accuracy, reducing latency, and enhancing the user experience.



Available match conditions:

- **Host** – Matches the HTTP `Host` header value using a case-insensitive regular expression (maximum length: 255 characters).
- **Username Domain** – Matches the domain portion of the username (`domain\username` or `username@domain`) using a case-insensitive regular expression (maximum length: 255 characters).
- **Source IP** – Matches the client's IPv4/IPv6 address or IP range. Supports up to 8 entries, separated by commas (maximum length: 1024 characters).

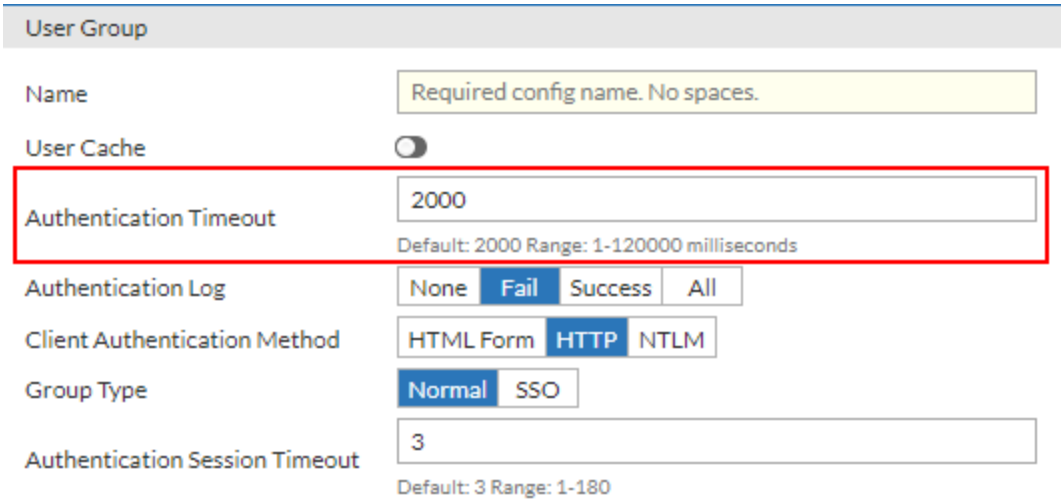
FortiADC evaluates all enabled conditions together. A user group member is selected only when every enabled condition matches the request. If no conditions are enabled, authentication proceeds as in previous versions with no condition-based filtering.

Extended maximum authentication timeout - 7.6.1

FortiADC has increased the maximum authentication timeout value to 120,000 milliseconds in the Authentication User Group configuration. The previous limit of 60,000 milliseconds restricted the implementation of authentication systems that require longer processing times, such as RADIUS-based two-factor authentication (2FA). This extension provides greater flexibility for managing authentication workflows with extended timeout requirements, ensuring compatibility with more complex authentication setups, including but not limited to RADIUS 2FA.

This information is also available in the FortiADC 7.6.1 Administration Guide and CLI Reference Guide:

- Configuring user groups
- config user user-group



User groups are authorized by the virtual server authentication policy. The user group configuration references the authentication servers that contain valid user credentials.

To configure a user group:

1. Go to **User Authentication > User Group**.
   The configuration page displays the **User Group** tab.
2. Click **Create New** to display the configuration editor.
3. Configure the following User Group settings:

Settings	Guidelines
Name	Configuration name. Valid characters are `A-Z`, `a-z`, `0-9`, `_`, and `-`. No spaces. After you initially save the configuration, you cannot edit the name.
User Cache	Enable to cache the credentials for the remote users (LDAP, RADIUS, TACACS+) once they are authorized.

Settings	Guidelines
User Cache Timeout	The **User Cache Timeout** option is available if **User Cache** is enabled. Timeout for cached user credentials. The default is 300 seconds. The valid range is 1-86,400 seconds.
Authentication Timeout	Timeout for query sent from FortiADC to a remote authentication server. The default is 2,000 milliseconds. The valid range is 1-120,000 milliseconds.
Authentication Log	Specify one of the following logging options for authentication events: • None — No logging. • Fail — Log failed attempts. • Success — Log successful attempts. • All — Log all (both failed and successful attempts).
Client Authentication Method	• HTML Form • HTTP • NTLM (only if you want to use NTLM server as a authentication server)
Use Default Form	The **Use Default Form** option is available if **Client Authentication Method** is **HTML Form**. Enabled by default to use the default authentication form. Disable to use a customized authentication form.
Customized Authentication Form	The **Customized Authentication Form** option is available if **Client Authentication Method** is **HTML Form** and **Use Default Form** is **disabled**. Select a Customized Authentication Form object or create new.
Group Type	• Normal — Default. No action is needed. • SSO — Select to enable Single Sign-On (SSO).
Authentication Relay	The **Authentication Relay** option is available if **Group Type** is **SSO**. Select an authentication relay profile.
Authentication Session Timeout	Specify the authentication session timeout. Valid values range from 1 to 180 minutes. The default is 3 (minutes).
SSO Cross Domain Support	The **SSO Cross Domain Support** option is available if **Group Type** is **SSO**. Disabled by default. When enabled, you must specify the SSO domain. **Note**: Authentication policies cannot be applied to multiple virtual servers. Due to security reasons, such as protection against XSS attacks, there is no shared mechanism between virtual servers to decrypt cookies. As a result, you cannot log into a second virtual server while already logged into the first virtual server as the virtual servers are independent from each other. SSO Cross Domain Support allows you to have multiple domain names on the same virtual server (the virtual host), where you can specify a first-level domain name to enable the second-level domain names on the virtual server to decrypt cookies at the same time.
SSO Domain	The **SSO Domain** option is available if **Group Type** is **SSO** and **SSO Cross Domain Support** is **enabled**.

Settings	Guidelines
	Specify the SSO domain.
Log Off URL	The **Log-off URL** option is available if **Group Type** is **SSO**. Specify the log-off URL.

4. Click **Save**.
   Once the **User Group** configuration is saved, the **Member** section becomes available for configuration.
5. Under the **Member** section, click **Create New** to display the configuration editor.
6. Configure the following Member settings and save the configuration:
   a. Select the **Type**: Local, LDAP, RADIUS, NTLM, or TACACS+.
   b. Select the corresponding configuration based on the selected Type.
7. Click **Save** again to save the Member added to the User Group configuration.

CLI update in `config user user-group`:

```
config user user-group
   edit <name>
      set set auth-timeout <integer>
   next
end
```

`auth-timeout`	Timeout for query sent from FortiADC to a remote authentication server. The default is 2000 milliseconds. The valid range is 1-120,000 milliseconds.

Log & Report

The FortiADC 7.6 release includes new features and enhancements in **Log & Report**:

Kafka Integration for Log Export 7.6.4 on page 186

FortiADC 7.6.4 extends its log export framework with native support for **Apache Kafka**, an open-source distributed event streaming platform used to aggregate and distribute high-volume telemetry data. This enhancement introduces a new **Plugin** configuration type that allows FortiADC to forward event, traffic, and security logs directly to Kafka for real-time processing by external systems such as Elastic or other SIEM/observability platforms.

Enhanced Syslog encryption via CLI 7.6.1 on page 189

FortiADC has strengthened Syslog security by introducing enhanced encryption through the TCP SSL protocol. New CLI options now allow administrators to apply either high and medium-level encryption algorithms for SSL communication, ensuring greater flexibility and control over security settings.

Syslog support for IPv6 FQDNs 7.6.1 on page 196

FortiADC now supports sending logs to syslog servers using IPv6 addresses resolved from FQDNs. Previously, only FQDNs resolving to IPv4 addresses were supported for syslog server communication. This enhancement improves flexibility by allowing users to configure syslog servers with IPv6 FQDNs.

Kafka Integration for Log Export - 7.6.4

FortiADC 7.6.4 extends its log export framework with native support for **Apache Kafka**, an open-source distributed event streaming platform used to aggregate and distribute high-volume telemetry data. This enhancement introduces a new **Plugin** configuration type that allows FortiADC to forward event, traffic, and security logs directly to Kafka for real-time processing by external systems such as Elastic or other SIEM/observability platforms.

Previous releases supported exporting logs only to syslog servers or FortiAnalyzer. With this enhancement, FortiADC can stream logs to Kafka without requiring FortiAnalyzer as an intermediary. Kafka is the first third-party destination supported through the Plugin framework, which is designed to accommodate additional platforms in future releases.

FortiADC implements this feature using **fluent-bit** as the log forwarder. The internal logging process (`miglogd`) streams logs to fluent-bit over a Unix socket, and fluent-bit publishes the data to Kafka brokers.

This information is also available in the FortiADC 7.6.4 Administration Guide and CLI Reference:

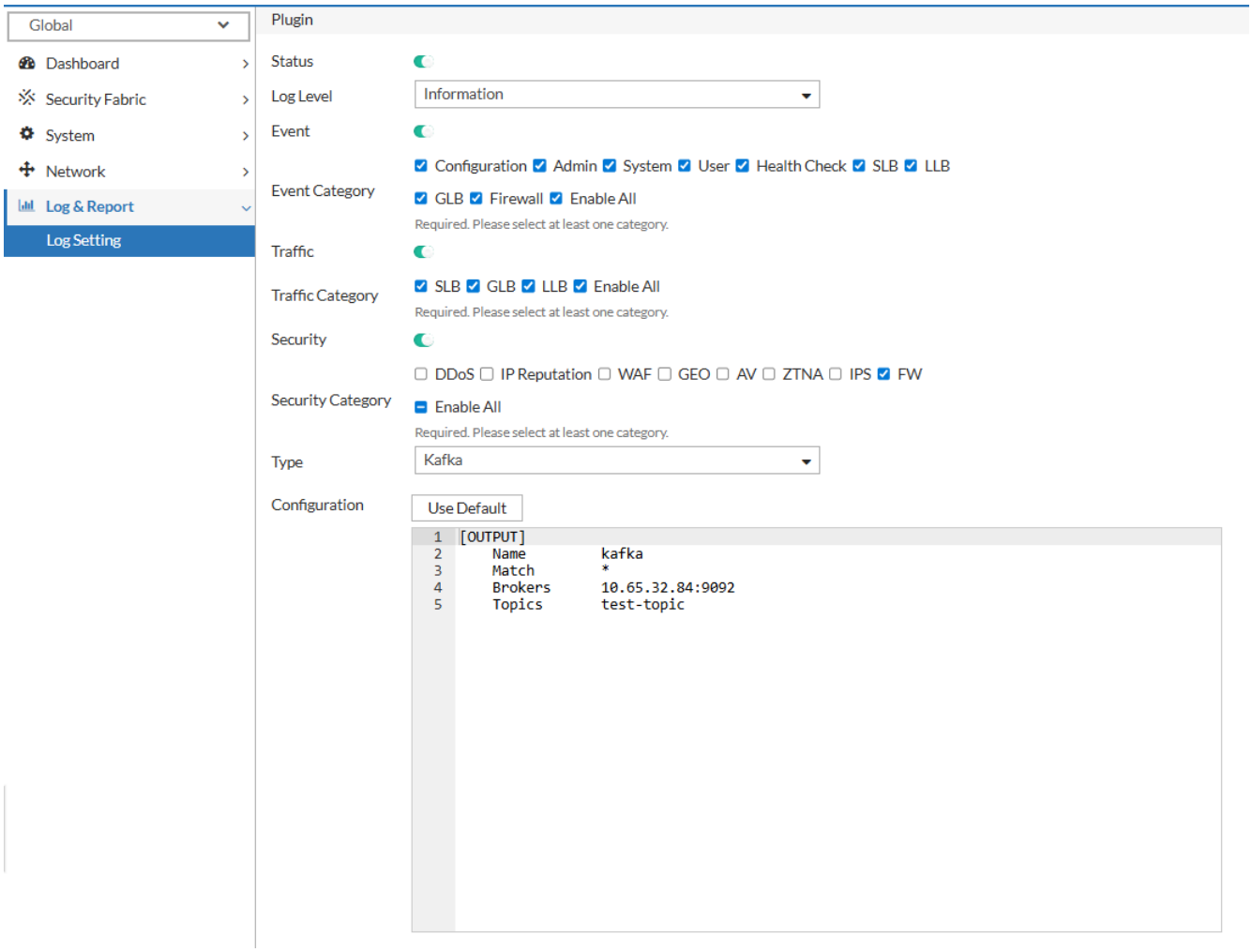
- Configuring Plugin log export settings
- config log setting global_plugin

Configuration

Kafka log export is managed from the new **Plugin** page under **Log & Report > Log Setting**. Configuration is available only in **Global Settings**, and all VDOM log data is exported through the root VDOM. Per-VDOM override is not supported in this release.

From the Plugin page, administrators can enable or disable the Kafka plugin, select which types of logs to forward, and define the connection parameters for their Kafka environment. The configuration can only be created in the **GUI**; CLI commands support editing but not creating a Kafka plugin entry.

The Plugin page provides both general log export controls (status, log level, and category selection) and a dedicated **Configuration** field where Kafka connection details are specified. The Configuration field uses **fluent-bit syntax** to define how FortiADC connects to Kafka brokers and publishes logs to topics. Administrators can either enter their own configuration or click **Use Default** to generate a baseline template and modify it to match their environment.



Plugin settings:

Setting	Description
Status	Enable or disable the Kafka plugin.
Log Level	Select the minimum severity level of logs to export: Critical, Error, Warning, Notification, Information, Debug.
Event	Enable or disable event log export.
Event Category	If event export is enabled, select one or more categories: Configuration, Admin, System, User, Health Check, SLB, LLB, GLB, Firewall, Enable All.
Traffic	Enable or disable traffic log export.
Traffic Category	If traffic export is enabled, select one or more categories: SLB, GLB, LLB, Enable All.
Security	Enable or disable security log export.

Setting	Description
Security Category	If security export is enabled, select one or more categories: DDoS, IP Reputation, WAF, GEO, AV, ZTNA, IPS, FW, Enable All. At least one category must be selected if Security is enabled.
Type	Plugin type. Only **Kafka** is supported in this release.
Configuration	Defines the Kafka output parameters using fluent-bit syntax. The **Use Default** button generates a baseline configuration, which can then be customized. • Name — Must be set to `kafka`. • Match — Determines which logs are exported. Use `*` to match all logs. • Brokers — One or more Kafka broker addresses in the format `<ip>:<port>`. Multiple brokers can be separated by commas, for example: `192.168.1.10:9092,192.168.1.11:9092`. Only IPv4 addresses are supported. • Topics — One or more Kafka topics to which logs will be published. Multiple topics can be specified, separated by commas. For example: `fad_traffic,fad_security`.

Example default configuration

```
[OUTPUT]
    Name         kafka
    Match        *
    Brokers      1.1.1.1:9092
    Topics       example_topic
```

Guidelines:

- Ensure that the broker IP address and port are reachable from FortiADC. Only plaintext port **9092** is supported; encrypted Kafka connections (port 9093) are not supported.
- Choose topic names that align with your monitoring pipeline. For example, use `fad_traffic` for traffic logs and `fad_security` for security logs.
- If multiple brokers are listed, fluent-bit uses them for redundancy and load distribution.
- Edit the configuration carefully — invalid syntax prevents logs from being exported.

For details on Kafka output configuration, see the Fluent Bit documentation.

Enhanced Syslog encryption via CLI - 7.6.1

FortiADC has strengthened Syslog security by introducing enhanced encryption through the TCP SSL protocol. New CLI options now allow administrators to apply either high and medium-level encryption algorithms for SSL communication, ensuring greater flexibility and control over security settings.

Two options are available:

- High-Medium Level: Offers up to 80 algorithm combinations for balanced security and performance.
- High Level: Provides 40 algorithm combinations for environments requiring the strongest encryption.

These improvements help secure sensitive log data and meet the demands of varied security environments.

This information is also available in the FortiADC 7.6.1 CLI Reference:

- config log setting remote

CLI updates in `config log setting remote`:

```
config log setting remote
   edit <name>
      set proto tcpssl
      set enc-algorithm {high-medium|high}
   next
end
```

`enc-algorithm`	The **enc-algorithm** option is available if **proto** is **tcpssl**. Select either the **high-medium** or **high** encryption algorithm options. The default option is **high-medium**. **Note**: Modifying the **enc-algorithm** setting triggers the initiation of a new SSL session negotiation with the syslog server, resulting in the disconnection of the current connection. The **High-Medium Level** contains the following 80 algorithm combinations: • TLS_AES_256_GCM_SHA384 • TLS_CHACHA20_POLY1305_SHA256 • TLS_AES_128_GCM_SHA256 • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-GCM-SHA384 • DHE-DSS-AES256-GCM-SHA384 • DHE-RSA-AES256-GCM-SHA384 • ECDHE-ECDSA-CHACHA20-POLY1305 • ECDHE-RSA-CHACHA20-POLY1305 • DHE-RSA-CHACHA20-POLY1305 • ECDHE-ECDSA-AES256-CCM8 • ECDHE-ECDSA-AES256-CCM

- DHE-RSA-AES256-CCM8
- DHE-RSA-AES256-CCM
- ECDHE-ECDSA-ARIA256-GCM-SHA384
- ECDHE-ARIA256-GCM-SHA384
- DHE-DSS-ARIA256-GCM-SHA384
- DHE-RSA-ARIA256-GCM-SHA384
- ADH-AES256-GCM-SHA384
- ECDHE-ECDSA-AES128-GCM-SHA256
- ECDHE-RSA-AES128-GCM-SHA256
- DHE-DSS-AES128-GCM-SHA256
- DHE-RSA-AES128-GCM-SHA256
- ECDHE-ECDSA-AES128-CCM8
- ECDHE-ECDSA-AES128-CCM
- DHE-RSA-AES128-CCM8
- DHE-RSA-AES128-CCM
- ECDHE-ECDSA-ARIA128-GCM-SHA256
- ECDHE-ARIA128-GCM-SHA256
- DHE-DSS-ARIA128-GCM-SHA256
- DHE-RSA-ARIA128-GCM-SHA256
- ADH-AES128-GCM-SHA256
- ECDHE-ECDSA-AES256-SHA384
- ECDHE-RSA-AES256-SHA384
- DHE-RSA-AES256-SHA256
- DHE-DSS-AES256-SHA256
- ECDHE-ECDSA-CAMELLIA256-SHA384
- ECDHE-RSA-CAMELLIA256-SHA384
- DHE-RSA-CAMELLIA256-SHA256
- DHE-DSS-CAMELLIA256-SHA256
- ADH-AES256-SHA256
- ADH-CAMELLIA256-SHA256
- ECDHE-ECDSA-AES128-SHA256
- ECDHE-RSA-AES128-SHA256
- DHE-RSA-AES128-SHA256
- DHE-DSS-AES128-SHA256
- ECDHE-ECDSA-CAMELLIA128-SHA256
- ECDHE-RSA-CAMELLIA128-SHA256
- DHE-RSA-CAMELLIA128-SHA256
- DHE-DSS-CAMELLIA128-SHA256
- ADH-AES128-SHA256
- ADH-CAMELLIA128-SHA256
- ECDHE-ECDSA-AES256-SHA

- ECDHE-RSA-AES256-SHA
- DHE-RSA-AES256-SHA
- DHE-DSS-AES256-SHA
- DHE-RSA-CAMELLIA256-SHA
- DHE-DSS-CAMELLIA256-SHA
- AECDH-AES256-SHA
- ADH-AES256-SHA
- ADH-CAMELLIA256-SHA
- ECDHE-ECDSA-AES128-SHA
- ECDHE-RSA-AES128-SHA
- DHE-RSA-AES128-SHA
- DHE-DSS-AES128-SHA
- DHE-RSA-CAMELLIA128-SHA
- DHE-DSS-CAMELLIA128-SHA
- AECDH-AES128-SHA
- ADH-AES128-SHA
- ADH-CAMELLIA128-SHA
- RSA-PSK-AES256-GCM-SHA384
- DHE-PSK-AES256-GCM-SHA384
- RSA-PSK-CHACHA20-POLY1305
- DHE-PSK-CHACHA20-POLY1305
- ECDHE-PSK-CHACHA20-POLY1305
- DHE-PSK-AES256-CCM8
- DHE-PSK-AES256-CCM
- RSA-PSK-ARIA256-GCM-SHA384
- DHE-PSK-ARIA256-GCM-SHA384
- AES256-GCM-SHA384
- AES256-CCM8
- AES256-CCM
- ARIA256-GCM-SHA384
- PSK-AES256-GCM-SHA384
- PSK-CHACHA20-POLY1305
- PSK-AES256-CCM8
- PSK-AES256-CCM
- PSK-ARIA256-GCM-SHA384
- RSA-PSK-AES128-GCM-SHA256
- DHE-PSK-AES128-GCM-SHA256
- DHE-PSK-AES128-CCM8
- DHE-PSK-AES128-CCM
- RSA-PSK-ARIA128-GCM-SHA256
- DHE-PSK-ARIA128-GCM-SHA256

- AES128-GCM-SHA256
- AES128-CCM8
- AES128-CCM
- ARIA128-GCM-SHA256
- PSK-AES128-GCM-SHA256
- PSK-AES128-CCM8
- PSK-AES128-CCM
- PSK-ARIA128-GCM-SHA256
- AES256-SHA256
- CAMELLIA256-SHA256
- AES128-SHA256
- CAMELLIA128-SHA256
- ECDHE-PSK-AES256-CBC-SHA384
- ECDHE-PSK-AES256-CBC-SHA
- SRP-DSS-AES-256-CBC-SHA
- SRP-RSA-AES-256-CBC-SHA
- SRP-AES-256-CBC-SHA
- RSA-PSK-AES256-CBC-SHA384
- DHE-PSK-AES256-CBC-SHA384
- RSA-PSK-AES256-CBC-SHA
- DHE-PSK-AES256-CBC-SHA
- ECDHE-PSK-CAMELLIA256-SHA384
- RSA-PSK-CAMELLIA256-SHA384
- DHE-PSK-CAMELLIA256-SHA384
- AES256-SHA
- CAMELLIA256-SHA
- PSK-AES256-CBC-SHA384
- PSK-AES256-CBC-SHA
- PSK-CAMELLIA256-SHA384
- ECDHE-PSK-AES128-CBC-SHA256
- ECDHE-PSK-AES128-CBC-SHA
- SRP-DSS-AES-128-CBC-SHA
- SRP-RSA-AES-128-CBC-SHA
- SRP-AES-128-CBC-SHA
- RSA-PSK-AES128-CBC-SHA256
- DHE-PSK-AES128-CBC-SHA256
- RSA-PSK-AES128-CBC-SHA
- DHE-PSK-AES128-CBC-SHA
- ECDHE-PSK-CAMELLIA128-SHA256
- RSA-PSK-CAMELLIA128-SHA256
- DHE-PSK-CAMELLIA128-SHA256

- AES128-SHA
- CAMELLIA128-SHA
- PSK-AES128-CBC-SHA256
- PSK-AES128-CBC-SHA
- PSK-CAMELLIA128-SHA256
- ECDHE-ECDSA-DES-CBC3-SHA
- ECDHE-RSA-DES-CBC3-SHA
- DHE-RSA-DES-CBC3-SHA
- DHE-DSS-DES-CBC3-SHA
- AECDH-DES-CBC3-SHA
- ADH-DES-CBC3-SHA
- ECDHE-PSK-3DES-EDE-CBC-SHA
- SRP-DSS-3DES-EDE-CBC-SHA
- SRP-RSA-3DES-EDE-CBC-SHA
- SRP-3DES-EDE-CBC-SHA
- RSA-PSK-3DES-EDE-CBC-SHA
- DHE-PSK-3DES-EDE-CBC-SHA
- DES-CBC3-SHA
- PSK-3DES-EDE-CBC-SHA

The **High Level** contains the following 40 algorithm combinations:

- TLS_AES_256_GCM_SHA384
- TLS_CHACHA20_POLY1305_SHA256
- TLS_AES_128_GCM_SHA256
- ECDHE-ECDSA-AES256-GCM-SHA384
- ECDHE-RSA-AES256-GCM-SHA384
- DHE-DSS-AES256-GCM-SHA384
- DHE-RSA-AES256-GCM-SHA384
- ECDHE-ECDSA-CHACHA20-POLY1305
- ECDHE-RSA-CHACHA20-POLY1305
- DHE-RSA-CHACHA20-POLY1305
- ECDHE-ECDSA-AES256-CCM8
- ECDHE-ECDSA-AES256-CCM
- DHE-RSA-AES256-CCM8
- DHE-RSA-AES256-CCM
- ECDHE-ECDSA-ARIA256-GCM-SHA384
- ECDHE-ARIA256-GCM-SHA384
- DHE-DSS-ARIA256-GCM-SHA384
- ADH-AES256-GCM-SHA384
- ECDHE-ECDSA-ARIA128-GCM-SHA256
- ECDHE-ARIA128-GCM-SHA256
- DHE-DSS-ARIA128-GCM-SHA256

- DHE-RSA-ARIA128-GCM-SHA256
- ECDHE-ECDSA-AES256-SHA384
- ECDHE-RSA-AES256-SHA384
- DHE-RSA-AES256-SHA256
- DHE-DSS-AES256-SHA256
- ECDHE-ECDSA-CAMELLIA256-SHA384
- ECDHE-RSA-CAMELLIA256-SHA384
- DHE-RSA-CAMELLIA256-SHA256
- DHE-DSS-CAMELLIA256-SHA256
- ADH-AES256-SHA256
- ADH-CAMELLIA256-SHA256
- ECDHE-ECDSA-CAMELLIA128-SHA256
- ECDHE-RSA-CAMELLIA128-SHA256
- DHE-RSA-CAMELLIA128-SHA256
- DHE-DSS-CAMELLIA128-SHA256
- ADH-CAMELLIA128-SHA256
- RSA-PSK-AES256-GCM-SHA384
- DHE-PSK-AES256-GCM-SHA384
- RSA-PSK-CHACHA20-POLY1305
- DHE-PSK-CHACHA20-POLY1305
- ECDHE-PSK-CHACHA20-POLY1305
- DHE-PSK-AES256-CCM8
- DHE-PSK-AES256-CCM
- RSA-PSK-ARIA256-GCM-SHA384
- DHE-PSK-ARIA256-GCM-SHA384
- AES256-GCM-SHA384
- AES256-CCM8
- AES256-CCM
- ARIA256-GCM-SHA384
- PSK-AES256-GCM-SHA384
- PSK-CHACHA20-POLY1305
- PSK-AES256-CCM8
- PSK-AES256-CCM
- PSK-ARIA256-GCM-SHA384
- RSA-PSK-ARIA128-GCM-SHA256
- DHE-PSK-ARIA128-GCM-SHA256
- ARIA128-GCM-SHA256
- PSK-ARIA128-GCM-SHA256
- AES256-SHA256
- CAMELLIA256-SHA256
- CAMELLIA128-SHA256

	• ECDHE-PSK-AES256-CBC-SHA384 • RSA-PSK-AES256-CBC-SHA384 • DHE-PSK-AES256-CBC-SHA384 • ECDHE-PSK-CAMELLIA256-SHA384 • RSA-PSK-CAMELLIA256-SHA384 • DHE-PSK-CAMELLIA256-SHA384 • PSK-AES256-CBC-SHA384 • PSK-CAMELLIA256-SHA384 • ECDHE-PSK-CAMELLIA128-SHA256 • RSA-PSK-CAMELLIA128-SHA256 • DHE-PSK-CAMELLIA128-SHA256 • PSK-CAMELLIA128-SHA256

Syslog support for IPv6 FQDNs - 7.6.1

FortiADC now supports sending logs to syslog servers using IPv6 addresses resolved from FQDNs. Previously, only FQDNs resolving to IPv4 addresses were supported for syslog server communication. This enhancement improves flexibility by allowing users to configure syslog servers with IPv6 FQDNs.

GUI

The FortiADC 7.6 release includes new features and enhancements in the **GUI**:

Improved firmware upgrade process with progress tracking 7.6.1 on page 198

FortiADC has improved the visibility of the firmware upgrade process by introducing a new progress bar that displays the upload percentage. Users will find enhanced information throughout the upgrade process, detailing current activities and setting clear expectations for what to anticipate.

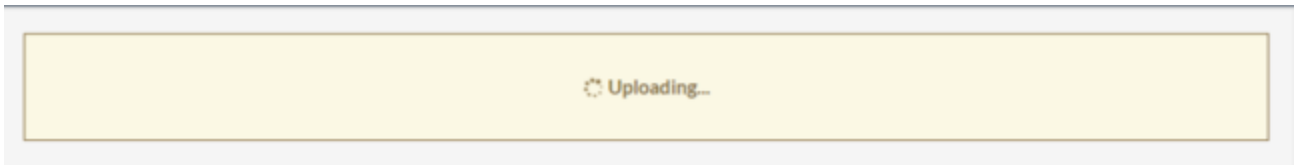
Improved firmware upgrade process with progress tracking - 7.6.1

FortiADC has improved the visibility of the firmware upgrade process by introducing a new progress bar that displays the upload percentage. Users will find enhanced information throughout the upgrade process, detailing current activities and setting clear expectations for what to anticipate.

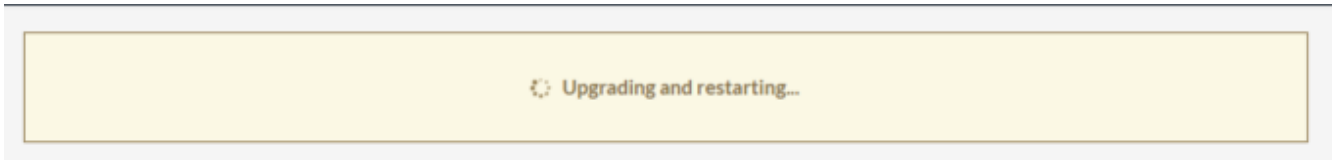
Previously, initiating a firmware upgrade resulted in a dialog message that provided minimal information and did not effectively convey the progress of the upgrade. In contrast, the new progress tracking feature displays the upload percentage in real time. Once the percentage reaches 100%, this indicates upload completion and signals a transition to the next steps, which include system updates and a restart.

Previous Firmware Upgrade Progress Dialog

After upgrade initiation and during the upload:

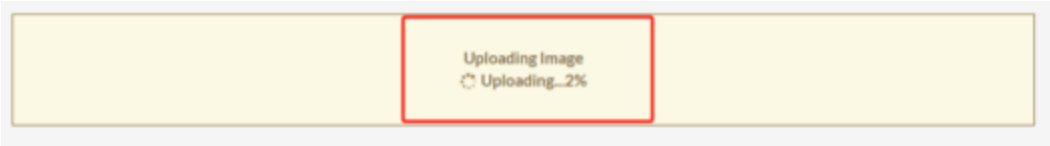


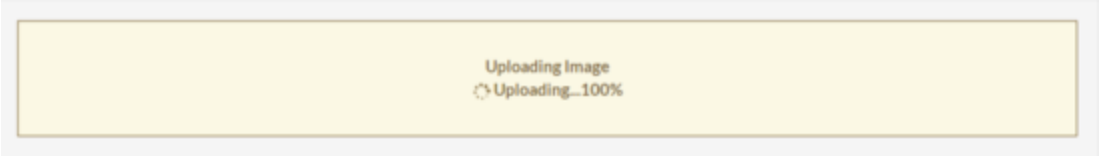
System update and restart:



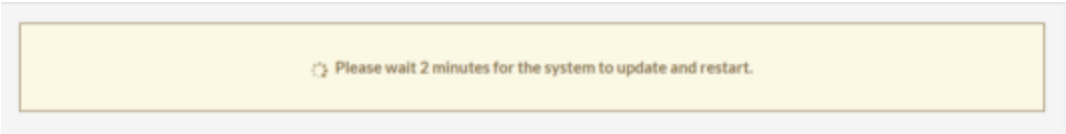
New Firmware Upgrade Progress Dialog

After upgrade initiation and during the upload:





System update and restart:



Platform

The FortiADC 7.6 release includes new features and enhancements to support **Platforms**:

Expanded Local Certificate Group Member Limit 7.6.4 on page 201

FortiADC 7.6.4 increases the maximum number of Local Certificate Group Members from 256 to 1024. This change provides greater flexibility for large-scale deployments that manage extensive sets of local certificates within a single group.

OpenSSL Upgrade to 3.1.8 7.6.4 on page 202

FortiADC 7.6.4 upgrades the OpenSSL library to version 3.1.8 to align with the latest security compliance requirements and upstream fixes.

OCI DRCC support 7.6.4 on page 203

FortiADC-VM is supported in OCI Dedicated Region Cloud@Customer (DRCC). For more information, see Dedicated Region Cloud@Customer.

New FortiADC-VMUL License Model and Expanded VDOM Support for Virtual Appliances 7.6.3 on page 204

FortiADC 7.6.3 introduces the **FortiADC-VMUL** license model, a new virtual appliance option designed to support large-scale deployments requiring high vCPU and VDOM capacity. Unlike other VM models, VMUL supports deployment on systems with significantly higher compute resources—for example, configurations with 64 vCPUs—providing improved scalability and flexibility for enterprise and service provider environments.

As part of this enhancement, the **maximum number of supported VDOMs** has been expanded for several FortiADC virtual machine models. Previously, VM04 and VM08 were limited to 10 VDOMs, but with FortiADC 7.6.3, VDOM limits now align more closely with system memory allocation. **VMUL** supports up to **90 VDOMs**, matching the capabilities of the high-end VM32 model.

Data Partition Expansion 7.6.2 on page 205

In FortiADC7.6.5, the data partition size is expanded to support larger firmware images and new feature implementations. The existing 400MB partition on most platforms has been a limiting factor for future enhancements. This update increases the partition size to the maximum allowable capacity based on the system's hardware, ensuring compatibility with upcoming releases.

Support FortiFlex Token in User Data for Public Cloud BYOL 7.6.2 on page 207

This enhancement enables FortiADC to parse and apply a FortiFlex license token provided through cloud-init user data during the initial deployment of BYOL instances on public cloud platforms—Azure, AWS, and GCP.

Instance type support for AWS/Azure/GCP 7.6.1 on page 208

FortiADC has expanded its compatibility to support a broader range of instance types on AWS, Azure, and GCP cloud platforms. This enhancement includes compatibility with additional CPU architectures, providing greater flexibility in

deployment options.

FortiFlex support for cloud-init in Proxmox (KVM) on page 208

FortiADC now supports FortiADC deployment with cloud-init in KVM on Proxmox using a valid FortiFlex license.

Expanded Local Certificate Group Member Limit - 7.6.4

FortiADC 7.6.4 increases the maximum number of Local Certificate Group Members from 256 to 1024. This change provides greater flexibility for large-scale deployments that manage extensive sets of local certificates within a single group.

This information is also available in the FortiADC 7.6.4 Administration Guide:

- Maximum Configuration Values

In environments where multiple certificates are required for diverse applications, services, or domains, the previous limit of 256 members could restrict configuration options. By expanding the capacity to 1024, FortiADC now supports larger deployments without requiring administrators to divide certificates across multiple groups, simplifying management and reducing configuration complexity.

The following lists the maximum certificate object values for all FortiADC platforms:

Certificate Object	Maximum Count
CA	1024
CA Group	256
CA Group Members	256
Intermediate CA	1024
Intermediate CA Group	256
Intermediate CA Group Members	256
Local Certificate	3072
Remote Certificate	1024
Local Certificate Group	3072
Local Certificate Group Members	**1024** (increased from 256)
CRL	1024
OCSP	1024
OCSP Stapling	256
Certificate Verify	256

OpenSSL Upgrade to 3.1.8 - 7.6.4

FortiADC 7.6.4 upgrades the OpenSSL library to version 3.1.8 to align with the latest security compliance requirements and upstream fixes.

OCI DRCC support - 7.6.4

FortiADC-VM is supported in OCI Dedicated Region Cloud@Customer (DRCC). For more information, see Dedicated Region Cloud@Customer.

New FortiADC-VMUL License Model and Expanded VDOM Support for Virtual Appliances - 7.6.3

FortiADC 7.6.3 introduces the **FortiADC-VMUL** license model, a new virtual appliance option designed to support large-scale deployments requiring high vCPU and VDOM capacity. Unlike other VM models, VMUL supports deployment on systems with significantly higher compute resources—for example, configurations with 64 vCPUs—providing improved scalability and flexibility for enterprise and service provider environments.

As part of this enhancement, the **maximum number of supported VDOMs** has been expanded for several FortiADC virtual machine models. Previously, VM04 and VM08 were limited to 10 VDOMs, but with FortiADC 7.6.3, VDOM limits now align more closely with system memory allocation. **VMUL** supports up to **90 VDOMs**, matching the capabilities of the high-end VM32 model.

The updated VDOM support matrix based on VM model and recommended memory size is as follows:

VM Model	Maximum VDOMs	Recommended System Memory
VM01	10	8 GB
VM02	10	8 GB
VM04	25	≥ 16 GB
VM08	45	≥ 32 GB
VM16	60	≥ 64 GB
VM32	90	≥ 128 GB
VMUL	**90**	**≥ 128 GB**

Note: While VMUL removes the CPU core limit (supports unlimited vCPUs), VDOM capacity remains primarily constrained by available system memory. To achieve the maximum number of supported VDOMs, ensure the FortiADC VM is provisioned with sufficient RAM according to the table above.

The maximum configuration values for VMUL are identical to those of the VM32 model. For more details, refer to the Maximum Configuration Values section of the FortiADC Administration Guide.

Data Partition Expansion - 7.6.2

In FortiADC 7.6.2, the data partition size is expanded to support larger firmware images and new feature implementations. The existing 200MB partition on most platforms has been a limiting factor for future enhancements. This update increases the partition size to the maximum allowable capacity based on the system's hardware, ensuring compatibility with upcoming releases.

This expansion applies only to hardware appliances and private cloud instances. Public cloud images will maintain the current partition size.

Key Enhancements

Benefit	Details
Increased Storage Capacity	Expands the data partition from 200MB to the maximum available space on supported hardware and private cloud platforms, allowing more room for firmware images, logs, and feature enhancements.
Seamless Future Upgrades	Eliminates storage-related upgrade failures, ensuring smooth transitions to newer firmware versions.
Enhanced System Longevity	Prevents storage limitations from restricting feature adoption, extending the platform’s scalability and maintainability.

Upgrade Considerations and Limitations

Expanding the data partition in FortiADC 7.6.2 introduces specific upgrade requirements and operational impacts. Administrators must follow a structured upgrade path to ensure a smooth transition while considering potential limitations.

Mandatory Upgrade Path

Upgrading beyond 7.6.2 (such as 7.6.3) requires installing 7.6.2 first. This ensures that the partition expansion is completed before applying a newer firmware version. Any attempt to upgrade directly to a post-7.6.2 release without first installing 7.6.2 will be blocked.

Longer Upgrade Duration

Because the upgrade includes a partition resizing process, the total upgrade time is longer than a typical firmware update. The duration depends on the platform and storage configuration, so administrators should plan accordingly to minimize downtime.

Irreversible Partition Change

Once the partition is expanded in 7.6.2, it cannot be reverted by downgrading to a previous firmware version. The partition remains in its expanded state even if an earlier release is installed. Before upgrading, ensure that your environment is compatible with 7.6.2 and later versions.

HA Cluster Upgrade Best Practices

For HA (High Availability) clusters, follow these guidelines to prevent service disruption:

- Do not toggle HA mode during the upgrade, as this can lead to downtime for all nodes in the process.
- Upgrade each node individually, rather than upgrading all nodes at once, to minimize potential issues.
- For Active-Passive (A-P) clusters, start by upgrading the secondary node. Once the secondary node is fully operational, proceed to upgrade the primary node to ensure continued availability.

Verifying Successful Data Partition Expansion

After performing an upgrade to FortiADC version 7.6.2 or later, the data partition will be expanded to provide increased storage capacity. To verify that the expansion has been successfully applied, you can use the following CLI command:

```
diagnose hardware get sysinfo partition
```

This command returns detailed information on the system's storage partitions, including the size of the data partition. By comparing the partition size values before and after the upgrade, you can confirm that the partition has been expanded as expected.

Example output comparison:

Platform	Before Upgrade to 7.6.2	After Upgrade to 7.6.2

Hardware (1200F)	`FortiADC-1200F # diagnose hardware get sysinfo partition`
	`Disk /dev/sda: 240.0 GB, 240057409536 bytes`
	`1 heads, 63 sectors/track, 7442256 cylinders`
	`Units = cylinders of 63 * 512 = 32256 bytes`
	`Device Boot Start End Blocks Id System`
	`/dev/sda1 * 2 7442256 234431032+ 83 Linux`
	`Partition 1 does not end on cylinder boundary`
	`Disk /dev/sdb: 2013 MB, 2013265920 bytes`
	`1 heads, 62 sectors/track, 63421 cylinders`
	`Units = cylinders of 62 * 512 = 31744 bytes`
	`Device Boot Start End Blocks Id System`
	`/dev/sdb1 * 197 6649 200000 83 Linux`
	`Partition 1 does not end on cylinder boundary`
	`/dev/sdb2 6649 13100 200000 83 Linux`
	`Partition 2 does not end on cylinder boundary`
	`/dev/sdb3 13100 45358 1000000 83 Linux`
	`Partition 3 does not end on cylinder boundary`
		`FortiADC-1200F # diagnose hardware get sysinfo partition`
		`Disk /dev/sda: 240.0 GB, 240057409536 bytes`
		`1 heads, 63 sectors/track, 7442256 cylinders`
		`Units = cylinders of 63 * 512 = 32256 bytes`
		`Device Boot Start End Blocks Id System`
		`/dev/sda1 * 2 7442256 234431032+ 83 Linux`
		`Partition 1 does not end on cylinder boundary`
		`Disk /dev/sdb: 2013 MB, 2013265920 bytes`
		`1 heads, 62 sectors/track, 63421 cylinders`
		`Units = cylinders of 62 * 512 = 31744 bytes`
		`Device Boot Start End Blocks Id System`
		`/dev/sdb1 197 13100 400000 83 Linux`
		`Partition 1 does not end on cylinder boundary`
		`/dev/sdb2 * 13100 26004 400000 83 Linux`
		`Partition 2 does not end on cylinder boundary`
		`/dev/sdb3 26004 58262 1000000 83 Linux`
		`Partition 3 does not end on cylinder boundary`

Virtual Machine	`FortiADC-VM # diagnose hardware get sysinfo partition`
	`Disk /dev/sda: 2147 MB, 2147483648 bytes`
	`1 heads, 63 sectors/track, 66576 cylinders`
	`Units = cylinders of 63 * 512 = 32256 bytes`
	`Device Boot Start End Blocks Id System`
	`/dev/sda1 * 194 6543 200000 83 Linux`
	`Partition 1 does not end on cylinder boundary`
	`/dev/sda2 6543 12892 200000 83 Linux`
	`Partition 2 does not end on cylinder boundary`
	`/dev/sda3 12892 25591 400000 83 Linux`
	`Partition 3 does not end on cylinder boundary`
	`Disk /dev/sdb: 32.2 GB, 32212254720 bytes`
	`1 heads, 63 sectors/track, 998643 cylinders`
	`Units = cylinders of 63 * 512 = 32256 bytes`
	`Device Boot Start End Blocks Id System`
	`/dev/sdb1 * 2 998644 31457248+ 83 Linux`
	`Partition 1 does not end on cylinder boundary`
		`FortiADC-VM # diagnose hardware get sysinfo partition`
		`Disk /dev/sda: 2147 MB, 2147483648 bytes`
		`1 heads, 63 sectors/track, 66576 cylinders`
		`Units = cylinders of 63 * 512 = 32256 bytes`
		`Device Boot Start End Blocks Id System`
		`/dev/sda1 194 22416 700000 83 Linux`
		`Partition 1 does not end on cylinder boundary`
		`/dev/sda2 * 22416 44638 700000 83 Linux`
		`Partition 2 does not end on cylinder boundary`
		`/dev/sda3 44639 57337 400000 83 Linux`
		`Partition 3 does not end on cylinder boundary`
		`Disk /dev/sdb: 32.2 GB, 32212254720 bytes`
		`1 heads, 63 sectors/track, 998643 cylinders`
		`Units = cylinders of 63 * 512 = 32256 bytes`
		`Device Boot Start End Blocks Id System`
		`/dev/sdb1 * 2 998644 31457248+ 83 Linux`
		`Partition 1 does not end on cylinder boundary`

Support FortiFlex Token in User Data for Public Cloud BYOL - 7.6.2

This enhancement enables FortiADC to parse and apply a FortiFlex license token provided through cloud-init user data during the initial deployment of BYOL instances on public cloud platforms—Azure, AWS, and GCP.

Previously, cloud-init user data supported only license files for automated license activation. With this update, FortiADC can now recognize and apply a FortiFlex token embedded in the user data input. During the initial boot sequence, cloud-init extracts the token and applies it automatically, allowing the VM to start with a valid FortiFlex license pre-installed.

This enhancement eliminates the need to run the manual CLI command `execute vm-license <token>` post-deployment and streamlines automated provisioning workflows for FortiFlex BYOL deployments in public cloud environments.

Instance type support for AWS/Azure/GCP - 7.6.1

FortiADC has expanded its compatibility to support a broader range of instance types on AWS, Azure, and GCP cloud platforms. This enhancement includes compatibility with additional CPU architectures, providing greater flexibility in deployment options.

FortiFlex support for cloud-init in Proxmox (KVM)

With a valid FortiFlex license, you can now deploy your FortiADC instance with cloud-init in KVM on Proxmox using a "cloud-init config drive".

Originally developed for OpenStack and other cloud environments, the cloud-init config drive is now integrated into FortiADC-VM/KVM. This functionality extends to deployments within VMware vCenter, standalone ESX, and KVM environments such as Proxmox. The config drive allows administrators to inject day-zero configuration scripts, FortiADC-VM licenses, and FortiFlex licenses directly into the FortiADC during the initial boot sequence. This feature streamlines the provisioning process by embedding critical configurations and licensing data within the initial boot parameters, ensuring immediate operational readiness and compliance. By automating these initial setup tasks, the Cloud-Init config drive significantly reduces deployment times and administrative overhead, while enhancing the flexibility and scalability of FortiADC implementations.

Troubleshooting

The FortiADC 7.6 release includes new features and enhancements to support Troubleshooting:

Health Check debug log enhancement in CLI on page 212

The new `diagnose debug pool-member-debug` command allows you to apply filtering at the VDOM, pool, and real server levels to specify the real server(s) you want to view the health check debug information of, instead of viewing debug messages for all real servers. With this enhancement, you can now focus on pertinent data that can help you troubleshoot the real server issues instead of being overwhelmed by irrelevant data.

New CLI Commands for Virtual Server and Pool Statistics - 7.6.4

FortiADC 7.6.4 introduces the new diagnostic command `diagnose server-load-balance vs_stat`, which allows administrators to view virtual server and real server pool statistics directly from the CLI. This command provides visibility into health states, last state changes, and session distribution, eliminating the need for manual health check scripts or reliance solely on the GUI.

Key capabilities:

- Display the availability state of virtual servers and pool members.
- Show the last state change (UP/DOWN transitions) with timestamps.
- Retrieve load balancing pool session distribution and performance statistics.
- Support for querying statistics across all VDOMs or scoped to a specific VDOM, virtual server, or real server.

Command usage

```
diagnose server-load-balance vs_stat vdom <vdom_name> vs <vs_name> rs <rs_name>
```

Parameter	Description
vs_stat	Displays virtual server and real server statistics. Required subcommand.
vdom <vdom_name>	Limits the output to the specified VDOM. If omitted, statistics for all VDOMs are shown.
vs <vs_name>	Limits the output to the specified virtual server within the given VDOM. If omitted, all virtual servers in the VDOM are shown.
rs <rs_name>	Limits the output to the specified real server within the virtual server's pool. If omitted, all pool members are shown.

Example

```
FortiADC-VM # diagnose server-load-balance vs_stat vdom root vs vs1
vdom name is root
    virtual-server vs1, availability: HEALTHY
    APP Response: 0
    Client RTT: 1
    Concurrent Connections: 10
    Inbound Throughput: 2611620
    Outbound Throughput: 1189284145
    Requests: 2569
    Server RTT: 1
    Total Sessions: 0

    real-server-pool: rsp1
    id:1, hc_state:UP, last-state-change-time: 2025-05-06 22:00:00
        APP Response: 0
```

```
        Concurrent Connections: 10
        Inbound Throughput: 2814350
        Outbound Throughput: 1281520059
        Requests: 2334
        Server RTT: 1
        Total Sessions: 0
    id:2, hc_state:UP, last-state-change-time: 2025-05-06 22:00:00
        APP Response: 0
        Concurrent Connections: 10
        Inbound Throughput: 2814350
        Outbound Throughput: 1281520059
        Requests: 2334
        Server RTT: 1
        Total Sessions: 0
```

This output shows both the virtual server statistics and the corresponding real server pool member statistics, including health status and the last state change time.

Health Check debug log enhancement in CLI

The new `diagnose debug pool-member-debug` command allows you to apply filtering at the VDOM, pool, and real server levels to specify the real server(s) you want to view the health check debug information of, instead of viewing debug messages for all real servers. With this enhancement, you can now focus on pertinent data that can help you troubleshoot the real server issues instead of being overwhelmed by irrelevant data.

After applying these filters, you can then enable debug in the health check module to view the applicable debug information for the specified real server(s).

This information is also available in the FortiADC 7.6.0 CLI Reference Guide:

- diagnose debug pool-member-debug

New `diagnose debug pool-member-debug`:

```
diagnose debug pool-member-debug add vdom <vdom> pool <pool_name> member <id>
diagnose debug pool-member-debug list
diagnose debug pool-member-debug clear
```

`add`	Add entries to specify the real servers you want to view the debug information of by providing the following information: • vdom <vdom> — You are required to specify the VDOM. If only the VDOM is specified, then each real server under this VDOM will be added as an entry. • pool <pool_name> — Specify the name of the real server pool. This real server pool must exist in the specified VDOM. • member <id> — Specify the ID of the real server pool member within the specified real server pool. This real server pool must exist in the specified VDOM.
`list`	Lists all the added entries.
`clear`	Clears all entries. This will revert the health check debug to print messages for all real servers.

Examples:

```
FortiADC-VM #diagnose debug pool-member-debug add vdom test

FortiADC-VM #diagnose debug pool-member-debug list
Pool member debug is enabled.
3 entries
vdom:test, pool:RSP3, member:1
vdom:test, pool:RSP2, member:1
vdom:test, pool:RSP2, member:2
```

```
FortiADC-VM #diagnose debug pool-member-debug add vdom test pool RSP2
```

```
FortiADC-VM #diagnose debug pool-member-debug list
Pool member debug is enabled.
2 entries
vdom:test, pool:RSP2, member:1
vdom:test, pool:RSP2, member:2
```

```
FortiADC-VM #diagnose debug pool-member-debug add vdom test pool RSP2 member 2

FortiADC-VM #diagnose debug pool-member-debug list
Pool member debug is enabled.
1 entries
vdom:test, pool:RSP2, member:2
```

www.fortinet.com

Copyright© 2025 Fortinet, Inc. All rights reserved. Fortinet®, FortiGate®, FortiCare® and FortiGuard®, and certain other marks are registered trademarks of Fortinet, Inc., and other Fortinet names herein may also be registered and/or common law trademarks of Fortinet. All other product or company names may be trademarks of their respective owners. Performance and other metrics contained herein were attained in internal lab tests under ideal conditions, and actual performance and other results may vary. Network variables, different network environments and other conditions may affect performance results. Nothing herein represents any binding commitment by Fortinet, and Fortinet disclaims all warranties, whether express or implied, except to the extent Fortinet enters a binding written contract, signed by Fortinet's Chief Legal Officer, with a purchaser that expressly warrants that the identified product will perform according to certain expressly-identified performance metrics and, in such event, only the specific performance metrics expressly identified in such binding written contract shall be binding on Fortinet. For absolute clarity, any such warranty will be limited to performance in the same ideal conditions as in Fortinet's internal lab tests. Fortinet disclaims in full any covenants, representations, and guarantees pursuant hereto, whether express or implied. Fortinet reserves the right to change, modify, transfer, or otherwise revise this publication without notice, and the most current version of the publication shall be applicable.