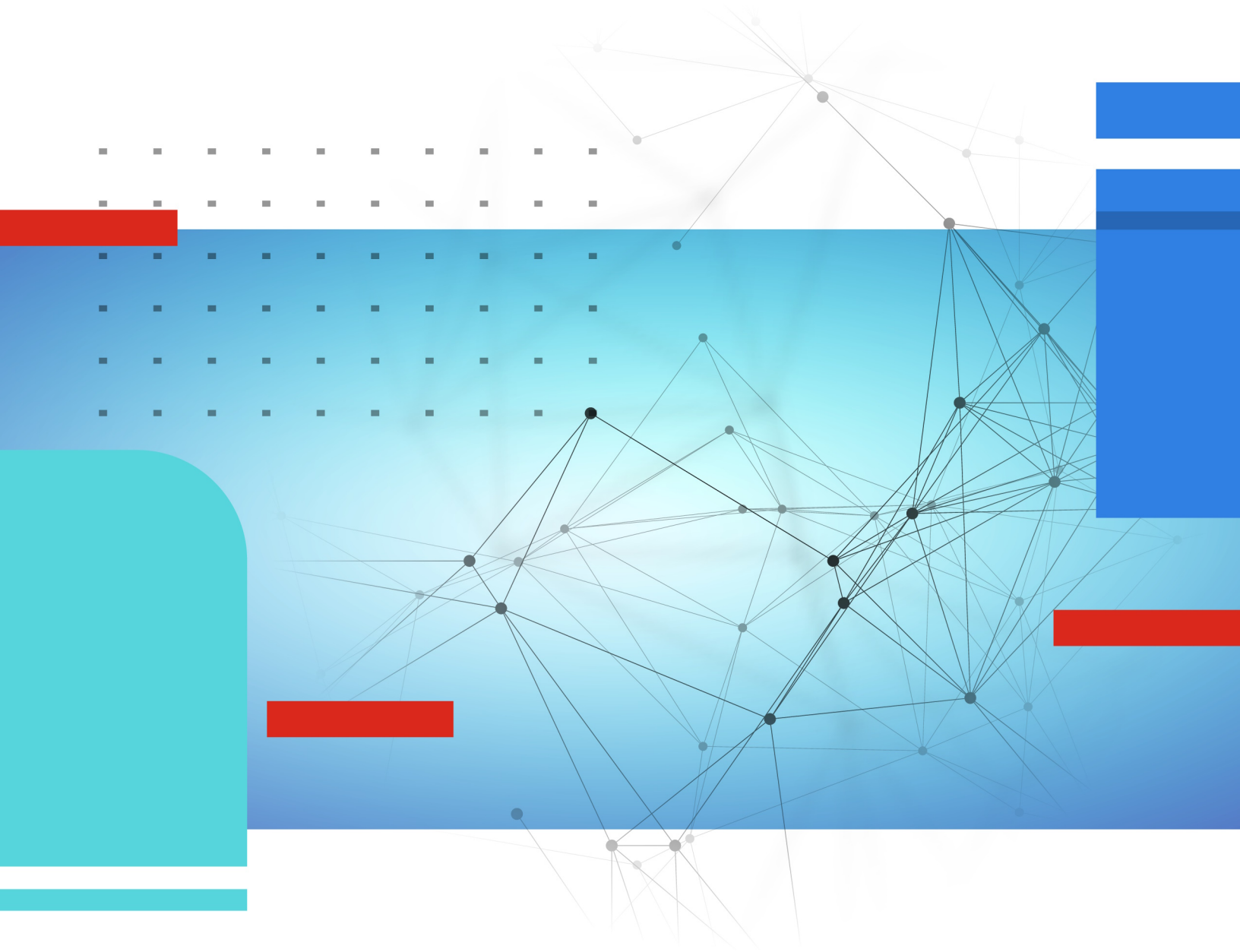
New Features

FortiADC 8.0.3



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

April 3, 2026
FortiADC 8.0.3 New Features

TABLE OF CONTENTS

Change Log

Date	Change Description
April 3, 2026	Added 8.0.3 new features.

Overview

This guide provides details of new features introduced in FortiADC 8.0. For each feature, the guide provides detailed information on configuration, requirements, and limitations, as applicable. Features are organized into the following sections:

Index

The following index provides a list of all new features added to FortiADC 8.0. 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:

8.0.0

Application Access Manager

Web Application Firewall

System

Server Load Balance

Network Security

GUI

Platform

Troubleshooting

8.0.1

Application Access Manager

Security Fabric

System

Server Load Balance

Global Load Balance

Network Security

Log & Report

Platform

8.0.2

FortiAI

- FortiAI Assistant for FortiADC 8.0.2 on page 15

Server Load Balance

- Increased Capacity for Content Routing and Health Check Objects 8.0.2 on page 133

Network

- VXLAN Support for Kubernetes Calico CNI 8.0.2 on page 121

8.0.3

FortiAI

Application Access Manager

Web Application Firewall

Security Fabric

Server Load Balance

Log & Report

GUI

FortiAI

The FortiADC 8.0 release includes new features and enhancements in **FortiAI**:

FortiAI Assistant Log Analysis Enhancements 8.0.3 on page 13

You can now use the **FortiAI Assistant** to perform deep-dive analysis on individual log entries for more granular operational and security insights. This update expands the assistant's capabilities to evaluate specific events within your traffic and security logs, transforming raw data into clear, actionable intelligence for faster troubleshooting and threat response.

FortiAI Assistant for FortiADC 8.0.2 on page 15

FortiADC 8.0.2 introduces FortiAI Assistant, an AI-powered assistant embedded directly in the FortiADC GUI. FortiAI Assistant enables administrators to use natural-language queries to obtain configuration guidance, inspect virtual server behavior, analyze system and security logs, and convert text-based prompts into Lua scripts through Text-to-Script. By allowing users to ask questions instead of manually navigating multiple pages or searching documentation, FortiAI Assistant streamlines common operational and troubleshooting workflows while keeping all configuration changes under administrator control.

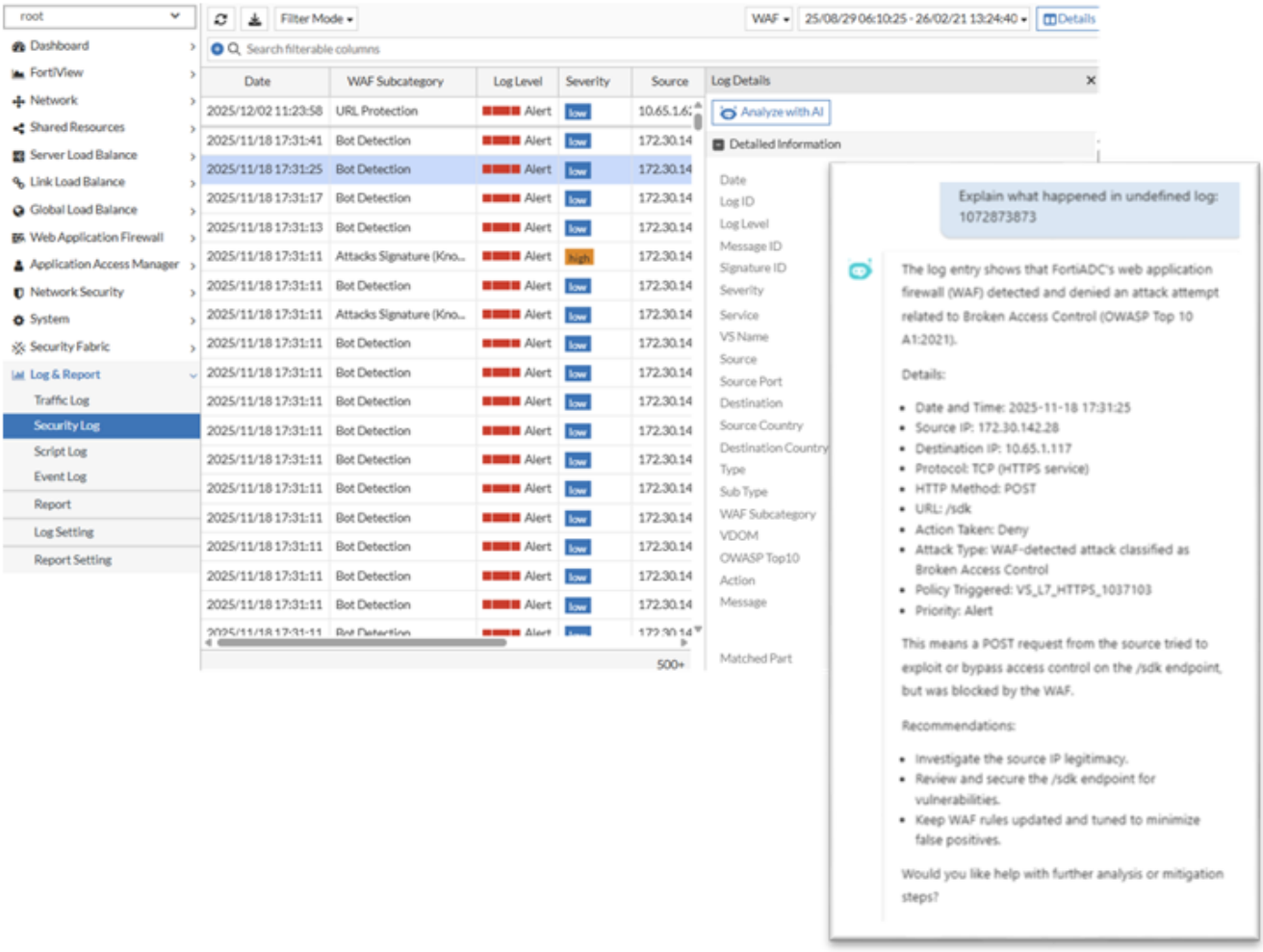
FortiAI Assistant Log Analysis Enhancements 8.0.3

You can now use the **FortiAI Assistant** to perform deep-dive analysis on individual log entries for more granular operational and security insights. This update expands the assistant's capabilities to evaluate specific events within your traffic and security logs, transforming raw data into clear, actionable intelligence for faster troubleshooting and threat response.

Integrated Workflow

The analysis capability is built directly into the log management interface for a seamless experience. When you view the details of a specific log, you can click the **Analyze with AI** button at the top of the Log Details sidebar. The FortiAI Assistant dialog window automatically launches and submits a contextual query based on the log's unique identifier.

For example, if you are reviewing a traffic log with Message ID 1071529342, clicking the button triggers the query: *"Explain what happened in traffic log: 1071529342"*. The assistant then processes the fields within that specific entry and displays the analysis result in the side pane.



Contextual Security Analysis

For security logs, the FortiAI Assistant analyzes individual threat events to provide immediate clarity on detected risks:

- **Risk Evaluation**: The assistant examines the specific details of a security event to deliver a focused risk assessment.
- **Response Guidance**: FortiAI suggests practical steps to mitigate identified threats based on the unique context of the log entry.
- **Event Interpretation**: The assistant translates complex security fields into plain-language summaries to explain the significance of an attack or anomaly.

Operational and Traffic Insights

For traffic logs, the assistant helps streamline network maintenance and performance tuning:

- **Behavioral Identification**: The assistant evaluates traffic patterns within a single log to help identify performance deviations.
- **Troubleshooting Assistance**: FortiAI correlates data points within the log record to help pinpoint the likely root cause of connectivity or performance issues.
- **Actionable Summaries**: By converting technical log data into readable insights, the assistant reduces the time required to interpret networking bottlenecks.

FortiAI Assistant for FortiADC 8.0.2

FortiADC 8.0.2 introduces **FortiAI Assistant**, an AI-powered assistant embedded directly in the FortiADC GUI. FortiAI Assistant enables administrators to use natural-language queries to obtain configuration guidance, inspect virtual server behavior, analyze system and security logs, and convert text-based prompts into Lua scripts through **Text-to-Script**. By allowing users to ask questions instead of manually navigating multiple pages or searching documentation, FortiAI Assistant streamlines common operational and troubleshooting workflows while keeping all configuration changes under administrator control.

This information is also available in the FortiADC 8.0.2 Administration Guide:

- FortiAI Assistant

FortiAI Assistant is built on **FortiAI**, Fortinet's generative AI platform that uses large language models trained on Fortinet product knowledge. Through this integration, FortiADC can combine documented product behavior with contextual analysis of local system data to deliver FortiADC-specific responses directly within the management interface.

FortiAI Assistant availability depends on the FortiCare support level associated with the FortiADC license. VM licenses **without an active FortiCare support subscription** are not entitled to use FortiAI Assistant. FortiAI login is supported only when the license includes **Premium Support** or **Elite Support**.

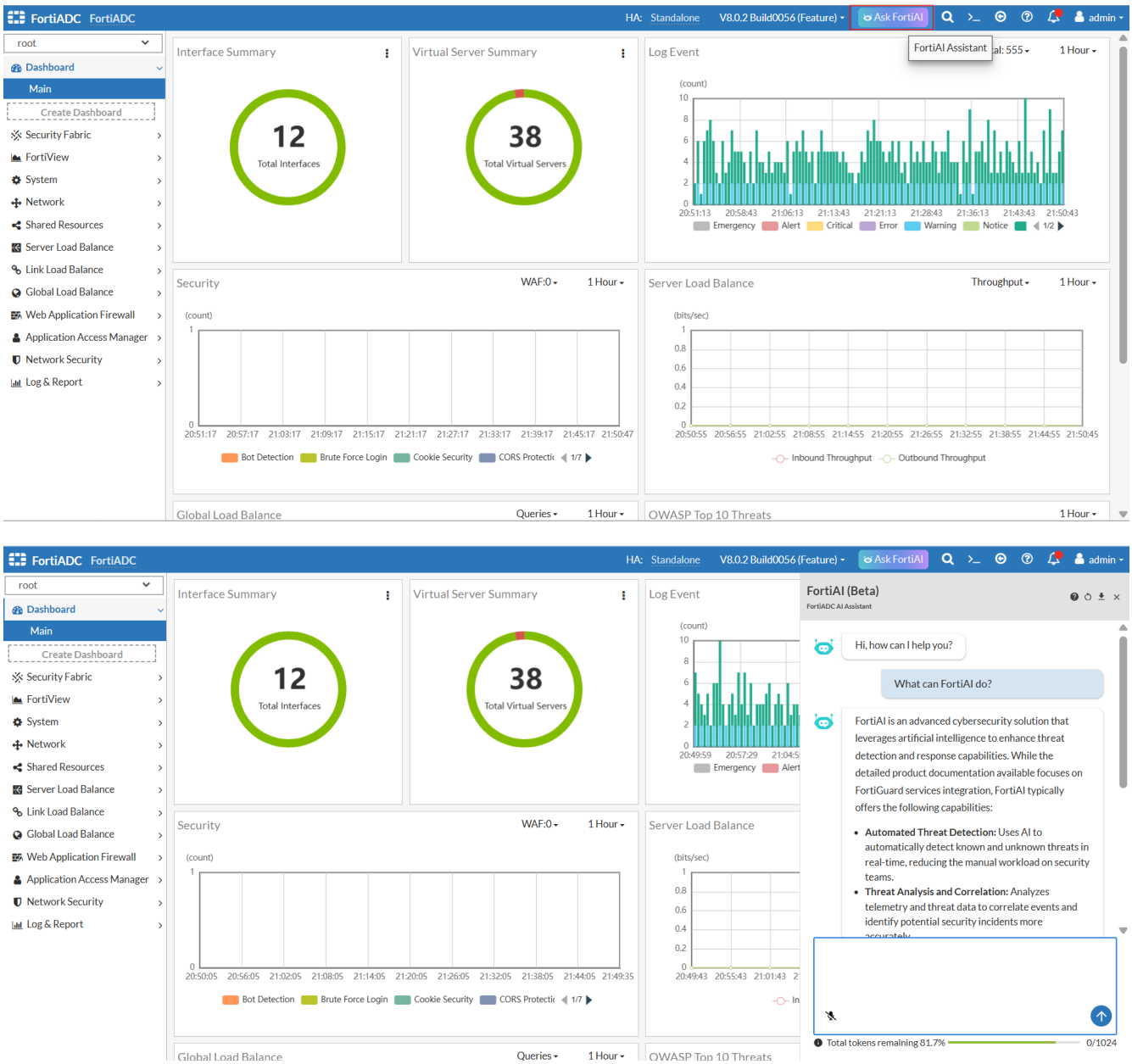
FortiAI Assistant is available on supported FortiADC platforms in FortiADC 8.0.2, including:

- FortiADC hardware models supported in 8.0.2
- FortiADC VM deployments on supported private hypervisors
- Public cloud BYOL deployments (all 8.0.2 supported BYOL instances except FortiADC-XENAWS)

FortiAI Assistant is not supported on on-demand or PAYG cloud instances, trial licenses, or VM deployments using Flex-VM licenses. For complete and up-to-date platform support information, refer to the FortiADC support matrix.

User interface and access

FortiAI Assistant is accessed from the **Ask FortiAI** icon in the FortiADC GUI header. Selecting this icon opens a dockable chat panel that remains available as you navigate configuration, monitoring, and log pages. The panel includes controls to submit prompts, reset the current session, download chat history, and monitor token usage.



The FortiAI Assistant panel can be resized and remains accessible across most FortiADC GUI pages, allowing administrators to interact with the assistant without disrupting their current task.

Access to FortiAI Assistant is permission-based and must be explicitly granted to administrator accounts. FortiAI Assistant is available only in **non-Global VDOMs** and supports **local administrator accounts only**. The **FortiAI User permission** can be assigned to a maximum of **three administrator accounts**, and only the **Global administrator** can grant or revoke FortiAI User permissions.

Architecture and data handling

FortiAI Assistant processes two primary types of requests:

- **Documentation-based requests**, where the assistant retrieves and summarizes relevant information from official FortiADC documentation.
- **Data-driven requests**, where FortiADC analyzes local system data such as virtual server configuration, health check status, and logs.

For data-driven requests, FortiADC applies data masking before sending information to the FortiAI service. After the response is generated, masked data is restored locally before being displayed in the FortiAI Assistant panel.

Functional capabilities

FortiAI Assistant supports the following operational use cases:

- **Product and configuration guidance**

  Explains FortiADC features and provides configuration guidance derived from official documentation.

  Example prompt:

  How do I configure advanced bot protection on FortiADC?
- **Virtual server and pool inspection**

  Displays virtual server details, associated real server pools, and historical health check status, and supports navigation to relevant FortiView pages.

  Example prompt:

  Show details for virtual server vs-http-8888
- **Log filtering and analysis**

  Enables natural-language filtering, summarization, and analysis of traffic, security, script, and system logs to help identify trends and anomalies.

  Example prompt:

  Show WAF attack logs with severity high from the past 24 hours
- **Text-to-Script (Lua script generation)**

  Converts natural-language descriptions into Lua scripts callable by virtual servers, helping accelerate script development and reduce scripting errors.

  Example prompt:

  Generate a Lua script to add a request-id header to HTTP requests

Licensing and token usage

FortiAI Assistant usage follows the FortiAI entitlement model used across Fortinet products that integrate with FortiAI. Usage depends on platform eligibility, FortiCare support level, and monthly token allocation.

Licensing requirements

FortiAI Assistant does not require a separate add-on license. Supported FortiADC platforms with a valid license are entitled to use FortiAI Assistant, subject to FortiCare support requirements.

For FortiADC VM deployments, FortiAI Assistant access requires an active **FortiCare Premium Support** or **Elite Support** subscription. VM licenses without an active FortiCare support subscription are not entitled to use FortiAI Assistant, and FortiAI login is not available for those deployments.

Token allocation

In FortiADC 8.0.2, FortiAI Assistant is provided as a **Beta feature** and operates with **platform-based monthly token allocations**.

Each supported FortiADC platform receives a fixed number of tokens per month. Tokens are consumed based on request complexity and response size. Token allocations are refreshed automatically at the beginning of each monthly cycle.

VM token allocation

VM license (vCPU)	Monthly token allocation
1 vCPU	350,000
2 vCPU	600,000
4 vCPU	1,100,000
8 vCPU	2,000,000
16 vCPU	3,200,000
32 vCPU	4,500,000
Unlimited vCPU	4,500,000

Hardware model token allocation

FortiADC model	Monthly token allocation
100F	200,000
120F / 200F	350,000
220F	1,000,000
300D	550,000
300F	750,000
320F	1,500,000
400F	1,250,000
420F	2,500,000
1000F / 1200F	3,250,000
2000F / 2200F	6,000,000

FortiADC model	Monthly token allocation
4000F / 4200F	9,000,000
5000F	15,000,000

When the monthly token allocation is exhausted, access to FortiAI Assistant is temporarily suspended until the next allocation cycle begins. At this time, additional token top-up options are not supported. The FortiAI entitlement model continues to evolve, and expanded token options, including token top-ups, may become available in future updates. For the latest entitlement and availability information, contact **Fortinet Support**.

Application Access Manager

The FortiADC 8.0 release includes new features and enhancements in **Application Access Manager**:

Agentless Application Gateway New Features 8.0.3 on page 21

Agentless Application Gateway (AAG) has been significantly enhanced to provide more flexible application delivery and granular access control:

- **SLB Virtual Server Integration**: AAG now leverages the full power of FortiADC by integrating the new WebAPP-Internal-Advanced bookmark type with SLB HTTP/S Virtual Servers. This enables enterprise-grade features such as WAF, advanced Load Balancing, Health Checks, and Scripting for bookmarked applications.
- **Shareable Bookmarks**: Bookmarks are now independent, shareable objects decoupled from App Groups. This allows for simplified object reuse and more efficient, centralized management across the platform.
- **User Group Matching**: The Authentication policy now supports **User Group Match** conditions. This allows administrators to define more granular access permissions based on specific user group memberships.
- **Unauthenticated URL Redirection**: For unauthenticated sessions, AAG now supports **URL auto-redirection**. This ensures a smoother user experience by automatically guiding users to the appropriate destination or login portal.

Agentless Application Gateway New Features 8.0.1 on page 25

FortiADC 8.0.1 introduces major new features to the Agentless Application Gateway (AAG), expanding its capabilities to publish internal web applications, enforce multi-factor authentication (MFA) at App Portal login, and improve portal usability with automatic language detection and customizable bookmark icons. These updates extend AAG to support browser-based access to internal resources such as intranet sites and collaboration platforms, providing secure, policy-driven delivery without the need for VPN software or client agents.

New Application Access Manager Module on page 33

FortiADC introduces the **Application Access Manager**, a centralized framework for managing user authentication and secure access to web-based applications. This enhancement consolidates all authentication-related configuration into a unified module and provides the foundation for the new Agentless Application Gateway (AAG).

Unified Access Policy for Application Access Control on page 38

FortiADC introduces a redesigned Access Policy framework that replaces the legacy Authentication Policy. This update expands authentication control beyond HTTP/HTTPS virtual servers to support the new Agentless Application Gateway (AAG) and provides a unified mechanism for managing user authentication across both standard and portal-based deployments.

Agentless Application Gateway (AAG) on page 43

The Application Access Gateway (AAG) on FortiADC provides secure, agentless access to internal applications through a centralized web portal. It allows users to access a variety of applications, including RDP, VDI, SSH, and web-based applications, without the need for client-side software installations (agentless)

Agentless Application Gateway New Features 8.0.3

FortiADC 8.0.3 introduces the following new features to the Agentless Application Gateway (AAG).

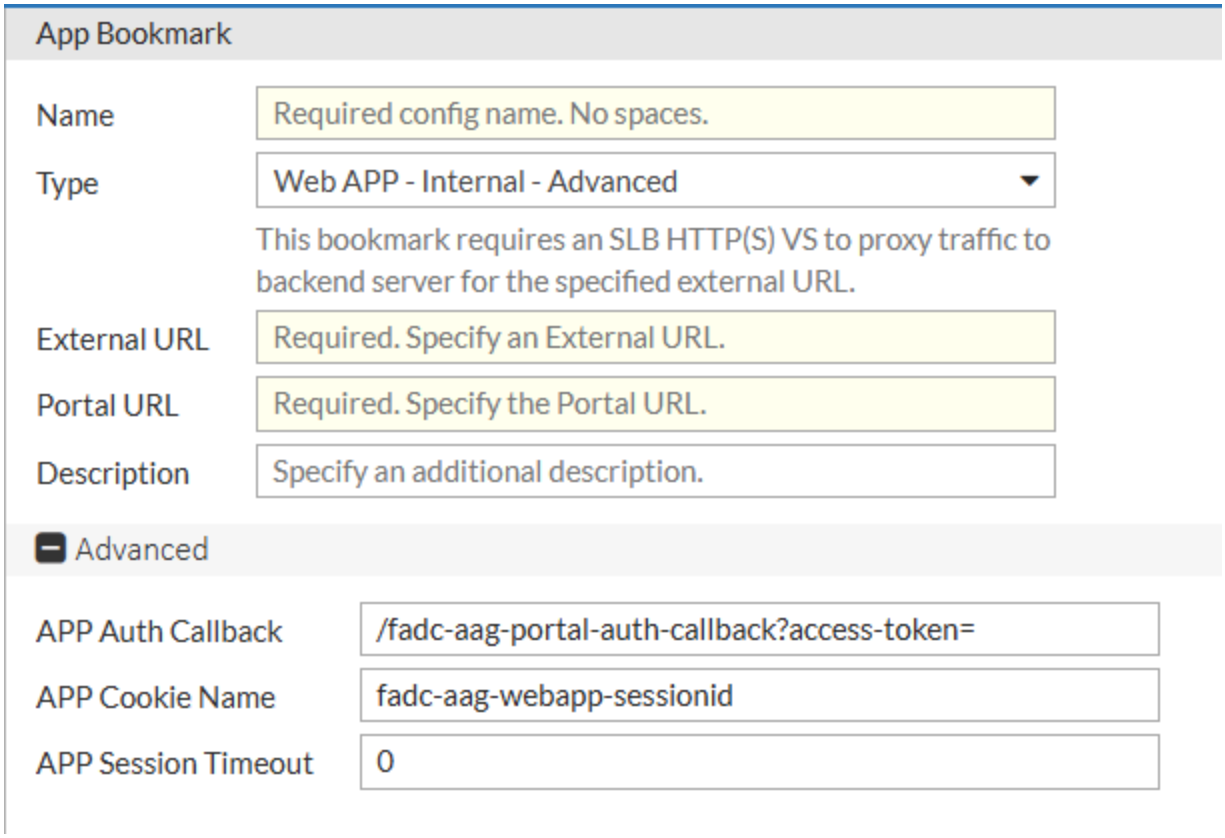
"Web App - Internal - Advanced" bookmarks

The “Web App – Internal - Advanced” option enables full utilization of FortiADC’s Layer 7 server load balancing, application delivery, and security capabilities, allowing administrators to apply the same advanced features available to standard Layer 7 HTTP virtual servers, including:

- **Load balancing** across multiple backend servers
- **Health checks** to ensure service availability
- **Web application firewall (WAF)** protection for enhanced security
- **Content routing** and **application optimization**
- **SSL/TLS processing** features

With this enhanced bookmark type, internal web applications accessed through the AAG portal are no longer limited to web proxy. Instead, they benefit from enterprise-grade traffic management, security enforcement, and performance optimization, just like externally published web applications.

The screenshot below shows the "Web App - Internal - Advanced" bookmark settings.



Related topics:

- Web App Publishing
- The "Web App - Internal - Advanced" bookmark in Configuring an App Bookmark.
- Configuring Virtual Severs for "Web App - Internal - Advanced" Applications

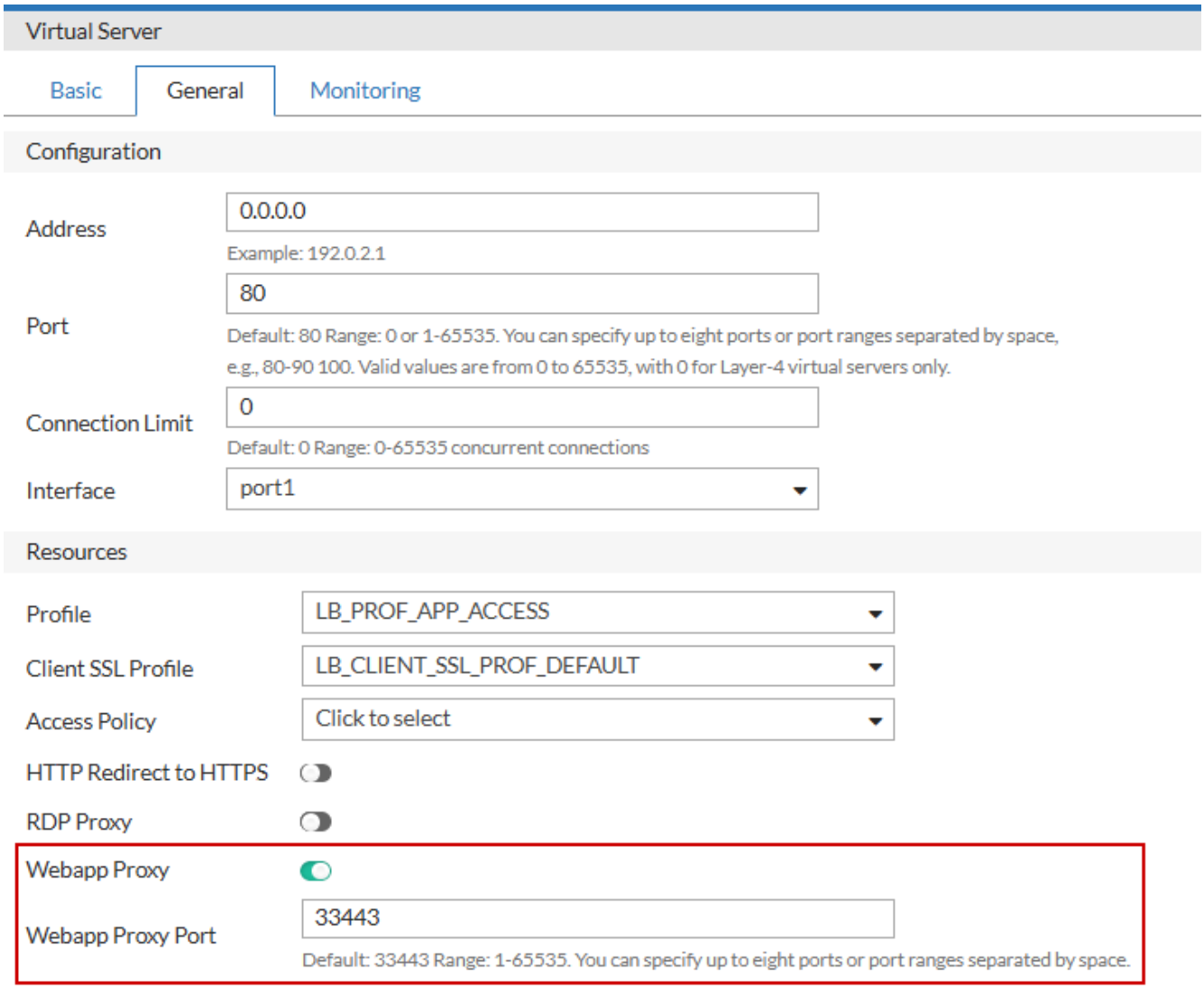
Webapp Proxy settings in virtual server

In previous releases, a dedicated virtual server was required for **Web App - Internal** application. Starting from version 8.0.3, a simplified configuration is available through **Webapp Proxy** settings within the portal virtual server.

With this enhancement, if a Web App - Internal application shares the same Internet-facing IP address (virtual server address) as the AAG portal, you no longer need to create a separate virtual server. Instead, you can enable **Webapp Proxy** when configuring the **AAG portal virtual server**.

This approach simplifies the overall configuration and reduces deployment complexity.

The screenshot below shows the **Webapp Proxy** settings in a virtual server.

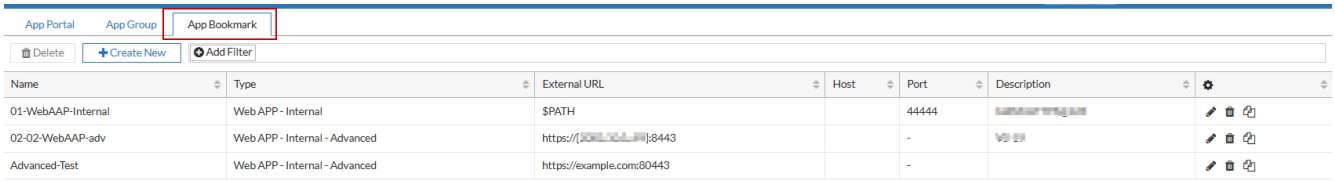


See Special Scenario: "Web App - Internal" Application Sharing the Same IP with AAG Portal.

App Bookmark moved to a separate tab

In previous releases, App Bookmarks were created under an App Group. Starting from version 8.0.3, this behavior has changed—App Bookmarks can now be created independently.

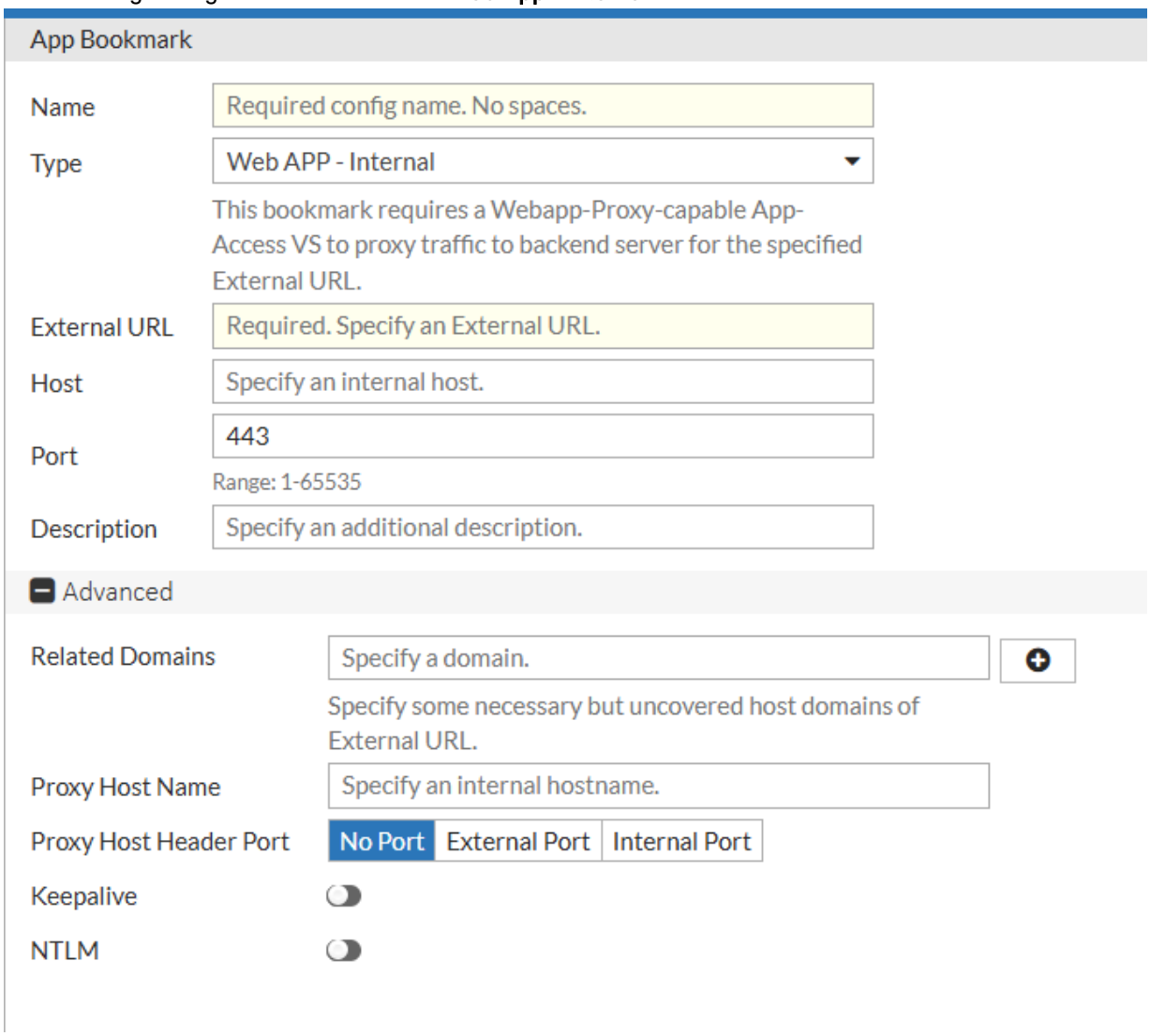
This allows a bookmark to be reused across multiple App Groups, instead of being tightly bound to a specific App Group.



Other enhancements

- WebSocket protocol is now supported for connections between FortiADC and the back-end hosts.
- When users enter the web application external URL directly in their browser, the AAG login page will appear if they are not already authenticated yet. After successful authentication, AAG automatically redirects the user to the requested application page.

- The following settings have been added in **Web App - Internal** bookmark.



  - **Keep Alive:** Keep the connection between FortiADC and the real server open until the server closes the connection.
  - **NTLM:** If the backend server requires NTLM authentication, enable this option so that FortiADC can proxy the NTLM authentication process to the client. The client is then prompted to enter credentials in a pop-up window to complete the authentication.
  - **Proxy Host Name:** Specifies a different hostname for backend server from the External URL hostname.
  - **Proxy Host Header Port:** Specifies whether the `Host` header in HTTP requests includes no port (default), the port of **External URL**, or the port from **Port** field.
  - The **URL path and query** components in **External URL** support LDAP user attributes.

For information, refer to the "Web App - Internal" bookmark in Configuring an App Bookmark.

Agentless Application Gateway New Features 8.0.1

FortiADC 8.0.1 introduces major new features to the Agentless Application Gateway (AAG), expanding its capabilities to publish internal web applications, enforce multi-factor authentication (MFA) at App Portal login, and improve portal usability with automatic language detection and customizable bookmark icons. These updates extend AAG to support browser-based access to internal resources such as intranet sites and collaboration platforms, providing secure, policy-driven delivery without the need for VPN software or client agents.

New AAG Features Overview

New Feature	Description	Impact
Internal Web Application Access	Support for publishing HTTP/HTTPS applications as **Web App - Internal** bookmarks. Applications are proxied through AAG virtual servers and accessed via the App Portal.	Extends AAG beyond predefined types, enabling secure, browser-based access to intranet sites, collaboration tools, and other internal services.
Web App Proxy See New Web App Proxy option in the App Access Profile on page 26.	New option in the App Access Profile to enable HTTP/HTTPS proxying on AAG virtual servers. Required for publishing internal web applications.	Provides the foundation for delivering browser-based applications through AAG.
Multi-Factor Authentication (MFA)	Enables MFA enforcement at **App Portal login** for Local and RADIUS users, requiring a FortiToken code or push approval.	Strengthens authentication for portal access and supports enterprise compliance requirements.
Automatic Language Detection	App Portal login and main pages automatically adapt to the browser's language preference (English, Simplified Chinese, Traditional Chinese, Japanese, Spanish, Portuguese).	Ensures a consistent, localized experience for global users.
Custom Bookmark Icons	Supports uploading administrator-defined icons for App Bookmarks in the portal.	Improves clarity and usability while aligning the portal with organizational branding.

Internal Web Application Access

This information is also available in the FortiADC 8.0.1 Administration Guide:

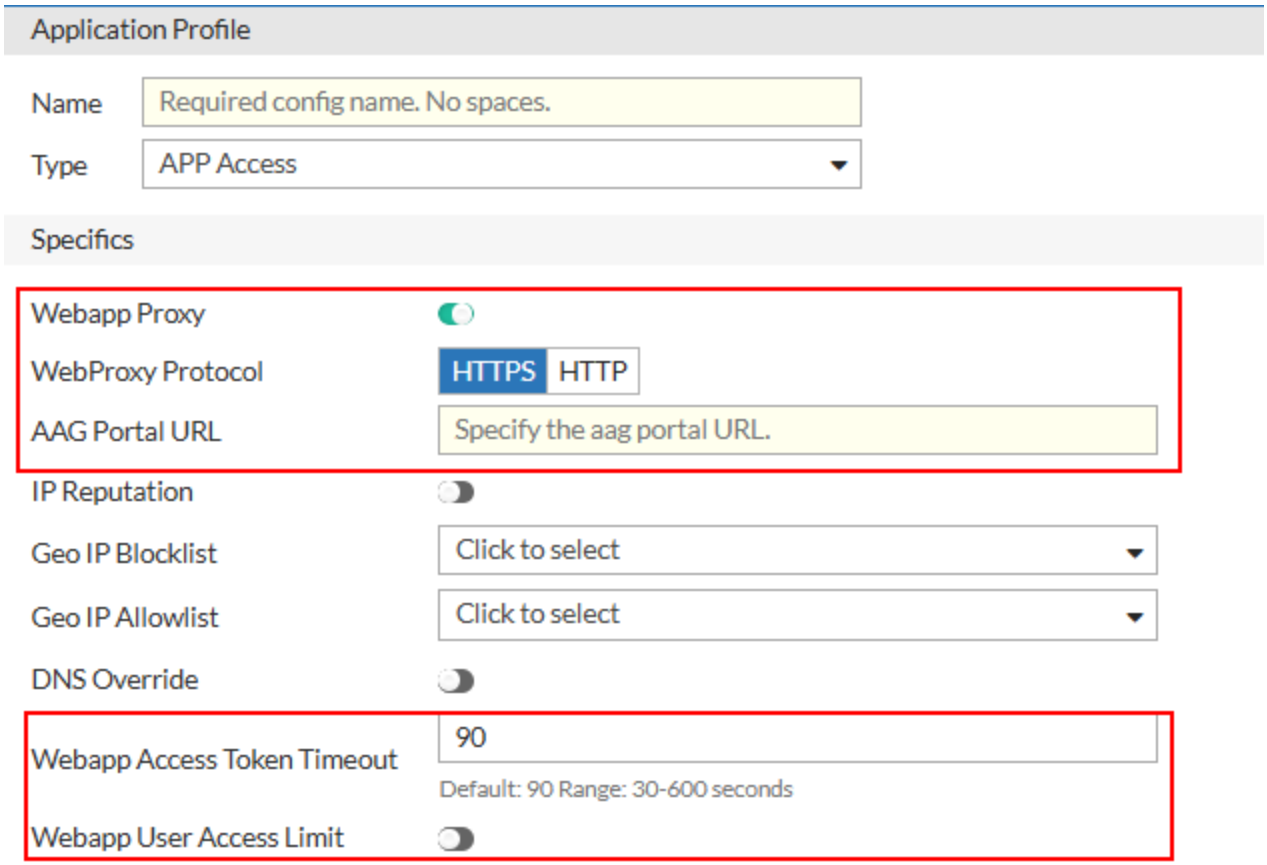
- Publishing Internal Web Applications for AAG Access
- Configuring an App Group

FortiADC 8.0.1 introduces support for Internal Web Application Access, allowing administrators to publish browser-based applications such as intranet sites, collaboration platforms, or other internal services through the AAG App Portal. Users authenticate once at the portal and can securely reach these applications without requiring VPN software or client agents. This enhancement extends AAG beyond predefined application types and makes it possible to deliver a wider range of enterprise web resources in a controlled, policy-driven way.

This new capability for internal web application publishing is implemented through a **Web App Proxy** option in the **App Access Profile**. When enabled, this option **allows FortiADC to terminate and proxy HTTP/HTTPS sessions for internal applications**. Web App Proxy is **applied to a separate virtual server created specifically for the internal resource**, while the App Portal VS continues to manage user authentication and bookmark presentation. Unauthenticated requests to the published application are redirected to the App Portal URL defined in the profile, ensuring that all access remains authenticated and centrally controlled. Administrators can configure these Internal App virtual servers and link them to the App Portal using **Web App – Internal bookmarks**, integrating web application publishing seamlessly into the existing AAG framework while maintaining consistent authentication and policy enforcement across all application types.

New Web App Proxy option in the App Access Profile

The **Web App Proxy** option extends the App Access Profile so it can be applied to a virtual server that publishes internal HTTP/HTTPS applications. In this configuration, the profile defines how FortiADC proxies browser-based sessions to backend servers and handles token-based user access.



When Web App Proxy is **disabled**, the App Access Profile is applied to the **App Portal virtual server** to control login behavior and session policies. When it is **enabled**, the profile is used on an **Internal App virtual server**, where it manages HTTP/HTTPS session handling for the published application. This allows administrators to use a single profile framework for both portal management and internal web application delivery.

Note: Currently, AAG Webapp VS supports HTTP/1.1 proxying only.

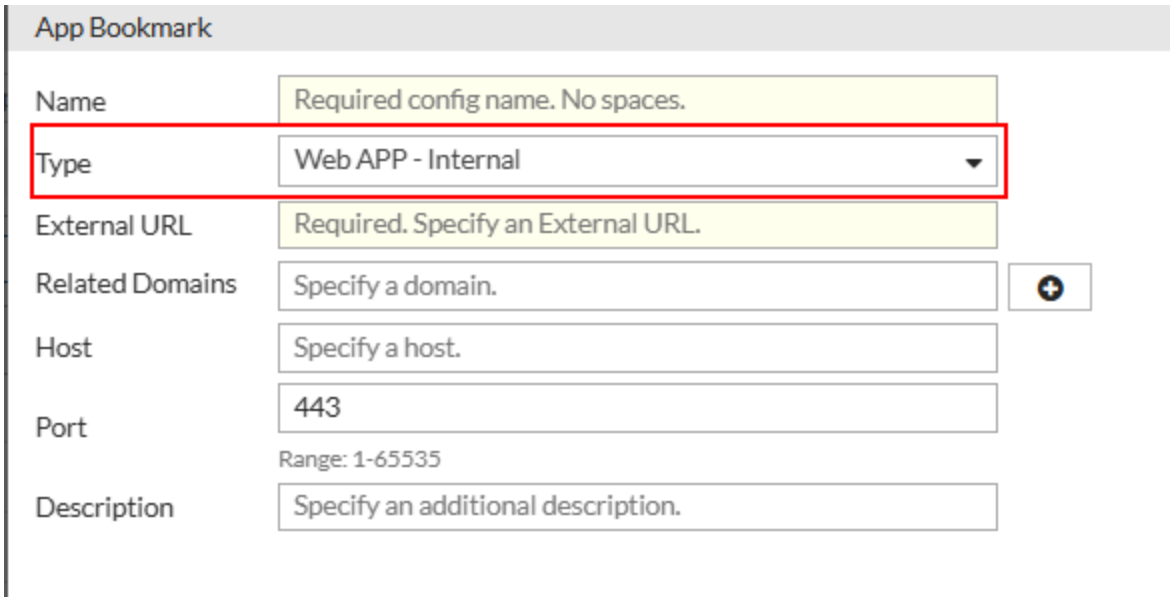
New APP Access Profile Parameters:

Parameter	Description
Webapp Proxy	Enable to activate Web App (HTTP/HTTPS) proxying for this App Access Profile. This is disabled by default.
WebProxy Protocol	Select **HTTP** or **HTTPS** according to the protocol used by the internal web application that the virtual server will proxy.
AAG Portal URL	Specifies the App Portal address (`http(s)://<fqdn>:<port>` from which the Internal App VS obtains user authentication information or redirects unauthenticated users to the portal login page.
Webapp Access Token Timeout	Defines the lifetime of the access token issued by the App Portal VS for an authenticated user. When the Internal App VS receives the token, it validates it against this timeout value; expired tokens are considered invalid. Default: 90, Range: 30-600 seconds.
Webapp User Access Limit	Enable to enforce user-session verification. If the user is no longer online, the Internal App VS redirects the session to the portal login page.

New Web App – Internal Bookmark type

FortiADC 8.0.1 adds a new bookmark type, **Web App – Internal**, which represents an internal web application published through the Agentless Application Gateway (AAG). This bookmark type links internal web application publishing with the App Portal, allowing administrators to provide seamless, browser-based access to internal services without requiring VPN or client software.

Administrators create Web App – Internal bookmarks to define how users connect to the published application. Each bookmark specifies the application's external FQDN, backend connection details, and any related domains used by the service. Once created, the bookmark is added to an App Group and assigned to an App Portal, where it appears as a selectable application for authenticated users.



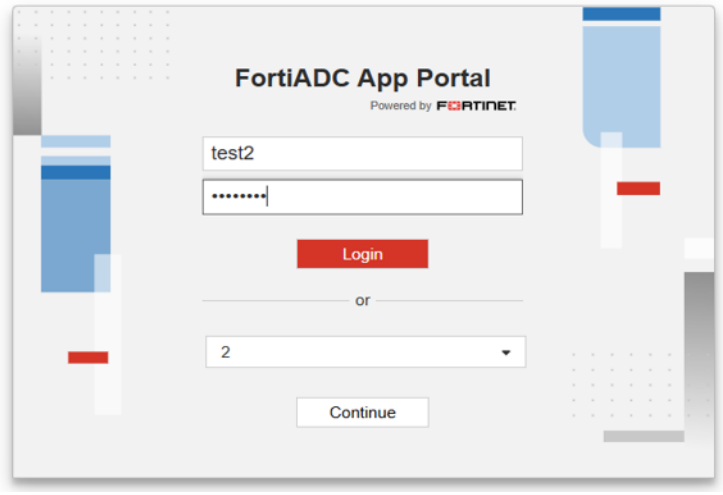
Parameter	Description
External URL	Specifies the bookmark URL — the homepage address of the internal web application, in the format `http(s)://<fqdn>:<port>`. The domain name must resolve to the Internal App virtual server configured with Web App Proxy enabled. This is the URL users access through the App Portal or directly in a browser.
Related Domains	Lists additional subdomains or domains used by the web application, in addition to the homepage domain (for example, for content delivery or APIs). Enter each entry in the format `${sub-domain}@${top-level domain}`. All specified domains must be included in the certificate's Subject Alternative Name (SAN) to ensure proper SSL validation. **Example**: For an application accessed at `https://portal.example.com` that loads content from `static.example.com` and `api.example.net`, enter `static@example.com` and `api@example.net`.
Host	(Optional) Specifies the backend server IP address or hostname (and optionally the port number) to which the Internal App virtual server routes traffic. If not set, FortiADC uses DNS resolution on the hostname portion of the URL to obtain the backend server IP address.
Port	(Optional) Specifies the TCP port (1-65535) used by the backend server for the application. This value must match the port configured on the Internal App virtual server that proxies the application.
Description	(Optional) Descriptive text for the bookmark. The label appears in the App Portal to help users identify the application (for example, Intranet Portal or SharePoint Site).

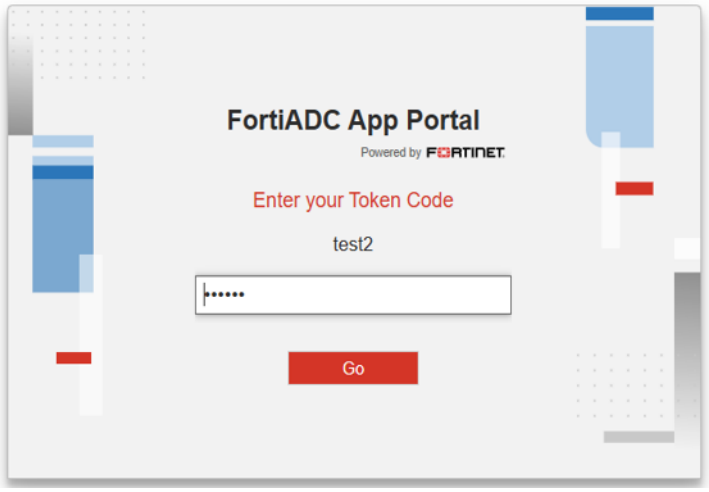
Multi-Factor Authentication (MFA)

FortiADC 8.0.1 adds support for **multi-factor authentication (MFA)** at the App Portal login, providing stronger verification for users accessing applications through the Agentless Application Gateway (AAG). This enhancement integrates MFA into the existing user authentication framework, allowing administrators to require additional authentication factors—such as FortiToken codes or push approvals—before granting portal access.

MFA enforcement is applied through the Access Policy by referencing **Local** or **RADIUS** user groups that are configured for multi-factor authentication. When an Access Policy with MFA-enabled user groups is assigned to the App Portal, users must first provide their credentials and then complete a second-factor challenge to complete the login process.

This capability strengthens access security for all applications published through AAG, including internal web applications, by ensuring that only verified users can reach the App Portal interface.





Usability Enhancements

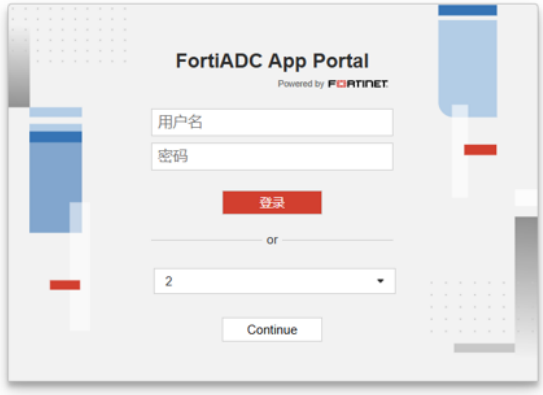
FortiADC 8.0.1 introduces several usability improvements to the Agentless Application Gateway (AAG) App Portal, designed to provide a more intuitive and personalized user experience. The enhancements include automatic detection

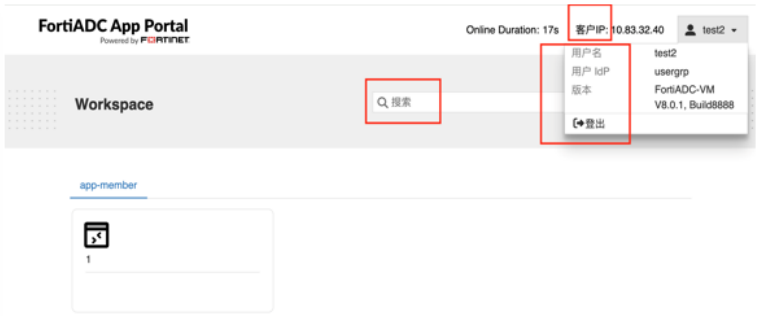
of the user's browser language and support for administrator-defined bookmark icons. Together, these updates improve accessibility for global users and allow greater flexibility in aligning the portal interface with organizational branding.

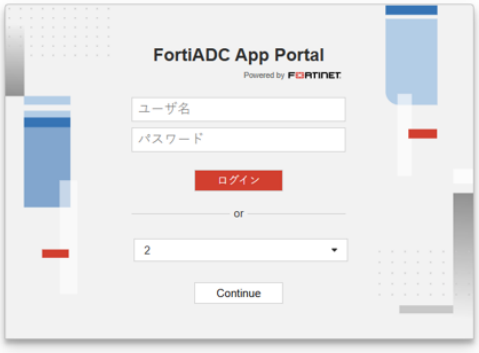
Automatic Language Detection

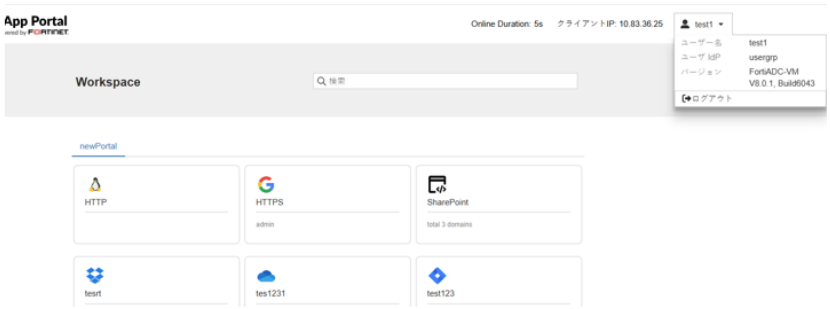
The App Portal now automatically adapts its interface language to match the client browser's language preference. No configuration is required–language detection occurs automatically based on the browser settings during login. Supported languages include **English**, **Simplified Chinese**, **Traditional Chinese**, **Japanese**, **Spanish**, and **Portuguese**.

When a user's browser requests one of the supported languages, the App Portal displays all pages and login prompts in that language. If the browser requests an unsupported language, the interface defaults to English. This enhancement simplifies the user experience for multilingual environments and ensures consistent localization across all AAG sessions.









Custom Bookmark Icons

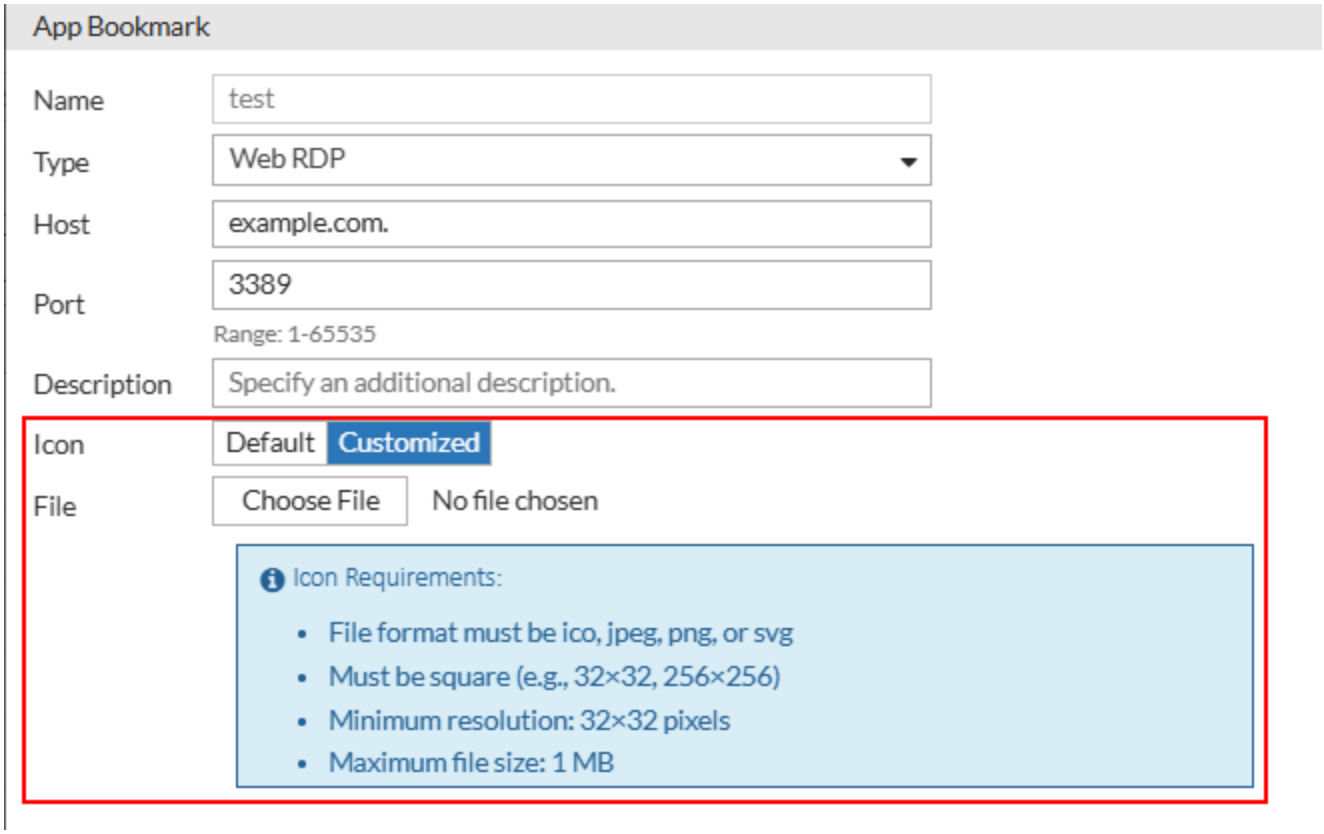
Administrators can now assign either default or custom icons to application bookmarks displayed in the App Portal. This enhancement allows organizations to visually distinguish critical applications, maintain consistent branding, and make commonly used resources easier to identify.

After completing the bookmark configuration, the **Icon** field becomes available when editing bookmarks under **Application Access Manager > Agentless Application Gateway > App Group**. Two options are supported:

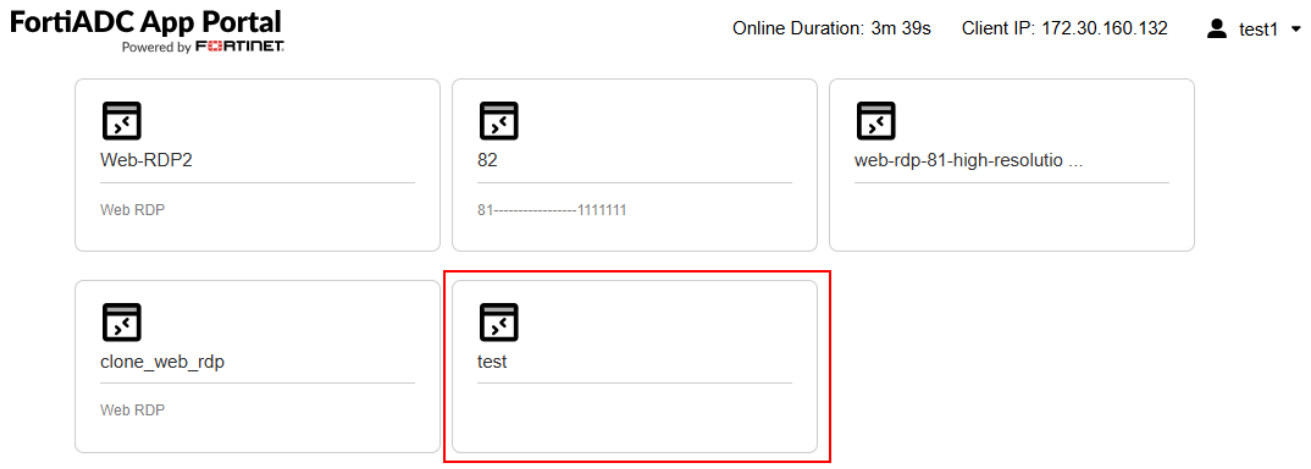
- **Default** - Uses the standard system icon.
- **Custom** - Allows the administrator to upload a custom image file.

Custom icons must meet the following requirements:

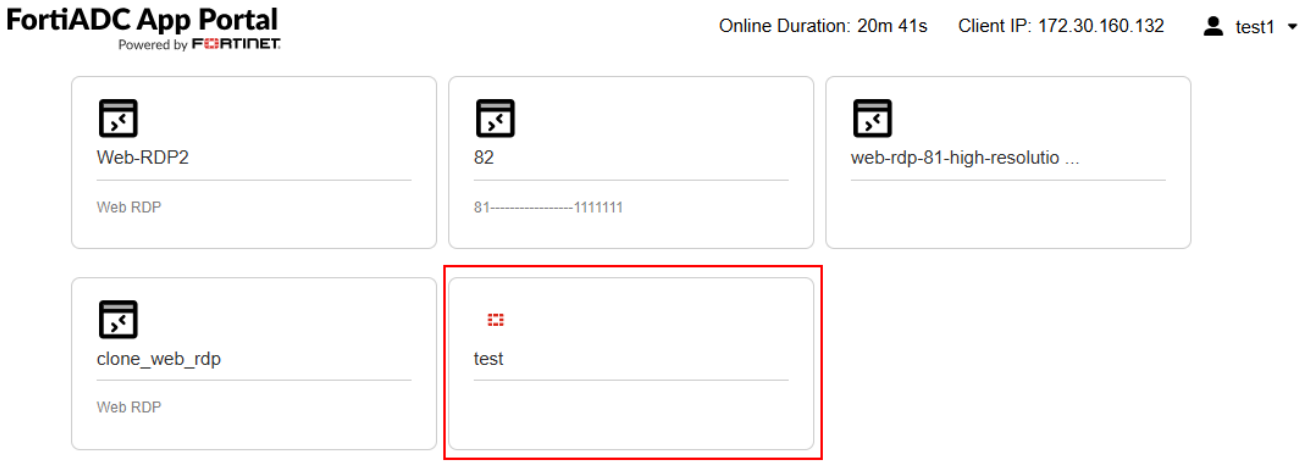
- **File format**: .ico, .jpeg, .png, or .svg
- **Shape**: Square (for example, 32×32 or 256×256)
- **Minimum resolution**: 32×32 pixels
- **Maximum file size**: 1 MB



Default icon:



Custom icon:



New Application Access Manager Module

FortiADC introduces the **Application Access Manager**, a centralized framework for managing user authentication and secure access to web-based applications. This enhancement consolidates all authentication-related configuration into a unified module and provides the foundation for the new Agentless Application Gateway (AAG).

This information is also available in the FortiADC 8.0.0 Administration Guide:

- Application Access Manager

Overview

The Application Access Manager replaces the former Authentication module and now appears as the User Authentication section in the navigation menu. All existing components—such as Access Policy (formerly Authentication Policy)—have been migrated under this new framework.

This modular design enables tighter integration between authentication services, access policies, and application publishing features. The following major capabilities are included:

Access Policy: Defines authentication requirements for protected applications and services. Supports rule-based enforcement and multiple authentication methods (Local User, LDAP, RADIUS, SAML).

Agentless Application Gateway (AAG): Enables secure, browser-based access to internal applications without the need for VPNs or endpoint agents.

Identity Integration: Streamlines user identity validation through integration with external Identity Providers (IdPs).

Benefits

- Centralizes all authentication and application access features in one logical module
- Enhances support for modern identity integration workflows
- Prepares the system for agentless access scenarios via the AAG
- Simplifies configuration and policy management for secure application publishing

This update provides the structural basis for scalable, policy-driven access control to enterprise applications while ensuring compatibility with FortiADC's existing SLB and WAF infrastructure.

Components of the Application Access Manager

The Application Access Manager consists of several modules, each dedicated to a specific aspect of access management:

Access Policy

The **Access Policy** module defines authentication and authorization rules that regulate user access to published applications. It specifies the required authentication method including:

- Local User accounts managed on FortiADC
- LDAP directory services (e.g., Active Directory)
- RADIUS authentication servers
- SAML 2.0 Identity Providers (e.g., Azure Entra ID, FortiAuthenticator)

Each Access Policy can also define session timeout, idle timeout, and reauthentication behavior. These policies are applied at the virtual server level—including servers configured for the Agentless Application Gateway (AAG).

Multi-factor authentication (MFA) is not currently supported for AAG. MFA is supported for standard HTTP/HTTPS virtual server access via Access Policies.

Agentless Application Gateway (AAG)

The **Agentless Application Gateway (AAG)** module enables secure, agentless access to enterprise applications without requiring endpoint agents. AAG operates as a reverse proxy, providing users with a web-based portal to access applications such as:

- Web-based RDP, VNC, SSH, and Telnet sessions
- Native RDP and RemoteApp connections
- Secure access to web applications

Users must first authenticate through the Access Policy applied to the AAG Virtual Server. Supported authentication for AAG includes:

- Local users
- LDAP and RADIUS-based authentication
- SAML 2.0 federated authentication (via providers such as Azure Entra ID and FortiAuthenticator)

OAuth Proxy and AD FS Proxy are not supported for AAG. These modules are only applicable to HTTP/HTTPS virtual server access scenarios.

AAG Configuration Components

AAG is configured through two primary components:

- **App Group** – Defines the applications available to users. Each App Group consists of one or more App Bookmarks, which specify individual applications. Each group can contain up to 256 bookmarks.
- **App Portal** – Provides users with a web-based interface to access applications. Each App Portal is associated with an App Group, controlling the set of applications available to users. Each portal supports up to 32 App Groups.

User Authentication

The **User Authentication** framework in FortiADC enables administrators to configure and manage how users are identified, authenticated, and grouped for access control. This section integrates with access policies, authentication workflows, and application access rules to ensure that only authorized users can access protected services.

It includes the following modules:

User Group

The **User Group** module allows you to define logical groupings of users and assign access privileges based on group membership. Groups serve as the main unit for applying access policies, particularly in environments using directory services or federated identity systems.

Key capabilities include:

- **Mixed membership**: Groups can consist of:
  - **Local users** defined in FortiADC.
  - **Remote users** authenticated through external identity sources (LDAP, RADIUS, NTLM, TACACS+).
- **Access assignment**: You can associate user groups with Access Policies or App Groups (in AAG) to determine which resources the users can access.
- **Group resolution**: For remote users, FortiADC can dynamically map users to groups based on attributes retrieved during authentication (e.g., LDAP group membership or RADIUS response attributes).

User Groups streamline user management by allowing scalable access rule configuration and simplifying identity integration across local and external systems.

Local User

The **Local User** module provides the ability to define and manage standalone user accounts directly on the FortiADC system. These accounts are stored in the local user database and authenticated without requiring an external identity provider.

Key features:

- **Independent identity management**: Useful in air-gapped environments, testing scenarios, or as a fallback when remote authentication systems are unavailable.
- **Password management**: Supports secure password storage with configurable password policies, expiration, and complexity requirements.
- **MFA support**: When enabled in the Access Policy, local users can be required to provide additional authentication factors such as time-based one-time passwords (TOTP).

While not scalable for large deployments, Local Users are ideal for small environments or administrative access during setup and recovery.

Remote Server

The **Remote Server** module defines how FortiADC integrates with external authentication infrastructure to validate user credentials. Supported server types include:

LDAP Server

Enables FortiADC to authenticate users against an LDAP directory, such as Microsoft Active Directory or OpenLDAP. Administrators can configure connection parameters (host, port, base DN, bind DN, and bind password), specify search filters, and choose attribute mappings to retrieve user group membership and identity attributes used in access policy evaluation. Both LDAP over TCP (port 389) and LDAPS (port 636) are supported.

RADIUS Server

Supports authentication through the Remote Authentication Dial-In User Service (RADIUS) protocol. FortiADC can be configured with one or more RADIUS servers, defining parameters such as shared secret, timeout, and retry count. When a user logs in, FortiADC sends an Access-Request to the configured RADIUS server and grants access upon receiving a valid Access-Accept response.

NTLM Server

Provides support for NT LAN Manager (NTLM) authentication, commonly used in Windows domain environments for single sign-on. FortiADC can act as an NTLM relay to authenticate users transparently against the Windows Domain Controller, enabling domain-joined users to access protected applications without manual credential entry.

TACACS+ Server

Integrates with Terminal Access Controller Access-Control System Plus (TACACS+) for centralized authentication, authorization, and accounting. FortiADC supports configuration of TACACS+ server address, port, and encryption key, and processes authentication transactions according to TACACS+ protocol standards.

Authentication Relay

The **Authentication Relay** module enables FortiADC to act as an intermediary between clients and external authentication services that are not directly supported by built-in methods. It is particularly useful in scenarios requiring:

- **Protocol translation**, such as transforming NTLM or Kerberos requests into LDAP or RADIUS transactions.
- **Custom authentication workflows**, where FortiADC forwards credentials to third-party systems via HTTP or HTTPS relay mechanisms.
- **Legacy system integration**, allowing FortiADC to support proprietary or environment-specific login flows.

FortiADC relays client credentials securely to the target authentication endpoint and processes the response to determine authentication success, enabling flexible interoperability across diverse enterprise environments.

SAML

The **SAML** module allows FortiADC to operate as a SAML 2.0 Service Provider (SP), supporting federated authentication with external Identity Providers (IdPs) such as FortiAuthenticator, Microsoft Entra ID (formerly Azure AD), Okta, and others.

Key technical capabilities include:

- **SSO support** – Users authenticate once with the IdP and gain access to multiple SAML-integrated services without repeated credential entry.
- **Metadata exchange** – FortiADC supports IdP metadata import and SP metadata export for simplified trust establishment.

- **Assertion validation** – FortiADC validates signed SAML assertions, enforces audience restrictions, and maps user attributes (such as group membership or username) to local access policies.

This integration is essential for organizations adopting centralized identity platforms and enabling secure, scalable user access to protected applications through the AAG App Portal or other HTTP/HTTPS services.

AD FS Proxy

The **AD FS Proxy** module enables FortiADC to function as an external proxy for Active Directory Federation Services (AD FS). In this role, FortiADC:

- Acts as a gateway between clients and the internal AD FS infrastructure.
- Terminates HTTPS on the DMZ interface while forwarding secure requests to the internal AD FS servers.
- Ensures federated authentication compatibility for applications relying on AD FS, including those using WS-Federation and SAML protocols.

This is particularly useful in segmented network deployments, where direct access to AD FS is restricted for security reasons, and FortiADC is positioned at the perimeter to securely mediate identity transactions. (**Note**: AD FS Proxy is not currently supported for AAG portal access.)

OAuth Proxy

The **OAuth Proxy** module allows FortiADC to integrate with **OAuth 2.0 Authorization Servers** for token-based access control. Acting as a proxy between the client application and the identity provider (IdP), FortiADC supports:

- **Authorization Code flow** for secure client-side authentication.
- **Token introspection** to validate bearer tokens (access tokens) with the authorization server.
- **Userinfo endpoint interaction** to retrieve user attributes from the IdP (if applicable).
- **Scope enforcement and attribute mapping** to align OAuth claims with FortiADC access policies.

This feature is essential for supporting modern authentication platforms like Google Identity, Microsoft Entra ID (via OAuth), and other standards-compliant OAuth 2.0 providers. (**Note**: OAuth Proxy is not currently supported for AAG portal access.)

Unified Access Policy for Application Access Control

FortiADC introduces a redesigned Access Policy framework that replaces the legacy Authentication Policy. This update expands authentication control beyond HTTP/HTTPS virtual servers to support the new Agentless Application Gateway (AAG) and provides a unified mechanism for managing user authentication across both standard and portal-based deployments.

This information is also available in the FortiADC 8.0.0 Administration Guide:

- Access Policy

Overview

The new Access Policy consolidates authentication configurations under a flexible and extensible policy model. Now located in the Application Access Manager, the Access Policy supports both traditional HTTP-based authentication and authentication for AAG-published applications.

Key enhancements include:

- Expanded application scope: Supports both standard HTTP/HTTPS virtual servers and AAG virtual servers.
- Modular member-based design: Define multiple user/group-based access rules per policy.
- Multiple authentication types: Supports Standard (local/LDAP/RADIUS), SAML, and OAuth (non-AAG only).
- App Portal association: In AAG mode, access policies determine user access to published apps in the portal.
- Fine-grained host and URI path control: In non-AAG mode, policies can match specific host headers and URI prefixes.

Use Modes

AAG Deployments

The Access Policy defines how users authenticate to the App Portal and which applications are made available after login. Authentication types include Standard and SAML. Each policy is associated with an App Portal configuration and applied to a Layer 7 virtual server with an APP Access profile.

Non-AAG Deployments

The Access Policy continues to enforce authentication for direct access to HTTP/HTTPS virtual servers. Each policy can match by host and URI and supports Standard, SAML, or OAuth authentication types. Policies are applied to Layer 7 or Layer 2 virtual servers configured for HTTP or HTTPS traffic.

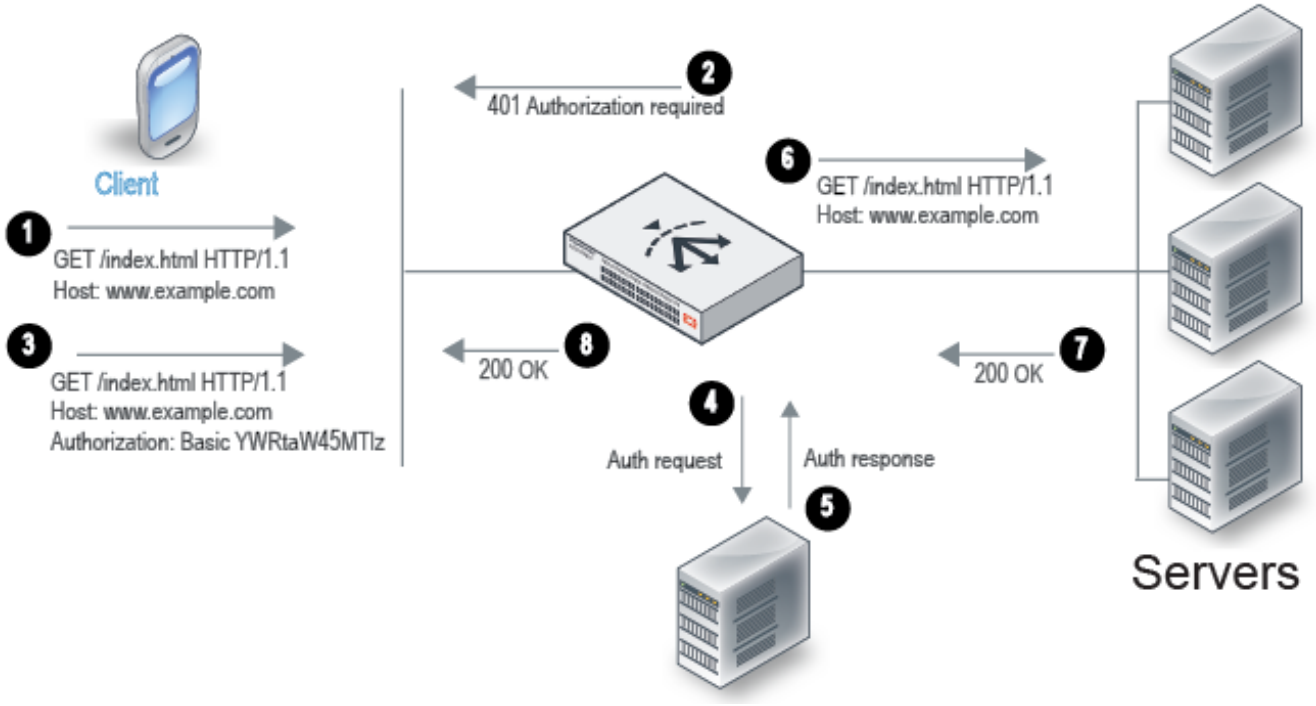
Migration Notes

- The **Authentication Policy** has been deprecated and replaced by the Access Policy. All existing configurations have been migrated under the **Application Access Manager > User Authentication** menu.
- Existing functionality remains supported; no behavioral changes are introduced for previously configured policies unless modified to include AAG-specific options.

How Access Policies Work

Authorization and authentication on page 39 illustrates the client-server communication when authorization is required.

Authorization and authentication



An Access Policy consists of authentication conditions and an associated user group. When a client attempts to access a virtual server that requires authentication, the following process occurs:

1. The client sends an HTTP request for a URL belonging to a FortiADC virtual server that has an authorization policy.
2. FortiADC replies with an HTTP 401 to require authorization. On the client computer, the user might be prompted with a dialog box to provide credentials.
3. The client reply includes an Authorization header that gives the credentials.
4. FortiADC sends a request to the server (local, LDAP, or RADIUS) to authenticate the user.
5. The authentication server sends its response, which can be cached according to your user group configuration.
6. If authentication is successful, FortiADC continues processing the traffic and forwards the request to the real server.
7. The real server responds with an HTTP 200 OK.
8. FortiADC processes the traffic and forwards the server response to the client.

For example, you can define an access policy that has the following logic: if the Host header matches `example.com` and the URI matches `/index.html`, then the group example-group is authorized. FortiADC supports the Basic Authentication Scheme described in RFC 2617.

Before you begin:

- You must have created the user groups to be authorized with the policy. You also configure users and authentication servers separately.
- You must have read-write permission for Server Load Balance settings.

After you have configured an access policy, you can select it in the applicable virtual server configuration.

Configuring an Access Policy for AAG

In an Agentless Application Gateway (AAG) deployment, the Access Policy controls authentication and authorization for users accessing the App Portal. This policy determines how users authenticate, which authentication providers are used, and what applications they can access after logging in.

This section provides the steps to configure an App Access User Auth Policy for an AAG virtual server, including defining authentication methods (Standard or SAML), associating user groups, and assigning an App Portal configuration. Once applied to the virtual server, the policy ensures that only authenticated users can access the applications presented in the App Portal.

Multi-factor authentication (MFA), OAuth Proxy, and AD FS Proxy are not currently supported for AAG. These authentication methods are only applicable to HTTP/HTTPS virtual server access scenarios.

1. Navigate to **Application Access Manager > Access Policy**.
2. Click **Create New** to display the configuration editor.
3. Configure the following App Access User Auth Policy settings:

Settings	Guidelines
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.
APP Portal Access	Enable APP Portal Access to configure the Access Policy for the AAG virtual server.

4. Save the configuration.
   Once the App Access User Auth Policy is saved, the Member section becomes configurable.
5. Under the Member section, click **Create New** to display the configuration editor.
6. Configure the following Access Policy Member settings:

Settings	Guidelines
Type	Select either of the following: • Standard – Uses local authentication or external authentication servers (LDAP, RADIUS). • SAML – Authenticates users via a SAML Identity Provider (IdP).

Settings	Guidelines
User Group	Select the user group that is authorized to access the protected resource. Available only if **Standard** is selected as the **Type**.
SP Name	Select the SAML SSO ID that is authorized to access the protected resource. Available only if **SAML** is selected as the **Type**. **Note**: SSO is not currently supported for AAG portal access.
App Portal	Associates the policy with a configured App Portal, defining the available applications.

7. Save the Member settings.
8. Click **Save** again to finalize the changes to the Access Policy.

You can now assign this Access Policy to a Layer 7 virtual server configured with an **APP Access** profile to enforce authentication and access control to your Agentless Application Gateway.

Configuring an Access Policy for a non-AAG virtual server

For non-AAG virtual servers, the Access Policy enforces authentication for users accessing protected HTTP/HTTPS resources. Unlike in AAG, where authentication applies to the App Portal, here, authentication is triggered when users attempt to access a specific host and URI path.

This section outlines the steps to configure an Access Policy for a standard HTTP/HTTPS virtual server, including defining authentication settings (Standard, SAML, or OAuth), specifying host-based access control, and assigning user groups or authentication proxies. Once applied, FortiADC challenges incoming requests with authentication before granting access to the backend servers.

1. Navigate to **Application Access Manager > Access Policy**.
2. Click **Create New** to display the configuration editor.
3. Configure the following App Access User Auth Policy settings:

Settings	Guidelines
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.
APP Portal Access	The APP Portal Access option is disabled by default. Keep the option disabled to configure the Access Policy for a non-AAG virtual server.

4. Save the configuration.
   Once the App Access User Auth Policy is saved, the Member section becomes configurable.
5. Under the Member section, click **Create New** to display the configuration editor.
6. Configure the following Access Policy Member settings:

Settings	Guidelines
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 with any Host header. Disabled by default.

Settings	Guidelines
Host	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`.
Type	Select either of the following: • Standard • SAML • OAuth
User Realm	Realm to which the Path URI belongs. The realm is included in the basic authentication header in the HTTP 401 message sent to the client. If a request is authenticated and a realm specified, the same credentials are deemed valid for other requests within this realm. Available only if **Standard** is selected as the **Type**.
Path	Require authorization only if the URI of the HTTP request matches this pathname. If none is specified, requests to any URI require authorization. The value is parsed as a match string prefix. For example, `/abc` matches `http://www.example.com/abcd` and `http://www.example.com/abc/11.html` but not `http://www.example.com/1abcd`.
User Group	Select the user group that is authorized to access the protected resource. Available only if **Standard** is selected as the **Type**.
SP Name	Select the SAML SSO ID that is authorized to access the protected resource. Available only if **SAML** is selected as the **Type**.
OAuth Policy	Select the OAuth policy that is authorized to access the protected resource. Available only if **OAuth** is selected as the **Type**.

7. Save the Member settings.
8. Click **Save** again to finalize the changes to the Access Policy.

You can now assign this Access Policy to a Layer 7 or Layer 2 virtual server configured with an HTTP or HTTPS profile to enforce authentication and access control to virtual server.

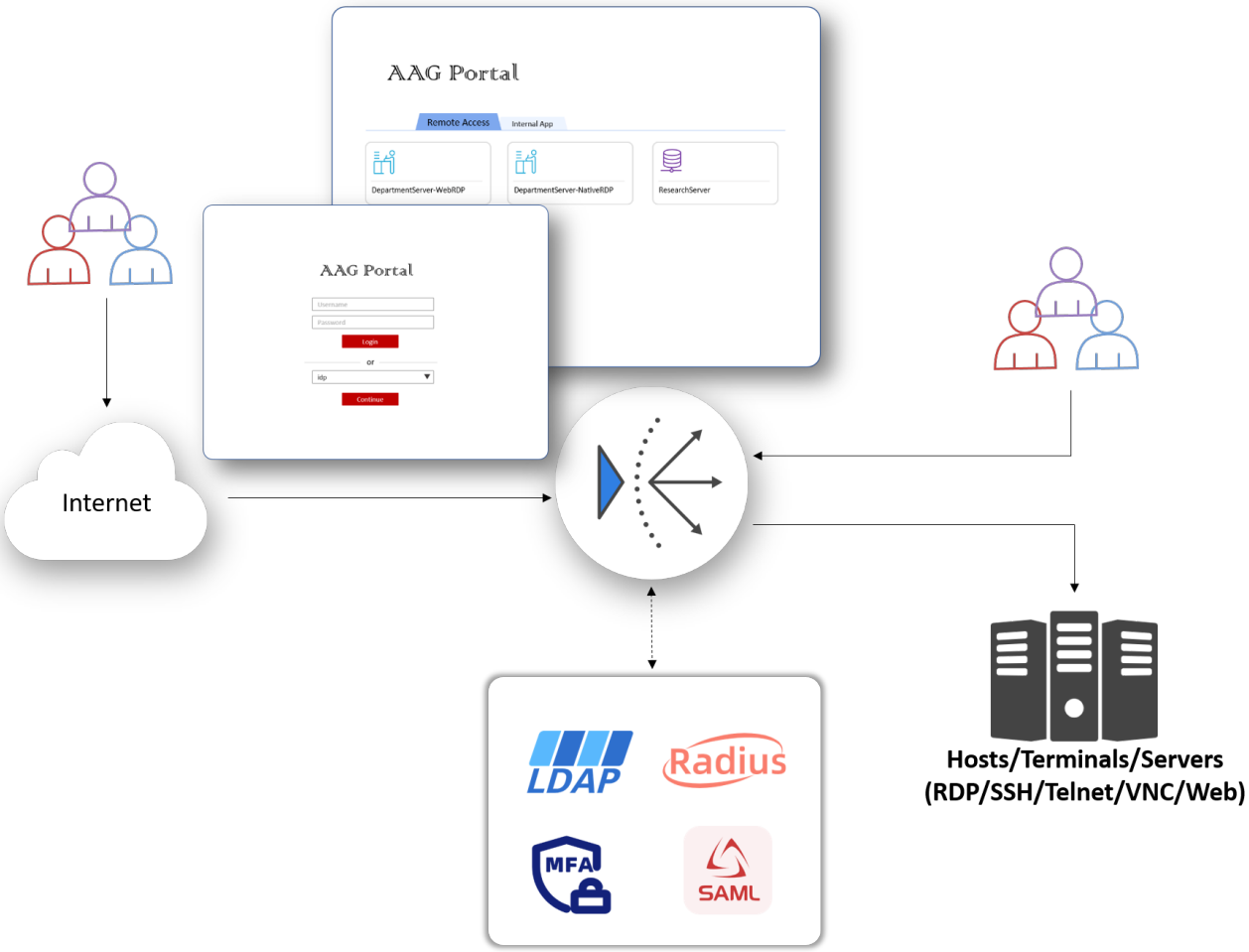
Agentless Application Gateway (AAG)

The Application Access Gateway (AAG) on FortiADC provides secure, agentless access to internal applications through a centralized web portal. It allows users to access a variety of applications, including RDP, VDI, SSH, and web-based applications, without the need for client-side software installations (agentless)

AAG supports multiple authentication methods, including Local User, LDAP, RADIUS, Azure EntraID, and SAML, enabling organizations to integrate with existing identity providers. It also offers fine-grained configuration of access policies, allowing administrators to enforce precise control over who can access specific applications and under what conditions.

This information is also available in the FortiADC 8.0.0 Administration Guide:

- Agentless Application Gateway (AAG)



Key Features and Benefits

AAG in FortiADC delivers secure and scalable remote access with the following capabilities:

- **Agentless Access** – Eliminates the need for client-side agents, reducing endpoint dependency and administrative burden.
- **Federated Authentication Support** – Supports SAML 2.0-based authentication using providers like Microsoft Entra ID and FortiAuthenticator.
- **Simplified Deployment** – Requires no software installation, enabling rapid implementation in secure environments.
- **Real-Time Monitoring** – Tracks user sessions and access logs for compliance and auditing.
- **Application-Layer Security Controls** – Enforces IP reputation filtering and geolocation restrictions through the Application Profile attached to the AAG Virtual Server.

Technical Overview

Traditional vs. Agentless Application Access

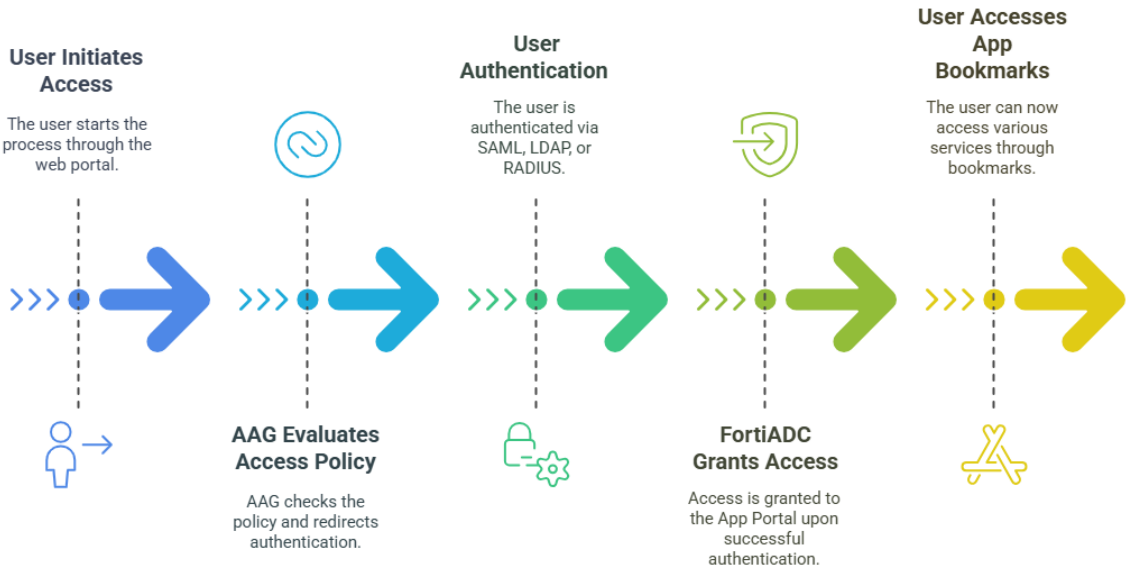
Traditional remote access solutions often require software agents installed on client devices or backend systems to enforce authentication and security policies. This introduces compatibility constraints, maintenance overhead, and security risks associated with unmanaged endpoints.

FortiADC's Agentless Application Gateway (AAG) eliminates these dependencies by acting as a reverse proxy, terminating client sessions at the gateway and forwarding traffic to backend resources based on authentication and policy enforcement mechanisms.

Authentication Flow and Identity Integration

AAG supports the following authentication workflow:

1. The user accesses the AAG App Portal via browser.
2. FortiADC evaluates the Access Policy and enforces the configured authentication method by:
   a. Validating credentials against Local User, LDAP, or RADIUS servers.
   b. Redirecting to a SAML 2.0 Identity Provider (e.g., FortiAuthenticator, Microsoft Entra ID) for federated login.
3. Upon successful authentication, FortiADC grants access to the App Portal associated with the user's session.
4. The user can view and launch their assigned application bookmarks for services such as RDP, SSH, VNC, Telnet, and web apps.



Multi-factor authentication (MFA), OAuth, and AD FS Proxy are not currently supported for AAG access.

Architecture and Components

AAG operates at Layer 7 as an SSL-terminating reverse proxy, inspecting client requests, performing authentication, and forwarding traffic to internal resources based on predefined access policies.

- Client-SSL termination ensures encrypted client communication while enabling inspection and policy enforcement at the gateway.
- Backend authentication passthrough allows seamless integration with existing enterprise identity providers.
- Session persistence mechanisms maintain continuity for authenticated sessions.
- Real-time logging and monitoring track user activity and enforce compliance requirements.

Control Plane Components

The control plane in AAG manages authentication, policy enforcement, session handling, and application publishing. It acts as the centralized management layer, integrating with identity providers (IdPs) and enforcing security policies based on user roles, device attributes, and contextual data. By decoupling authentication and access control from the data plane, the control plane ensures efficient session management while maintaining high security.

App Portal Virtual Server (L7 Reverse Proxy)

- Handles user login, authentication enforcement, and session initiation.
- Performs SSL termination and supports IPv4/IPv6.
- Integrates with Access Policies and federated identity providers.

Authentication and Authorization Services

- Supports local and remote identity verification through Access Policies.
- Compatible with LDAP, RADIUS, and SAML IdPs (e.g., FortiAuthenticator, Microsoft Entra ID).

Application Portal and Bookmark Management

- Hosts a centralized, web-based portal for users to access published applications.
- Supports dynamic bookmarks mapped via LDAP attributes for personalized access.

Data Plane Components

The data plane in AAG is responsible for handling user traffic, forwarding requests to backend applications, and enforcing security measures in real time. It processes encrypted sessions, inspects traffic for potential threats, and applies policy-based access controls. By managing the actual data exchange between users and applications, the data plane ensures that only authorized requests reach enterprise resources while maintaining performance and security.

Application Publishing and Session Management

- Delivers HTML5-based clientless access for RDP, VNC, SSH, and Telnet.
- Supports direct RDP and RemoteApp connections for seamless interaction.
- Enforces RDP attribute-based policies to regulate session parameters.

Access Control and Security Enforcement

- Enforces IP Reputation, Geo IP Blocklist, and Allowlist through Application Profiles.
- Logs session details and access records for diagnostics and compliance.

This architecture ensures a scalable, policy-driven remote access solution while maintaining strong security controls.

AAG Configuration Components in FortiADC

The Agentless Application Gateway (AAG) configuration in FortiADC consists of two primary components: App Groups and App Portals. These components define how applications are published and accessed by authenticated users.

App Portal Configuration

An App Portal defines the web-based authentication interface where users log in and access applications. Each App Portal is associated with an App Group, controlling which applications are available to authenticated users.

- Defines the authentication portal for end users.
- Associates App Groups to control user access.
- Supports TLS encryption for secure access.

For details, see Step 3: Configuring an App Portal on page 54.

App Group Configuration

An App Group is a logical container that organizes multiple application bookmarks, each corresponding to a backend resource. Users assigned to an App Group can access only the applications explicitly defined within it.

- Defines application bookmarks for structured access.
- Maps applications to RDP, VNC, SSH, Telnet, and Web applications.
- Supports dynamic bookmark mapping via LDAP attributes.

For details, see Step 2: Configuring an App Group on page 52.

Supported Application Types in FortiADC AAG:

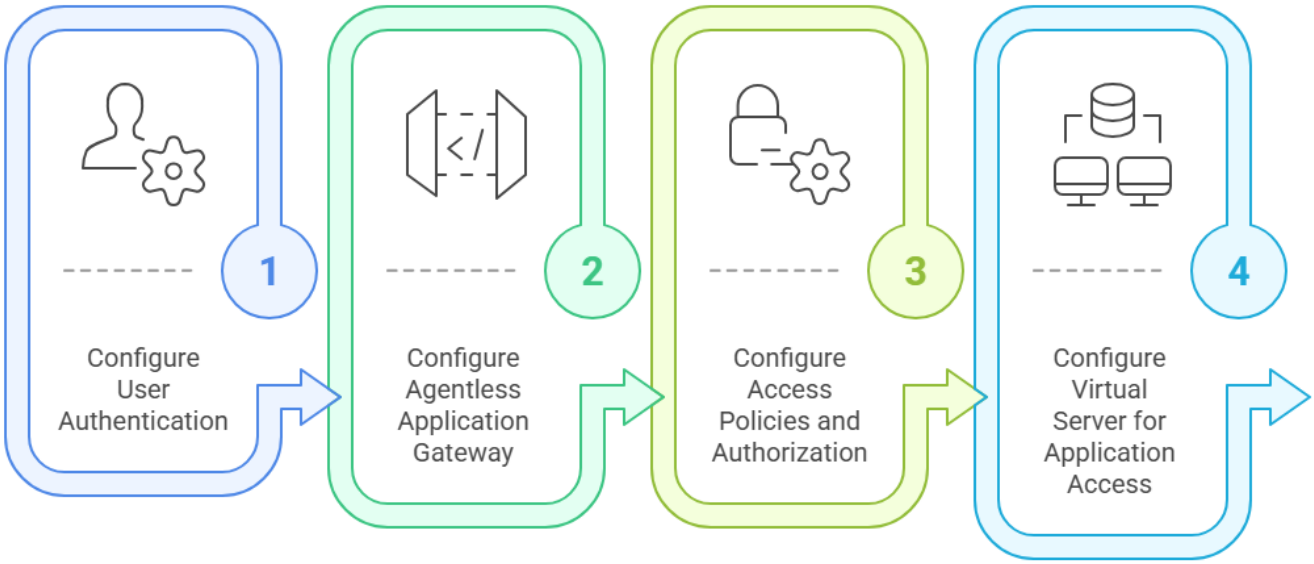
- **Web RDP** – Provides full desktop access via Remote Desktop Protocol (RDP) through an HTML5 browser session. No RDP client is required on the endpoint.
- **Native RDP** – Enables traditional RDP access using the native RDP client installed on the user's device. Requires endpoint support for Microsoft RDP.
- **RemoteApp** – Launches individual Windows applications published via Remote Desktop Services, rather than a full desktop session.
- **Web VNC** – Offers browser-based access to VNC-enabled systems for remote control of graphical desktops.
- **Web SSH** – Delivers secure command-line access to Linux and Unix systems through a browser-based SSH terminal.
- **Web Telnet** – Enables text-based terminal sessions to legacy devices or servers using the Telnet protocol, directly within the browser.

Capacity Guidelines (Per VDOM)

To ensure efficient performance and prevent configuration errors, observe the following capacity limits per VDOM:

Item	Maximum Count
App Portals	1024
App Groups	1024
App Groups per App Portal (sections)	32
Bookmarks per App Group	256
IdPs per Access Policy	8

Workflow for Configuring AAG in FortiADC



Step 1: Configure User Authentication

Select the appropriate authentication method based on deployment requirements.

- For local authentication, create a Local User and assign it to a User Group.
- For remote authentication, configure a Remote Server (e.g., LDAP, RADIUS) and create a User Group.
- For SAML authentication, set up SAML IDP and SAML SP for federated authentication.

For recommendations, see Step 4: Configuring an Authentication Server for AAG on page 50.

Step 2: Configure the Agentless Application Gateway (AAG)

- Create an App Group and add relevant Application Bookmarks. For details, see Step 2: Configuring an App Group on page 52.
- Configure an App Portal and associate it with the App Group. For details, see Step 3: Configuring an App Portal on page 54.

Step 3: Configure Access Policies and Authorization

Apply policies to enforce authentication and authorization.

- Define an Access Policy that enforces the selected authentication method. For details, see Access Policy.
- Assign User Groups to the App Portal based on access requirements. For details, see Configuring user groups.

Step 4: Configure Virtual Server for Application Access

Set up application delivery through a virtual server.

- Use the **APP Access** Application Profile to configure IP Reputation, Geo IP Blocklist, and Geo IP Allowlist.
- Deploy a Layer 7 Virtual Server and assign the APP Access profile, Access Policy, and enable RDP Proxy (if required).
- Associate the App Portal with the virtual server.

For details, see Step 7: Configuring a Virtual Server for the AAG Portal on page 57.

Step 4: Configuring an Authentication Server for AAG

Choosing the appropriate authentication method for FortiADC's Agentless Application Gateway (AAG) is critical for securing access to published applications while ensuring compatibility with existing identity management frameworks. This section provides a structured approach to determining the most suitable authentication mechanism based on user types, security policies, and infrastructure requirements.

This information is also available in the FortiADC 8.0.0 Administration Guide:

- Selecting the Optimal Authentication Method for AAG

FortiADC 8.0.1 introduces support for **Multi-factor authentication (MFA)** at AAG App Portal login when using Local or RADIUS user authentication.

For other authentication types, MFA can be applied to FortiADC administrative access or managed externally through the identity provider.

Authentication Methods and Deployment Considerations

FortiADC supports multiple authentication mechanisms for AAG access, all enforced through Access Policies. Each authentication method is suited to specific deployment scenarios. The table below outlines the key characteristics of each method:

Authentication Method	Best Suited For	Key Advantages
Local User Authentication	Small-scale or standalone deployments without external authentication infrastructure	Simple configuration and self-contained user management directly on FortiADC. Supports MFA through FortiToken or push approval.
Remote User Authentication (LDAP, RADIUS)	Enterprises with centralized authentication systems such as Active Directory or corporate RADIUS servers	Enables centralized credential management and integration with existing infrastructure. RADIUS-based users can leverage MFA for additional verification.
SAML Authentication	Federated authentication across multiple domains or cloud environments	Seamless login experience via trusted Identity Providers (IdPs), including FortiAuthenticator and Microsoft Entra ID

Authentication Deployment Considerations

Local User Authentication

Local user accounts are defined and stored directly on FortiADC. This method is suitable for small deployments or administrative access when external identity systems are unavailable. Local users can be assigned to User Groups and linked to App Portals via Access Policies.

- **Pros**: Simplifies configuration, requires no external dependencies, and allows MFA enforcement through FortiToken or push notification.
- **Cons**: Requires manual user provisioning and offers limited scalability.

LDAP and RADIUS Authentication

AAG integrates with LDAP directories (e.g., Microsoft Active Directory) and RADIUS servers to authenticate users based on corporate credentials. This allows centralized user account management and simplifies integration into existing identity infrastructures.

- **Pros**: Supports secure credential validation and dynamic group mapping through Access Policies. RADIUS authentication can also enforce multi-factor authentication (MFA).
- **Cons**: Both methods require connectivity to external servers and rely on network availability for login verification.

SAML-Based Authentication

SAML is the preferred method for organizations using federated identity platforms such as FortiAuthenticator or Microsoft Entra ID (formerly Azure AD). FortiADC acts as a SAML 2.0 Service Provider (SP), redirecting authentication requests to the Identity Provider (IdP).

Benefits:

- Centralized authentication and single sign-on (SSO)
- Passwordless experience through external IdP
- Support for cross-domain identity federation

Multi-Factor Authentication (MFA) Support

FortiADC supports multi-factor authentication (MFA) for AAG App Portal login when using Local or RADIUS user authentication. After successful primary credential validation, users are prompted for a second factor, such as a FortiToken one-time passcode (OTP) or push notification.

For deployments using these authentication types, MFA can still be applied to FortiADC administrative login or managed externally through the connected identity provider.

Aligning Authentication with Security Policies

Authentication settings should be configured in conjunction with FortiADC's access control policies, session management rules, and logging mechanisms to ensure a secure and compliant deployment.

By selecting an appropriate authentication method—such as Local User, LDAP, RADIUS, or SAML—and integrating with centralized identity services where applicable, organizations can strengthen authentication workflows and improve user experience in their AAG deployments.

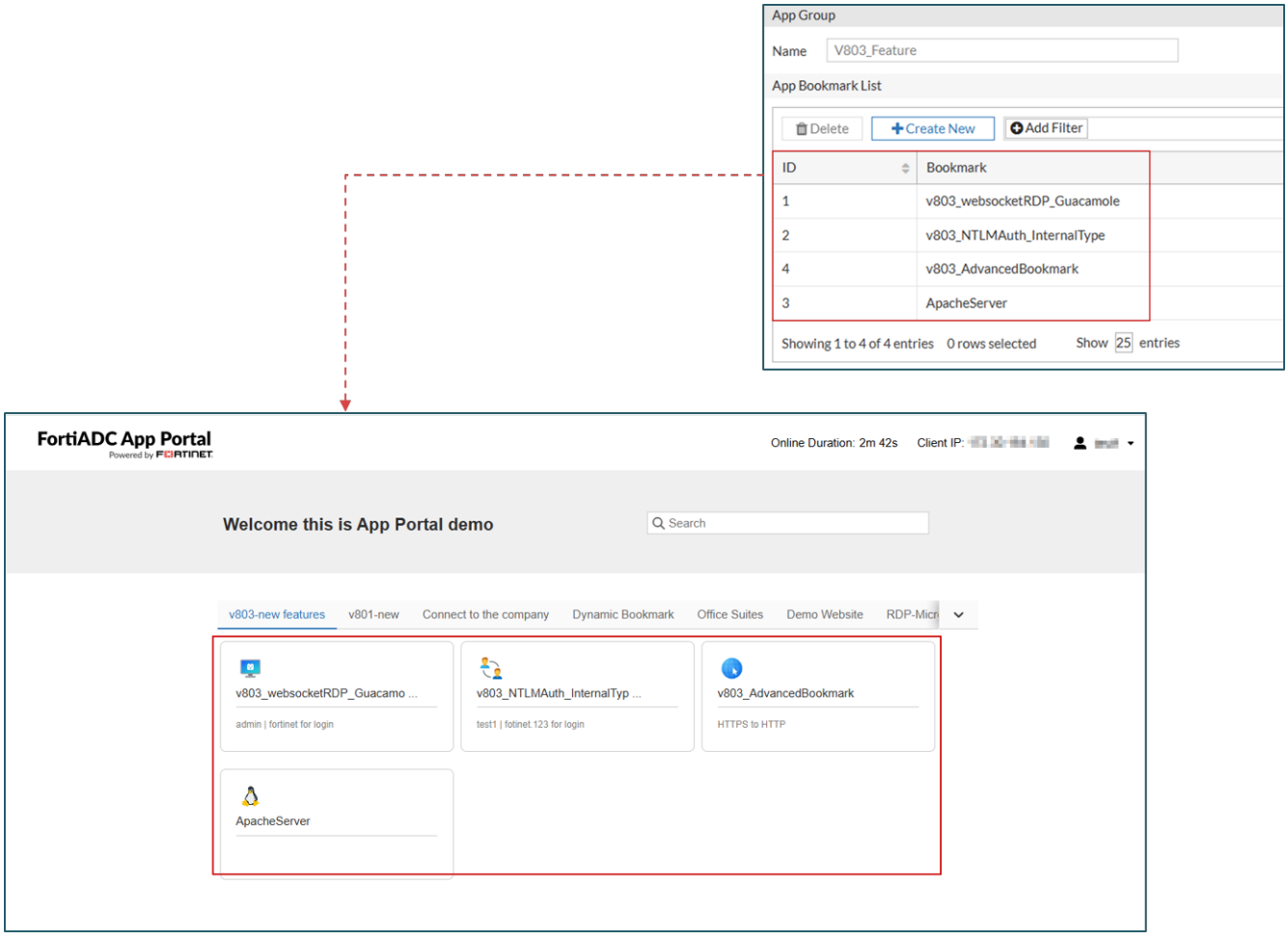
Step 2: Configuring an App Group

An App Group is a logical container that organizes application bookmarks. Each App Group can contain multiple bookmarks of various types (for example, Web RDP, Web SSH, or Web App - Internal).

This information is also available in the FortiADC 8.0.0 Administration Guide and CLI Reference:

- Configuring an App Group
- config user app-group

FortiADC 8.0.1 introduces the **Web App - Internal** bookmark type, which allows administrators to link internal web applications published through FortiADC to the AAG App Portal, and adds support for **custom bookmark icons** for App Portal display customization.



To create an App Group:

1. Navigate to **Application Access Manager > Agentless Application Gateway**.
   The configuration page displays the **App Portal** tab.
2. Click the **App Group** tab.
3. Click **Create New** to display the configuration editor.
4. In the **Name** field, specify a unique name for the App Group configuration object. Valid characters are `A-Z`, `a-z`, `0-9`, `_`, and `-`. No space is allowed.
5. Click **Save**.
   Once the App Group is created, the **App Bookmark** section becomes configurable.
6. Click **Create New**.
7. Select the bookmark to add in the App group.
8. Click **Save**.
9. Repeat step 6-8 until you add all the bookmarks for this group.

Each VDOM supports up to 1024 App Groups, but a single App Portal can be associated with a maximum of 32 App Groups.

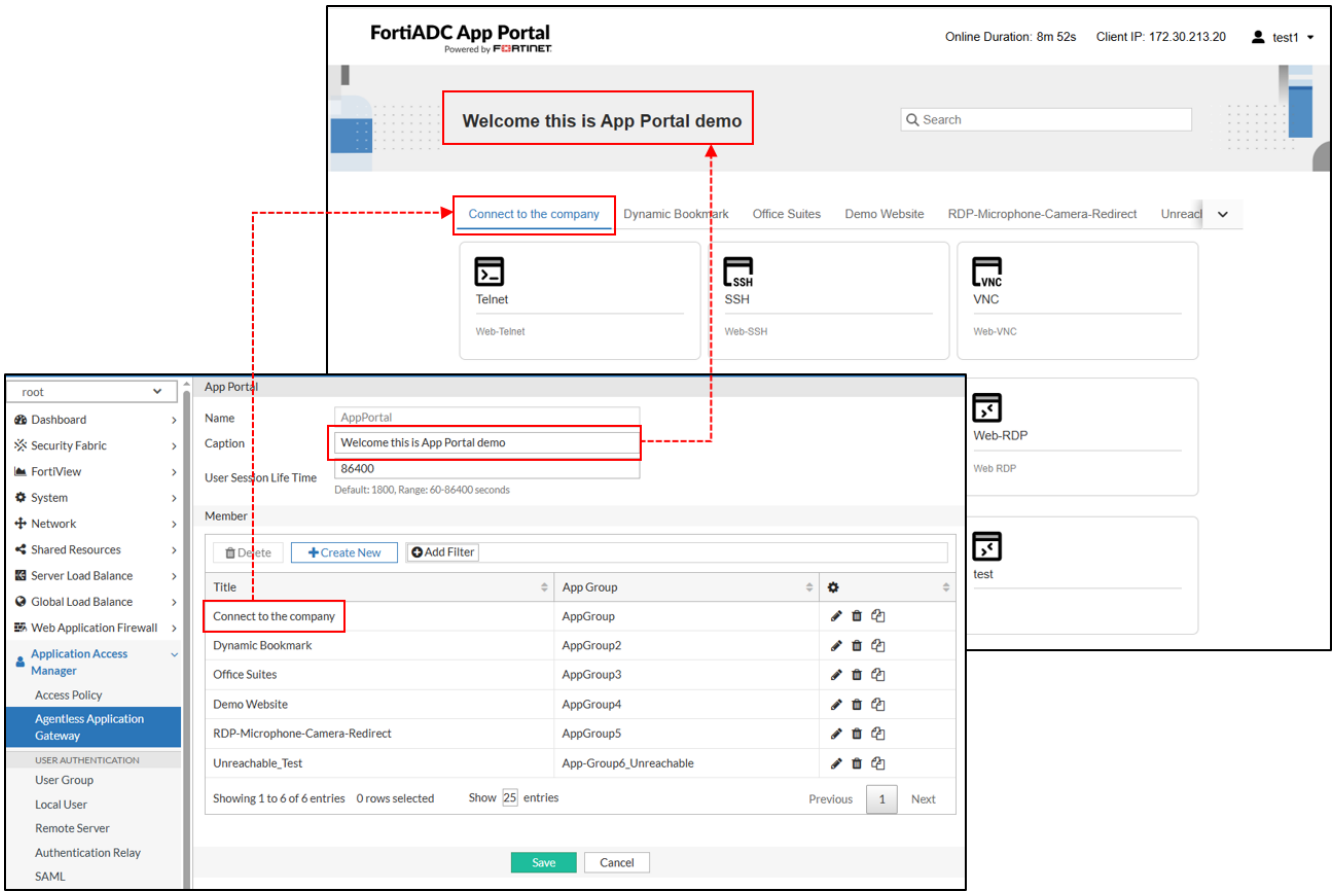
Step 3: Configuring an App Portal

An App Portal in FortiADC serves as a centralized, browser-based interface that allows end-users to securely access applications published through the Agentless Application Gateway (AAG). It aggregates applications from one or more App Groups and integrates with the configured Access Policy to enforce authentication and authorization based on user identity and group membership.

This information is also available in the FortiADC 8.0.0 Administration Guide and CLI Reference:

- Configuring an App Portal
- config user app-portal

From the App Portal configuration page, administrators can customize the portal's appearance and structure. A **Caption** can be entered to display descriptive text in the top banner of the portal interface. App Group configurations are added as **Members** of the App Portal, with each Member appearing as a dedicated tab within the portal. The **Title** field within each Member configuration allows administrators to define the label that appears on the corresponding tab. Each tab provides access to the set of application bookmarks defined in its associated App Group.

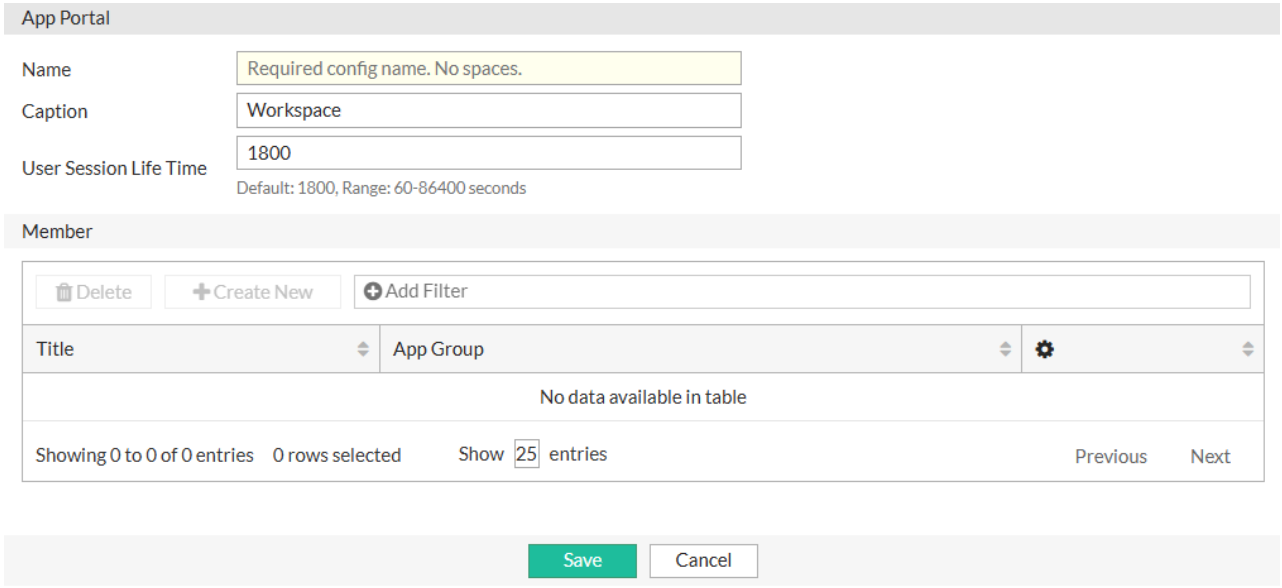


Before you begin:

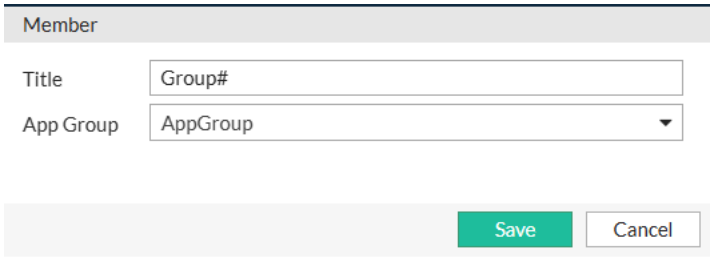
- Ensure that you have already configured the necessary App Groups containing the application bookmarks that will be made available to end-users through the App Portal. For details, see Step 2: Configuring an App Group on page 52.

To configure an App Portal:

1. Navigate to **Application Access Manager > Agentless Application Gateway**.
   The configuration page displays the **App Portal** tab.
2. Click **Create New** to display the configuration editor.



3. In the **Name** field, specify a unique name for the App Portal configuration object. Valid characters are `A-Z`, `a-z`, `0-9`, `_`, and `-`. No space is allowed.
4. In the **Caption** field, enter the text you want to display in the top banner of the App Portal. This can be used to provide a welcome message, organization name, or other identifying information for users.
5. In the **User Session Life Time** field, specify the maximum duration (in seconds) that a user session can remain idle before automatic logout. The default is 1800 seconds. Valid values range from 60 to 86400 seconds.
6. Click **Save**.
   Once the App Portal is created, the **Member** section becomes configurable.
7. Under the **Member** section, click **Create New** to display the configuration editor.



8. In the **Title** field, specify the label that will appear as the tab header for the associated App Group within the App Portal interface.
9. In the **App Group** field, select a predefined App Group configuration from the drop-down list.

10. Click **Save** to apply the configuration and close the App Portal Member dialog. You can add up to 32 members to a single App Portal.
11. Click **Save** to apply all changes to the App Portal and its member settings.

Each VDOM supports up to 1024 App Groups, but a single App Portal can be associated with a maximum of 32 App Groups.

Step 7: Configuring a Virtual Server for the AAG Portal

To enable users to access the AAG portal, you must **create an APP Access application profile and layer 7 virtual server for the portal**. It acts as the central access point for the web application, routing traffic through the App Portal and enforcing user authentication through access policy.

The Virtual Server specifies:

- The IP address of the AAG portal that users will connect to.
  This is typically the public IP address associated with the AAG portal domain name.
- The AAG portal Application Profile.
- Client SSL profile.
- The Access Policy for user authentication and authorization.
- The RDP Proxy and the port number for RDP traffic
- Webproxy and Port for "Web App-internal" applications traffic, if applicable.

This section covers the following topics:

- Configuring the application profile for AAG portal on page 57
- Configuring the virtual server for AAG portal on page 60
- What's the External RDP Address? on page 62

To help you better understand AAG virtual server configuration, we've prepared the following video: FortiADC Virtual Server Configurations for AAG.

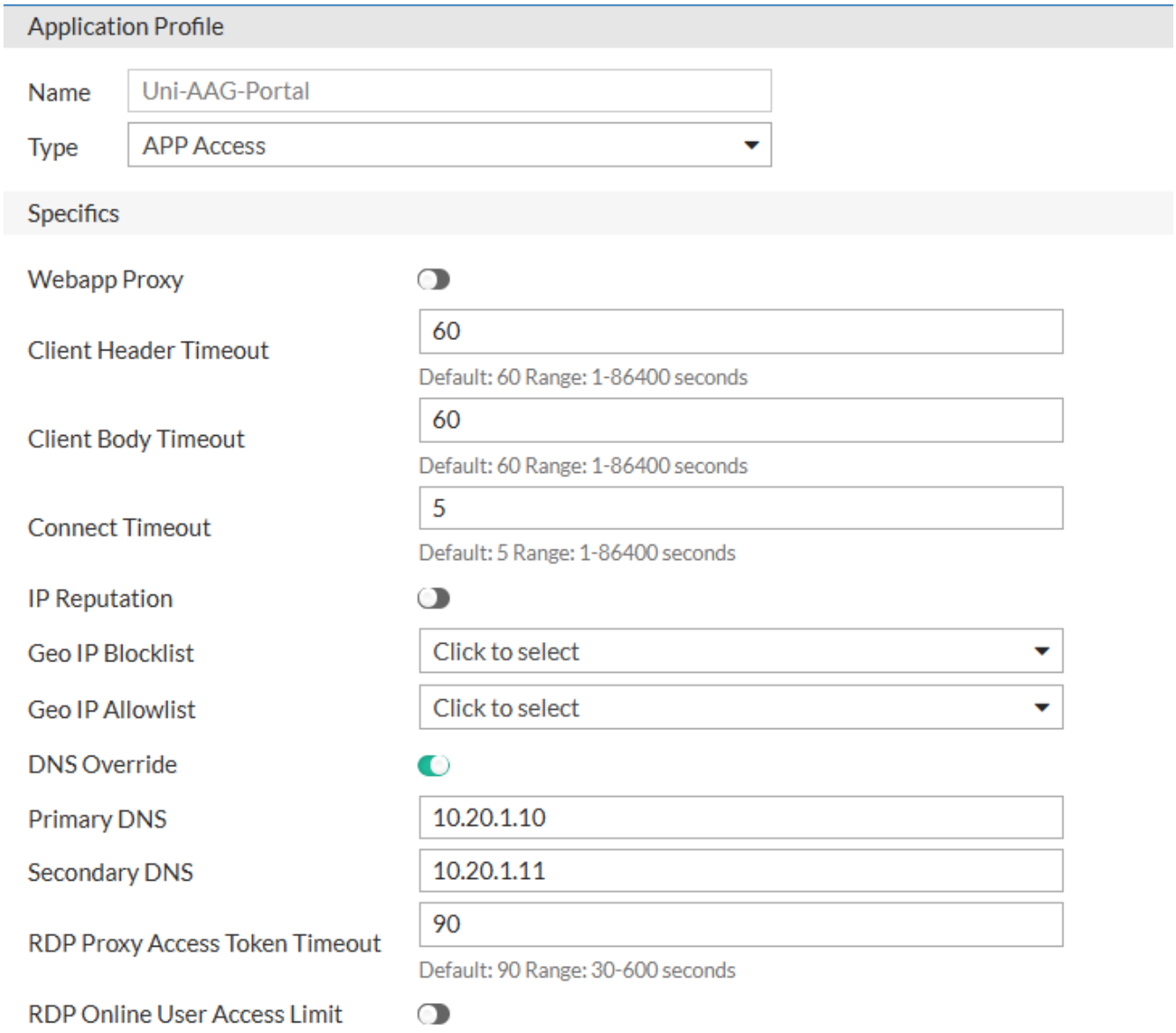
This information is also available in the FortiADC 8.0.0 Administration Guide and CLI Reference:

- Deploying the AAG Virtual Server
- Configuring Application profiles
- Configuring virtual servers
- config load-balance profile
- config load-balance virtual-server

Configuring the application profile for AAG portal

FortiADC introduces a new application profile type–**APP Access**–to support AAG use cases. This type is specifically used for AAG and operates natively over HTTP/HTTPS. When bound to a Virtual Server, it enables AAG-specific capabilities.

The application profile for the AAG portal should have the **Webapp Proxy** option disabled.



To configure an APP Access Profile:

1. Navigate to **Server Load Balance > Application Resources**.
   The configuration page displays the **Application Profile** tab.
2. Click **Create New** to display the configuration editor.
3. In the **Name** field, specify a unique name for the custom Application Profile configuration object. Valid characters are `A-Z`, `a-z`, `0-9`, `_`, and `-`. No space is allowed.
   Once saved, the name of a Application Profile configuration cannot be changed.
4. From the **Type** field, select **APP Access**.

5. Configure the settings for the APP Access profile type. Note the following:
   - Leave the **Webapp Proxy** disabled. This option is only used for "Web App – Internal" applications.
   - If an FQDN is used for the destination host in the bookmark settings, verify where the corresponding DNS records are defined. If the records are hosted on a custom DNS server, enable **DNS Override** and specify the server address.

Parameter	Description
Client Header Timeout	Maximum time (in seconds) FortiADC waits to receive the complete HTTP headers from the client. The default value is 60 seconds, and the valid range is 1-86400 seconds.
Client Body Timeout	Maximum time (in seconds) FortiADC waits to receive the complete HTTP body after headers are received. The default value is 60 seconds, and the valid range is 1-86400 seconds.
Connect Timeout	Maximum time (in seconds) to wait for a connection attempt to a backend server to succeed. The default value is 5 seconds, and the valid range is 1-86400 seconds.
IP Reputation	Enable to apply FortiGuard's IP reputation service to block or allow traffic based on IP threat scores.
Geo IP Blocklist	Select a Geo IP block list configuration object to deny access from specific countries or regions.
Geo IP Allowlist	Select a Geo IP allowlist configuration object to explicitly permit access from specific regions.
DNS Override	If an FQDN is used for the destination host in the bookmark settings, FortiADC must query a DNS server to resolve the IP address. Whether to enable **DNS Override** depends on where the corresponding DNS records are configured. • If the records are available on the system DNS server, **DNS Override** is not required. • If the records are hosted on a different DNS server, enable **DNS Override** and specify that server so FortiADC can successfully resolve the FQDN.
Primary DNS	The **Primary DNS** option is available if **DNS Override** is **enabled**. Specify the IP address of the primary DNS server.
Secondary DNS	The **Secondary DNS** option is available if **DNS Override** is **enabled**. Specify the IP address of the secondary DNS server.
RDP Proxy Access Token Timeout	When the bookmark type is **Native RDP** or **Remote App**, FortiADC operates as an RDP proxy and uses a short-lived access token to securely authorize the client's RDP session to the backend server. **This setting defines how long the access token remains valid, in seconds.** The default value is 90 seconds, and the valid range is 30-600 seconds.

Parameter	Description
	The access token is generated when the user launches an RDP session from the AAG portal. It is embedded in the RDP connection (for example, in the downloaded .rdp file) and is used by FortiADC to validate that the session request is legitimate and associated with an authenticated user. This mechanism prevents unauthorized or replayed connection attempts.
RDP Online User Access Limit	Enable to terminate the RDP connection when user is not at logged-in state.

6. Click **Save** to save the configuration.

Predefined APP Access Profile: LB_PROF_APP_ACCESS

Parameter	Default value
Client Header Timeout	60
Client Body Timeout	60
Connect Timeout	5
IP Reputation	Disabled
Geo IP Blocklist	None
Geo IP Allowlist	None
DNS Override	Disabled
RDP Proxy Access Token Timeout	90
RDP Online User Access Limit	Disabled

Configuring the virtual server for AAG portal

Once the APP Access profile and Access Policy are in place, configure the Virtual Server that delivers the App Portal functionality.

To configure the Virtual Server:

1. Navigate to **Server Load Balance > Virtual Server**.
   The configuration page displays the **Virtual Server** tab.
2. Click **Create New > Advanced Mode** to display the configuration editor.
3. Configure the following key Virtual Server settings:
   a. In the **Basic** tab, select the virtual server **Type** as **Layer 7**.
   b. In the **General** tab, configure the following **Configuration** settings:
      - **Address** – Specifies the IP address to which the virtual server binds, serving as the entry point for portal traffic.

        This is typically the IP address associated with the AAG portal's domain name in the DNS record. When clients initiate requests to the portal interface, this IP address is used as the destination IP.

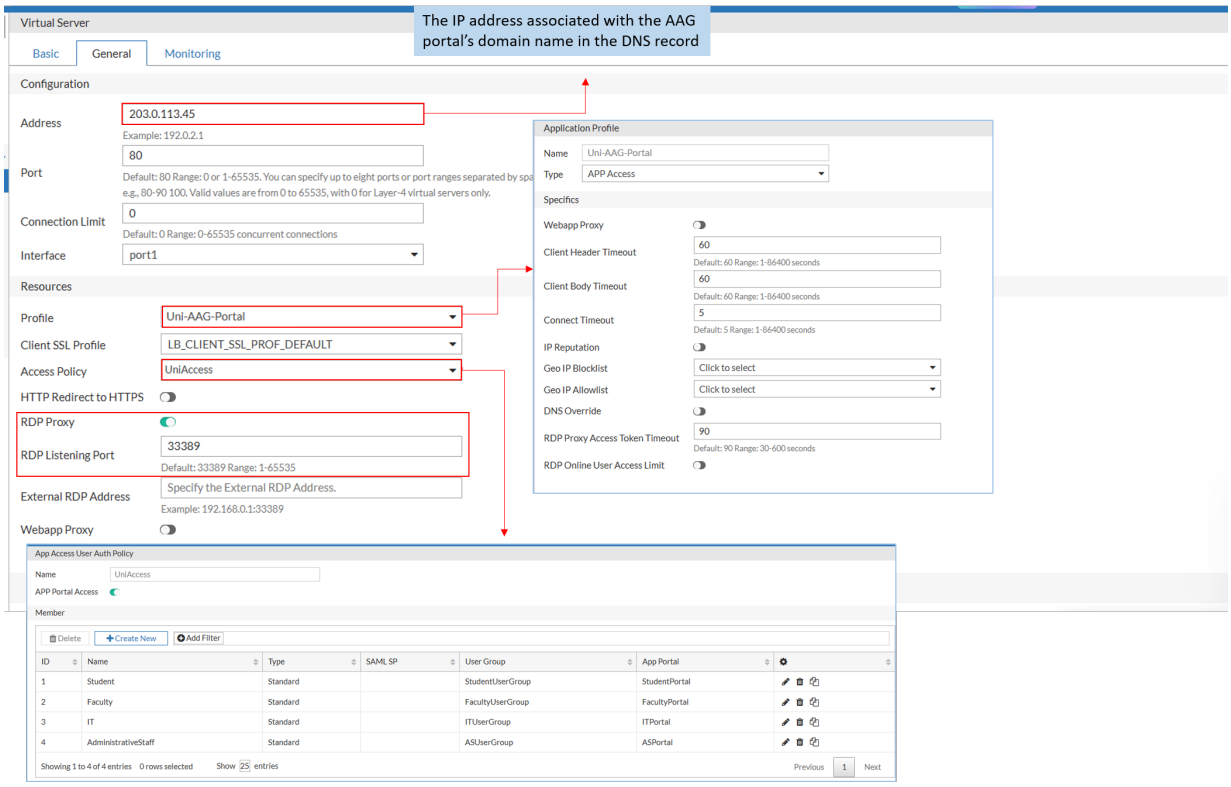
In some deployments, this IP address resides on an upstream device, such as a perimeter firewall. In such cases, the firewall may perform DNAT to translate the destination IP to a different internal IP address before forwarding the traffic. Configure the **Address** field with this translated (NAT) IP address so that FortiADC can receive the forwarded traffic.

- **Port** – Enter the listening port for user connections (e.g., 443 for HTTPS).
- **Connection Limit** – Optionally, define a maximum number of concurrent connections.
- **Interface** – Select the network interface on which the virtual server will listen.

c. In the **General** tab, configure the following **Resources** settings:

- **Profile** – Select a predefined or user-defined APP Access profile. This enables agentless access features and exposes AAG-specific settings. The configuration options will adjust to what is applicable to this profile type.
- **Access Policy** – Attach the corresponding Access Policy that enforces authentication and authorization rules for App Portal access.
- **RDP Proxy** – If **Native RDP** and **Remote App** bookmarks are included in the portal, enable the **RDP Proxy** option. This activates RDP-specific settings configured in the APP Access profile. The **Native RDP** and **Remote App** bookmarks appear on the portal only when **RDP Proxy** is enabled.
- **RDP Listening Port** – Defines the TCP port on which FortiADC listens for inbound RDP client connections. This setting is only visible when **RDP Proxy** is enabled and applies specifically to Native RDP and Remote App bookmarks. The default port is 3389, and the valid range is 1-65535.
- **External RDP Address**: Optional. The format should be `IP:Port`.

  Its IP part is typically the public IP associating with the domain name of the AAG portal. This ensures that RDP sessions can be successfully established in deployments where FortiADC is located behind a NAT device. See What's the External RDP Address? on page 62
- **Webapp Proxy**: Refer to Special Scenario: "Web App - Internal" Application Sharing the Same IP with AAG Portal.



The **Client SSL Profile** field is available when configuring an AAG App Portal virtual server, but only partial functionality is supported.

Supported settings:

- SSL Cipher Suite List
- TLS v1.3 Cipher Suite List
- Allowed SSL Versions
- Local Certificate Group

All other parameters configured in the Client SSL Profile are not supported in this context and have no effect when applied to the App Portal virtual server.

4. Click **Save**.
   The configured Virtual Server will now route users through the App Portal, enabling secure, agentless access to internal web and desktop applications.

RDP Proxy Requirement for Native RDP and RemoteApp

The visibility of Native RDP and RemoteApp bookmarks in the App Portal depends on whether **RDP Proxy** is enabled on the associated Virtual Server.

By default, RDP Proxy is **disabled**, and any configured Native RDP or RemoteApp bookmarks will not appear in the App Portal until it is enabled.

To ensure these application types are accessible to users, **enable RDP Proxy** in the Virtual Server configuration.

What's the External RDP Address?

When the bookmark type is **Native RDP** or **Remote App**, FortiADC operates as a RDP proxy.

- When a user clicks the **Native RDP** or **Remote App** bookmark, an `.rdp` file is downloaded to the user's device.
- This file uses the **Address** of the portal virtual server as the destination IP for the RDP session.
- As a result, the RDP connection is first established to FortiADC, which then proxies the session to the backend RDP session host.

This mechanism raises an important consideration: how the **Address** of the portal virtual server is configured directly affects whether the RDP session can be successfully established.

The following sections describe two common deployment scenarios.

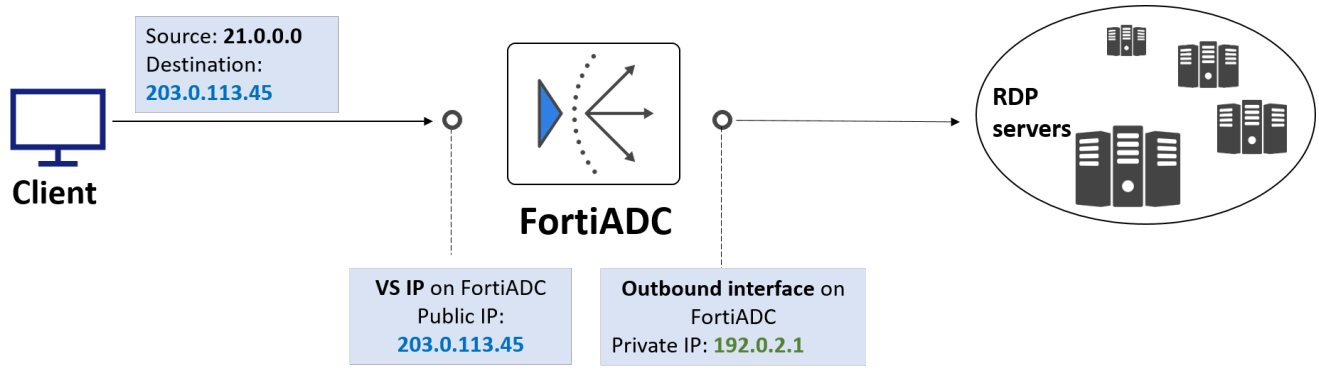
Scenario 1: Public IP address resides on FortiADC

In this scenario, the public IP address (for example, `203.0.113.45`) associated with the AAG portal domain name (for example, `example.com`) is configured directly on FortiADC.

In this case:

- The `.rdp` file uses `203.0.113.45` as the destination IP.
- The RDP client connects directly to FortiADC.
- FortiADC receives the connection and proxies it to the backend RDP session host.

No additional configuration is required.

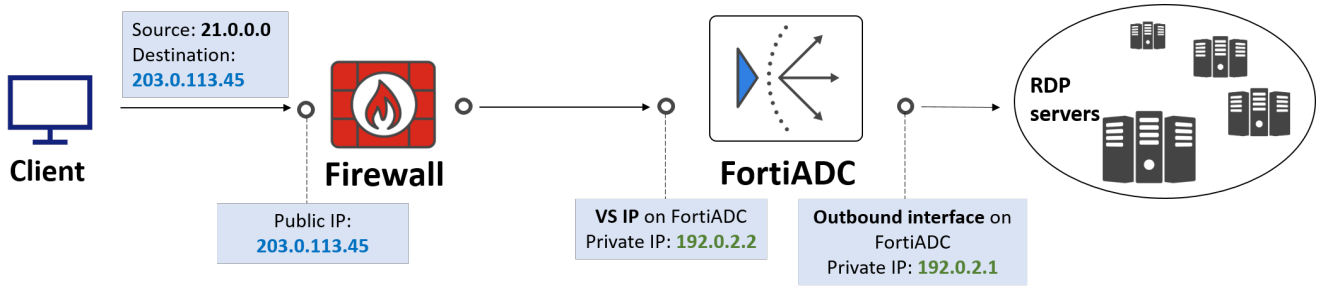


Scenario 2: Public IP address resides on an upstream device

In a more common deployment, a perimeter firewall (for example, FortiGate) is deployed in front of FortiADC. In this case, the public IP address (`203.0.113.45`) is configured on the firewall, not on FortiADC.

The firewall performs DNAT to forward traffic to FortiADC:

```
Public IP (FortiGate): 203.0.113.45 → FortiADC AAG VS Private IP: 192.0.2.2
```



Web access behavior

This architecture works correctly for web-based access such as Web SSH, Web VNC:

1. The user accesses `https://example.com`.
2. DNS resolves to `203.0.113.45`.
3. The client connects to the firewall.
4. The firewall translates the destination IP to `192.0.2.2`.
5. FortiADC receives the request and proxies it to the backend server.

RDP access issue

However, this setup introduces a problem for RDP access.

When a user launches **Native RDP** or **Remote App**:

1. FortiADC generates an `.rdp` file.
2. The Destination IP in the file is the **Address** of the portal virtual server (`192.0.2.2`)
3. This is a private IP address and is not reachable from the Internet.

As a result, the RDP client cannot establish the connection.

Solution: External RDP Address

To resolve this issue, configure the **External RDP Address** on the AAG portal virtual server.

In this example:

- The RDP proxy on FortiADC listening on `192.0.2.2:33389`
- Set **External RDP Address** to `203.0.113.45:63389`

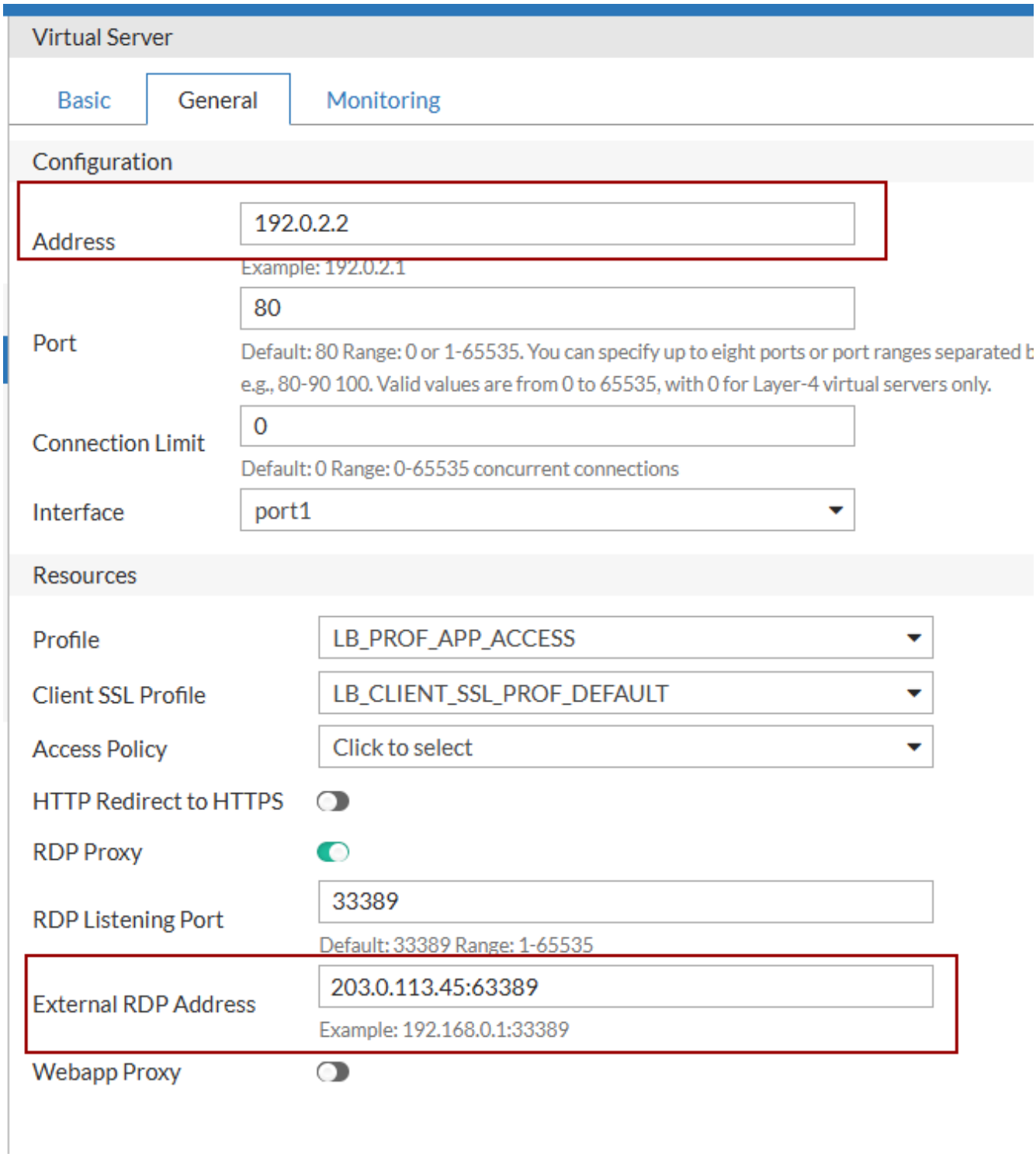
After configuration:

- The `.rdp` file uses `203.0.113.45:63389` as the destination IP and port
- The RDP client connects to Public/External address: `192.0.2.2:33389`
- The firewall performs DNAT from `203.0.113.45:63389` to `192.0.2.2:33389`.

FortiADC receives the connection and proxies it to the backend RDP session host

This ensures that RDP sessions can be successfully established in deployments where FortiADC is located behind a NAT device.

Below is the AAG Portal virtual server settings when the VS IP uses a private IP address `192.0.2.2`, and the **External RDP Address** is configured with the public IP `203.0.113.45:63389` that associates with the AAG portal domain name.


Virtual Server
Basic
General
Monitoring
Configuration
Address
192.0.2.2
Example: 192.0.2.1
Port
80
Default: 80 Range: 0 or 1-65535. You can specify up to eight ports or port ranges separated b
e.g., 80-90 100. Valid values are from 0 to 65535, with 0 for Layer-4 virtual servers only.
Connection Limit
0
Default: 0 Range: 0-65535 concurrent connections
Interface
port1
Resources
Profile
LB_PROF_APP_ACCESS
Client SSL Profile
LB_CLIENT_SSL_PROF_DEFAULT
Access Policy
Click to select
HTTP Redirect to HTTPS
RDP Proxy
RDP Listening Port
33389
Default: 33389 Range: 1-65535
External RDP Address
203.0.113.45:63389
Example: 192.168.0.1:33389
Webapp Proxy

Accessing the AAG App Portal

After setting up the App Portal and deploying the Agentless Application Gateway (AAG) Virtual Server with the appropriate Access Policy, users can securely log in through their browser to access internal enterprise applications. The AAG App Portal serves as a centralized interface for launching web, desktop, and terminal-based resources—no client agents required.

This information is also available in the FortiADC 8.0.0 Administration Guide:

- Accessing the AAG App Portal

FortiADC 8.0.1 adds support for multi-factor authentication (MFA) at App Portal login for Local and RADIUS users and introduces automatic language detection for the App Portal interface.

This section provides an overview of the end-user experience, including authentication flow, portal navigation, application access, and session handling.

Topics Covered:

- **Signing In and User Session Management** – Authentication flow, MFA, and portal language behavior.
- **Exploring the AAG Portal Interface** – Layout and navigation overview.
- **Launching Applications** – Accessing applications including Web RDP, Native RDP, Remote App, `Web SSH, Web VNC, Web Telnet, and web apps.
- **Managing Sessions and Logging Out** – Viewing active sessions and secure logout.

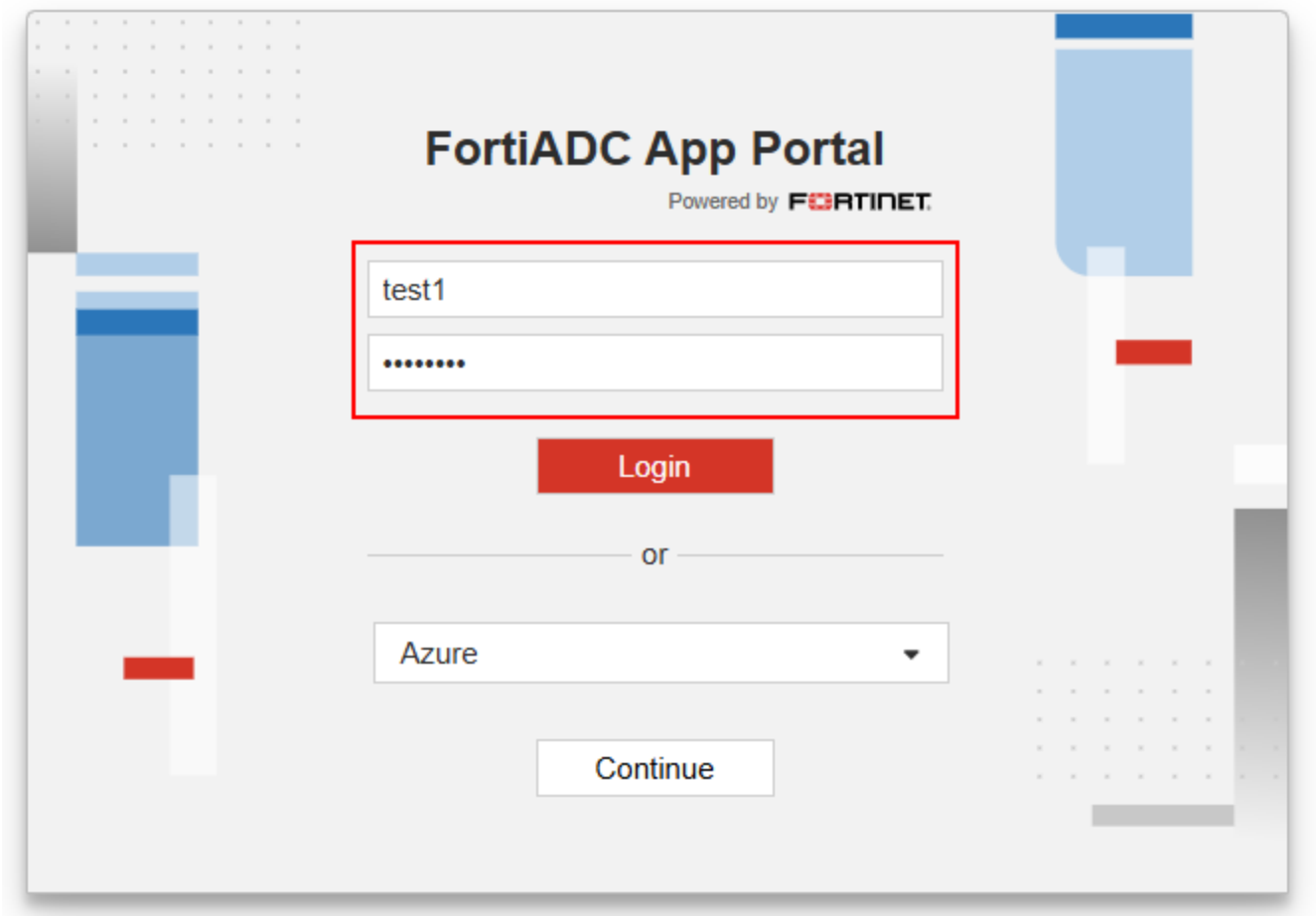
Signing In and User Session Management

Users authenticate based on the method defined in the Access Policy bound to the AAG Portal Virtual Server. FortiADC supports a range of authentication mechanisms, including:

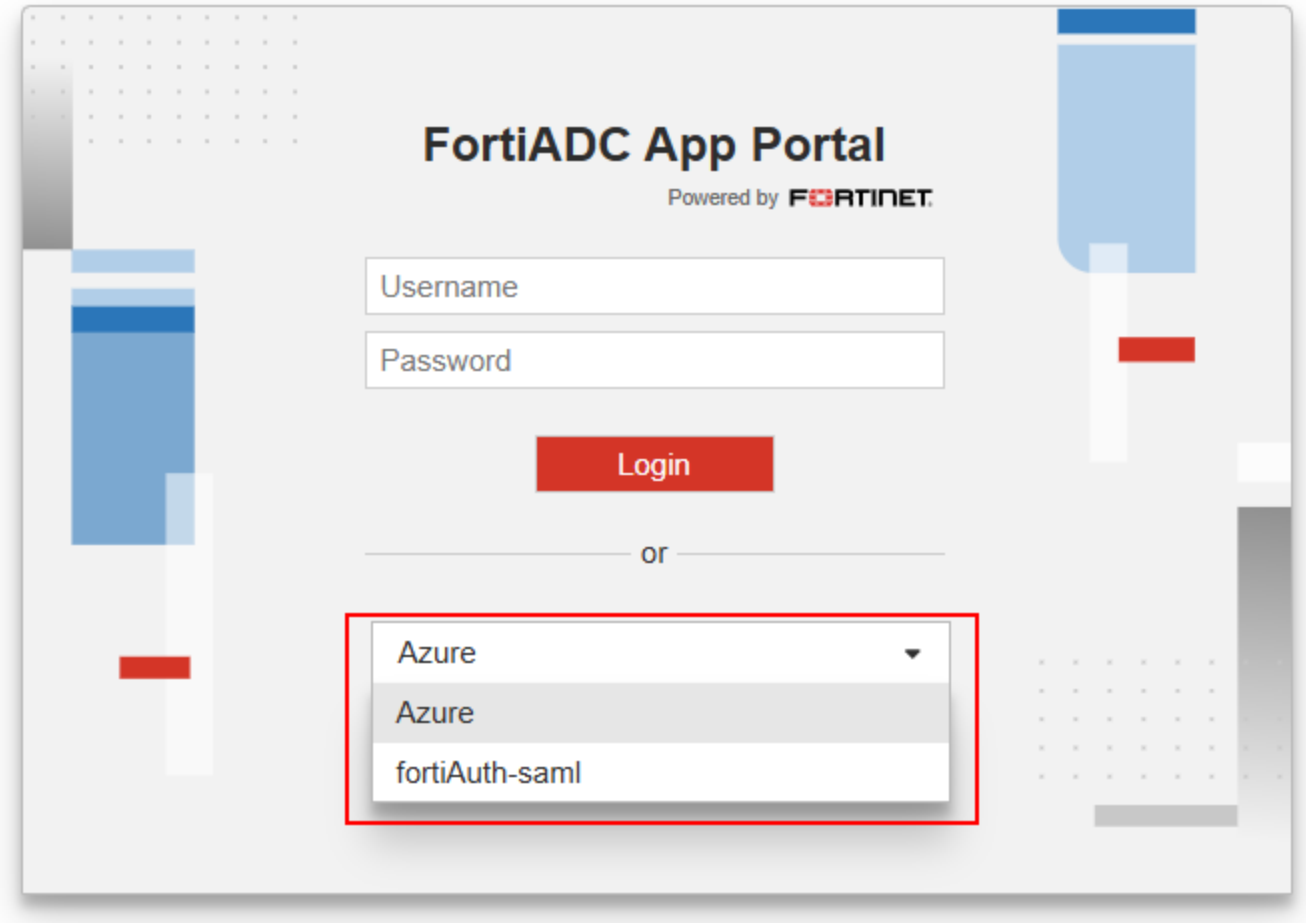
- **Local authentication** – Users enter credentials defined on FortiADC.
- **Remote authentication** – Users authenticate via LDAP, RADIUS, or SAML-based identity providers.

Once authenticated, FortiADC initializes a user session and enforces idle timeouts.

Local, LDAP, or Radius user login page



SAML user login page



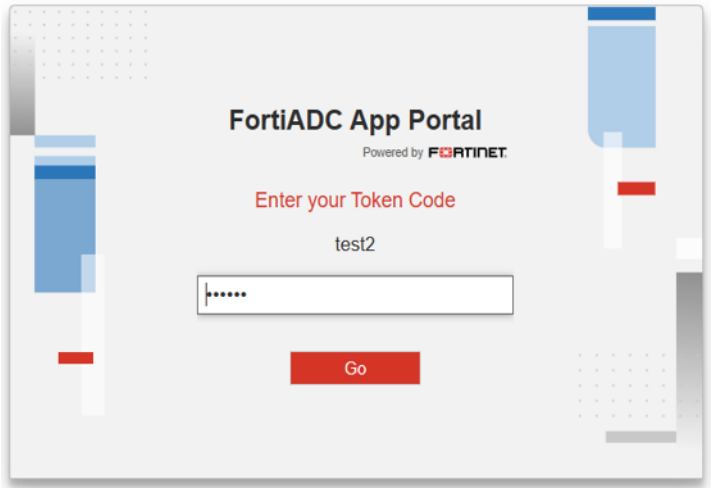
Multi-Factor Authentication (MFA)

FortiADC AAG depends on 3rd party MFA capability for LDAP and SAML users. If **multi-factor authentication (MFA)** is enabled for Local or RADIUS users, the App Portal adds an additional verification step after successful primary credential entry.

Users are prompted to verify their identity using one of the following methods:

- Enter a FortiToken one-time passcode (OTP).
- Approve a push notification on their registered device.

Portal access is granted only after successful completion of the second-factor verification.



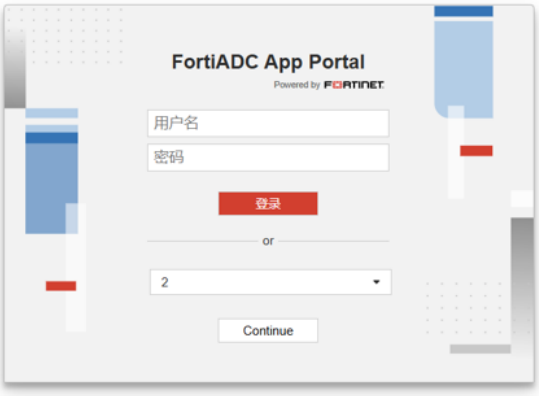
Automatic Language Detection

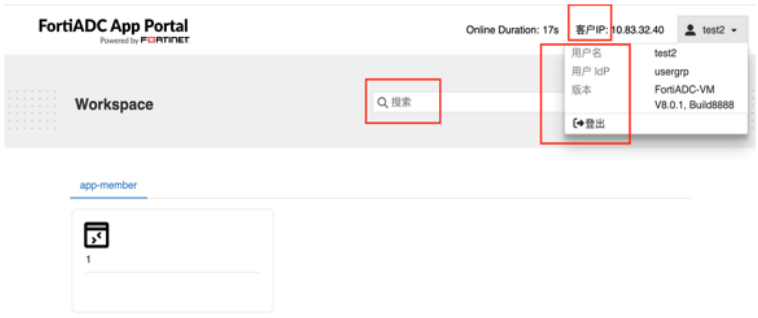
FortiADC automatically adapts the App Portal interface language to match the user's browser settings. No configuration is required–language detection occurs automatically at login.

When a browser requests one of the supported languages, the App Portal displays all pages and login prompts in that language. If the browser requests an unsupported language, the interface defaults to English.

Supported languages:

- English
- Simplified Chinese
- Traditional Chinese
- Japanese
- Spanish
- Portuguese



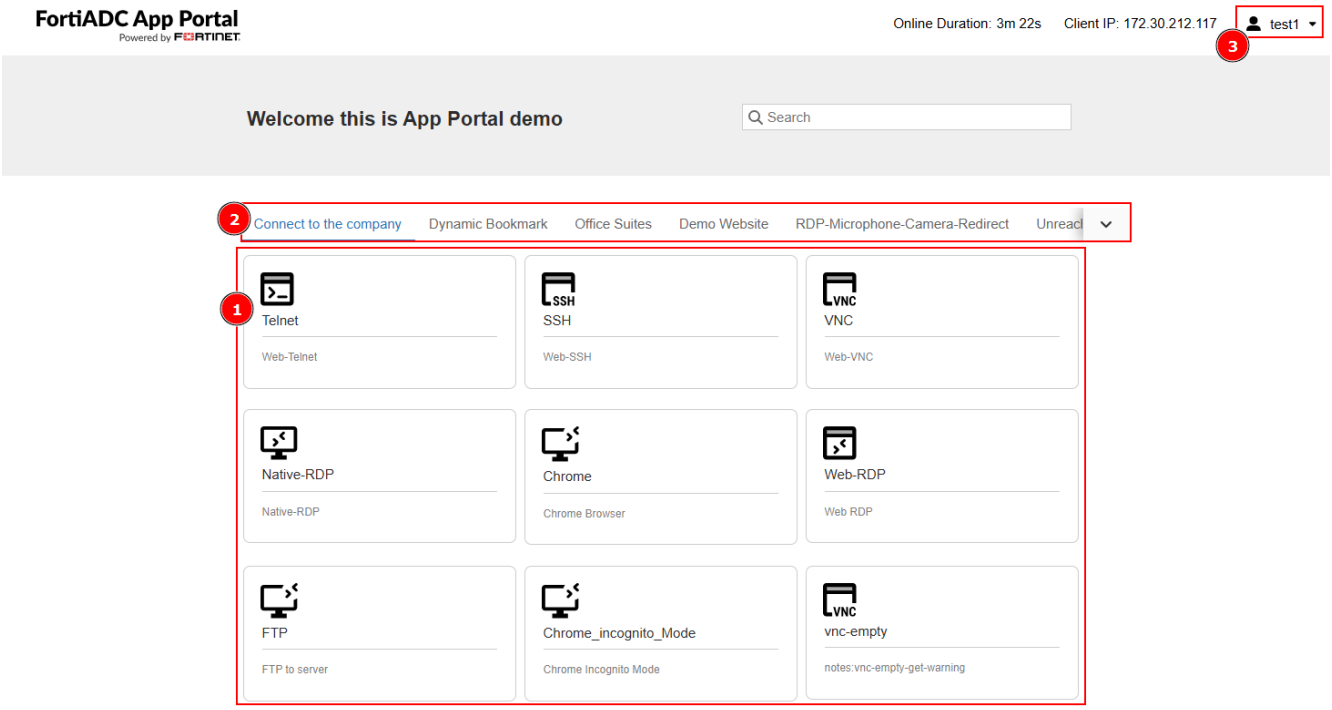


Exploring the AAG Portal Interface

Upon successful login, users are presented with the AAG App Portal interface. The interface is browser-based, intuitive, and dynamically reflects the user's access permissions.

The portal includes:

- **Application Tiles** – Icons representing Web RDP, Native RDP, Remote App, Web SSH, Web VNC, Web Telnet, and web application resources.
- **App Grouping** – Applications grouped by assigned App Group.
- **User Information** – Includes user information and option to logout.



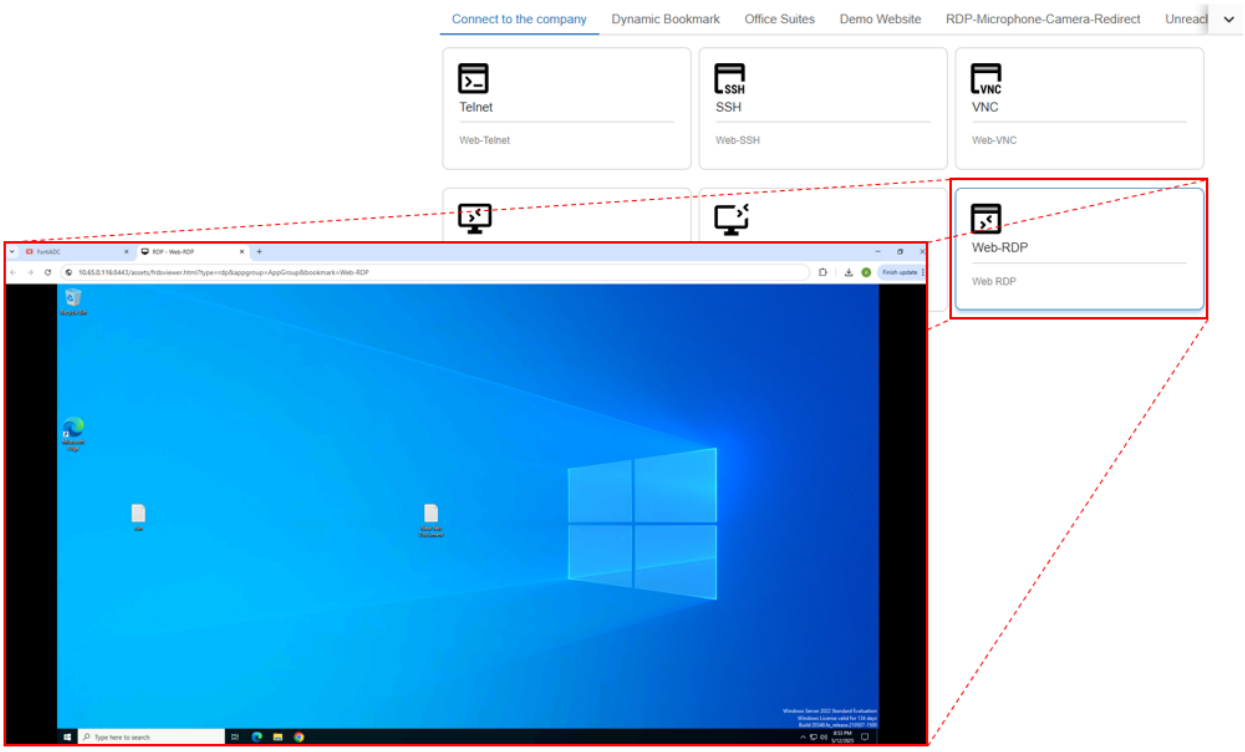
Launching Applications

Users can launch applications directly from the portal without requiring any local agents. Supported application type include:

- **Web Applications** – Opens in a new tab or embedded frame.
- **Native RDP / Remote App** – RDP access via the installed Remote Desktop Client.
- **Web RDP / SSH / Telnet / VNC** – Browser-based host or terminal access, rendered using HTML5 viewer.

RDP connections honor the settings configured in the APP Access profile:

- **RDP Proxy Access Token Timeout** – Controls how long a token remains valid.
- **RDP Online User Access Limit** – Terminate the RDP connection when user is not at logged-in state.

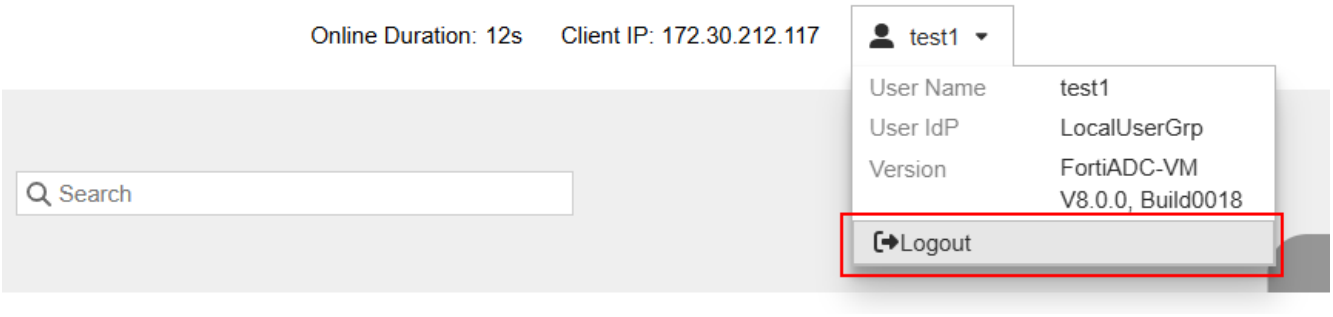


Managing User Sessions and Logging Out

Users can track their active sessions within the portal and safely log out once done. FortiADC enforces session cleanup on logout.

Session-related capabilities:

- **Switching Between Applications** - Seamless navigation without needing to log in again for each application.
- **Manual Logout** - Enforce session cleanup.



AAG Implementation Scenarios

FortiADC's Agentless Application Gateway (AAG) is designed to provide secure, browser-based access to internal applications without requiring client-side agents or VPNs. AAG is especially effective in environments where users access sensitive resources from remote locations, unmanaged devices, or hybrid cloud infrastructures.

This information is also available in the FortiADC 8.0.0 Administration Guide:

- AAG Implementation Scenarios

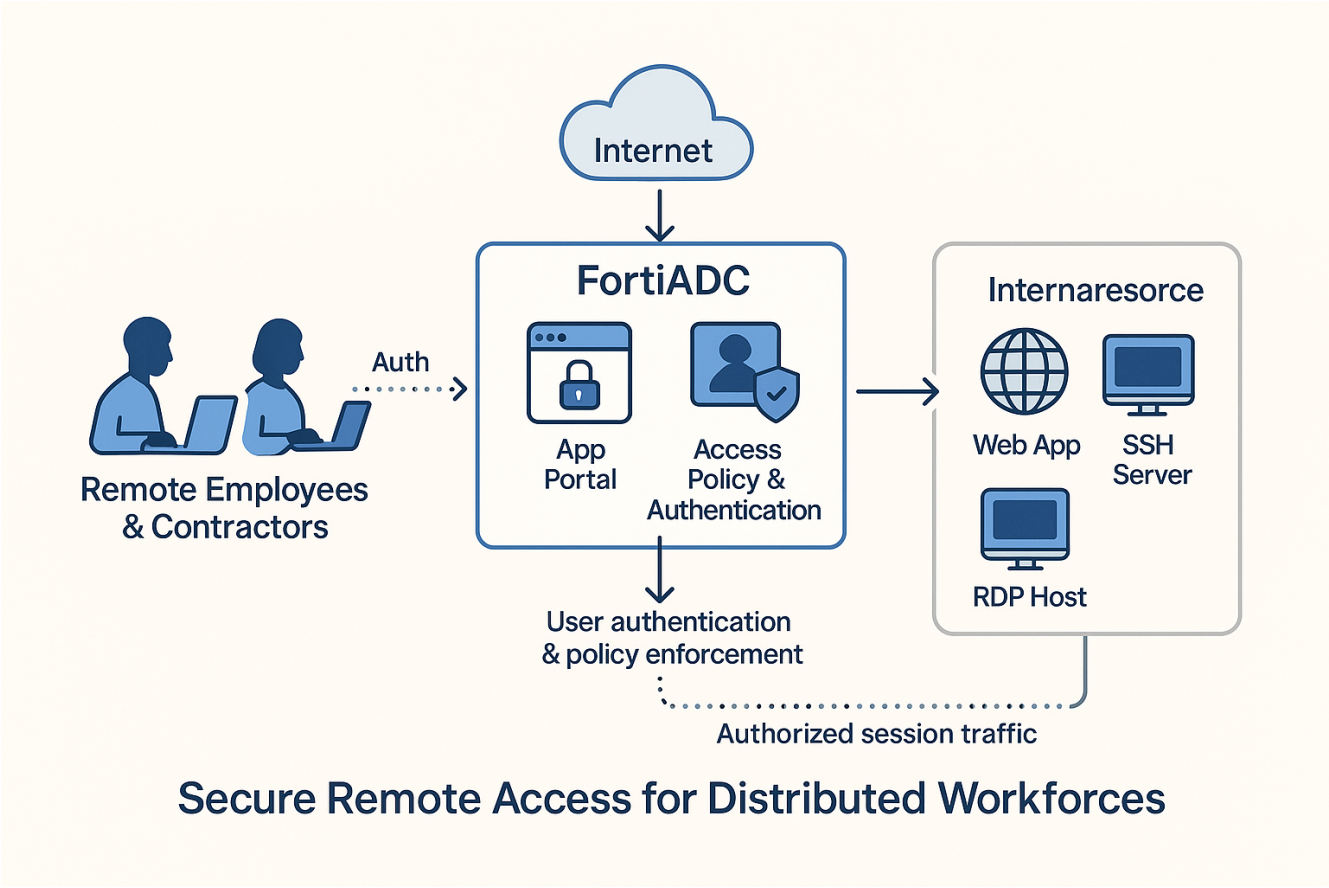
FortiADC 8.0.1 introduces a new deployment scenario for internal web application access through the **Web App Proxy** feature and **Web App – Internal** bookmarks.

This section outlines three practical deployment scenarios where FortiADC AAG can be applied to meet specific organizational needs:

- Scenario 1: Secure Remote Access for Distributed Workforces on page 73
- Scenario 2: Unified Access to Hybrid Cloud Applications on page 74
- Scenario 3: Secure, Clientless RDP Access for Remote and BYOD Users on page 75
- Scenario 4: Secure Browser-Based Access to Internal Web Applications on page 76

Each scenario presents specific challenges, followed by a technical walkthrough of the AAG solution and the resulting operational benefits.

Scenario 1: Secure Remote Access for Distributed Workforces



Challenge:

Organizations need to grant secure, browser-based access to internal applications for remote employees and contractors—without relying on VPN clients or exposing internal systems to unmanaged endpoints.

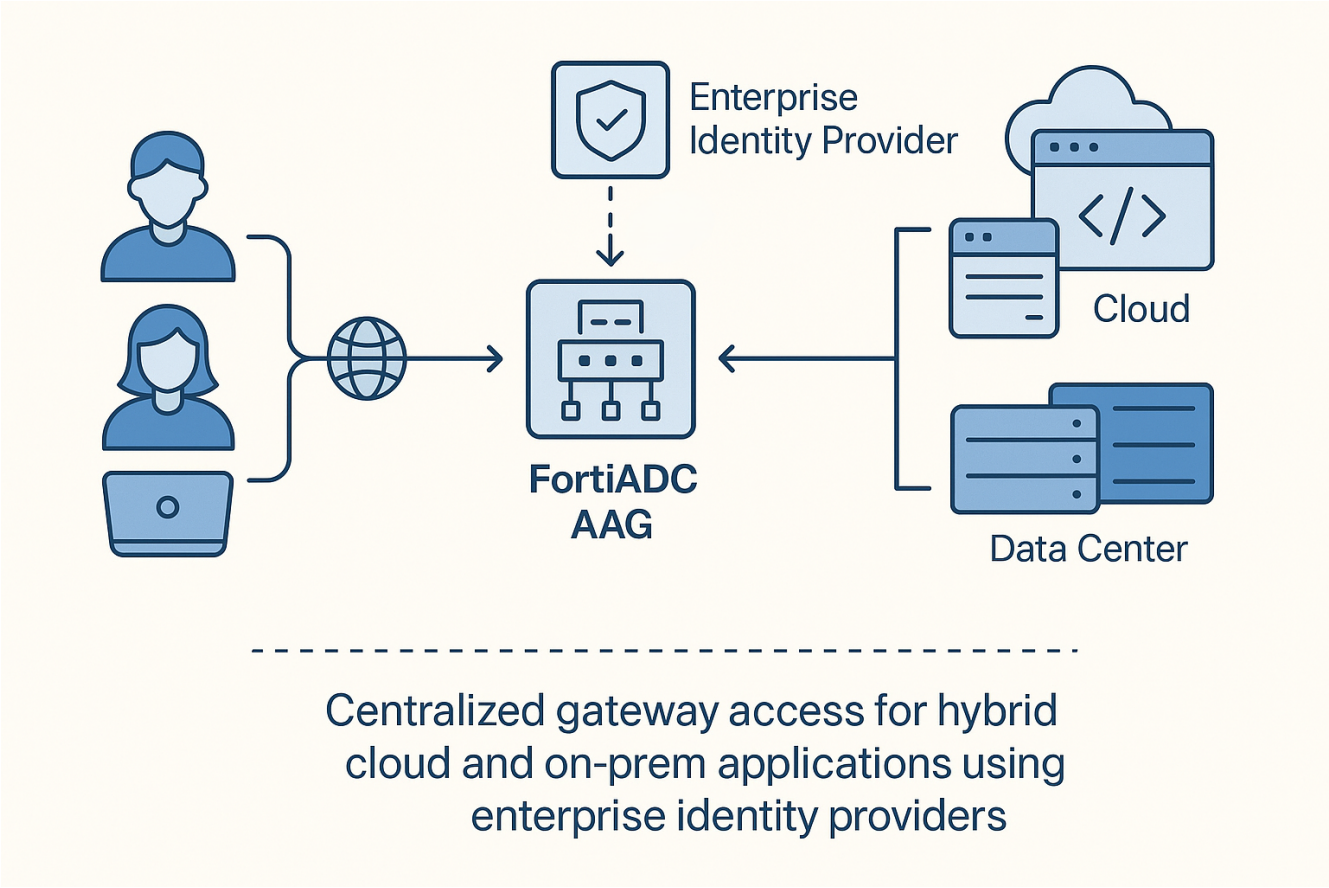
AAG Solution:

FortiADC AAG rovides secure, clientless access to internal web, SSH, and RDP services through a unified browser-based App Portal. A Layer 7 virtual server acts as the central gateway, enforcing authentication and session control based on Access Policies. AAG supports Local, RADIUS, and SAML-based authentication, with optional MFA for Local and RADIUS users. Application access is defined through App Groups and bookmarks, ensuring users see only the applications they are authorized to use.

Outcome:

- Simplified access from unmanaged or non-compliant devices
- Reduced VPN complexity and administrative overhead
- Centralized authentication and session-level visibility

Scenario 2: Unified Access to Hybrid Cloud Applications



Challenge:

Enterprises operating in hybrid environments struggle to provide secure, seamless access to both cloud and on-premises applications while maintaining consistent identity and policy enforcement.

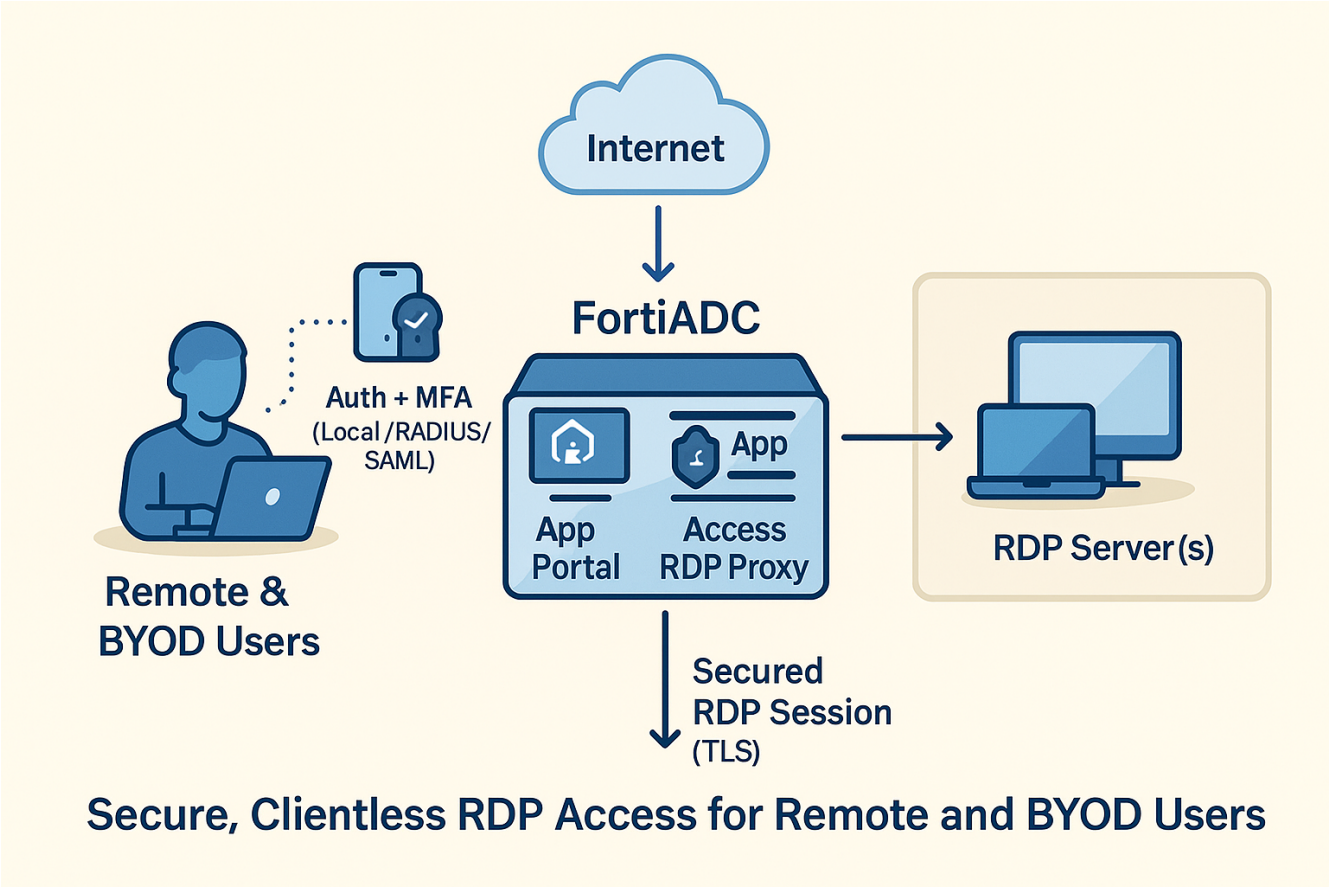
AAG Solution:

FortiADC AAG centralizes access across environments using multiple Virtual Servers grouped by application location. Each is configured with APP Access profiles and Access Policies tied to federated authentication methods such as SAML. Unified access control policies ensure identity-aware enforcement and consistent auditing.

Outcome:

- Single point of access for hybrid app environments
- Reduced operational complexity via centralized policy management
- Strengthened compliance through unified logging and identity-based controls

Scenario 3: Secure, Clientless RDP Access for Remote and BYOD Users



Challenge:

Organizations need to provide secure remote desktop access for users on BYOD (Bring Your Own Device) or unmanaged endpoints—without relying on installed RDP clients or introducing the complexity of VPN deployments.

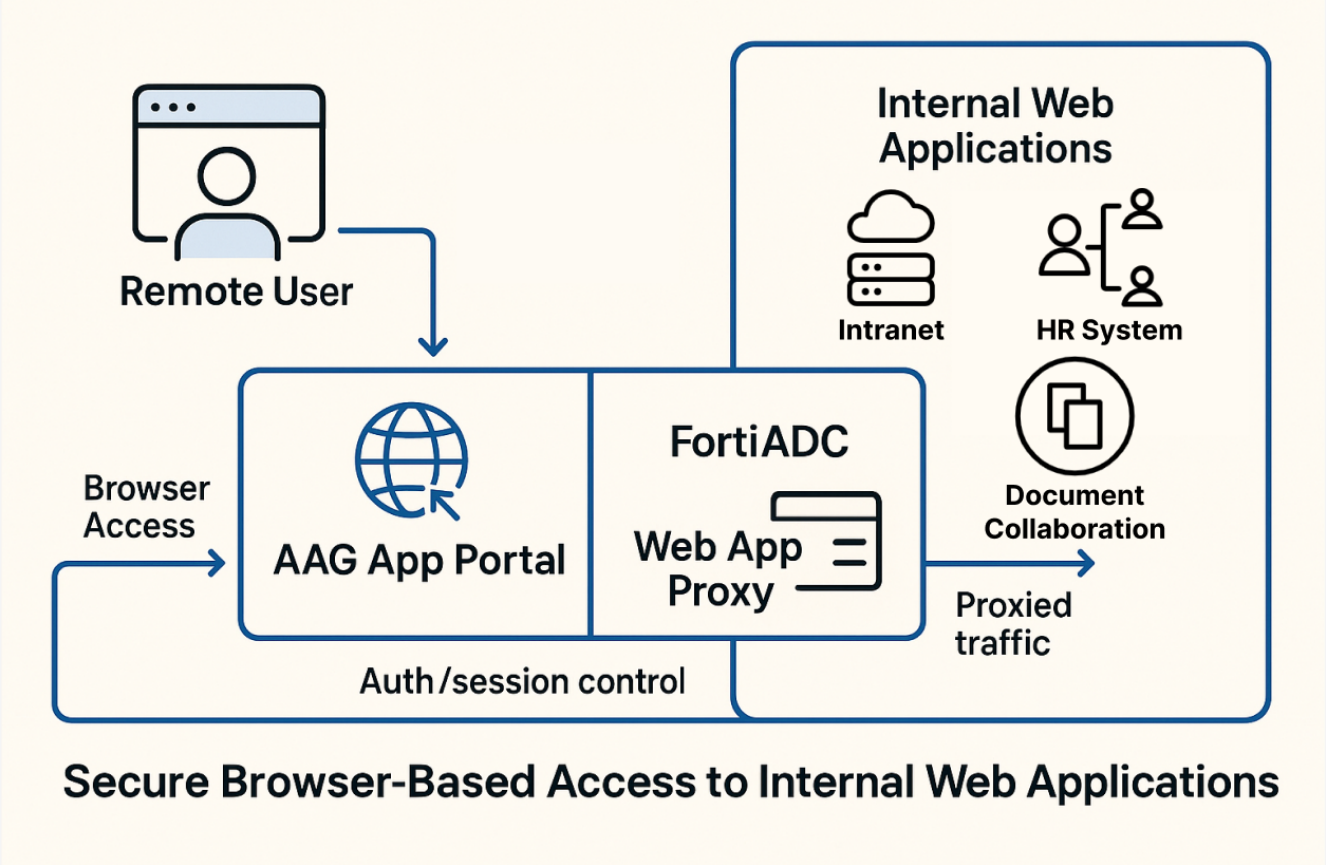
AAG Solution:

FortiADC AAG delivers fully browser-based RDP access through an integrated HTML5 viewer. A Layer 7 virtual server with an App Access Profile and RDP Proxy handles user authentication, session control, and traffic forwarding. AAG supports Local, RADIUS, and SAML authentication, with optional MFA for Local and RADIUS users. Additional endpoint controls—such as clipboard, drive, or file transfer restrictions—can be enforced on the backend RDP session host to enhance security.

Outcome:

- Fully browser-based RDP access from unmanaged devices
- Reduced client and VPN management overhead
- Secure session handling with enforced timeouts and session visibility

Scenario 4: Secure Browser-Based Access to Internal Web Applications



Challenge:

Internal business applications—such as intranet portals, HR systems, and document collaboration platforms—are often hosted inside private networks and not directly reachable by remote users. Traditional access solutions typically rely on VPNs, which expose internal addresses, increase administrative overhead, and create inconsistent user experiences across devices. Organizations need a way to deliver these internal web applications securely to remote and external users—without compromising network security or requiring client software.

AAG Solution:

FortiADC's Agentless Application Gateway (AAG) provides secure, browser-based access to internal web applications published through a FortiADC virtual server configured with Web App Proxy. Acting as a reverse proxy, FortiADC authenticates users through the AAG App Portal, inspects HTTP/HTTPS traffic, and forwards authorized sessions to backend servers.

Internal web applications are presented in the App Portal as Web App - Internal bookmarks, allowing authenticated users to launch them directly from a browser while maintaining centralized authentication, session control, and logging. Whether accessed from the App Portal or directly through the published URL, AAG enforces the same authentication flow—redirecting unauthenticated users to the portal login page before granting access.

Outcome:

- Provides secure, browser-based access to internal web applications without requiring VPN software or client agents
- Centralizes authentication, authorization, and logging through the AAG App Portal
- Maintains full control and visibility over user sessions with consistent policy enforcement
- Reduces exposure of internal resources by proxying all traffic through FortiADC's Web App Proxy layer

AAG Debugging and Troubleshooting

FortiADC provides both GUI and CLI-based diagnostic tools to monitor and troubleshoot Agentless Application Gateway (AAG) activity. These tools allow administrators to inspect authentication events, session status, traffic usage, and connection behavior in real time.

This information is also available in the FortiADC 8.0.0 Administration Guide and CLI Reference:

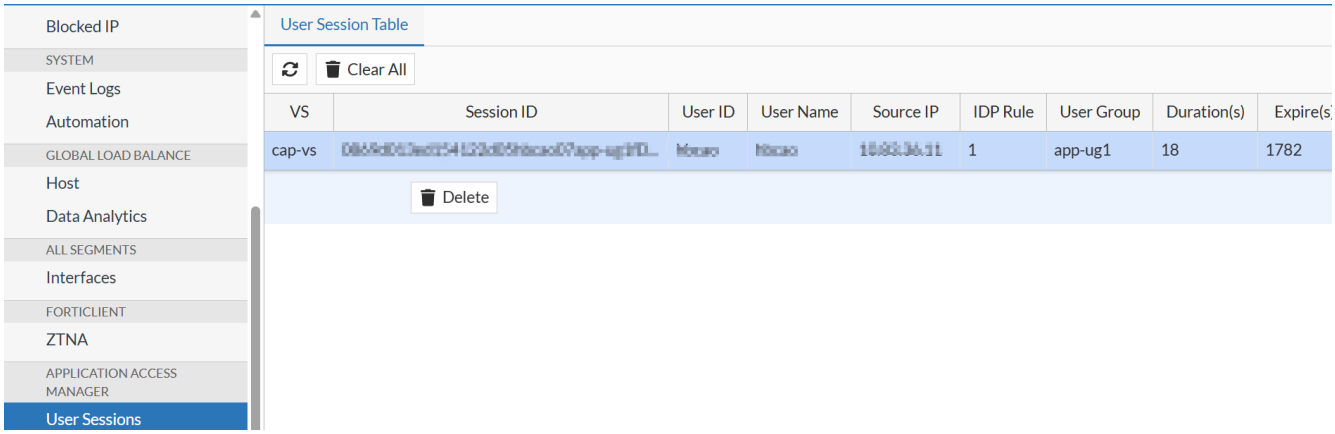
- AAG Debugging and Troubleshooting
- AAG User Sessions
- diagnose app-publish
- diagnose debug module fnginx

GUI-Based Monitoring and Logging

FortiView – User Session

Navigate to **FortiView > User Session** to view real-time information about users currently authenticated through the AAG App Portal.

You can clear a specific user session, or clear all user sessions.



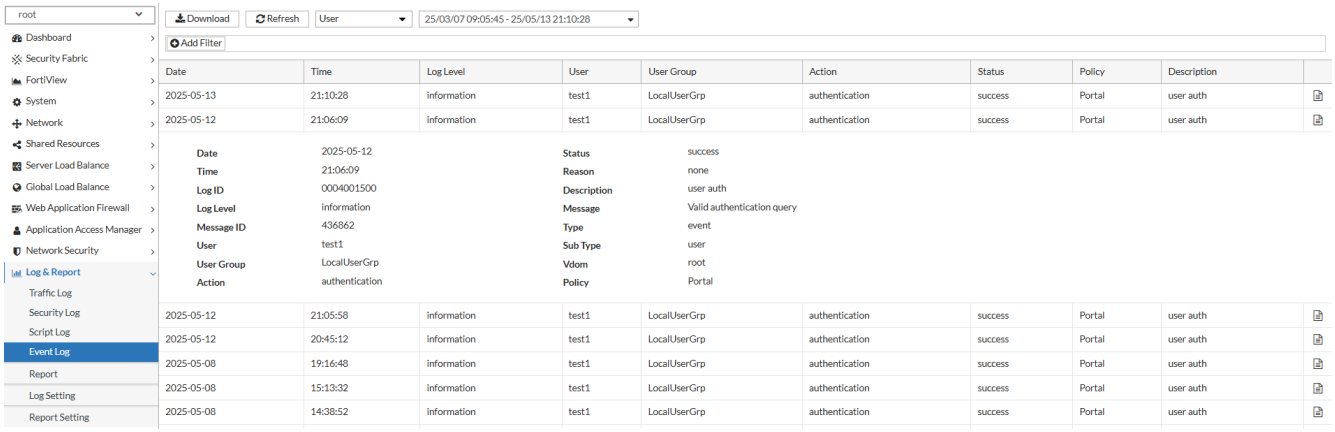
The table displays the following fields:

- VS – Virtual Server name handling the session
- Session ID – Unique identifier for the user session
- User ID – Internal user identifier
- User Name – Authenticated username
- Source IP – Origin IP address of the client
- IDP Rule – Identity provider rule that applied
- User Group – Assigned group membership
- Duration (s) – Time the session has been active
- Expire (s) – Remaining time before session expiration

This dashboard is useful for tracking session state, identifying active users, and confirming authentication success.

Event Logs – Authentication Activity

To review authentication behavior and diagnose login issues, navigate to **Log & Report > Event Log** and filter by **User**.



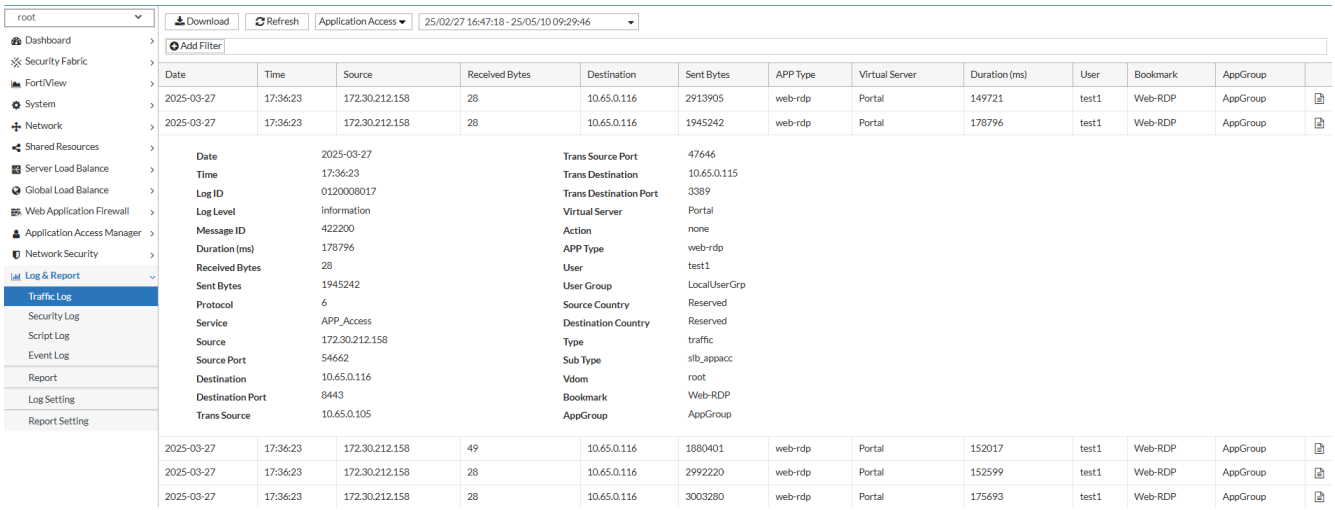
Each log entry provides detailed contextual information, including:

- Date – The date the event occurred
- Time – The timestamp of the event
- Log Level – Severity or verbosity of the log (e.g., notice, warning)
- User – Username of the individual attempting access
- User Group – Group membership associated with the user
- Action – Type of authentication event (e.g., login, logout, failure)
- Status – Result of the event (e.g., success, failed)
- Reason – High-level reason code for the event (e.g., invalid credentials)
- Description – Summary of the action performed
- Message – Detailed message or explanation from the system
- Vdom – Virtual domain where the event occurred
- Policy – Access Policy name that governed the authentication flow

Use this log view to validate user authentication behavior, troubleshoot policy misconfigurations, or verify enforcement across user groups and virtual domains.

Traffic Logs – Application Access

Navigate to **Log & Report > Traffic Log** and filter by **Application Access** to analyze usage and connection details for all applications accessed through the AAG App Portal.



Each entry includes:

- Date, Time, Log Level
- Duration (ms)
- Received Bytes, Sent Bytes
- Protocol, Service
- Source, Source Port
- Destination, Destination Port
- Translated Source/Destination Port
- Virtual Server
- Action (accept, deny)
- APP Type - Application type (e.g., RDP, SSH)
- User Group
- Source Country, Destination Country
- Bookmark - The application bookmark used
- AppGroup - The App Group containing the bookmark

This log provides visibility into user activity, bandwidth usage, and resource access patterns.

CLI-Based Diagnostic Tools

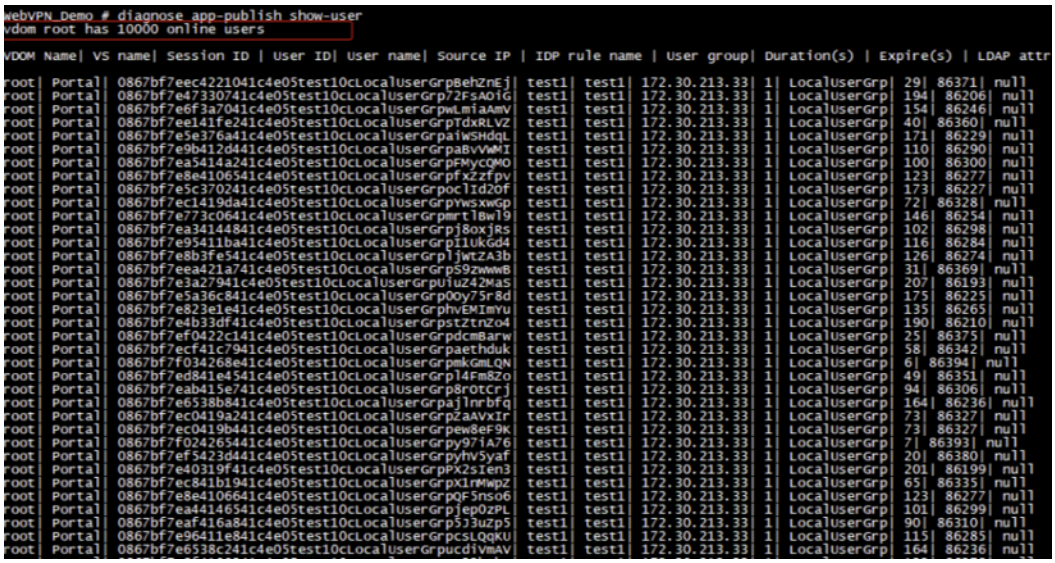
App-Publish Diagnostic Commands

Use the `diagnose app-publish` command set to inspect AAG configuration and manage user sessions directly from the CLI:

Command	Description
`show-config`	Displays the current AAG configuration, including App Portals, App Groups, and bookmarks.
`show-user`	Lists all currently authenticated AAG users and session details.
`show-connection`	Displays active application connections established through the AAG portal (e.g.,

Command	Description
	RDP, SSH sessions).
`kickoff-user`	Terminates the session of a specific AAG user. Useful for forced logout or session resets.
`clear-user`	Clears all tracked session state for a specified user.
`clear-connection`	Terminates all AAG application connections for a specific user or session.

Example:



Module-Level Debug Logging

- To enable verbose logging for the AAG module (app-publish), use the following command to view runtime behavior and debug issues:

  ```
  diagnose debug module fnginx app-publish
  ```

  This command provides backend insight into:

  - Portal rendering behavior
  - Bookmark resolution and visibility issues
  - Session negotiation
  - Authentication handoff and result mapping
  - Application proxy session establishment and teardown

  Enable debug output when troubleshooting issues related to user login failures, bookmark rendering problems, or connection errors.

  Example:

```
webVPN_Demo # diagnose debug  module  fnginx app-publish

webVPN_Demo # diagnose  debug  enable

webVPN_Demo # [ngx_http_cap_limit_conn_handler:234]
[ngx_http_cap_limit_conn_handler:238]
[ngx_http_adc_auth_handler:3597] No valid app cookie for request URL /api/login/auth
[ngx_http_adc_auth_handler:3613] found auth cookie kcfjAVAwItRkLJCkIBlvk3jCDPY8oPqc5YzytRn3k+ZDTjAWFe4VZlGe
[ngx_http_adc_auth_handler:3627] auth cookie timestamp: 1740196736, current timestamp: 1740196780
[ngx_http_adc_auth_request_body_handler:3110]
[ngx_http_adc_auth_request_body_handler:3127] data is in memory, size: 72
[ngx_http_adc_auth_request_body_handler:3138] body total length is 72
[ngx_http_adc_auth_request_body_handler:3185] auth type is standard
[ngx_http_adc_auth_request_body_handler:3230] found user group RemoteUserGrp
[ngx_http_adc_auth_request_body_handler:3255] username: user100, password: XXX
[ngx_http_adc_auth_send_to_authd:1392]
[ngx_http_adc_auth_send_to_authd:1415] The current vdom is root
[ngx_http_authd_write_handler:1352] username&&password in query: user100, XXX
[ngx_http_authd_read_handler:1159] result: Success
[ngx_http_authd_read_handler:1223] Retrieved attributes for user [user100]
[ngx_http_authd_read_handler:1228] [title] -> [10.65.0.114]
[ngx_http_authd_read_handler:1228] [Test2] -> []
[ngx_http_authd_read_handler:1228] [department] -> [10.65.0.113]
[ngx_http_authd_read_handler:1228] [Test4] -> []
```

- To view AAG configurations, run the following command:

```
diagnose debug module fnginx conf
```

- To view the authentication process, run the following command:

```
diagnose debug module authd
```

- To view the Web App-Internal-Advanced bookmark process, run the following command:

```
diagnose debug module httproxy
```

-

Web Application Firewall

The FortiADC 8.0 release includes new features and enhancements in **Web Application Firewall**:

WAF Signature Support for HTTP/3 and HTTP/2 8.0.3 on page 84

FortiADC 8.0.3 now supports Web Application Firewall (WAF) Web Attack Signature scanning for HTTP/3 and HTTP/2 Virtual Servers. This update allows you to apply robust security policies to modern high-performance traffic, ensuring your services are protected against known web-based vulnerabilities.

RESTful API Input Security Check 8.0.3 on page 85

You can now benefit from automated, deep-level security inspections for the FortiADC RESTful API to detect and prevent sophisticated exploits such as command injection and path traversal. FortiADC 8.0.3 introduces a specialized, independent signature database that performs real-time validation of API requests before business logic is executed. This feature ensures robust protection for the management interface by inspecting multiple input locations including headers, request bodies, query arguments, and URL paths without requiring any manual configuration.

WAF Adaptive Learning 2.0 on page 86

WAF Adaptive Learning 2.0 enhances the original engine with expanded module coverage, exception-aware tuning, and improved operational control. It introduces support for new protection types, including Credential Stuffing Defense, CSRF Protection, HTTP Protocol Constraints, and SQL/XSS Injection Detection. Administrators can now bind exception policies directly to learned recommendations, export model data and recommendations in PDF format, and control when to activate the 30-day trial license for evaluation. These improvements enable more accurate policy tuning and better alignment with production workflows.

WAF Signature Support for HTTP/3 and HTTP/2 8.0.3

FortiADC 8.0.3 now supports Web Application Firewall (WAF) Web Attack Signature scanning for HTTP/3 and HTTP/2 Virtual Servers. This update allows you to apply robust security policies to modern high-performance traffic, ensuring your services are protected against known web-based vulnerabilities.

Supported Security Enhancements

The following capabilities are now available for Virtual Servers handling modern HTTP protocols:

- **Web Attack Signature Scanning**: You can now enable the "Web Attack Signature" sub-profile within a WAF profile to inspect HTTP/3 and HTTP/2 traffic.
- **Active Threat Mitigation**: The system supports the use of **Deny** and **Alert** actions when a signature match is identified within a stream.
- **Integrated Security Logging**: All detected attacks are recorded in the standard Security Log, providing full visibility into threats targeting your modern web services.
- **Diagnostic Tools**: Real-time debugging is available via the CLI to help you monitor how the WAF engine evaluates and protects traffic.

Limitations

For this enhancement, the focus is on providing specialized WAF coverage for Web Attack Signatures. The following limitations apply:

- **Signature-Only Profiles**: The WAF profile applied to the Virtual Server must be configured exclusively with the **Web Attack Signature** sub-profile.
- **Profile Validation**: Including other WAF features such as AI, Protocol Constraints, or Captcha within the same profile will result in a configuration error.
- **Custom Responses**: The system does not support custom WAF error pages or Captcha challenges for these protocols.
- **Management & Reporting**: The FortiView "WAF Blocked IP" console and WAF-specific performance statistics are currently not supported for these protocols.

RESTful API Input Security Check 8.0.3

You can now benefit from automated, deep-level security inspections for the FortiADC RESTful API to detect and prevent sophisticated exploits such as command injection and path traversal. FortiADC 8.0.3 introduces a specialized, independent signature database that performs real-time validation of API requests before business logic is executed. This feature ensures robust protection for the management interface by inspecting multiple input locations including headers, request bodies, query arguments, and URL paths without requiring any manual configuration.

Multi-Layered Inspection and Validation

The security check employs a dual-layered JSON-based rule system to ensure comprehensive coverage:

- **Global Pre-checks**: The system applies baseline rules to every incoming HTTP request to identify known attack patterns.
- **API-Specific Rules**: Granular protection is applied when a matching rule file exists for a specific requested API path.
- **Deep Decoding**: The engine supports multi-layer URL decoding to identify dangerous patterns hidden within percent-encoded strings.
- **Path Traversal Prevention**: The check specifically monitors requests for directory traversal sequences to block unauthorized system access attempts.

Intelligent Traffic Handling

The inspection engine is optimized to maintain high system performance while ensuring constant security:

- **Large Request Processing**: The system uses efficient data handling to process very large request bodies without impacting system stability.
- **Automated Blocking**: Requests that trigger a match are immediately terminated with a 403 Forbidden response.
- **Data Privacy**: Standardized error responses ensure that sensitive match context or internal data is never exposed to the client.

Operational Consistency

The system is designed for high reliability and seamless background operation:

- **Zero Configuration**: Security checks are enabled automatically in version 8.0.3 and later.
- **Post-Login Protection**: Inspections focus on protecting the REST API after successful administrative authentication.
- **Signature Database Maintenance**: The signature database is integrated into the WAF package and receives updates centrally via FortiGuard.

WAF Adaptive Learning 2.0

WAF Adaptive Learning 2.0 enhances the original engine with expanded module coverage, exception-aware tuning, and improved operational control. It introduces support for new protection types, including Credential Stuffing Defense, CSRF Protection, HTTP Protocol Constraints, and SQL/XSS Injection Detection. Administrators can now bind exception policies directly to learned recommendations, export model data and recommendations in PDF format, and control when to activate the 30-day trial license for evaluation. These improvements enable more accurate policy tuning and better alignment with production workflows.

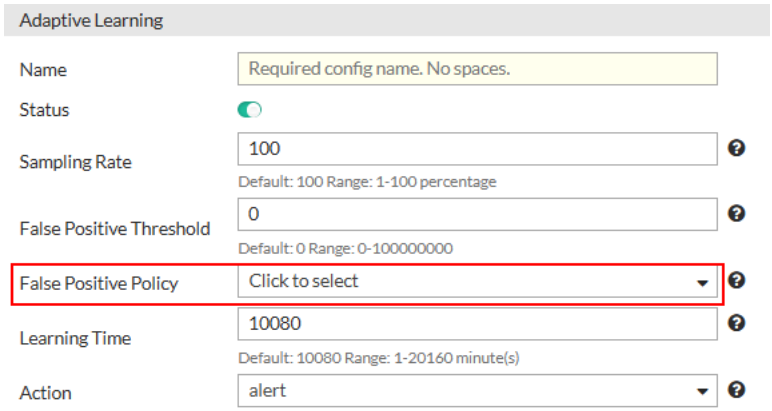
This information is also available in the FortiADC 8.0.0 Administration Guide:

- WAF Adaptive Learning

Key Features and Improvements

Enhancements	Description
False Positive Policy Integration	Adaptive Learning now includes a field to assign a false positive policy (mapped to a WAF Exception object). If the protection module supports exception logic, this policy is applied during rule deployment. Unsupported modules will ignore the setting.
Expanded Module Coverage	Adaptive Learning now generates recommendations for additional inspection types: • Credential Stuffing Defense • CSRF Protection • HTTP Protocol Constraints • SQL/XSS Injection Detection Each module applies its own logic for pattern recognition and response.
30-Day Trial License Activation Control	You can now begin the 30-day trial license for the WAF Adaptive Learning feature at your discretion. In earlier versions, the trial started automatically upon upgrade. Manual activation allows administrators to align evaluation with testing or deployment readiness.
PDF Report Export	A new reporting mechanism allows administrators to export PDF summaries of Adaptive Learning models, recommendations, and related statistics per VDOM. Reports include up to one year of history and are accessible through the GUI.

False Positive Policy Integration



A dedicated False Positive Policy field is now available in the Adaptive Learning settings pane. This policy maps directly to a WAF Exception object and is evaluated during rule acceptance. Its behavior depends on the target module's ability to process exceptions:

Supported Modules:

- JSON/XML Protection – Policy is applied as structured rule exceptions.
- SQL/XSS Injection Detection – Policy is applied to rule subtypes.
- Bot Detection – Entries are converted into Bot Allowlist configurations.

Unsupported Modules:

- WAF Signature
- Input Validation (parameter validation and hidden field validation)
- HTTP Protocol Constraints
- CSRF Protection
- Credential Stuffing Defense

If the target module does not support exception handling, the policy field is ignored. This ensures alignment between recommendation logic and enforcement capabilities.

How the Adaptive Learning False Positive Policy is mapped to the Bot Allowlist rules:

False Positive Policy Parameter	Mapped Parameters: WAF Exception to Bot Allowlist Rule		Description
	WAF Exception	Bot Allowlist	
URL	URL Pattern	URL Pattern	Only `URL Pattern` is mapped; other attributes are ignored.
Source IP	Source IP	IPv4/Netmask	If the same network address already exists, or if a broader subnet mask is present, the new entry is not added.

False Positive Policy Parameter	Mapped Parameters: WAF Exception to Bot Allowlist Rule		Description
	WAF Exception	Bot Allowlist	
HTTP Header	Name Pattern: User-Agent	User-Agent	Mapped only if the header name matches "User-Agent".
	Value Pattern: Value	User-Agent Value	
Cookie	Name Pattern: Cookie	Cookie Name	Cookie value must be disabled; otherwise, the entry is not deployed.
	Value Pattern (*should be disabled, otherwise will **not** be deployed*)		
Parameter	Name Pattern: Cookie	URL Parameter Name	Parameter value must be disabled; otherwise, the entry is not deployed.
	Value Pattern (*should be disabled, otherwise will **not** be deployed*)		
Source IPv6	Source IPv6	Not Supported	IPv6 addresses are not supported and will be ignored.
HTTP Method	HTTP Method	Not Supported	HTTP method is not supported and will be ignored.

Limitations:

- False Positive Recommendations are supported for: Web Attack Signature, Bot Detection, Parameter Validation, JSON/XML, HTTP Protocol Constraint, and SQL/XSS Injection Detection.
- If an Exception is configured in the WAF profile, matching traffic bypasses Adaptive Learning.
- If a False Positive Policy is configured in Adaptive Learning, exceptions will be automatically applied to supported function policies.
- If an Exception is directly configured in a supported function policy, Adaptive Learning will still process traffic and generate recommendations.

CLI update:

```
config security waf adaptive-learning
  edit <name>
    set false-positive-policy <datasource>
```

```
  next
end
```

Expanded Module Coverage

The adaptive learning engine now supports analysis and recommendation generation for the following modules. Each module may or may not support exception-based tuning. Refer to the module capabilities when interpreting recommendations.

Credential Stuffing Defense

Detects Basic Authentication in the Authorization header. Generates a recommendation to enable Credential Stuffing Defense when matched. This feature does not support false positive policy exceptions.

Credential Stuffing Defense examples

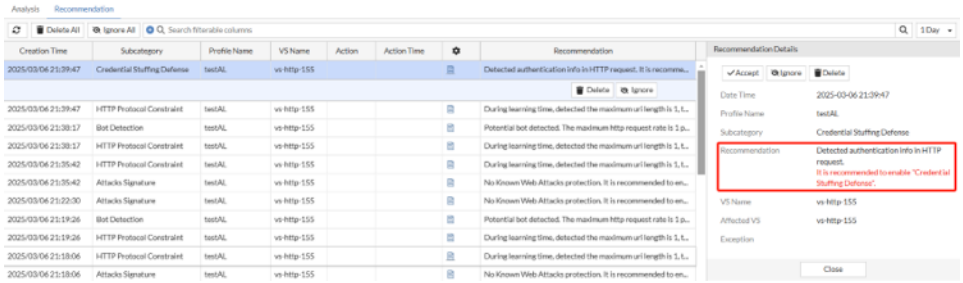
Triggering recommendations to set a new policy

Recommendations to set a Credential Stuffing Defense policy are triggered when HTTP requests include an Authorization: Basic header containing a decodable base64-encoded credential string.

Example:

```
curl -iv http://10.65.1.113/ -H "Authorization: Basic cGtsYW5nZG9uNEBtc24uY29tOnBwbDExMjY2"
```

When the recommendation is accepted, a Credential Stuffing Defense policy is created and attached to the WAF profile.



Cases where no recommendation is generated

Recommendations are **not** generated in the following cases:

- The `Authorization` header uses a method that is **not** Basic. Methods such as Bearer, Digest, and NTLM are not supported.
- A recommendation has already been generated for the same traffic pattern. Additional matching requests do not trigger new recommendations.
- A Credential Stuffing Defense policy was previously attached and then removed from the WAF profile. In this case, the system assumes the removal was intentional and does not generate another recommendation. To allow recommendations again, the Adaptive Learning policy must be removed and re-bound to the WAF profile, or the system must be restarted.

Adaptive Learning does not generate false positive recommendations for Credential Stuffing Defense.

CSRF Protection

Identifies potential CSRF risks using the Referer header. Triggers recommendations when the request's host does not match the Referer domain. False positive policies are not applicable.

CSRF Protection examples

Triggering recommendations to set a new policy

Recommendations to configure a CSRF protection policy are triggered when an HTTP request includes a Referer header, and the host portion of the Referer URL differs from the host of the request URL.

Example:

```
curl -iv http://10.65.1.155/index.html -H "Referer: http://demosite.com"
```

In this case:

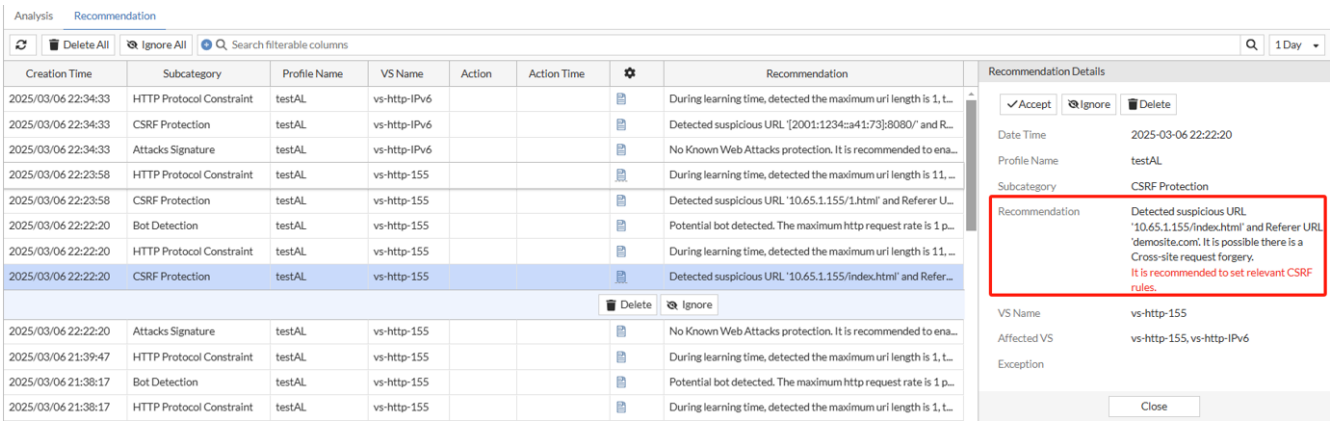
- A `csrf-page-list` entry is created for `10.65.1.155/index.html` (from the request URL).
- A `csrf-url-list` entry is created for `demosite.com` (from the Referer header).

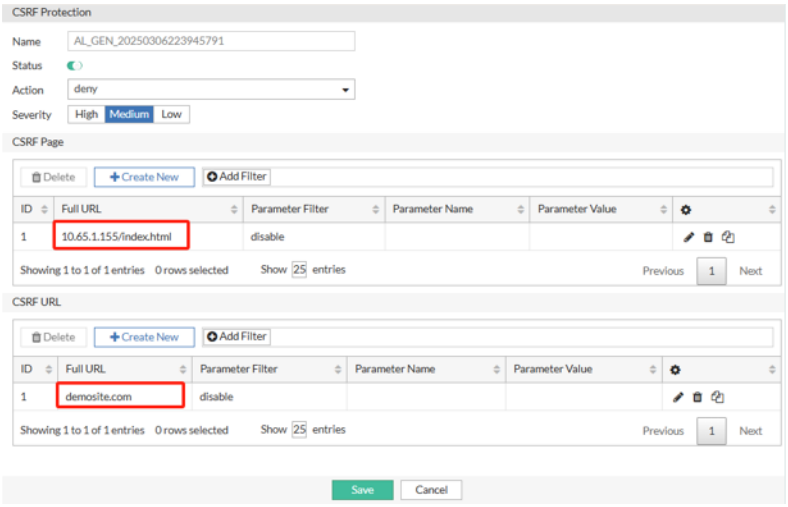
To trigger a recommendation:

- The request must include a `Referer:` header with an HTTP or HTTPS URL.
- The `Referer:` header name is case-insensitive, but URL values are case-sensitive.

When the recommendation is accepted, a CSRF protection policy is created and attached to the WAF profile automatically.

If a CSRF protection recommendation is triggered but not accepted, similar future requests will continue to trigger recommendations. Unlike Credential Stuffing Defense, which generates a recommendation only once, CSRF protection allows for repeated recommendations because the `Referer` URLs may vary across different requests.





Triggering recommendations to adjust a policy

If a CSRF protection policy is present but disabled in the WAF profile, a recommendation to enable it will be generated when a request is received with a `Referer` header whose domain differs from the request URL.

If an active CSRF protection policy exists, and a request includes a new request URL or a new domain in the `Referer` header, the system will generate a recommendation to update the policy by appending the entries to the `csrf-page-list` and/or `csrf-url-list`.

When CSRF policy recommendations are not triggered or do not modify policy settings

If both the request URL and the `Referer` domain already match existing entries in the `csrf-page-list` or `csrf-url-list`, no recommendation is generated—or only a recommendation to append unmatched URLs may appear.

Accepting a CSRF recommendation for an existing policy does not alter the configured **Severity** or **Action** settings.

HTTP Protocol Constraints

Learns request characteristics like header lengths and field sizes. Recommends constraint settings based on observed traffic patterns. Updates are only triggered if values change; no support for false positive policies.

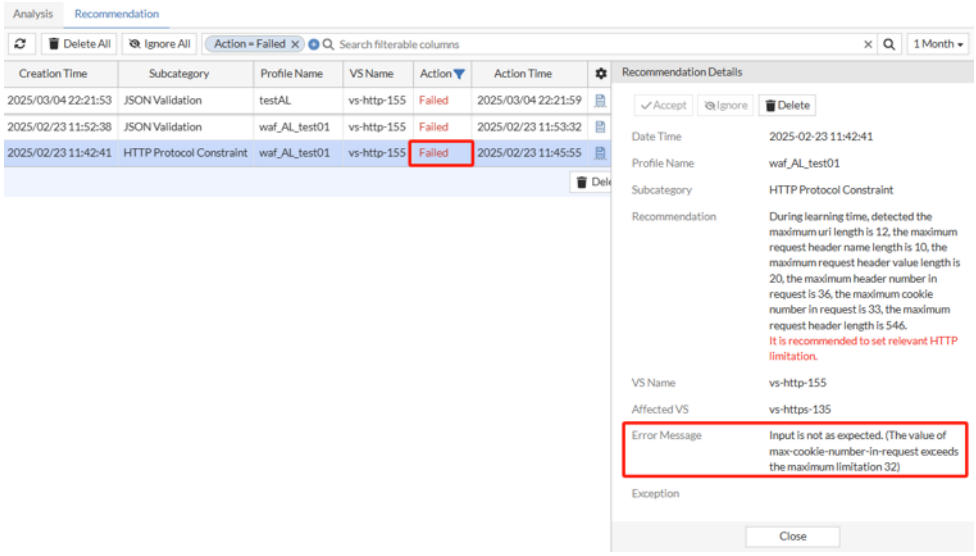
HTTP Protocol Constraints examples

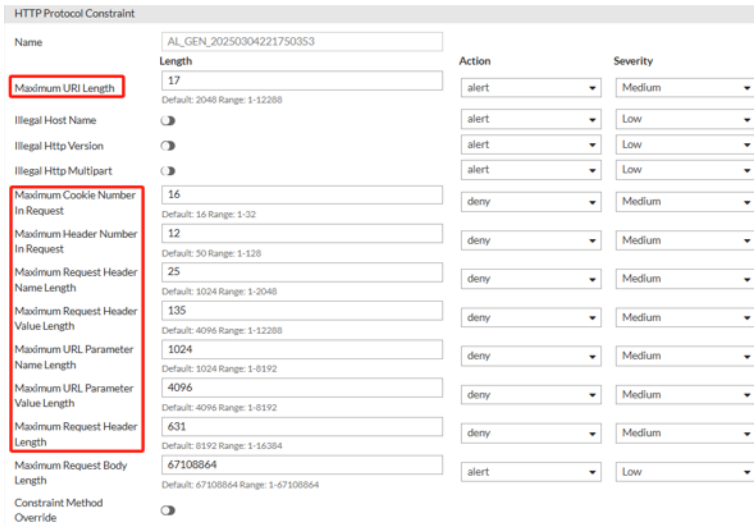
Triggering recommendations to set a new policy

Recommendations for HTTP Protocol Constraint policies are generated when HTTP requests contain recognizable protocol elements. Upon acceptance, a policy is created and attached to the WAF profile.

- Maximum length values for detected elements are set to match the actual observed value in the request. This differs from JSON/XML protection, where the configured length must exceed the actual value by one.
- The `max-request-body-length` setting is excluded from this auto-configuration.
- Elements not detected during learning are assigned default length values.
- For supported elements, **Action** and **Severity** are derived from the associated Adaptive Learning policy.
- Unsupported elements—such as Illegal Host Name, Illegal HTTP Version, Illegal HTTP Multipart, Constraint Method Override, Request Method Rule, and Response Code Rule—retain their default configuration, including **Action** and **Severity**.

If any detected element exceeds the maximum value allowed by the system, accepting the recommendation will fail, and an error message will appear in the recommendation details.





Triggering recommendations to adjust a policy

If an HTTP Protocol Constraint policy is already applied, Adaptive Learning continues to monitor request element lengths and may generate recommendations to adjust the configured maximum values:

- If a request contains an element shorter than the current configured maximum but longer than previously recorded values, a recommendation is generated to reduce the configured maximum length.
- If a request contains an element longer than the current configured maximum and longer than any previously observed value, a recommendation is generated to increase the configured maximum length.

When the active constraint policy is a predefined profile (e.g., High-Level-Security), accepting a recommendation will clone the profile and create a new policy with the updated length values. For example, if the original maximum cookie name length was 16 and a request includes a 32-character name, a new profile (e.g., AL_GEN_...) is generated with the updated length set to 32.

False Positive Recommendation

If traffic from multiple distinct sources (as determined by the false positive threshold) repeatedly exceeds a configured maximum length in the HTTP Protocol Constraint policy, a false positive recommendation is triggered. This suggests adjusting the element length limit to a more appropriate value based on observed traffic patterns.

If the current policy is based on a predefined profile, accepting the recommendation will clone the profile and create a new policy with the updated maximum length setting.

SQL/XSS Injection Detection

Detects injection attempts in request elements (URI, headers, etc.). Triggers recommendations to enable protection and subtypes. Supports false positive policies for more granular tuning.

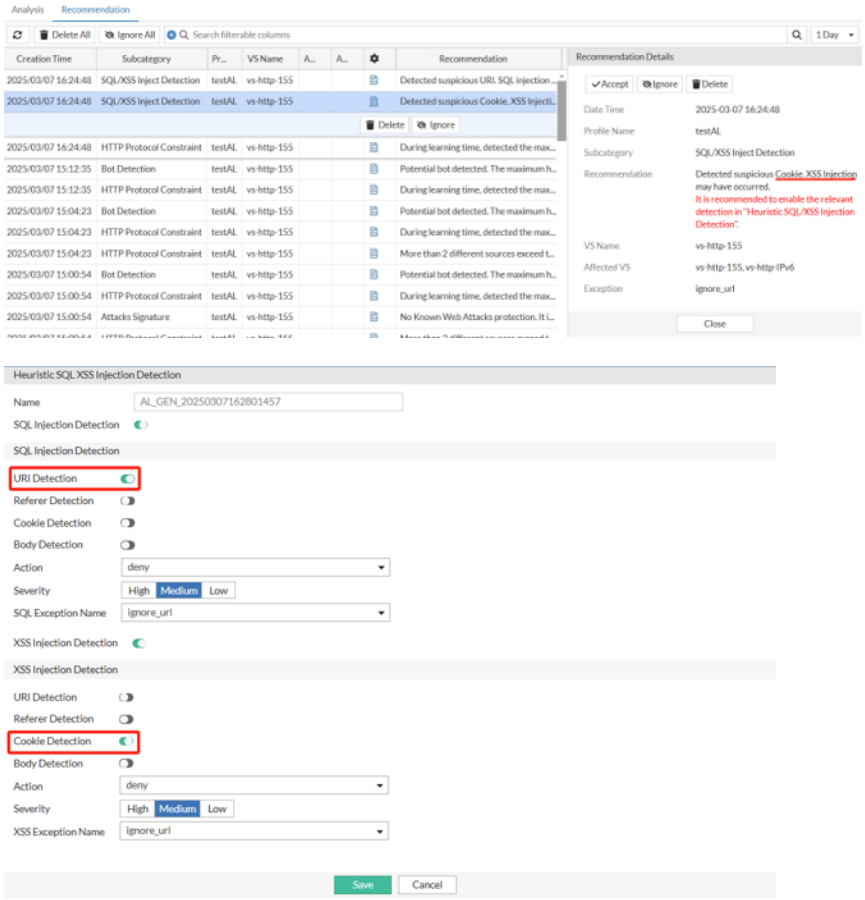
SQL/XSS Injection Detection examples

Triggering recommendations to set a new policy

Recommendations to configure a SQL/XSS Injection Detection policy are generated when potential SQL or XSS injection patterns are identified in incoming HTTP requests. Upon acceptance, a corresponding policy is created and attached to the WAF profile.

- Detection types (SQL or XSS) are enabled based on the observed payload.
- Sub-type detection (URI, Referer, or Cookie) is selectively enabled depending on the injection vector.
- Body-based detection is not evaluated by Adaptive Learning and does not trigger recommendations.
- Action and Severity values are applied as defined in the AL policy.

For example, a request triggering SQL in the URI and XSS in the Cookie header will enable both detection types accordingly.



Cases where no recommendation is generated

Recommendations for SQL/XSS Injection Detection are generated only once. Subsequent requests that match the same injection patterns do not produce additional recommendations. If a SQL/XSS policy created by Adaptive Learning is later disabled or removed from the WAF profile, no further recommendations are generated, since the system assumes the change was intentional.

Triggering recommendations to adjust a policy

Adaptive Learning monitors configured SQL/XSS policies for new injection vectors. When a request contains an injection type that is currently disabled–such as XSS in a Cookie header–a recommendation is generated to enable that sub-detection.

If the active policy is a predefined profile, accepting the recommendation clones the profile and activates the newly detected sub-type(s).

Example:

Given the predefined profile Medium-Level-Security with only SQL URI detection enabled, the request:

```
curl -iv "http://10.65.1.155/index.html?x=1--" \
     -H "Cookie: src=<script>d</script>"
```

triggers a recommendation to enable XSS Cookie detection in a new cloned policy.

False Positive Recommendation

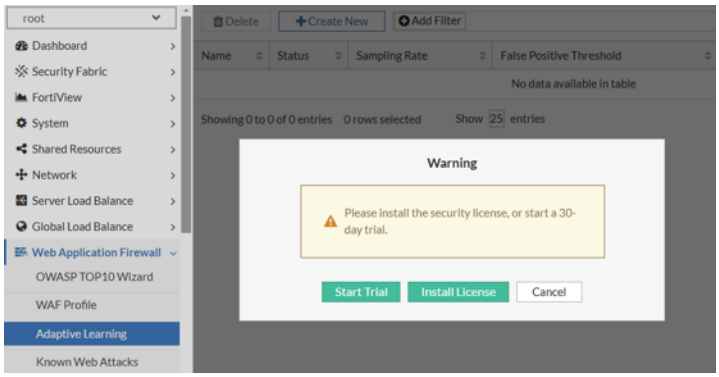
When traffic from multiple distinct clients (exceeding the configured false-positive threshold) violates a specific SQL or XSS detection sub-type, Adaptive Learning generates a recommendation to disable that sub-type.

If the active policy is a predefined profile, accepting the recommendation clones the profile and disables the identified sub-type(s) in the new policy.

30-Day Trial License Activation Control

Adaptive Learning no longer initiates a trial period automatically upon upgrading. Instead, administrators can now trigger the 30-day trial license manually from the GUI, allowing greater control over when evaluation begins.

Activation is performed in the root VDOM by navigating to **Web Application Firewall > Adaptive Learning**, where a prompt will appear to either start the trial or upload a valid license.

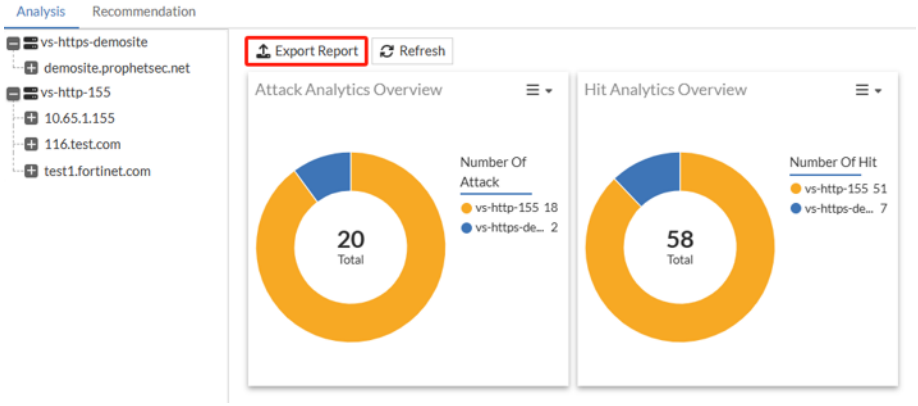


Once activated, the license enables full functionality:

- Traffic-based model training
- Recommendation analysis and acceptance
- Exporting and reporting
- Exception-aware rule deployment

After activation, the 30-day trial runs continuously and cannot be paused or restarted. Upon expiration, the Adaptive Learning feature is automatically disabled unless a valid license is installed. A full, permanent license is required to re-enable functionality beyond the trial period.

PDF Report Export



A GUI-based export function is now available to generate PDF reports of Adaptive Learning data. Reports include:

- Model metadata
- Rule recommendations
- Learning activity logs and metrics

Exports are generated per VDOM and can include up to 12 months of data. The feature is designed for auditing, reporting, and cross-team sharing. Note that exporting does not impact enforcement logic or system configuration.

Security Fabric

The FortiADC 8.0 release includes new features and enhancements in **Security Fabric**:

External ICAP Server Support 8.0.3 on page 98

FortiADC 8.0.3 introduces support for external Internet Content Adaptation Protocol (ICAP) servers through a new Fabric Connector to provide an additional layer of file inspection. This feature allows FortiADC to act as an ICAP client, sending files to a third-party ICAP server for deep scanning after the local AntiVirus (AV) engine has performed its initial check. To use this feature, you enable ICAP scanning within an AntiVirus profile and associate it with a Virtual Server. If the ICAP server detects a threat, FortiADC automatically blocks the request and logs the event.

FortiSandbox Cloud Connectivity Enhancements 8.0.3 on page 100

FortiADC 8.0.3 simplifies the integration with FortiSandbox by introducing an automated connection process and expanded regional support. These updates eliminate the need for manual account configuration, allowing the system to establish a secure link and retrieve available service regions automatically.

Cisco ACI External Connector 8.0.1 on page 102

FortiADC now supports direct integration with **Cisco ACI 5.2** through a new **Cisco ACI SDN** connector in the **Security Fabric > External Connectors** framework.

This connector establishes a northbound API connection to the Cisco Application Policy Infrastructure Controller (APIC), enabling FortiADC to automatically discover and synchronize ACI tenants, application profiles, and endpoint groups (EPGs) with its own load-balancing configuration.

By linking the application-centric visibility of Cisco ACI with FortiADC's traffic management engine, this feature delivers adaptive, SDN-driven load balancing that evolves automatically with your data-center topology.

FortiGate Security Fabric-Based Admin SSO 8.0.1 on page 106

FortiADC now supports administrator Single Sign-On (SSO) through **FortiGate Security Fabric integration**. When connected to the Security Fabric where FortiGate acts as the root, FortiADC can use the FortiGate as its **SAML Identity Provider (IdP)** for administrator authentication.

External ICAP Server Support 8.0.3

FortiADC 8.0.3 introduces support for external Internet Content Adaptation Protocol (ICAP) servers through a new Fabric Connector to provide an additional layer of file inspection. This feature allows FortiADC to act as an ICAP client, sending files to a third-party ICAP server for deep scanning after the local AntiVirus (AV) engine has performed its initial check. To use this feature, you enable ICAP scanning within an AntiVirus profile and associate it with a Virtual Server. If the ICAP server detects a threat, FortiADC automatically blocks the request and logs the event.

Multi-Layered Threat Detection

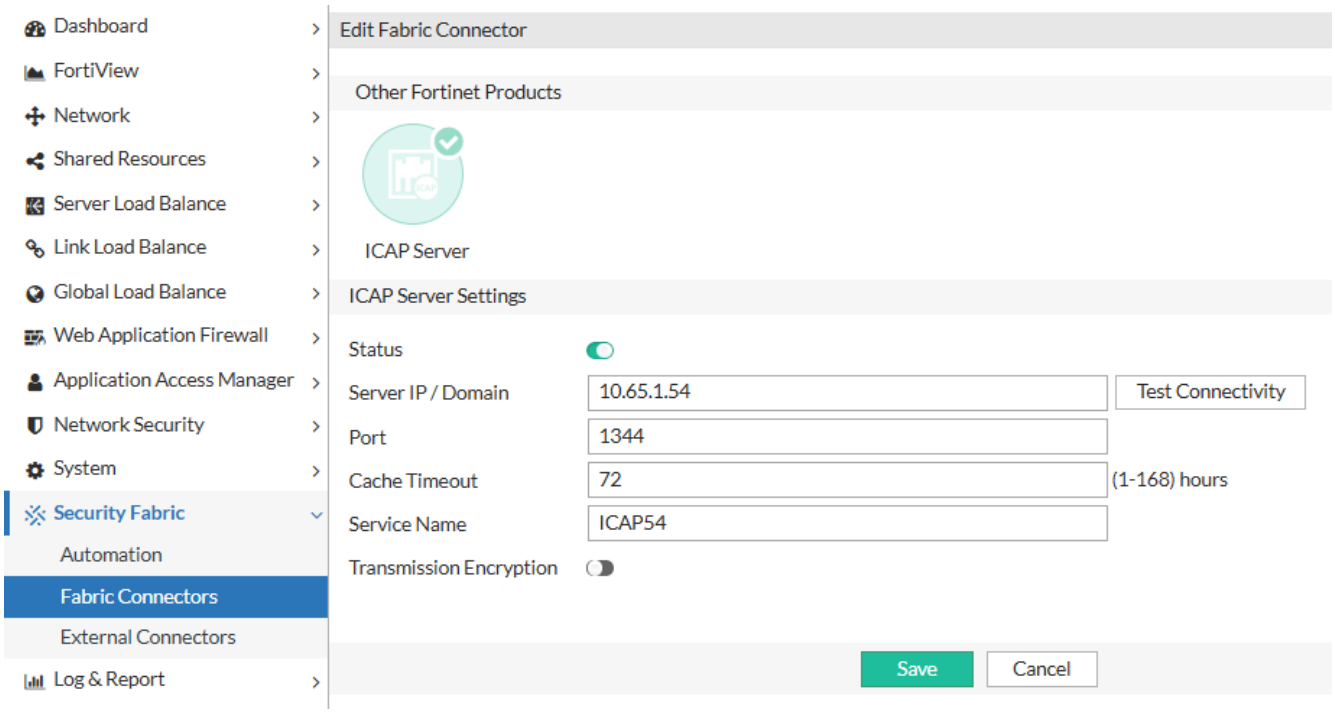
The integration of an external ICAP server enhances your defense-in-depth strategy by coordinating local and remote scans:

- **Offloaded Inspection**: If the local AV engine identifies a file as clean, FortiADC can transmit the file to a configured ICAP server for a second, independent scan.
- **Flexible Scanning Workflow**: You can configure the system to perform local scans, ICAP scans, or both to suit specific security requirements.
- **Unified Security Logging**: All scan results from the external ICAP server are integrated directly into the FortiADC AV security logs, ensuring centralized visibility for threat monitoring.

Fabric Connector Integration

The connection to an external ICAP server is managed through a new dedicated Fabric Connector:

- **Streamlined Configuration**: You can define the ICAP server details, including the IP address, port, and service path, within the **Security Fabric > Fabric Connector** menu.
- **Service Health Monitoring**: The Fabric Connector provides real-time status updates to ensure the ICAP service is available before offloading traffic.


Dashboard
FortiView
Network
Shared Resources
Server Load Balance
Link Load Balance
Global Load Balance
Web Application Firewall
Application Access Manager
Network Security
System
Security Fabric
Automation
Fabric Connectors
External Connectors
Log & Report
Edit Fabric Connector
Other Fortinet Products
ICAP Server
ICAP Server Settings
Status
Server IP / Domain
10.65.1.54
Test Connectivity
Port
1344
Cache Timeout
72
(1-168) hours
Service Name
ICAP54
Transmission Encryption
Save
Cancel

FortiSandbox Cloud Connectivity Enhancements 8.0.3

FortiADC 8.0.3 simplifies the integration with FortiSandbox by introducing an automated connection process and expanded regional support. These updates eliminate the need for manual account configuration, allowing the system to establish a secure link and retrieve available service regions automatically.

Automated Cloud Integration

The connection to FortiSandbox is now more streamlined and user-friendly:

- **Zero-Configuration Connectivity**: You no longer need to manually input FortiCloud account credentials or create an account to establish a link.
- **Automatic Service Discovery**: FortiADC automatically connects to the cloud service to retrieve a real-time list of supported regions.
- **Continuous Synchronization:** The system periodically updates its local database with results from the cloud every 8 hours, ensuring identified threats are blocked locally during future scans.

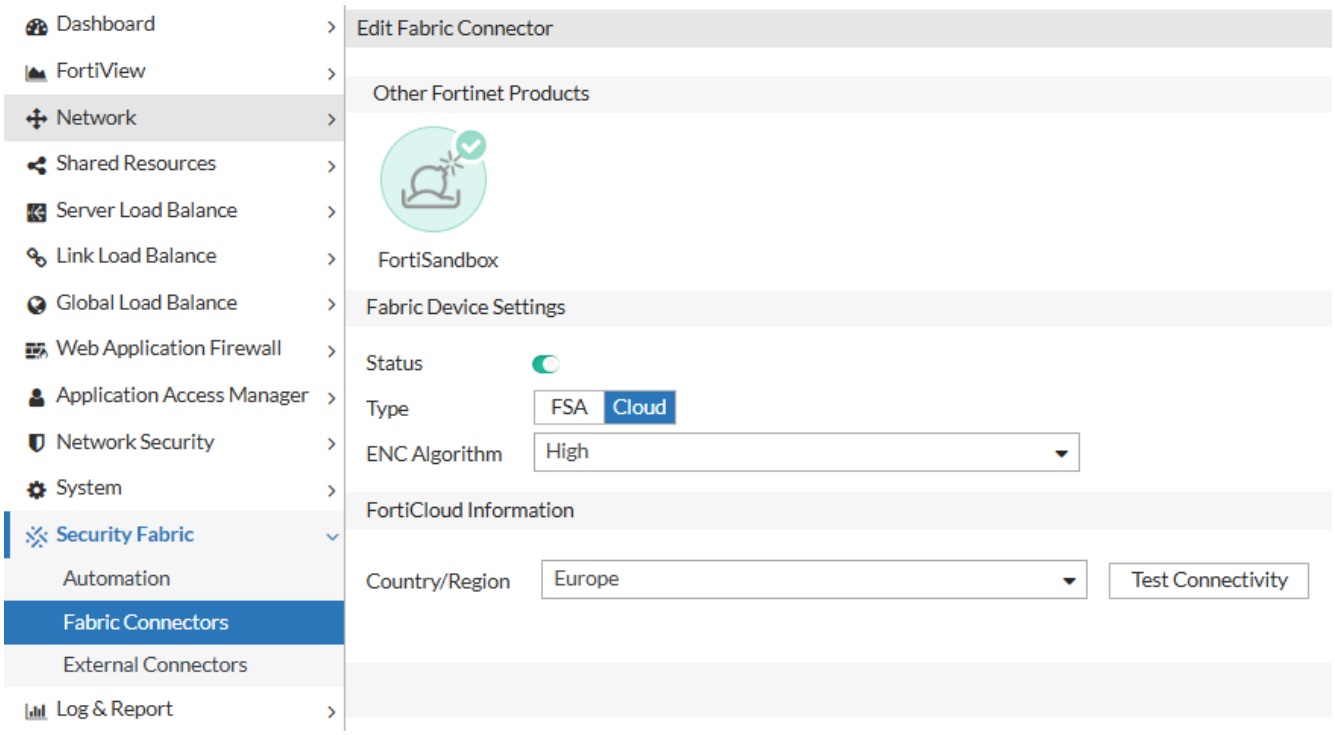
Global Region Selection

New regional settings allow administrators to control where their data is processed to meet local compliance and performance needs:

- **Data Residency Control**: You can select a specific region, such as Europe, Global, US, or Japan, for file analysis.
- **Compliance Alignment**: Providing multiple regional options helps organizations meet different countries' compliance requirements regarding data storage locations.
- **Smart Defaults**: If no specific region is selected, the system automatically uses the **Global** default setting to ensure uninterrupted protection.
- **Connection Validation**: A **Test Connectivity** button in the interface allows you to immediately verify the status of the connection to the selected region.

Configuration: FortiSandbox Fabric Connector

These settings are configured under **Security Fabric > Fabric Connectors**.



Parameter	Description
Status	Enables or disables the FortiSandbox connector. When enabled, the system maintains a control link to the cloud for contract updates and file analysis.
Type	Specifies the sandbox service type. This must be set to Cloud to utilize the new automated connectivity features.
ENC Algorithm	Sets the SSL encryption strength (High, Default, or Low) for the mandatory secure connection between FortiADC and FortiSandbox. Note that the Disable option is not supported, as an encrypted link is required for the service to function.
Country/Region	Defines the geographic data center used for file processing. Available options include Global (default), US, Europe, and Japan to help meet local data residency requirements.

Limitations

- **Initial Processing**: Files uploaded for the first time that are not identified as a virus locally are forwarded to the real server while cloud analysis continues in the background.
- **Reporting Features**: Real-time detection statistics from the cloud are not supported in version 8.0.3 and are planned for a future release.

Cisco ACI External Connector 8.0.1

FortiADC now supports direct integration with **Cisco ACI 5.2** through a new **Cisco ACI SDN** connector in the **Security Fabric > External Connectors** framework.

This connector establishes a northbound API connection to the Cisco Application Policy Infrastructure Controller (APIC), enabling FortiADC to automatically discover and synchronize ACI tenants, application profiles, and endpoint groups (EPGs) with its own load-balancing configuration.

By linking the application-centric visibility of Cisco ACI with FortiADC's traffic management engine, this feature delivers adaptive, SDN-driven load balancing that evolves automatically with your data-center topology.

This information is also available in the FortiADC 8.0.1 Administration Guide and CLI Reference:

- Cisco ACI Connector
- config system sdn-connector

Integration Model

The Cisco ACI connector allows FortiADC to act as an external service consumer within the ACI fabric. It retrieves application hierarchy information from the APIC and maps it to ADC objects as follows:

- **Tenant** – Logical grouping of applications within the ACI fabric.
- **Application Profile** – Defines application contexts that organize related EPGs.
- **Endpoint Group (EPG)** – Represented on FortiADC as a dynamic real server pool, where each endpoint node in the EPG becomes a pool member.

FortiADC maintains a read-only connection to the APIC using ACI's northbound REST API. When endpoints are added, removed, or reassigned within ACI, FortiADC automatically updates the corresponding dynamic pool so that real server membership always reflects the live fabric topology.

The connector supports up to **four APIC hosts** in an active/standby sequence. FortiADC maintains a single active session with the first reachable APIC and continuously monitors server availability. If the active controller becomes unreachable, FortiADC automatically fails over to the next server in the configured order, ensuring uninterrupted synchronization with the ACI fabric.

Configuration and Operation

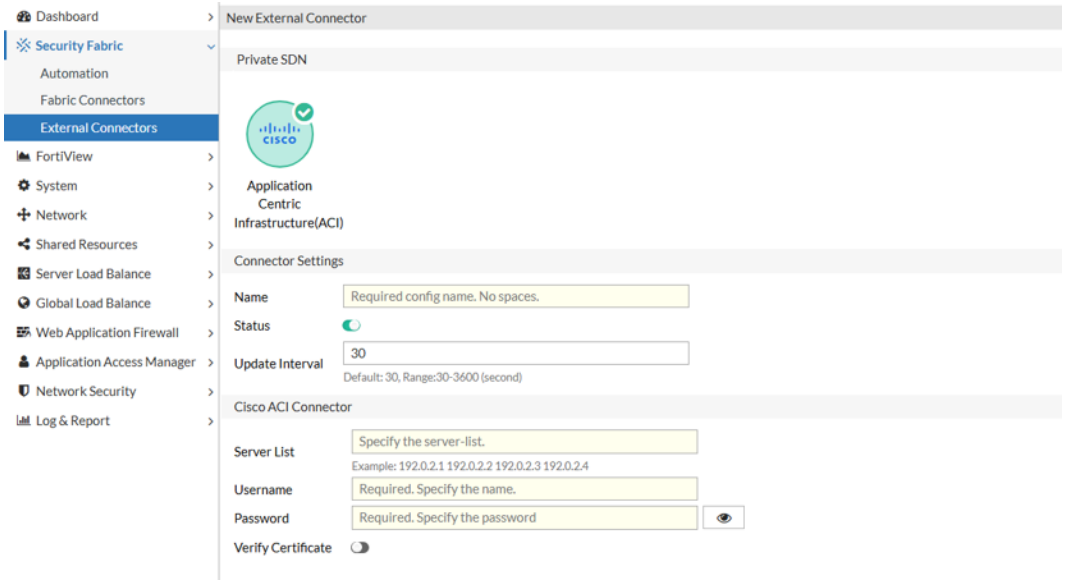
Configuring the Cisco ACI connector involves two stages:

1. Creating the Cisco ACI Connector on page 103 - defines how FortiADC communicates with the Cisco APIC cluster.
2. Creating the Dynamic Real Server Pool through the Cisco APIC on page 104 - maps ACI endpoint groups (EPGs) to FortiADC load-balancing objects.

Creating the Cisco ACI Connector

The Cisco ACI Connector defines how FortiADC communicates with the Cisco APIC and synchronizes topology information.

It authenticates with the APIC cluster, retrieves application hierarchy data, and maintains a continuous polling session to track endpoint changes.



To configure the Cisco ACI connector:

1. Go to **Security Fabric > External Connectors** and click **Create New**.
2. Under Private SDN, select **Application Centric Infrastructure (ACI)**.
3. Configure the following settings:

Setting	Description
Name	A unique identifier for the connector configuration. This name appears in the SDN Connector list when creating dynamic pools. The name must be alphanumeric (`A–Z`, `a–z`, `0–9`, `_`, `-`) with no spaces.
Status	Enables or disables the connector. When enabled, FortiADC immediately attempts to establish communication with the APIC servers in the configured list. Disabling the connector suspends synchronization and retains the last cached topology data until re-enabled.
Update Interval	Specifies the polling frequency (in seconds) for topology updates from the APIC cluster. Each interval triggers a REST API query to retrieve changes in tenants, application profiles, and EPGs. Default: 60, Valid range: 10-3600 seconds. Shorter intervals increase synchronization responsiveness but generate higher API request volume.
Server List	Lists up to four Cisco APIC controller IP addresses (IPv4 or IPv6). FortiADC connects to the first reachable host in sequence and monitors the connection.

Setting	Description
	If the active controller becomes unavailable, FortiADC automatically switches to the next server in the list. At least one reachable APIC host is required for successful synchronization.
Username	Specifies the APIC account used for REST API authentication. The account should have **read-only** privileges to the ACI tenant and fabric objects. Higher privileges are not required or recommended.
Password	Password for the APIC user account. The password is stored in encrypted form and used only for API authentication.
Verify Certificate	When enabled, FortiADC verifies the SSL certificate presented by the APIC server during HTTPS negotiation. This should be enabled when using CA-signed or trusted certificates. Disable this option if the APIC uses self-signed certificates or an internal CA not trusted by FortiADC. Disabling verification bypasses certificate validation but does not affect encryption. This is disabled by default.

4. Click **Save**.

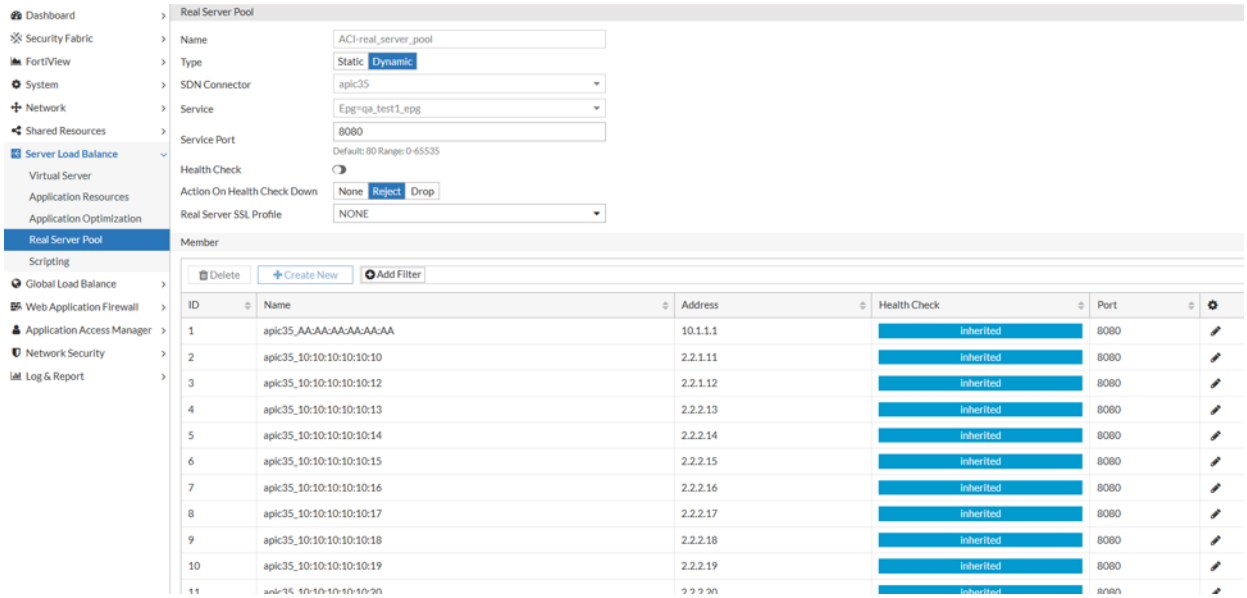
After the connector is created, FortiADC establishes an API session with the APIC cluster and begins synchronizing tenants, application profiles, and EPGs.

Connector health and synchronization status are displayed in the GUI and can also be verified through system diagnostics.

Creating the Dynamic Real Server Pool through the Cisco APIC

Dynamic pools use ACI data to define and maintain pool members automatically.

Each endpoint group (EPG) discovered from the APIC is represented as a dynamic pool on FortiADC, with its endpoint nodes reflected as pool members.



To create the dynamic pool:

1. Go to **Server Load Balance > Real Server Pool** and click **Create New**.
2. Set **Type** to **Dynamic**.
3. In **SDN Connector**, select the Cisco ACI connector you created.
4. From the **Service** drop-down list, select the ACI endpoint group (EPG) discovered by the connector. The list is automatically populated with EPGs retrieved from the APIC, often displayed with identifiers such as `Epg=<name>`. Selecting an EPG binds the pool to that group, and FortiADC immediately queries the APIC to populate the pool with its current endpoint members.
5. Specify the **Service Port** in which FortiADC will use when creating pool members for this EPG. If the service port is modified, allow several seconds for the change to propagate and appear in the GUI.
6. Configure other pool settings (method/monitoring, if applicable to your policy). Load-balancing method and health checks (if configured) apply uniformly to all auto-discovered members. This lets you keep operational policy while the connector manages membership.
7. Click **Save**.

FortiADC immediately queries the selected EPG and populates the pool with endpoint IP addresses learned from ACI.

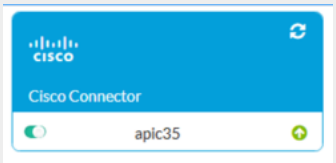
These members are auto-generated and **read-only**, ensuring configuration consistency with the ACI fabric.

When endpoints are added, removed, or migrated in ACI, FortiADC updates the corresponding pool membership in real time.

Monitoring and Diagnostics

FortiADC automatically records connector activity and synchronization events.

Administrators can monitor the connector's operational state and troubleshoot communication issues using the following tools:

Method	Description
GUI Status View	Displays connector health, last synchronization time, and APIC connectivity under **Security Fabric > External Connectors**. 
CLI Diagnostics	Run `diagnose system sdn status` to view real-time connection status and synchronization state. To view detailed event and error messages, enable `diagnose debug` and use `diagnose debug module acid`.

FortiGate Security Fabric-Based Admin SSO 8.0.1

FortiADC now supports administrator Single Sign-On (SSO) through **FortiGate Security Fabric integration**. When connected to the Security Fabric where FortiGate acts as the root, FortiADC can use the FortiGate as its **SAML Identity Provider (IdP)** for administrator authentication.

This information is also available in the FortiADC 8.0.1 Administration Guide and CLI Reference:

- FortiGate Security Fabric Connector
- Configuring SSO admin accounts
- config system csf
- config system sso-admin

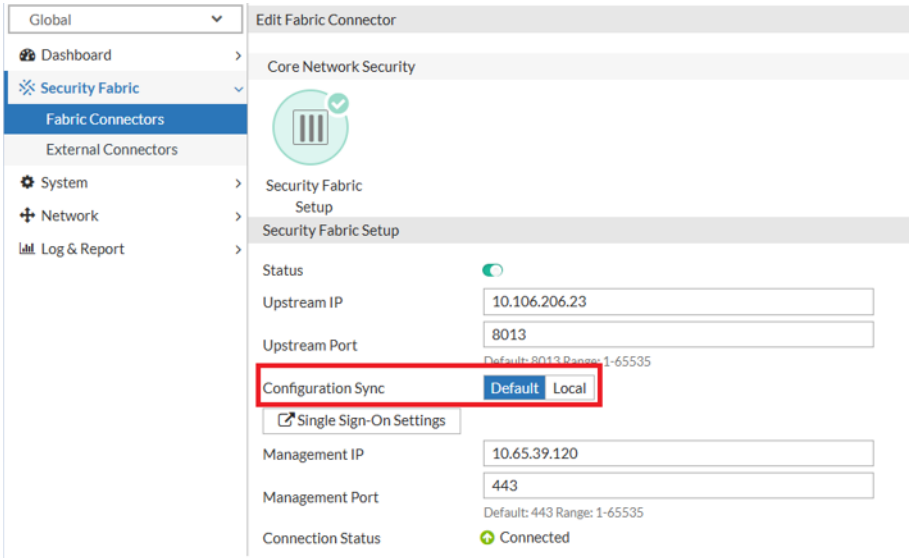
This feature consists of two integrated components:

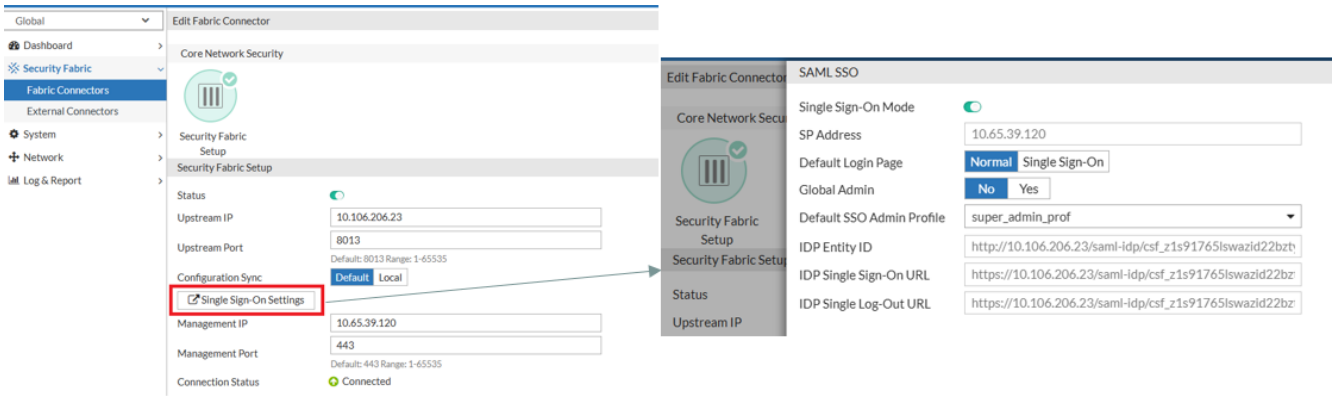
- The **FortiGate Security Fabric Setup connector**, which establishes the Fabric relationship and synchronizes IdP parameters from the root FortiGate.
- The **SSO Admin account mechanism**, which enables FortiADC to recognize and authenticate administrators through FortiGate SSO, creating corresponding accounts automatically when needed.

In the connector, the new **Configuration Sync** option controls how IdP settings are obtained:

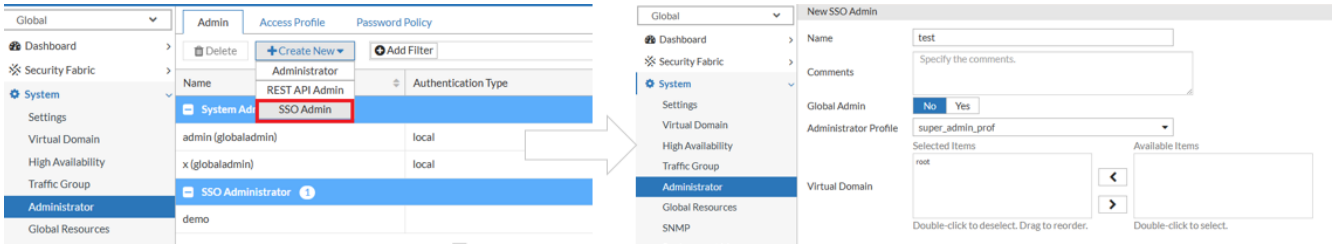
- **Default** – Synchronize IdP settings from FortiGate.
- **Local** – Use locally configured settings.

When synchronization is enabled, FortiADC automatically retrieves the IdP metadata–specifically the **IdP Entity ID**, **IdP Single Sign-On URL**, and **IdP Single Log-Out URL**–from the Fabric root FortiGate and populates these values in the **Single Sign-On Settings** which is the SAML service provider configuration.





SSO administrator accounts can be created manually or automatically. If not pre-created, FortiADC automatically generates an SSO admin account upon the user's first login through FortiGate SSO. The automatically created account uses the **Default SSO Admin Profile** and is placed under the **root VDOM** if VDOMs are enabled. Manually created SSO admin accounts, on the other hand, can specify a different access profile or VDOM as needed.



System

The FortiADC 8.0 release includes new features and enhancements in the following **System** modules:

WAF Signature Staging 8.0.1 on page 109

FortiADC introduces support for **WAF Signature Staging**, providing a controlled process to evaluate newly released or modified FortiGuard attack signatures before they are enforced. With this capability, newly added or updated signatures are first placed in a **Signature Staging** list. Administrators can monitor these signatures as they trigger on live traffic and review their Matched status before deciding whether to apply or disable them—reducing false positives and smoothing production rollouts. This capability is supported with **WAF Signature Database version 1.00063 and later**.

Disable Default Admin Account via CLI 8.0.1 on page 112

Administrators now have the option to disable the built-in **admin** account using the new CLI command `set default-admin` under `config system global`. This enhancement improves security and compliance by allowing organizations to prevent login with the default account once alternate administrator accounts have been created. When disabled, the admin account cannot log in, and any active sessions are immediately terminated.

Socket Selection Hash Control via CLI 8.0.1 on page 113

FortiADC introduces a new CLI option, `sip-to-same-sock`, under `config system global` to control how sessions are hashed across sockets. By default, sessions with the same source IP, destination IP, and destination port (`sip+dip+dport`) are consistently directed to the same CPU, `httproxy` process, and listening socket. This ensures that related sessions remain on the same processing path.

Feature Visibility on page 114

You can now selectively control the visibility of configurable features in the GUI via **System > Feature Visibility** or through the CLI. This allows administrators to streamline the interface by hiding features that are unused, inactive, or not relevant to the current deployment.

Enhancement to WAF Signature Telemetry Reporting to FortiGuard on page 119

FortiADC 8.0.0 enhances its threat telemetry capabilities by adding support for uploading **Web Attack Signature** statistics to FortiGuard. This builds on the existing telemetry framework introduced in 7.6.1, which previously supported only IPS and Antivirus (AV) threat data.

WAF Signature Staging 8.0.1

FortiADC introduces support for **WAF Signature Staging**, providing a controlled process to evaluate newly released or modified FortiGuard attack signatures before they are enforced. With this capability, newly added or updated signatures are first placed in a **Signature Staging** list. Administrators can monitor these signatures as they trigger on live traffic and review their Matched status before deciding whether to apply or disable them—reducing false positives and smoothing production rollouts. This capability is supported with **WAF Signature Database version 1.00063 and later**.

This information is also available in the FortiADC 8.0.1 Administration Guide and CLI Reference:

- Using WAF Signature Staging
- config waf staging-signature-list

Key benefits include:

- **Controlled activation**: New and modified signatures are staged for observation instead of being automatically applied.
- **Granular tuning**: Each signature can be set to **Applied**, **Disabled**, or **Unapplied**, with **Undo** support for rollback.
- **Operational efficiency**: Bulk operations are supported to approve or disable multiple staged signatures at once.
- **Transparent monitoring**: Unapplied signatures that match traffic are marked as **Matched** for visibility and review.

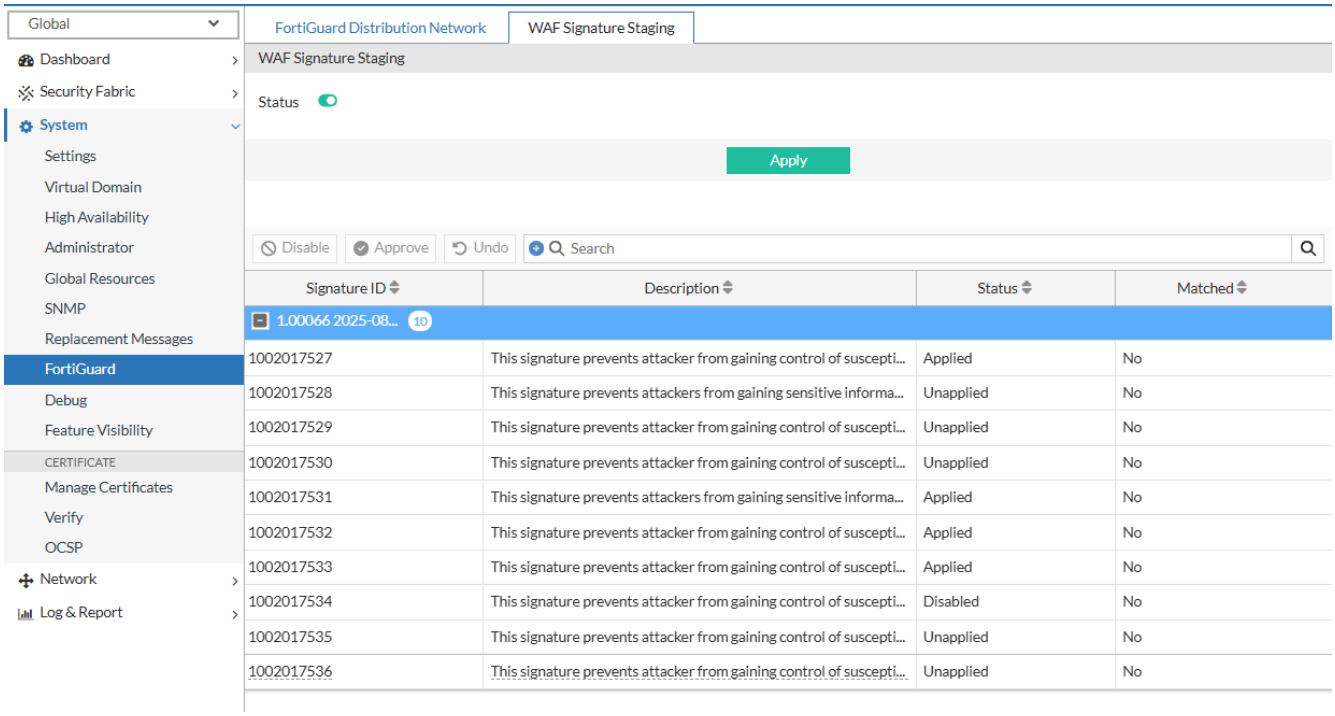
Unapplied signatures that are not explicitly disabled are automatically applied in the next version of the **WAF signature database**.

Staging model and workflow

The **Signature Staging** feature functions as a validation layer between FortiGuard signature updates and the active WAF inspection engine. When a new WAF signature database is downloaded, FortiADC identifies all newly added or modified entries and places them in a temporary staging list for review. This ensures that administrators can observe and fine-tune signature behavior before it affects production traffic.

Each entry in the staging list includes essential metadata—such as signature ID, name, revision, and description—along with its operational state and whether it has triggered on observed traffic.

Lifecycle and behavior



1. Initialization

- Upon receiving a signature update, FortiADC compares it with the existing local database.
- Any new or modified signatures are automatically added to the staging list in the **Unapplied** state.
- Signatures that were explicitly **Disabled** in prior versions also appear here for continued visibility and management.

2. Observation phase

- Unapplied signatures are evaluated passively against live traffic.
- If a match occurs, FortiADC records an Alert log entry and updates the signature's **Matched** indicator to **Yes**.
- No blocking or enforcement occurs at this stage. The **Matched** field helps administrators identify which staged signatures are relevant to their environment.

3. Administrative action

- After observing behavior, administrators can:
  - **Apply** the signature to enable enforcement.
  - **Disable** the signature if it generates false positives or is not applicable.
  - **Undo** recent changes to revert to a previous state.
- Actions can be performed individually or in bulk from the Signature Staging page.

4. **Post-activation and persistence**

   - **Applied** signatures become part of the active WAF signature set and remain visible in the staging list. If FortiADC later receives traffic matching an applied signature, the **Matched** status no longer updates and preserves its last recorded result. To view detections related to an applied signature, the WAF Security Log.
   - **Disabled** entries retain their state across future database updates.
   - **Unapplied** signatures that are not manually disabled are automatically applied in the next version of the WAF signature database. This behavior assumes that, in the absence of administrator-specific intervention, the newly introduced signatures are considered safe for deployment in the customer environment.

After each FortiGuard update, administrators are encouraged to review the **Signature Staging** list to assess newly introduced or modified signatures. The **Matched** field provides a quick indication of which entries have already detected relevant traffic, helping prioritize which signatures to apply first. Leaving critical signatures in the **Unapplied** state for extended periods may reduce effective protection, while disabling signatures should be reserved for confirmed false positives or conflicts. Bulk approval and disable operations further streamline the management of large signature updates.

Disable Default Admin Account via CLI 8.0.1

Administrators now have the option to disable the built-in **admin** account using the new CLI command `set default-admin` under `config system global`. This enhancement improves security and compliance by allowing organizations to prevent login with the default account once alternate administrator accounts have been created. When disabled, the admin account cannot log in, and any active sessions are immediately terminated.

This information is also available in the FortiADC 8.0.1 CLI Reference:

- config system global

Configuration

```
config system global
  set default-admin {enable|disable}
end
```

By default, the **admin** account remains enabled. Disabling it requires another **global administrator**, ensuring that at least one administrator retains unrestricted control of the system. The admin account cannot disable itself, and enabling or disabling must be performed through the CLI. The GUI displays the current status of the account but does not provide configuration controls for this option.

Behavior

- Login attempts with a disabled admin account return the standard “incorrect username or password” error.
- Improper disable attempts (for example, by a non-global admin) return an error message.
- If the account is disabled while logged in, the session is immediately terminated.

Socket Selection Hash Control via CLI 8.0.1

FortiADC introduces a new CLI option, `sip-to-same-sock`, under `config system global` to control how sessions are hashed across sockets. By default, sessions with the same source IP, destination IP, and destination port (`sip+dip+dport`) are consistently directed to the same CPU, `httproxy` process, and listening socket. This ensures that related sessions remain on the same processing path.

With this option, administrators can change the behavior so that sessions with the same `sip+dip+dport` tuple may be distributed across different CPUs, processes, and sockets. This provides additional flexibility for scaling workloads in high-connection environments.

This information is also available in the FortiADC 8.0.1 CLI Reference:

- config system global

Configuration

```
config system global
  set sip-to-same-sock {enable|disable}
end
```

- **enable** – (default) Sessions with the same `sip+dip+dport` are hashed to the same CPU, `httproxy`, and socket.
- **disable** – Sessions with the same `sip+dip+dport` may be distributed across multiple CPUs, processes, and sockets.

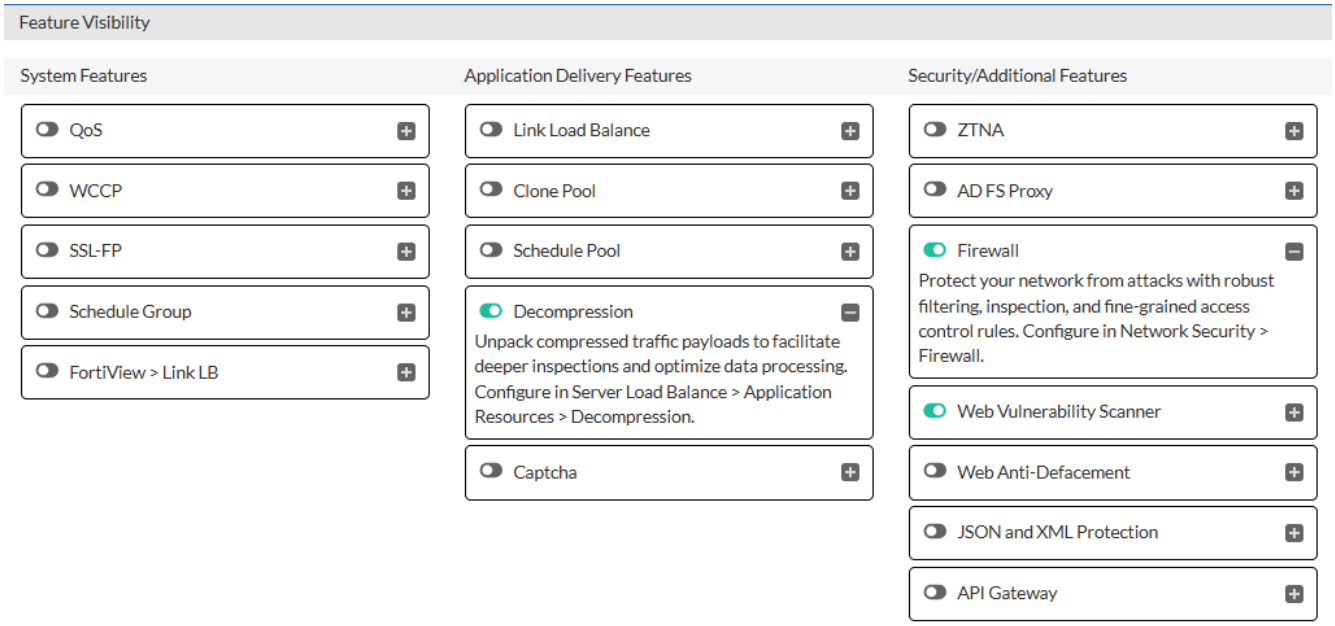
Feature Visibility

You can now selectively control the visibility of configurable features in the GUI via **System > Feature Visibility** or through the CLI. This allows administrators to streamline the interface by hiding features that are unused, inactive, or not relevant to the current deployment.

This information is also available in the FortiADC 8.0.0 Administration Guide and CLI Reference:

- Feature Visibility
- config system feature-visibility/feature-visibility-global

Feature visibility settings affect only the graphical interface and do not control feature activation. A hidden feature remains fully operational and can still be configured through the CLI.

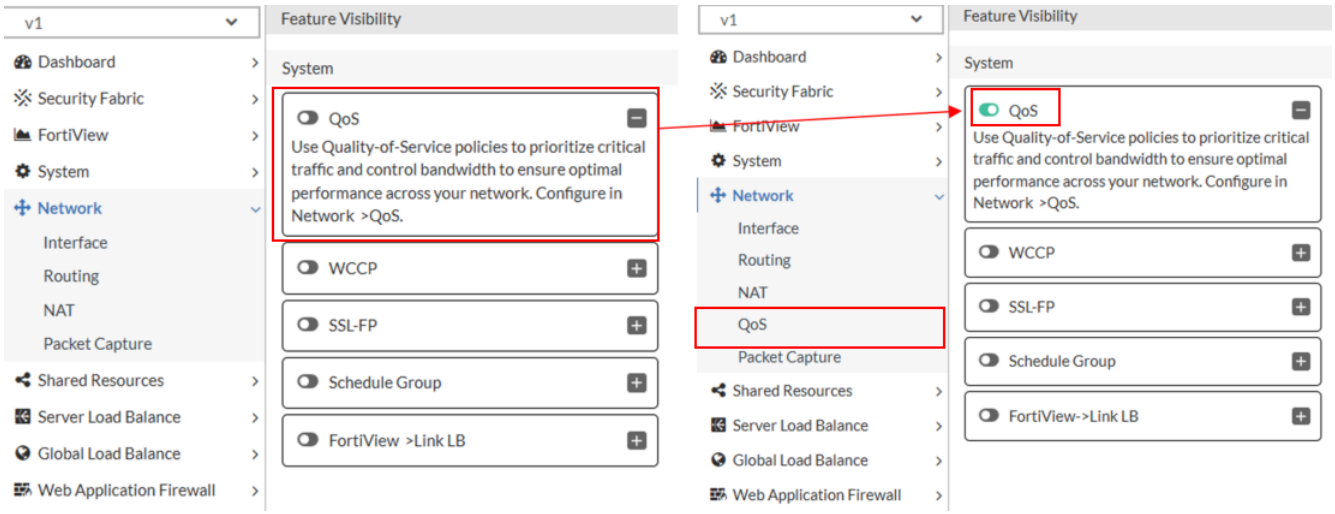


To configure feature visibility settings in the GUI:

1. Navigate to **System > Feature Visibility**.
2. Enable or disable the visibility of individual features as needed.
3. Click **Apply** to save your changes.

Once applied, the GUI navigation menu is refreshed to reflect the updated visibility settings.

For example, if **QoS** is enabled, its corresponding configuration options will become visible in the navigation pane.



Limitations and Tips:

- **Read-only access restrictions**: Users with read-only privileges cannot modify settings on the **System > Feature Visibility** page.
- **Reauthentication required**: After making and saving changes to the feature visibility list, the system automatically logs out the current session. You must log in again to continue working.

Feature Categories

The features available in the Feature Visibility page are grouped into three categories: **System**, **Application Delivery**, and **Security/Additional Feature**. These categories help administrators locate and manage GUI options more efficiently, especially in complex environments.

Some features are visible by default to support common deployment scenarios. Others can be enabled as needed, based on the requirements of the current configuration.

The following features are visible by default:

- Decompression
- Firewall
- Web Vulnerability Scanner

System	Application Delivery	Security/Additional Feature
QoS	Link Load Balance	ZTNA
WCCP	Clone Pool	AD FS Proxy
SSL-FP	Schedule Pool	Firewall
Schedule Group	Decompression	Web Vulnerability Scanner
FortiView->Link LB	Captcha	Web Anti-Defacement
Debug		JSON and XML Protection
		API Gateway

Feature Visibility View by Configuration Context

The features available in the Feature Visibility page vary depending on the system's **VDOM status**, the current **administrative context** (Global vs. VDOM), and the **VDOM mode** (Independent Network or Shared Network).

- **VDOM Disabled**: A single set of features is available and managed system-wide.
- **VDOM Enabled – Independent Network Mode**:
  - The **Global context** shows **only** Debug.
  - Each **VDOM context** (root or non-root) displays the full set of configurable features **except** Debug
- **VDOM Enabled – Shared Network Mode (ADOM Mode)**:
  - The **Global context** shows **only** Debug.
  - The **root ADOM** displays the full set of configurable features **except** Debug.
  - The **non-root ADOM** displays the same set of configurable features as the root ADOM, **except** QoS and Firewall.

Configuration Context	System	Application Delivery	Security/Additional Feature
VDOM Disabled	QoS, WCCP, SSL-FP, Schedule Group, FortiView > Link LB, Debug	Link Load Balance, Clone Pool, Schedule Pool, Decompression, Captcha	ZTNA, AD FS Proxy, Firewall, Web Vulnerability Scanner, Web Anti-Defacement, JSON and XML Protection, API Gateway
VDOM Enabled – VDOM Mode – Global	Debug	*(Not shown)*	*(Not shown)*
VDOM Enabled – VDOM Mode – Root VDOM	QoS, WCCP, SSL-FP, Schedule Group, FortiView > Link LB	Link Load Balance, Clone Pool, Schedule Pool, Decompression, Captcha	ZTNA, AD FS Proxy, Firewall, Web Vulnerability Scanner, Web Anti-Defacement, JSON and XML Protection, API Gateway
VDOM Enabled – VDOM Mode – Non-root VDOM	*(Same as Root VDOM)*	*(Same as Root VDOM)*	*(Same as Root VDOM)*
VDOM Enabled – ADOM Mode – Global	Debug	*(Not shown)*	*(Not shown)*
VDOM Enabled – ADOM Mode – Root ADOM	QoS, WCCP, SSL-FP, Schedule Group, FortiView > Link LB	Link Load Balance, Clone Pool, Schedule Pool, Decompression, Captcha	ZTNA, AD FS Proxy, Firewall, Web Vulnerability Scanner, Web Anti-Defacement, JSON and XML Protection, API Gateway
VDOM Enabled – ADOM Mode – Non-root ADOM	WCCP, SSL-FP, Schedule Group, FortiView > Link LB	*(Same as Root ADOM)*	ZTNA, AD FS Proxy, Web Vulnerability Scanner, Web Anti-Defacement,

Configuration Context	System	Application Delivery	Security/Additional Feature
			JSON and XML Protection, API Gateway

CLI Configuration

Feature visibility can also be controlled through the CLI. The commands differ depending on whether VDOM is enabled and whether you are operating in the Global or VDOM context.

VDOM Disabled

```
config system feature-visibility
  set API_gateway {enable|disable}
  set ad-fs-proxy {enable|disable}
  set captcha {enable|disable}
  set clone-pool {enable|disable}
  set debug {enable|disable}
  set decompression {enable|disable}
  set firewall {enable|disable}
  set json_protection {enable|disable}
  set llb {enable|disable}
  set llb_list {enable|disable}
  set qos {enable|disable}
  set schedule-pool {enable|disable}
  set schedule_group {enable|disable}
  set ssl-fp-resources {enable|disable}
  set wad_profile {enable|disable}
  set wccp {enable|disable}
  set web_vulnerability_scanner {enable|disable}
  set xml_protection {enable|disable}
  set ztna {enable|disable}
end
```

VDOM Enabled

Global context (requires `config global`):

```
config global
config system feature-visibility-global
  set debug {enable|disable}
end
```

VDOM context (requires entering a specific VDOM):

```
config vdom
  edit <vdom_name>
```

```
  config system feature-visibility-global
    set API_gateway {enable|disable}
    set ad-fs-proxy {enable|disable}
    set captcha {enable|disable}
    set clone-pool {enable|disable}
    set decompression {enable|disable}
    set firewall {enable|disable}
    set json_protection {enable|disable}
    set llb {enable|disable}
    set llb_list {enable|disable}
    set qos {enable|disable}
    set schedule-pool {enable|disable}
    set schedule_group {enable|disable}
    set ssl-fp-resources {enable|disable}
    set wad_profile {enable|disable}
    set wccp {enable|disable}
    set web_vulnerability_scanner {enable|disable}
    set xml_protection {enable|disable}
    set ztna {enable|disable}
  next
end
```

Enhancement to WAF Signature Telemetry Reporting to FortiGuard

FortiADC 8.0.0 enhances its threat telemetry capabilities by adding support for uploading **Web Attack Signature** statistics to FortiGuard. This builds on the existing telemetry framework introduced in 7.6.1, which previously supported only IPS and Antivirus (AV) threat data.

What's New in 8.0.0

- **WAF Signature Telemetry**:
  FortiADC now reports statistics related to Web Attack Signature detections. These include signature ID, detection counts, and associated WAF activity observed during HTTP/HTTPS inspection.
- **Revised Data Format for IPS and AV**:
  The data structures used to transmit IPS and Antivirus telemetry have been updated to match the latest FortiGuard backend specifications, ensuring compatibility and future extensibility.

Note: While FortiADC now sends WAF signature statistics to FortiGuard, backend processing and analysis of WAF telemetry is not yet supported. This enhancement enables upload functionality only; FortiGuard does not currently parse or display this data.

Network

The FortiADC 8.0 release includes new features and enhancements to support the **Network**:

VXLAN Support for Kubernetes Calico CNI 8.0.2 on page 121

FortiADC 8.0.2 adds support for VXLAN-based networking with the Kubernetes Calico CNI, enabling FortiADC to integrate with Calico environments that use VXLAN for pod-to-pod and service traffic.

VXLAN Support for Kubernetes Calico CNI 8.0.2

FortiADC 8.0.2 adds support for **VXLAN-based networking with the Kubernetes Calico CNI**, enabling FortiADC to integrate with Kubernetes clusters that use Calico VXLAN for pod and service networking, expanding compatibility with commonly deployed Kubernetes network architectures.

This information is also available in the FortiADC 8.0.2 Administration Guide and CLI Reference:

- Configuring virtual overlay networks
- config system overlay-tunnel

For further information on implementing Calico CNI in the FortiADC Kubernetes Controller, see the Deployment Guide topic on Calico VXLAN CNI.

Enhancements

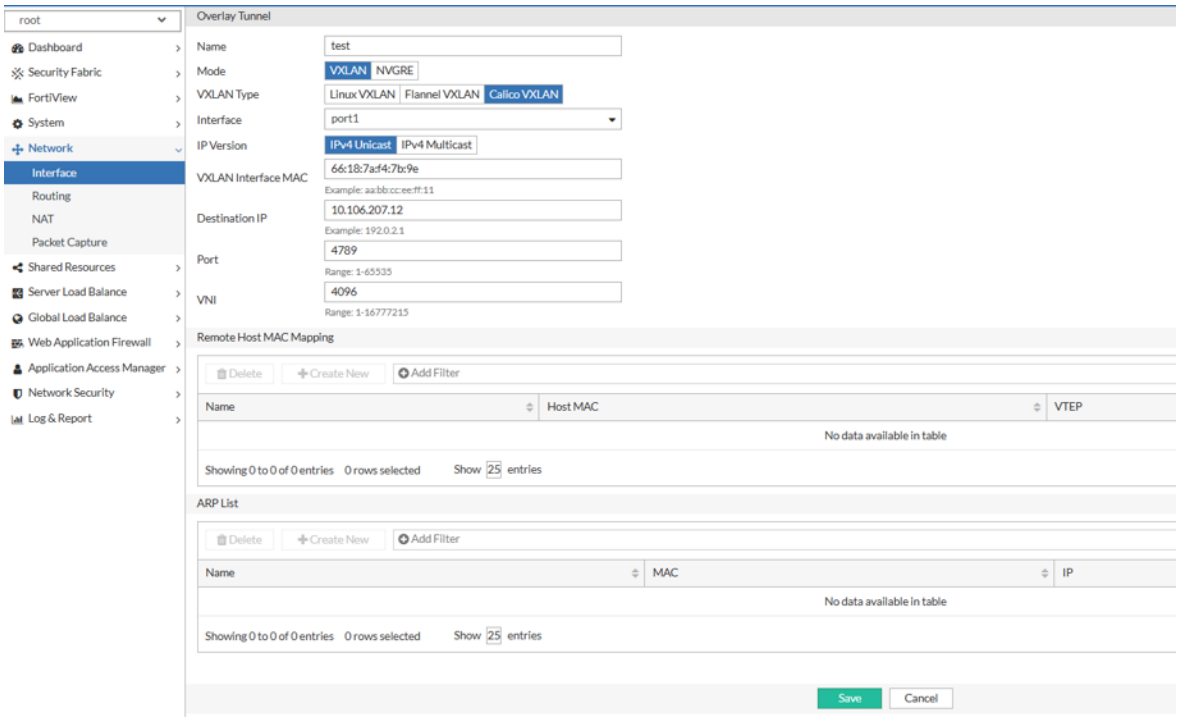
This release introduces the following enhancements to support Calico VXLAN environments:

- Added a new **VXLAN Type**, **Calico VXLAN**, for **Overlay Tunnel** configuration
- Enabled manual configuration of the **VXLAN Interface MAC** address to align with Calico-assigned MAC addresses
- Ability to configure Address Resolution Protocol (ARP) entries associated with Calico VXLAN overlay tunnels

These enhancements allow FortiADC to participate correctly in Calico-managed VXLAN networks and ensure proper data plane connectivity between FortiADC and Kubernetes pods.

Configuration updates

To support Calico VXLAN deployments, updates have been made to the **Overlay Tunnel** configuration.



A new **Calico VXLAN** option is now available in the **VXLAN Type** field when creating or editing an overlay tunnel (**Network > Interface > Overlay Tunnel**). When this VXLAN type is selected, additional configuration parameters required by Calico VXLAN networking are displayed.

Calico VXLAN configuration parameters

Parameter	Description
Interface	Specifies the FortiADC interface used for the VXLAN overlay tunnel.
IP Version	Specifies whether the VXLAN tunnel uses IPv4 or IPv6 addressing.
VXLAN Interface MAC	Specifies the MAC address of the VXLAN interface. This value must match the MAC address assigned by Calico for proper data plane connectivity.
Destination IP	Specifies the remote VXLAN endpoint IP address used for tunnel encapsulation.
Port	Specifies the UDP port used for VXLAN traffic.
VNI	Specifies the VXLAN Network Identifier (VNI) used to identify the VXLAN segment.

After the overlay tunnel is created, additional configuration sections become available for the Calico VXLAN overlay tunnel.

Remote Host MAC Mapping and ARP configuration

For Calico VXLAN overlay tunnels, FortiADC provides dedicated **Remote Host MAC Mapping** and **ARP List** configuration sections. These sections allow FortiADC to maintain the IP-to-MAC mappings required for VXLAN data plane connectivity in Calico environments.

These configuration areas may be updated automatically by the FortiADC Kubernetes Controller as Kubernetes services are deployed.

Server Load Balance

The FortiADC 8.0 release includes new features and enhancements in **Server Load Balance**:

TLS 1.3 Hardening and Post-Quantum Cryptography Support 8.0.3 on page 126

FortiADC 8.0.3 introduces advanced hardening for TLS 1.3 handshakes and adds support for Post-Quantum Cryptography (PQC) to protect data against future quantum computing threats. These enhancements provide granular control over cryptographic parameters through new security level settings and customizable signature and group selections.

FQDN Real Server DNS Cache and Refresh Configuration 8.0.3 on page 129

FortiADC 8.0.3 introduces enhanced control over Domain Name System (DNS) resolution for Real Servers configured with Fully Qualified Domain Names (FQDN). This update allows you to manually configure the Time to Live (TTL) for cached DNS responses and define a minimum refresh interval, ensuring more predictable traffic steering when backend IP addresses change. By customizing these settings, you can prevent premature cache expiration or reduce the frequency of DNS queries to your nameservers.

Load Balance Pool Support in Stream Scripts 8.0.3 on page 131

You can now use Stream Scripting to load balance an entire real server pool, instead of only individual real servers, by leveraging the new `LB:routing()` script command and Layer 7 content routing capabilities for non-HTTP protocols. This enhancement allows scripts to inspect the application-layer payload and programmatically select a destination real server pool based on real-time traffic analysis.

Increased Capacity for Content Routing and Health Check Objects 8.0.2 on page 133

FortiADC 8.0.2 increases the maximum number of Content Routing (CR) and Health Check (HC) objects that can be configured on the system. The increased limits allow administrators to define more CR and HC objects overall, improving scalability for complex traffic distribution and service monitoring scenarios. Existing constraints on how many health checks can be referenced by a single pool, and how many content routes can be referenced by a single virtual server, remain unchanged.

Advanced mTLS Support with Enhanced Client Authentication and C3D 8.0.1 on page 135

FortiADC expands its **mutual TLS (mTLS)** capabilities with advanced features that strengthen security and improve deployment flexibility. mTLS requires both the client and server to authenticate each other using certificates, ensuring trusted, bidirectional communication.

While FortiADC already supported basic mTLS, this release introduces advanced functions for greater control and interoperability:

- **Enhanced Client Authentication** – Configurable authentication frequency and selective advertisement of trusted certificate authorities (CAs).
- **Client Certificate Constrained Delegation (C3D)** – Allows FortiADC to issue delegated client certificates when forwarding traffic to backend servers, maintaining mutual TLS authentication while still enabling SSL decryption and inspection.

Together, these enhancements provide administrators with fine-grained control over certificate handling, ensuring secure, verifiable mTLS chains on both the client- and server-facing sides of FortiADC.

Advanced TCP Optimization and Transparent Proxy Support for L7 TCP Virtual Servers 8.0.1 on page 139

FortiADC extends its L7 TCP virtual server capabilities with support for transparent TCP proxying and advanced TCP optimization features. While transparent proxy modes (Layer 2 and Layer 3) were already available for other types of virtual servers, this enhancement makes them available for **L7 TCP virtual servers**, enabling inline deployments with full application-layer visibility and control. In addition, L7 TCP profiles now include per-connection tuning parameters and congestion control options, giving administrators precise control over throughput, efficiency, and reliability.

Content Rewriting support for HTTP/3 and Backend HTTP/2 8.0.1 on page 144

FortiADC 8.0.1 extends the content rewriting functionality to HTTPS Virtual Servers that have HTTP/3 enabled on the frontend or Backend HTTP/2 enabled. Previously, content rewriting was limited to Virtual Servers with HTTP profiles, which meant services delivered over HTTP/3 or full end-to-end HTTP/2 could not take advantage of the same traffic manipulation policies.

Support Proxy Protocol for L4 TCP on page 147

FortiADC now supports Proxy Protocol v1 and v2 in Layer 4 TCP server load balancing (SLB) deployments. This enhancement allows client connection metadata—such as source IP address and port—to be preserved across NAT boundaries and forwarded to backend servers. Proxy Protocol insertion ensures that real servers can accurately log or respond to the original client source, even in NAT or multi-hop environments.

Clear Session and Persistence Table for HTTP/S Virtual Servers on page 150

FortiADC extends session and persistence clearing functionality to Layer 7 virtual servers (VS), including those that handle HTTP, HTTPS, TCPS, and RDP traffic. Prior to this enhancement, the CLI command sets, `diagnose server-load-balance session` and `diagnose server-load-balance persistence`, supported clearing operations only for Layer 4 (TCP/UDP) virtual servers. This update extends both command sets to support Layer 7 virtual servers by introducing the l7-http keyword, which targets httproxy-managed session and persistence tables.

Support Multi-Process Mode for Up to 64 Processes per Virtual Server on page 153

FortiADC now supports up to 64 processes per virtual server in multi-process mode, significantly increasing the previous limit of 15. This enhancement is designed to improve performance and scalability on high-core platforms by allowing better distribution of traffic handling across CPU cores.

Scripting Support for Persistence Functions in HTTP Data Events on page 154

FortiADC now supports calling persistence-related scripting functions—**HTTP:persist()** and **HTTP:lookup_tbl()**—within HTTP data phase events. This enhancement allows scripting-based persistence decisions to be made using values extracted from HTTP payloads, such as session tokens embedded in POST request bodies.

RFC 7919 Compliance Support for TLS 1.3 in SSL Profiles on page 157

FortiADC now supports the **RFC 7919 Comply** option when **TLS 1.3** is selected in the allowed SSL versions of **Client SSL Profiles** and **Real Server SSL Profiles**. This enhancement resolves a previous limitation where enabling RFC 7919 compliance would result in a configuration error if SSLv3 or TLS 1.3 was also selected.

TLS 1.3 Hardening and Post-Quantum Cryptography Support 8.0.3

FortiADC 8.0.3 introduces advanced hardening for TLS 1.3 handshakes and adds support for Post-Quantum Cryptography (PQC) to protect data against future quantum computing threats. These enhancements provide granular control over cryptographic parameters through new security level settings and customizable signature and group selections.

Post-Quantum Cryptography (PQC) Support

To defend against the evolving threat of quantum computing, FortiADC now supports PQC algorithms for both key exchange and digital signatures:

- **Quantum-Resistant Algorithms**: Support has been added for Module-Lattice-Based Key-Encapsulation Mechanism (ML-KEM) and Module-Lattice-Based Digital Signature Algorithm (ML-DSA).
- **Hybrid Key Exchange**: Users can configure hybrid groups, such as `x25519mlkem768`, which combine traditional elliptic curve cryptography with post-quantum algorithms for robust security.
- **Signature Schemes**: New PQC signature algorithms, including `mldsa44`, `mldsa65`, and `mldsa87`, are now available for TLS authentication.
- **Future-Proof Protection**: These algorithms are designed to protect TLS sessions from "harvest now, decrypt later" attacks initiated by future quantum computers.

Predefined Security Levels and Customization

New settings allow you to simplify TLS configuration or manually harden specific parameters to meet strict compliance needs:

- **Security Level Profiles**: You can select from `default`, `medium`, or `high` security levels to automatically apply recommended sets of supported groups and signature algorithms.
- **Manual Hardening**: By selecting the `customized` security level, you can manually select and order the specific signature algorithms and key exchange groups permitted for a profile.
- **Expanded Group Support**: Supported groups now include a wide range of options such as the FFDHE series (`ffdhe2048` to `ffdhe8192`), `X25519`, `X448`, and the new `MLKEM` series.
- **Signature Algorithm Control**: Administrators can restrict authentication to modern schemes like `RSA-PSS` or specific `ECDSA` curves while excluding weaker options.

Hardware Acceleration Considerations

When using hardware-accelerated platforms, specific settings may be required to support advanced algorithms:

- **Hardware SSL Load Provider**: For hardware platforms with SSL cards, this new parameter allows the system to support specific exchange algorithms like `x25519mlkem768` and the `FFDHE` series.

- **Performance Impact**: Enabling the hardware SSL load provider for these specific algorithms may significantly decrease overall SSL performance.
- **Compatibility Logic**: Certain groups, such as the FFDHE series on the Real Server side, require the hardware load provider to be enabled to function correctly.

Configuration

The following updates are available to manage TLS hardening and PQC settings.

Client and Server SSL Profiles

The `supported-groups-sigalg-security-level` parameter (CLI) determines the security baseline for both Client SSL Profiles and Real Server SSL Profiles.

- **Security Levels (CLI)**:
  - `default`, `medium`, `high`: Automatically applies predefined sets of groups and algorithms.
  - `customized`: Required to manually select and order supported-groups and sigalgs via CLI.
- **Supported Groups (GUI & CLI)**: Includes standard elliptic curves (`secp256r1`, `x25519`), finite field groups (`ffdhe2048` through `ffdhe8192`), and the new PQC options (`mlkem512`, `mlkem768`, `mlkem1024`, `secp256r1mlkem768`, `secp384r1mlkem1024`, and `x25519mlkem768`).
- **Signature Algorithms (sigalgs - CLI)**: Includes traditional `RSA` and `ECDSA` schemes alongside new PQC options (`mldsa44`, `mldsa65`, `mldsa87`).

CLI Example:

```
config load-balance client-ssl-profile
    edit <name>
        set supported-groups-sigalg-security-level customized
        set supported-groups x25519mlkem768 x25519 secp256r1
        set sigalgs mldsa65 rsa_pss_rsae_sha256 ecdsa_secp256r1_sha256
    next
end
```

Virtual Server

For hardware platforms equipped with an SSL card, the `hardware-ssl-load-provider` parameter must be enabled via CLI to support specific advanced algorithms.

- **Hardware SSL Load Provider**:
  - **Enabled**: Necessary to support specific key exchange algorithms such as `x25519mlkem768` and the `FFDHE` series on hardware platforms. If this is disabled, attempting to select these unsupported algorithms in the GUI or CLI will trigger an error message.
  - **Performance Note**: Enabling this option may lead to a significant decrease in overall SSL performance.

CLI Example:

```
config load-balance virtual-server
    edit <name>
```

```
        set hardware-ssl-load-provider enable
    next
end
```

FQDN Real Server DNS Cache and Refresh Configuration 8.0.3

FortiADC 8.0.3 introduces enhanced control over Domain Name System (DNS) resolution for Real Servers configured with Fully Qualified Domain Names (FQDN). This update allows you to manually configure the Time to Live (TTL) for cached DNS responses and define a minimum refresh interval, ensuring more predictable traffic steering when backend IP addresses change. By customizing these settings, you can prevent premature cache expiration or reduce the frequency of DNS queries to your nameservers.

Optimized DNS Resolution Control

The new configuration parameters provide granular management of how FortiADC handles FQDN-based Real Servers:

- **Configurable DNS Cache TTL**: You can now specify the duration that DNS resolution results are stored in the local cache, overriding the TTL provided by the DNS server if necessary.
- **Minimum Refresh Interval**: The system supports a minimum refresh timer to prevent the load balancer from querying DNS servers too frequently, which helps reduce network overhead and protects DNS infrastructure.
- **Dynamic IP Handling**: When an FQDN resolves to multiple IP addresses, FortiADC continues to map these to the Real Server pool based on the refined cache and refresh logic.

CLI Configuration

This feature is managed exclusively through the Command Line Interface (CLI) when the Real Server type is set to `fqdn` or `fqdn_populate_more`.

```
config load-balance real-server
    edit <name>
        set type {fqdn | fqdn_populate_more}
        set fqdn <domain_name>
        set fqdn-cache-ttl <integer>
        set fqdn-min-refresh <integer>
    next
end
```

Parameter	Description	Default
`fqdn-cache-ttl`	Sets the time-to-live (TTL) for cached FQDN records, in seconds. A value of 0 means the TTL from the DNS server is used. This value must not be lower than `fqdn-min-refresh`, except when set to 0. The valid range is 0-86400.	0

Parameter	Description	Default
`fqdn-min-refresh`	Sets the minimum refresh interval for FQDN queries, in seconds. This value must be lower than or equal to `fqdn-cache-ttl`, except when `fqdn-cache-ttl` is set to 0. The valid range is 60-3600.	60

Load Balance Pool Support in Stream Scripts 8.0.3

You can now use Stream Scripting to load balance an entire real server pool, instead of only individual real servers, by leveraging the new `LB:routing()` script command and Layer 7 content routing capabilities for non-HTTP protocols. This enhancement allows scripts to inspect the application-layer payload and programmatically select a destination real server pool based on real-time traffic analysis.

Advanced Layer 7 Stream Traffic Steering

The integration of load-balance pools into Stream Scripts provides sophisticated control over application-layer traffic for non-HTTP protocols:

- **Layer 7 Content Routing**: You can now perform deep packet inspection on TCP and UDP streams to route traffic based on specific strings or patterns identified within the payload.
- **Pool-Based Load Balancing**: This feature allows the load balancer to target a complete real server pool rather than a single real server, providing improved redundancy and resource distribution for stream-based traffic.
- **Dynamic Routing Logic**: The new `LB:routing()` command allows the scripting engine to dynamically assign traffic to a specific backend pool based on the identified content.
- **Configuration Requirements**: Script-based pool routing only functions when content routing is enabled on the virtual server; otherwise, the virtual server forwards traffic to its default real server pool.

New Stream Script Command

The introduction of the `LB:routing()` command allows you to programmatically assign traffic to a specific real server pool by referencing a predefined content routing profile. This provides a flexible way to manage Layer 7 TCP and UDP traffic steering through the scripting engine.

Syntax

```
LB:routing("content_routing_profile_name")
```

Argument

Name	Description
`content_routing_profile_name`	Specifies the name of the content routing profile to be used for the connection. **Note**: • The profile must be previously configured in the system; otherwise, the script returns an error. • If the referenced profile contains parameters unsupported by Layer 7 (such as `LB_METHOD_URI`), an error message is returned.

Events

- **STREAM_CLIENT_INIT**: Triggered when a client connection is first initialized.
- **STREAM_REQUEST_DATA**: Triggered when the system receives data from the client stream.

Example

```
when STREAM_CLIENT_INIT{
    LB:routing("content routing profile")
}
```

Increased Capacity for Content Routing and Health Check Objects 8.0.2

FortiADC 8.0.2 increases the maximum number of **Content Routing (CR)** and **Health Check (HC)** objects that can be configured on the system. The increased limits allow administrators to define more CR and HC objects overall, improving scalability for complex traffic distribution and service monitoring scenarios. Existing constraints on how many health checks can be referenced by a single pool, and how many content routes can be referenced by a single virtual server, remain unchanged.

This information is also available in the FortiADC 8.0.2 Administration Guide:

- Maximum Configuration Values

Supported platform limits

The following tables list the supported limits for **Content Routing (CR)** and **Health Check (HC)** objects by platform. All limits apply **per VDOM**. Existing limits on the number of health checks that can be referenced by a single pool, and the number of content routes that can be referenced by a single virtual server, remain unchanged. **Bold values indicate increased limits introduced in FortiADC 8.0.2**.

Virtual Platforms

VM Platform	Max Health Checks (HC)	Max Content Routes (CR)
VM00	512	128
VM01	512	128
VM02	512	256
VM04	512	512
VM08	**2048**	**2048**
VM16	**4096**	**4096**
VM32	**4096**	**4096**

Hardware Platforms

Hardware Model	Max Health Checks (HC)	Max Content Routes (CR)
60F	512	256

Hardware Model	Max Health Checks (HC)	Max Content Routes (CR)
100F	512	256
120F	512	256
200F	512	256
200D	512	256
220F	512	256
300D	512	512
300E	512	512
300F	512	512
320F	512	512
400D	512	512
400F	**1024**	**1024**
420F	**1024**	**1024**
700D	**1024**	**1024**
1000F	**1024**	**1024**
1200F	**1024**	**1024**
1500D	**1024**	**1024**
2000D	**1024**	**1024**
2000F	**1024**	**1024**
2200F	**2048**	**2048**
4000D	**2048**	**2048**
4000F	**2048**	**2048**
4200F	**4096**	**4096**
5000F	**8192**	**8192**

Advanced mTLS Support with Enhanced Client Authentication and C3D 8.0.1

FortiADC expands its **mutual TLS (mTLS)** capabilities with advanced features that strengthen security and improve deployment flexibility. mTLS requires both the client and server to authenticate each other using certificates, ensuring trusted, bidirectional communication.

While FortiADC already supported basic mTLS, this release introduces advanced functions for greater control and interoperability:

- **Enhanced Client Authentication** – Configurable authentication frequency and selective advertisement of trusted certificate authorities (CAs).
- **Client Certificate Constrained Delegation (C3D)** – Allows FortiADC to issue delegated client certificates when forwarding traffic to backend servers, maintaining mutual TLS authentication while still enabling SSL decryption and inspection.

Together, these enhancements provide administrators with fine-grained control over certificate handling, ensuring secure, verifiable mTLS chains on both the client- and server-facing sides of FortiADC.

This information is also available in the FortiADC 8.0.1 Administration Guide and CLI Reference:

- Validating certificates
- Configuring real server SSL profiles
- config system certificate certificate_verify
- config load-balance real-server-ssl-profile

Understanding the Workflow

Advanced mTLS in FortiADC consists of two complementary functions:

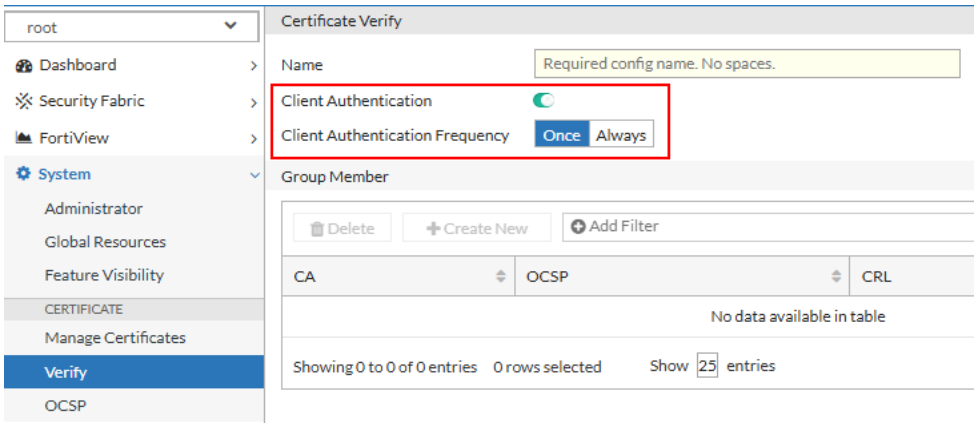
- **Client-side authentication** – FortiADC validates certificates presented by external clients using a **Certificate Verify** object referenced in a **Client SSL profile**.
- **Server-side delegation** – When FortiADC forwards traffic to backend servers, **C3D** enables it to present delegated client certificates signed by an approved local CA, ensuring backend servers can still perform mutual authentication.

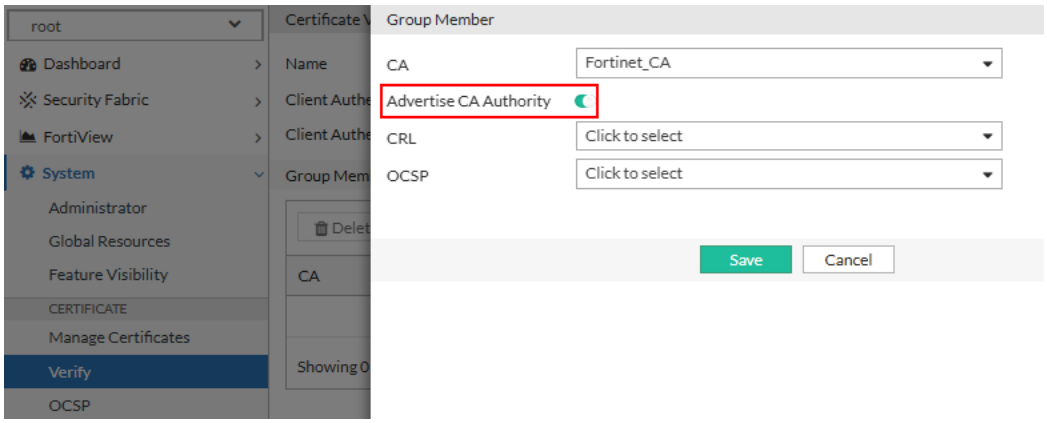
The two mechanisms operate independently but can be combined for end-to-end security. Certificate Verify secures **inbound connections**, while C3D secures **outbound TLS connections** toward real servers.

Enhanced Client Authentication (Client SSL)

Enhanced Client Authentication extends standard mTLS by allowing administrators to control how FortiADC authenticates client certificates and which trusted authorities are advertised during the TLS handshake.

Configuration is available in **System > Verify**, under the **Verify** tab.





To configure Enhanced Client Authentication in the Certificate Verify object:

1. **Enable Client Authentication** - Enable or disable advanced mode for mutual TLS (mTLS) client authentication.
   - **Disabled (default)**
     FortiADC performs standard mTLS client authentication,
   - **Enabled:**
     FortiADC enables advanced mTLS capabilities, including:
     - **Client Certificate Configuration Delegation (C3D):**
       FortiADC extracts client identity information from the client certificate, dynamically re-signs a delegated client certificate, and uses it to establish an mTLS connection with the backend server. This allows client identity information to be securely propagated to the backend server.
       This is configured under **Server SSL** using the **C3D** settings. See Client Certificate Constrained Delegation (Server SSL) on page 137.
     - **Client Authentication Frequency:**
       Allows control over how often clients are required to present their certificates for authentication.
     - **Advertise CA Authority:**
       Allows FortiADC to advertise trusted CA authorities to clients during the TLS handshake.
2. **Set Authentication Frequency** - Once client authentication is enabled, FortiADC allows administrators to decide how often certificates are checked:

- **Once** (default) – The client certificate is validated only at session establishment.
- **Always** – The client certificate is validated for the initial handshake and every time the session is resumed, providing stronger security in long-lived or reused sessions.

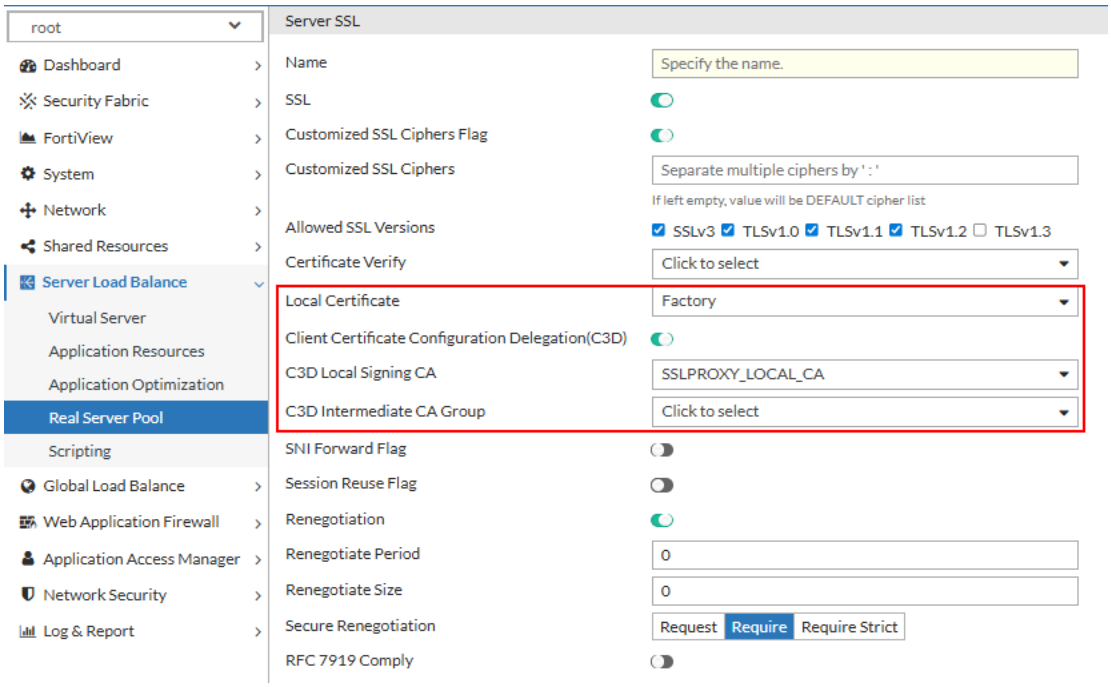
3. **Advertise Certificate Authorities (Group Member)** – Within the **Group Member** configuration, enable this option to control whether FortiADC advertises the selected CA list to clients when requesting a certificate. This indicates which CAs are trusted by FortiADC. If disabled, the selected CAs are still used for validation but are not displayed to clients.

By requiring explicit trust anchors and controlling certificate revalidation, FortiADC strengthens the mutual TLS handshake while giving administrators flexibility in balancing performance and security.

Note: When Client Authentication is enabled, Certificate Verify objects are restricted to Client SSL profiles. Application in Real Server SSL profiles is unsupported and has no effect.

Client Certificate Constrained Delegation (Server SSL)

C3D is configured in **Server Load Balance > Real Server Pool > Server SSL** tab. This feature allows FortiADC to re-issue validated client certificates and present them to backend servers, ensuring mutual TLS is preserved even when FortiADC decrypts and inspects the traffic.



To configure C3D in the Real Server SSL:

1. **Select a Local Certificate** – The local certificate serves as a template for delegation. FortiADC copies its identifying information and uses it as the basis for a new delegated certificate issued to the backend server. Without a local certificate, C3D cannot be enabled because FortiADC lacks a signing identity.
2. **Enable Client Certificate Constrained Delegation (C3D)** – Once a local certificate is defined, this option becomes available.
   When enabled, FortiADC issues a delegated client certificate to establish a new mTLS connection with the real server, allowing the backend server to enforce mutual TLS authentication while still enabling FortiADC to decrypt,

inspect, and re-encrypt traffic between the client and server. Disabled by default.

3. **Configure C3D Parameters** – With C3D enabled, two additional fields appear:
   - **C3D Local Signing CA** (required) – Specifies the local CA used by FortiADC to sign the delegated client certificate.
   - **C3D Intermediate CA Group** (optional) – Specifies an intermediate CA chain to include with the delegated certificate, ensuring that backend servers can build a complete trust path.

This dependency-based configuration ensures that C3D is enabled only when FortiADC has both a valid local certificate and signing authority, maintaining continuous mutual authentication from client to server.

Advanced TCP Optimization and Transparent Proxy Support for L7 TCP Virtual Servers 8.0.1

FortiADC extends its L7 TCP virtual server capabilities with support for transparent TCP proxying and advanced TCP optimization features. While transparent proxy modes (Layer 2 and Layer 3) were already available for other types of virtual servers, this enhancement makes them available for **L7 TCP virtual servers**, enabling inline deployments with full application-layer visibility and control. In addition, L7 TCP profiles now include per-connection tuning parameters and congestion control options, giving administrators precise control over throughput, efficiency, and reliability.

This information is also available in the FortiADC 8.0.1 Administration Guide and CLI Reference:

- Configuring Application profiles
- config load-balance profile

Key improvements include support for **BBR (Bottleneck Bandwidth and Round-trip propagation time) congestion control, TCP window scaling (RFC 1323)**, and per-connection tuning of **receive windows, send buffers, and maximum segment size (MSS)**. Together, these enhancements allow FortiADC to deliver optimized TCP performance in high-bandwidth or high-latency environments while maintaining deployment flexibility for service providers and enterprises.

Transparent TCP Proxy Support for L7 TCP Virtual Servers

L7 TCP virtual servers now support deployment in both transparent proxy modes:

- **Layer 2 Transparent Proxy**: The client and server reside on the same subnet, and FortiADC operates at Layer 2. The virtual server binds to a soft-switch interface, intercepting and proxying TCP flows without IP address translation. This mode simplifies inline insertion where routing changes are not desired.
- **Layer 3 Transparent Proxy**: The client and server are located on different subnets, and FortiADC operates at Layer 3. The virtual server binds to an inbound interface, and packet forwarding is handled through routing entries. FortiADC becomes a routed hop while maintaining transparency at the TCP session level.

In both deployment models, the L7 TCP virtual server terminates and re-establishes TCP sessions end-to-end while preserving client IP visibility and applying the configured optimization policies.

TCP Optimization Features

The L7 TCP profile has been enhanced with new fields for tuning TCP performance. These options are applied independently on the client and server sides of a connection, providing asymmetric optimization where needed:

- **Receive and send buffer control** for both client and server connections.
- **Maximum Segment Size (MSS)** tuning to prevent fragmentation.
- **Congestion control algorithm selection**, including **CUBIC** (default), **Reno**, and the new **BBR** for high-throughput, low-latency optimization.

BBR Congestion Control

BBR is a modern congestion control algorithm designed to maximize throughput and minimize latency by modeling available bandwidth and RTT. It is especially effective in long-distance or high-bandwidth environments where traditional loss-based algorithms (such as Reno) are less efficient.

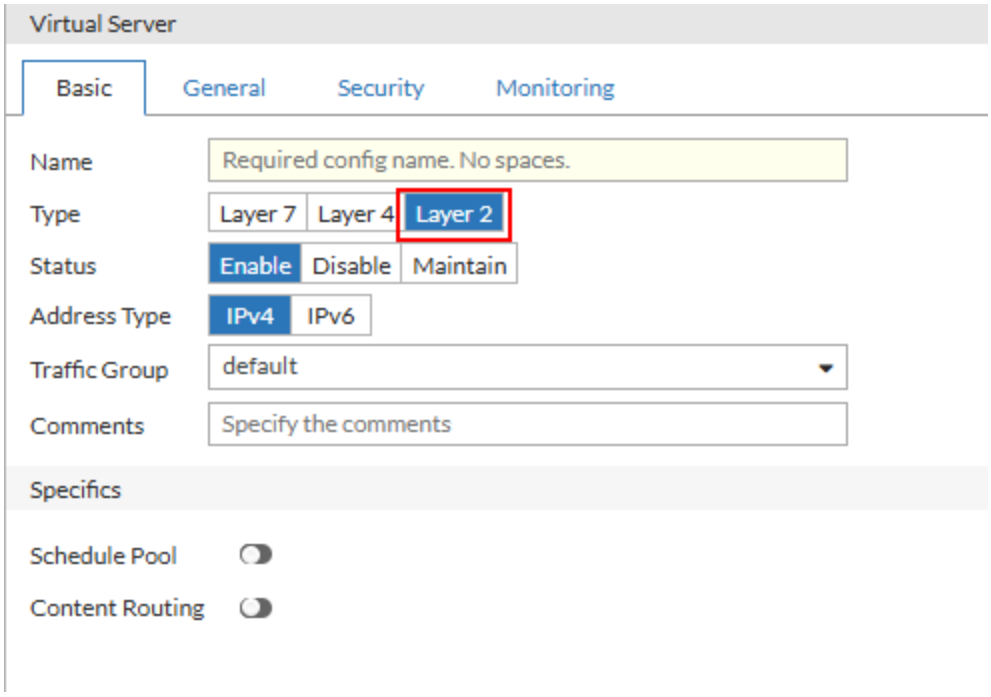
TCP Window Scaling

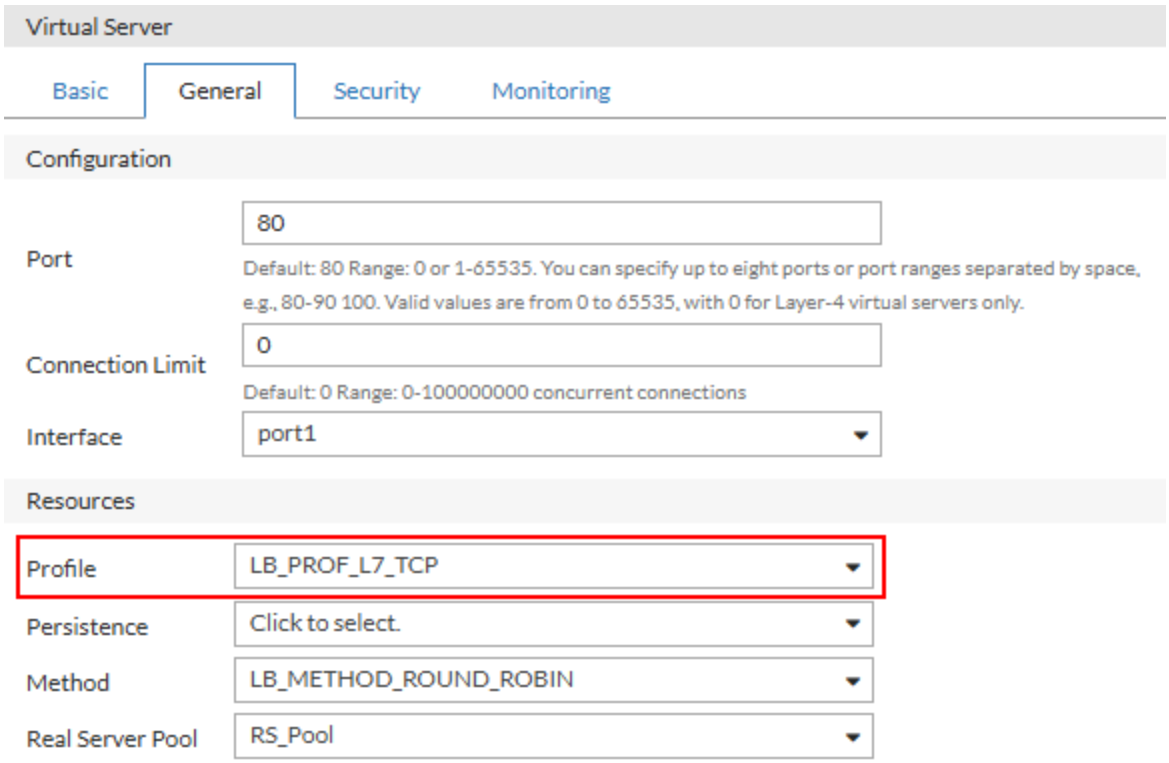
When buffer sizes exceed 65,535 bytes, FortiADC automatically enables window scaling (RFC 1323). This allows advertised windows up to 1 GB, ensuring efficient transmission over high-delay networks.

Configuration

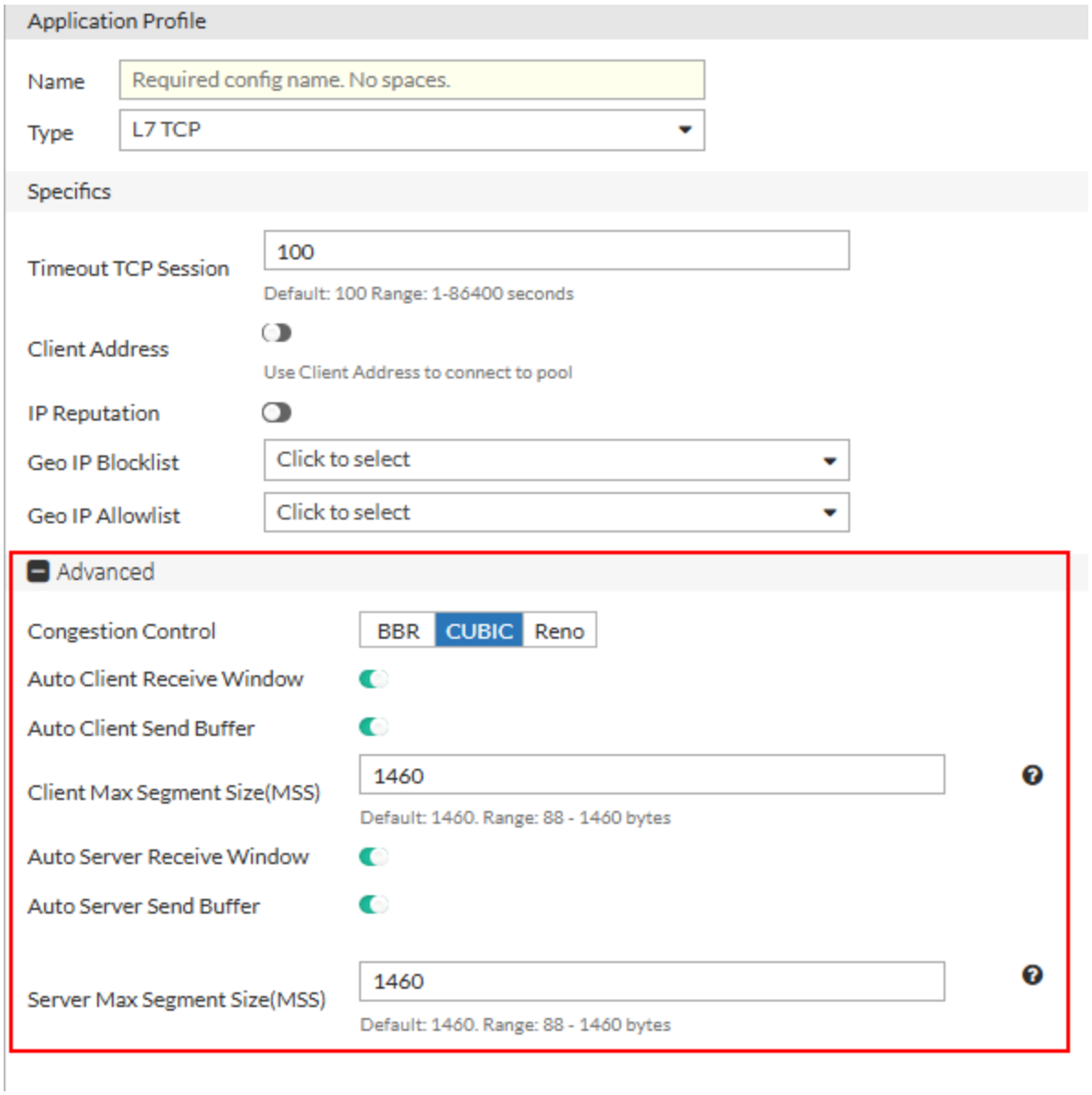
The new enhancements are fully integrated into the FortiADC GUI and are configured from the **Virtual Server** and **Application Profile** configuration pages under **Server Load Balance**.

When creating a new virtual server, the **Type** field now supports **Layer 2** when paired with an **L7 TCP profile**. In this mode, the virtual server is bound to a soft-switch interface and requires only **Interface** and **Port** to be defined. All other Layer 7 parameters remain unchanged. This enables FortiADC to function as a transparent inline proxy without IP address translation, preserving client IPs and simplifying deployment into existing topologies. TCP optimization is applied transparently to the proxied traffic.





The **L7 TCP Application Profile** page introduces additional fields for tuning TCP performance. These options are available for both client and server sides of a connection:



Option	Description	Range / Values	Notes
Receive Window	Maximum advertised receive window size.	1184-16,777,216 bytes	Effective value is doubled internally by Linux.
Send Buffer	Socket send buffer size.	2368-16,777,216 bytes	Applied on accepted sockets.
Maximum Segment Size (MSS)	Maximum TCP payload size per segment (excluding headers).	88-1460 bytes	Helps prevent fragmentation.
Congestion Control	Congestion control algorithm.	CUBIC (default), BBR, Reno	Must match system-supported algorithms.

Example workflow:

1. Go to **Server Load Balance > Application Resources > Application Profile** and create a new profile of type **L7 TCP**.
2. Configure TCP optimization settings, such as:
   - Set **Client Receive Window** and **Send Buffer** to match the expected bandwidth-delay product.
   - Specify **MSS** to match the network MTU.
   - Select a **Congestion Control Algorithm**, such as **BBR** for high-bandwidth, high-latency links.
3. Assign the profile to a virtual server configured for Layer 2 or Layer 3 transparent proxy mode.

Limitations

Each parameter can be configured independently for client and server, allowing asymmetric optimization. Note that some TCP features remain limited: per-virtual-server controls for **maximum congestion window**, **SACK**, and **dynamic buffer scaling** are not exposed in the GUI and remain global. Additionally, **TCP connection multiplexing is not supported**, and only the listed congestion control algorithms are available.

Content Rewriting support for HTTP/3 and Backend HTTP/2 8.0.1

FortiADC 8.0.1 extends the content rewriting functionality to HTTPS Virtual Servers that have HTTP/3 enabled on the frontend or Backend HTTP/2 enabled. Previously, content rewriting was limited to Virtual Servers with HTTP profiles, which meant services delivered over HTTP/3 or full end-to-end HTTP/2 could not take advantage of the same traffic manipulation policies.

This information is also available in the FortiADC 8.0.1 Administration Guide:

- Configuring virtual servers
- Using content rewriting rules

Content rewriting allows administrators to modify HTTP request and response content by adding, removing, or replacing headers and rewriting URLs or message bodies. By extending support to HTTP/3 and HTTP/2, FortiADC ensures that rewriting rules can be applied consistently across all major HTTP protocol versions, eliminating configuration gaps and maintaining operational consistency as organizations adopt newer protocols.

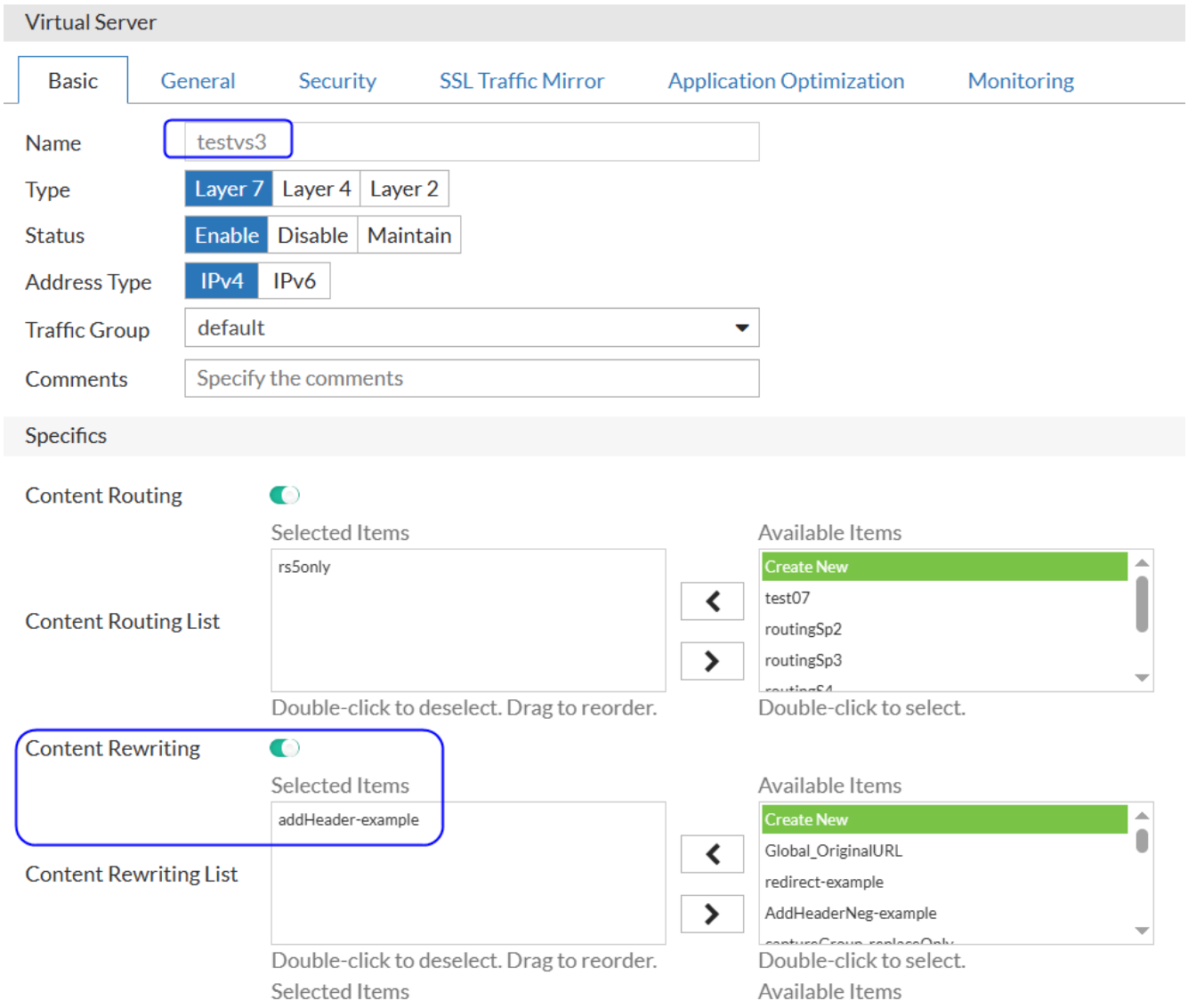
Protocol-Specific Considerations

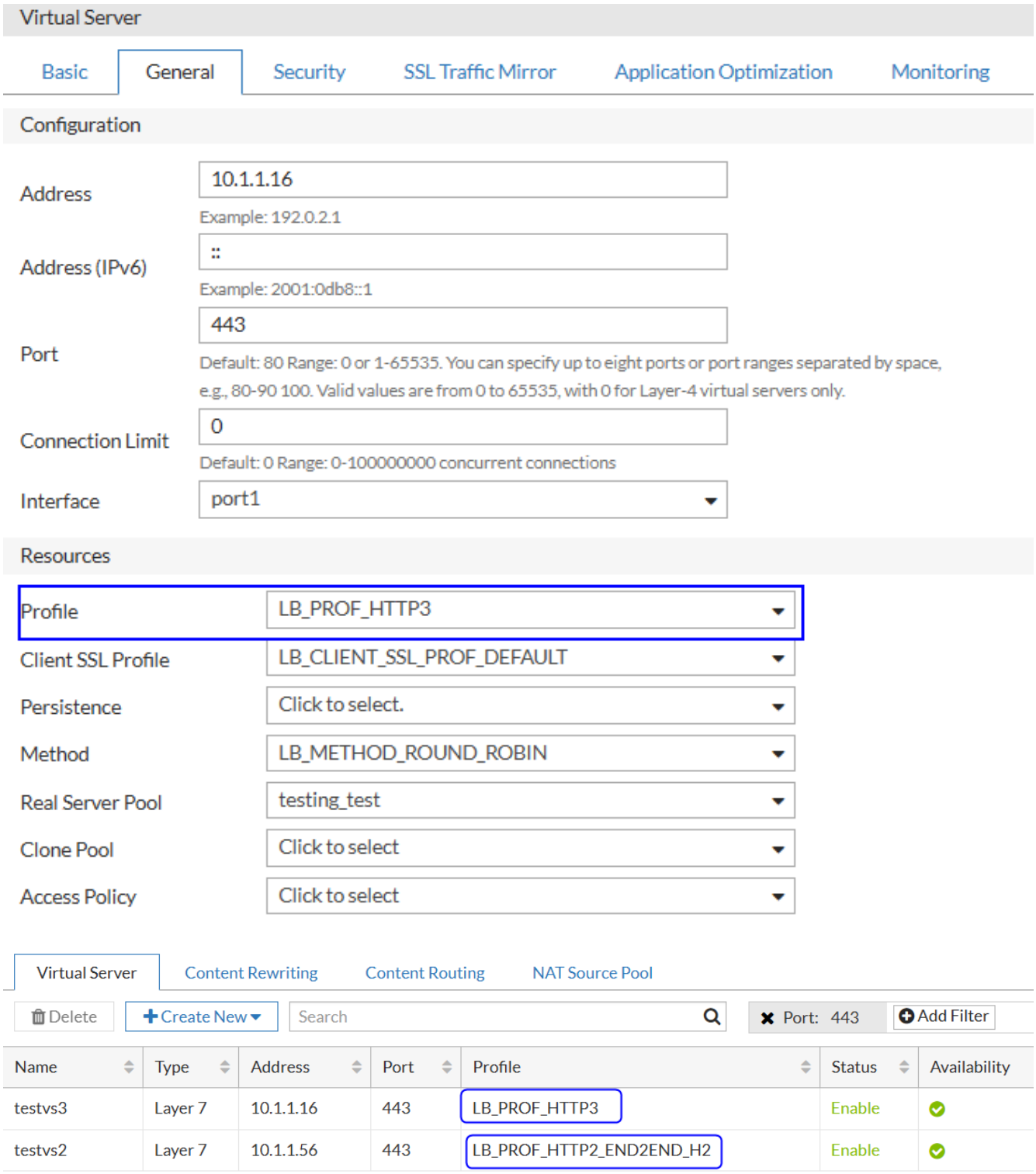
Behavior	HTTP/1.1	HTTP/2 (Backend HTTP/2 enabled)	HTTP/3
Header normalization	Initial-capital format	All lowercase	All lowercase
Phase scoping	Not restricted	Certain fetches request-only; use variables	Certain fetches request-only; use variables
Pseudo-headers	N/A	Reserved (`:method`, `:scheme`, `:path`, `:authority`, `:status`) cannot be rewritten	N/A
Rule interaction	Rules can chain	Headers added by one rule not visible to subsequent rules	Headers added by one rule not visible to subsequent rules

These behaviors reflect protocol specifications and should be considered when creating rewriting policies.

Configuration

The configuration for applying content rewriting to a Virtual Server is unchanged. You can now enable content rewriting on HTTPS-type profiles that have HTTP/3 or Backend HTTP/2 enabled. For example, the predefined profiles **LB_PROF_HTTP3** and **LB_PROF_HTTP2_END2END_H2** can be used.


Virtual Server
Basic
General
Security
SSL Traffic Mirror
Application Optimization
Monitoring
Name
testvs3
Type
Layer 7
Layer 4
Layer 2
Status
Enable
Disable
Maintain
Address Type
IPv4
IPv6
Traffic Group
default
Comments
Specify the comments
Specifics
Content Routing
Selected Items
Available Items
Content Routing List
rs5only
Create New
test07
routingSp2
routingSp3
Double-click to deselect. Drag to reorder.
Double-click to select.
Content Rewriting
Selected Items
Available Items
addHeader-example
Create New
Global_OriginalURL
redirect-example
AddHeaderNeg-example
Content Rewriting List
Double-click to deselect. Drag to reorder.
Double-click to select.
Selected Items
Available Items


Virtual Server
Basic
General
Security
SSL Traffic Mirror
Application Optimization
Monitoring
Configuration
Address
10.1.1.16
Example: 192.0.2.1
Address (IPv6)
::
Example: 2001:0db8::1
Port
443
Default: 80 Range: 0 or 1-65535. You can specify up to eight ports or port ranges separated by space, e.g., 80-90 100. Valid values are from 0 to 65535, with 0 for Layer-4 virtual servers only.
Connection Limit
0
Default: 0 Range: 0-100000000 concurrent connections
Interface
port1
Resources
Profile
LB_PROF_HTTP3
Client SSL Profile
LB_CLIENT_SSL_PROF_DEFAULT
Persistence
Click to select.
Method
LB_METHOD_ROUND_ROBIN
Real Server Pool
testing_test
Clone Pool
Click to select
Access Policy
Click to select
Virtual Server
Content Rewriting
Content Routing
NAT Source Pool
Delete
Create New
Search
Port: 443
Add Filter
Name
Type
Address
Port
Profile
Status
Availability
testvs3
Layer 7
10.1.1.16
443
LB_PROF_HTTP3
Enable
testvs2
Layer 7
10.1.1.56
443
LB_PROF_HTTP2_END2END_H2
Enable

Support Proxy Protocol for L4 TCP

FortiADC now supports Proxy Protocol v1 and v2 in Layer 4 TCP server load balancing (SLB) deployments. This enhancement allows client connection metadata—such as source IP address and port—to be preserved across NAT boundaries and forwarded to backend servers. Proxy Protocol insertion ensures that real servers can accurately log or respond to the original client source, even in NAT or multi-hop environments.

This information is also available in the FortiADC 8.0.0 Administration Guide and CLI Reference:

- Using real server pools
- config load-balance pool

Previously, Proxy Protocol was only supported in Layer 7 virtual servers. In 8.0.0, support is extended to Layer 4 TCP virtual servers configured with DNAT or Full NAT packet forwarding.

Proxy Protocol provides a standardized method to prepend connection metadata (source and destination IP/port) to the start of a TCP stream. This is especially useful in SLB environments where source NAT masks the original client IP. Supported applications—such as NGINX, Apache, and HAProxy—can extract and use this information for logging, session tracking, and access control.

FortiADC inserts the Proxy Protocol header before application data, leaving the application payload unmodified. Both IPv4 and IPv6 traffic are supported.

Note: Proxy Protocol is not defined by an IETF RFC but is widely adopted by major proxy and web server implementations.

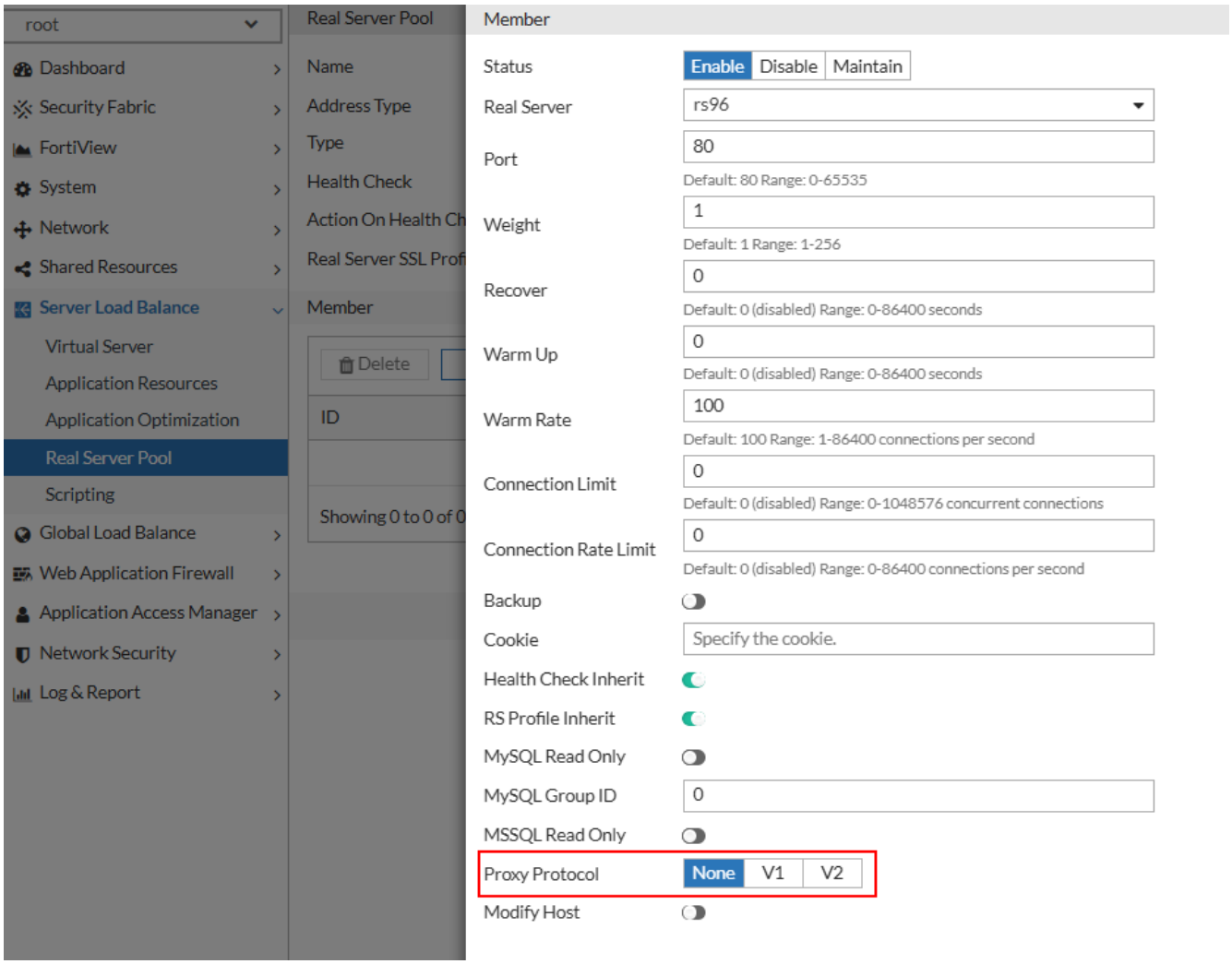
Configuring Proxy Protocol for Layer 4 TCP server load-balancing

This enhancement uses the existing Proxy Protocol configuration options, now extended to Layer 4 TCP virtual servers. Configuration is performed at the real server pool member level.

Each pool member can be individually set to one of the following modes:

- None (default) – no Proxy Protocol header is inserted
- V1 – use human-readable, newline-terminated format (Proxy Protocol v1)
- V2 – use binary-encoded format (Proxy Protocol v2)

Note: Ensure that the backend application explicitly supports and is configured to accept Proxy Protocol headers.



Example Scenario: Preserving Client IP in Full NAT Deployments

In a typical Layer 4 SLB deployment using SNAT or Full NAT, the backend server sees only the source NAT address—making it impossible to log or act on the real client IP. Enabling Proxy Protocol resolves this by including connection metadata in the TCP stream.

In this scenario, FortiADC is deployed as a Layer 4 TCP load balancer using Full NAT. The goal is to ensure the backend server (e.g., NGINX) receives the original client IP address for logging and application logic, even though SNAT is used for egress.

Setup:

- **Client IP**: 192.100.0.2
- **Virtual Server (VIP)**: 192.100.0.3
- **FortiADC SNAT IP**: 10.0.0.3
- **Real Server IP**: 10.0.0.2
- **Packet Forwarding Method**: Full NAT

- **Proxy Protocol Setting**: disabled (default) vs enabled (v1 or v2)

Without Proxy Protocol

Flow Component	IP/Port Observed
Client → FortiADC VIP	192.100.0.2:54566 → 192.100.0.3:80
FortiADC → Real Server	10.0.0.3:12345 → 10.0.0.2:80
Backend Observes	Client IP = 10.0.0.3 (SNAT IP)

FortiADC performs SNAT to forward traffic to the real server. The backend server sees only the SNAT source address (10.0.0.3), with no visibility into the original client IP.

With Proxy Protocol Enabled

Flow Component	IP/Port Observed
FortiADC Prepends	Proxy Protocol header (v1 or v2)
Backend Receives	PROXY TCP4 192.100.0.2 192.100.0.3 ... + payload
Backend Observes	Client IP = 192.100.0.2 (from Proxy header)

With Proxy Protocol enabled, FortiADC prepends the header containing client and server metadata to the first TCP packet with a payload. The backend must be explicitly configured to accept Proxy Protocol connections on the specified port. Once parsed, the backend can restore visibility into the original client IP and port.

Clear Session and Persistence Table for HTTP/S Virtual Servers

FortiADC extends session and persistence clearing functionality to Layer 7 virtual servers (VS), including those that handle HTTP, HTTPS, TCPS, and RDP traffic. Prior to this enhancement, the CLI command sets, `diagnose server-load-balance session` and `diagnose server-load-balance persistence`, supported clearing operations only for Layer 4 (TCP/UDP) virtual servers. This update extends both command sets to support Layer 7 virtual servers by introducing the l7-http keyword, which targets httproxy-managed session and persistence tables.

- **Session Clearing**: Enables targeted removal of active Layer 7 sessions using `diagnose server-load-balance session clear l7-http`. This includes support for common filter parameters such as source IP, destination IP, and virtual server name.
- **Persistence Clearing**: Enables removal of persistence entries using `diagnose server-load-balance persistence clear l7-http`. Only specific filters–such as source IP and virtual server name–are supported, depending on the underlying persistence method.

Both operations utilize the existing filter mechanism to define match criteria, enabling administrators to perform selective clearing without disrupting unrelated sessions. These capabilities enhance operational control over session management and traffic distribution, especially during maintenance or debugging.

This information is also available in the FortiADC 8.0.0 CLI Reference:

- diagnose server-load-balance session
- diagnose server-load-balance persistence

Configuring Session Clearing for HTTP/S Virtual Servers

You can clear active HTTP/S sessions using the `diagnose server-load-balance session clear l7-http` command. This action targets sessions managed by the httproxy engine and applies only to Layer 7 virtual servers. To avoid indiscriminate clearing, filters should always be applied before issuing the clear operation.

1. **Apply a session filter to target specific sessions**:

   ```
   diagnose server-load-balance session filter <criteria>
   ```

   Example:

   ```
   diagnose server-load-balance session filter vs-name VS1 source-ip 192.0.2.10
   ```

   This filter restricts matching to sessions that belong to virtual server `VS1` and originate from the client IP `192.0.2.10`.

   To verify the active filter state:

   ```
   diagnose server-load-balance session filter show
   ```

Supported filter criteria:

- `source-ip`—Single IP address or specify start and end addresses of a range.
- `source-port`—Single port number or start and end port numbers of a range.
- `dest-ip`—Single IP address or specify start and end addresses of a range.
- `dest-port`—Single port number or start and end port numbers of a range.
- `trans-source-ip`—Single IP address or specify start and end addresses of a range.
- `trans-source-port`—Single port number or start and end port numbers of a range.
- `trans-dest-ip`—Single IP address or specify start and end addresses of a range.
- `trans-dest-port`—Single port number or start and end port numbers of a range.
- `type`—Specify `ipv4`, `ipv6`, `ipv4v6`, or `ipv6v4`.
- `protocol`—Specify `tcp` or `udp`.
- `vs-name`—Specify a space-separated list of up to 8 virtual server configuration names.
- `rs-name`—Specify a space-separated list of up to 8 real server configuration names.

Note: `httproxy` sessions without a server-side connection may not populate translated address fields (e.g., `trans-dest-ip`), so those filters may have no effect on such entries.

2. **Clear the filtered sessions**:

```
diagnose server-load-balance session clear l7-http
```

This command will immediately remove all matching Layer 7 sessions. Connections are forcefully closed, and any in-progress transactions will be dropped. On systems with large session counts, this operation may introduce temporary delays or impact forwarding while `httproxy` synchronizes. Use with care during active traffic.

Configuring Persistence Clearing for HTTP/S Virtual Servers

Layer 7 HTTP/S virtual servers support multiple persistence mechanisms (e.g., Source Address, Passive Cookie, or script-based). These are managed in a shared memory table by the httproxy engine. FortiADC provides the ability to clear selected persistence entries via:

```
diagnose server-load-balance persistence clear l7-http
```

Before clearing, filters should be defined to scope the operation.

1. **Apply a persistence filter**:

```
diagnose server-load-balance persistence filter <criteria>
```

Example:

```
diagnose server-load-balance persistence filter vs-name VS1 source-ip 192.0.2.10
```

To check the current filter state:

```
diagnose server-load-balance persistence filter show
```

Supported filter criteria:

- `source-ip` – Single IP address or specify start and end addresses of a range.
- `source-port` – Single port number or start and end port numbers of a range.
- `dest-ip` – Single IP address or specify start and end addresses of a range.
- `dest-port` – Single port number or start and end port numbers of a range.
- `vs-name` – Specify a space-separated list of up to 8 virtual server configuration names.

Note: Only source IP and virtual server name filters are valid for L7 persistence clearing. Other fields are not present in the `httproxy` persistence table and will be ignored silently.

2. **Once the filters are verified, clear the filtered persistence entries**:

```
diagnose server-load-balance persistence clear l7-http
```

This operation deletes matching entries from the Layer 7 persistence table. The most common use case is clearing Source Address-based persistence entries during testing or when troubleshooting client stickiness issues.

Note: Passive Cookie and script-based persistence methods also share the same internal table but may not have externally visible identifiers in CLI output. Use caution when clearing in mixed configurations.

Support Multi-Process Mode for Up to 64 Processes per Virtual Server

FortiADC now supports up to 64 processes per virtual server in multi-process mode, significantly increasing the previous limit of 15. This enhancement is designed to improve performance and scalability on high-core platforms by allowing better distribution of traffic handling across CPU cores.

Multi-process mode enables each virtual server process to operate independently, which increases concurrency and reduces contention in multi-core systems. With the new 64-process limit, administrators can match the number of processes to the number of available CPU cores more precisely, optimizing throughput and resource utilization.

Example: On a platform with 32 CPU cores, users can now assign 32 processes to a single HTTP or HTTPS virtual server for balanced load distribution across all cores.

This information is also available in the FortiADC 8.0.0 CLI Reference:

- config load-balance virtual-server

Configuring Multi-Process Mode

This feature is available via the CLI and applies to Layer 7 and Layer 2 HTTP/HTTPS virtual servers.

```
config load-balance virtual-server
  edit <name>
    set type l7-load-balance
    set multi-process <1-64>
  next
end
```

- The `multi-process` field now accepts values from **1 to 64**.
- If the configured number exceeds the number of CPU cores, a warning is shown:

  ```
  It is recommended that the number of processes (32) does not exceed the actual CPU cores (2).
  ```

  Note: Overprovisioning process counts beyond available CPU cores may degrade performance due to increased scheduling overhead and memory usage. It is recommended to align the process count with hardware capacity and traffic demands.

Scripting Support for Persistence Functions in HTTP Data Events

FortiADC now supports calling persistence-related scripting functions—**HTTP:persist()** and **HTTP:lookup_tbl()**—within HTTP data phase events. This enhancement allows scripting-based persistence decisions to be made using values extracted from HTTP payloads, such as session tokens embedded in POST request bodies.

This information is also available in the FortiADC 8.0.0 Script Reference Guide:

- HTTP Persistence commands

Enhanced Scripting Functionality:

Function	Supported Events in FortiADC 8.0.0
HTTP:persist()	HTTP_REQUEST, HTTP_RESPONSE, HTTP_DATA_REQUEST, HTTP_DATA_RESPONSE
HTTP:lookup_tbl()	HTTP_REQUEST, HTTP_DATA_REQUEST

This update extends persistence scripting capabilities beyond header-based processing, allowing decisions to be made based on content in the HTTP body (e.g., XML/JSON tokens, session identifiers).

Application Scope

- Supported on SLB-L7 and SLB-L2 virtual servers using HTTP, HTTPS, TCPS, or RDP.
- Fully backward-compatible: existing use of HTTP:persist() and HTTP:lookup_tbl() in header-phase events remains unchanged.
- Enables advanced persistence for deployments involving:
  - Encrypted session tokens
  - XML/JSON-based payload structures
  - Dynamic routing and session stickiness requirements

Use Cases

This feature is particularly useful in scenarios where persistence must rely on application-layer payload content:

- Federated identity systems using SAML tokens or embedded session assertions
- RESTful or SOAP-based APIs carrying user or session information in the body
- Middleware applications requiring data-driven stickiness

Example: Persistence Using a Token from HTTP Payload

The following Lua script demonstrates how to extract a session ID from the body of an HTTP POST request and use it as the persistence key. If the session ID is unavailable, the client IP is used as a fallback.
This scenario is particularly useful in deployments where tokens or session identifiers are embedded deep in request payloads (e.g., SAML, SOAP, or REST-based APIs).

```
-- Collect full payload during request phase
when HTTP_REQUEST {
    local t = {}
    HTTP:collect(t)
}

-- Inspect and process the payload in the data phase
when HTTP_DATA_REQUEST {
    local cip = IP:client_addr()
    local t = { operation = "content" }
    local body_str = HTTP:payload(t)

    local persist_key = nil
    if body_str ~= nil then
        local sessionID = extract_id(body_str, "ssoSessionId xsi:type=", 35, 32)
        if sessionID then
            debug("Extracted session ID: %s\n", sessionID)
            persist_key = sessionID
        end
    else
        debug("No data payload detected in request\n")
    end

    if persist_key == nil then
        persist_key = cip
    end

    -- Calculate server assignment from hashed key
    t = {
        operation = "cal_server_from_hash",
        hash_value = sha1_hex(persist_key)
    }
    local rs = HTTP:persist(t)

    if rs then
        debug("Mapped hash to server: %s → %s\n", t.hash_value, rs)

        -- Save hash-to-server mapping for session affinity
        t = {
            operation = "save_tbl",
            hash_value = sha1_hex(persist_key),
            srv_name = rs
        }
        local ret = HTTP:persist(t)
        if ret then
```

```
            debug("Saved mapping: %s → %s\n", t.hash_value, rs)
        else
            debug("Failed to save mapping: %s → %s\n", t.hash_value, rs)
        end
    else
        debug("No server resolved for hash: %s\n", t.hash_value)
    end
}
```

RFC 7919 Compliance Support for TLS 1.3 in SSL Profiles

FortiADC now supports the **RFC 7919 Comply** option when **TLS 1.3** is selected in the allowed SSL versions of **Client SSL Profiles** and **Real Server SSL Profiles**. This enhancement resolves a previous limitation where enabling RFC 7919 compliance would result in a configuration error if SSLv3 or TLS 1.3 was also selected.

RFC 7919 defines the use of **Ephemeral Diffie-Hellman (DHE) key exchange parameters** that are fixed and standardized to improve the security and interoperability of TLS handshakes. Prior to this enhancement, FortiADC restricted the use of RFC 7919 Comply to earlier TLS versions only, excluding TLS 1.3. With this update:

- TLS 1.3 is now fully compatible with the RFC 7919 Comply setting.
- The system will no longer block configurations that include both TLS 1.3 and RFC 7919 Comply.
- SSLv3 remains unsupported due to its inherent security weaknesses and deprecation.

This enhancement ensures standards compliance and improves compatibility with modern security policies that require the use of predefined DHE parameters in TLS 1.3 sessions.

Global Load Balance

The FortiADC 8.0 release includes new features and enhancements in **Global Load Balance**:

New Secondary Zone Type with Secure AXFR Synchronization via TSIG Authentication 8.0.1 on page 159

FortiADC introduces a new **Secondary zone type** to expand its DNS role beyond primary-only operation. Previously, FortiADC could act only as a primary DNS server, serving zone data to other secondaries. With this enhancement, it can also function as a secondary, synchronizing its DNS zone data from an upstream primary server using **AXFR (Authoritative Zone Transfer)**. To enable secure synchronization, this release also adds support for **TSIG (Transaction SIGnature) authentication**, ensuring that AXFR transfers and NOTIFY messages are validated and accepted only from trusted servers. Together, these enhancements provide a more flexible, interoperable, and secure foundation for Global Server Load Balancing (GSLB).

User-Defined Certificates and CA Verification for GSLB 8.0.1 on page 164

FortiADC now supports user-defined certificates and peer certificate verification for Global Server Load Balancing (GSLB). This enhancement strengthens authentication between GLB and SLB, mitigates man-in-the-middle (MITM) risks, and enables integration with enterprise PKI infrastructures. It also extends cipher suite support to include FIPS-compliant options, ensuring compliance with stricter security requirements.

New Secondary Zone Type with Secure AXFR Synchronization via TSIG Authentication 8.0.1

FortiADC introduces a new **Secondary zone type** to expand its DNS role beyond primary-only operation. Previously, FortiADC could act only as a primary DNS server, serving zone data to other secondaries. With this enhancement, it can also function as a secondary, synchronizing its DNS zone data from an upstream primary server using **AXFR (Authoritative Zone Transfer)**. To enable secure synchronization, this release also adds support for **TSIG (Transaction SIGnature) authentication**, ensuring that AXFR transfers and NOTIFY messages are validated and accepted only from trusted servers. Together, these enhancements provide a more flexible, interoperable, and secure foundation for Global Server Load Balancing (GSLB).

This information is also available in the FortiADC 8.0.1 Administration Guide and CLI Reference:

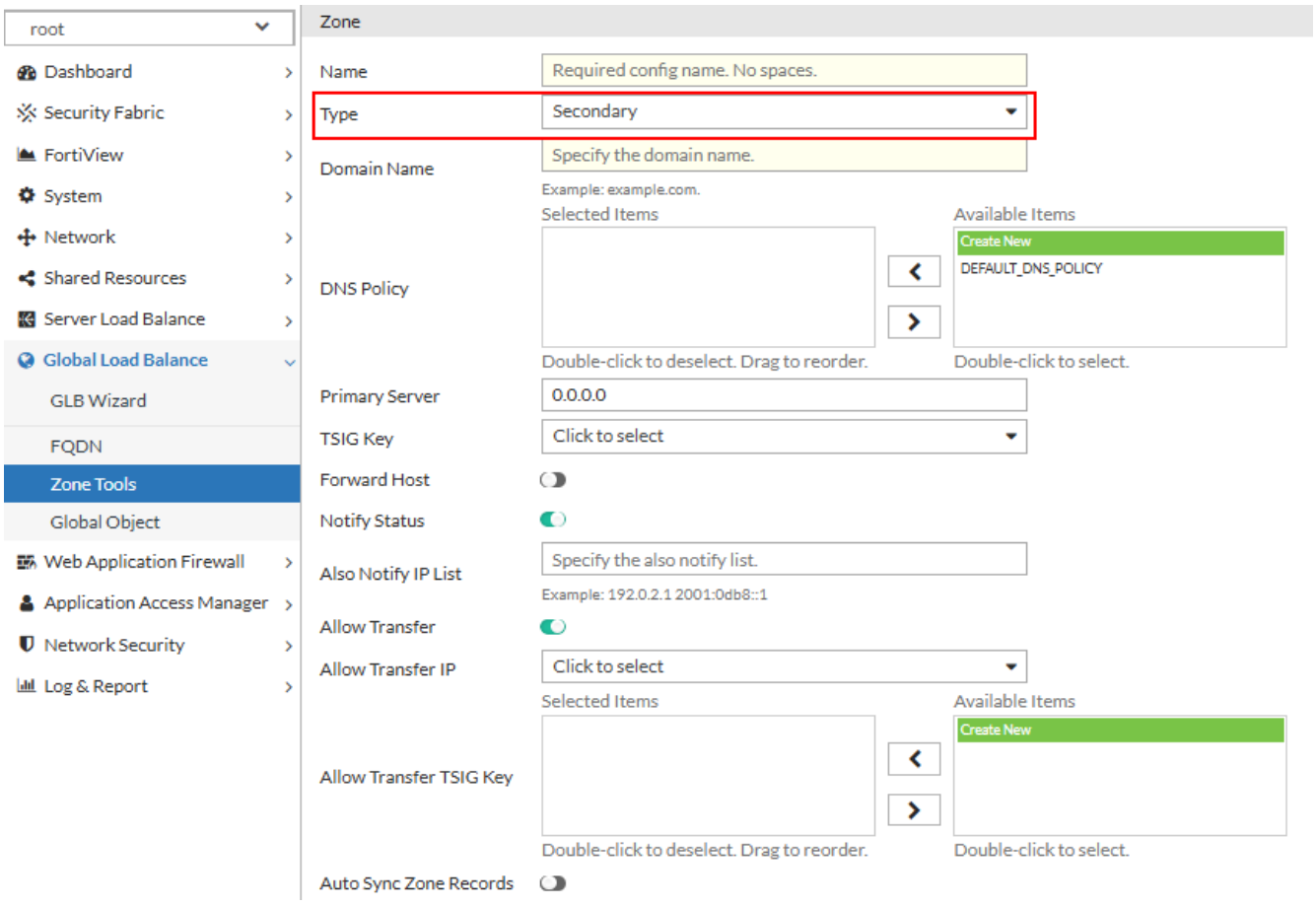
- Configuring DNS zones
- Managing TSIG Keys
- config global-dns-server zone
- config global-dns-server tsig-key

AXFR is the standard DNS protocol for replicating entire zone files from a primary server to one or more secondaries. By supporting both outbound AXFR (when FortiADC is a primary) and inbound AXFR (when FortiADC is a secondary), FortiADC ensures that DNS zone data remains consistent across servers. Synchronization is triggered either through scheduled SOA checks or immediately upon receiving a NOTIFY message from the primary.

TSIG authentication adds a security layer to this process. TSIG uses shared secret keys to sign AXFR and NOTIFY traffic, preventing unauthorized or spoofed servers from injecting false data. In FortiADC, administrators import TSIG key files through Zone Tools > TSIG Key and assign them to zones to enforce authenticated synchronization. This approach ensures that DNS transfers are accepted only from trusted peers, maintaining both data integrity and operational security.

New Secondary Zone Type

The new **Secondary** type allows FortiADC to operate as a synchronized replica of an upstream Primary DNS server. Instead of defining records locally, Secondary zones pull their content from the Primary through AXFR transfers.



FortiADC initiates synchronization in two ways:

- By sending periodic SOA queries to the Primary. If the serial number is newer than the local copy, FortiADC requests a full AXFR transfer. The refresh interval is based on the SOA record, with built-in minimum (300 seconds) and maximum (4 weeks) limits.
- By responding to **NOTIFY messages** from the Primary. When a NOTIFY is received, FortiADC verifies it, checks the SOA serial, and requests an AXFR if the zone has been updated.

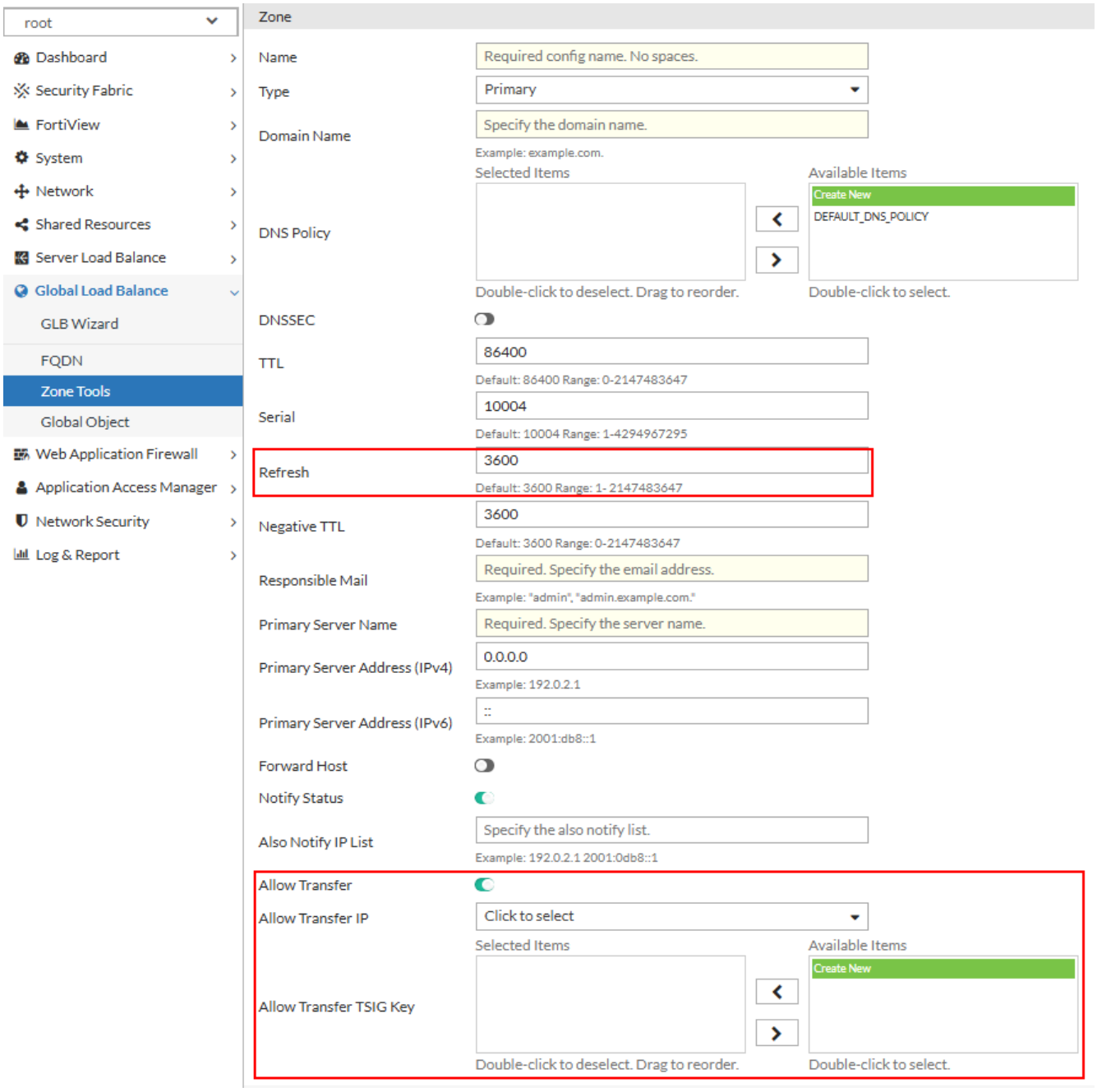
Secondary zones are strictly read-only. Administrators cannot add or edit records directly. The option **Auto Sync Zone Records** controls whether records are applied automatically when transfers occur:

- **Enabled**: synchronized records are written to the configuration.
- **Disabled**: existing records are cleared, and new records from the Primary are not applied.

Only FortiADC-supported record types are synchronized. Unsupported types are ignored during AXFR, ensuring that the zone remains usable but may not include every record type present on the Primary.

This implementation ensures that FortiADC can participate as a secondary in standard DNS hierarchies while maintaining consistency with upstream authoritative servers. It allows Global Server Load Balancing (GSLB) responses to be based on accurate, up-to-date DNS data without requiring FortiADC to be the authoritative Primary for every zone.

Updates to Primary Zone Type configuration



When FortiADC operates as a Primary DNS server, it remains responsible for originating and serving authoritative zone data to secondaries. In this release, the configuration model has been enhanced with finer-grained controls over how transfers are permitted and authenticated. These changes improve security, interoperability with external secondaries, and administrative flexibility.

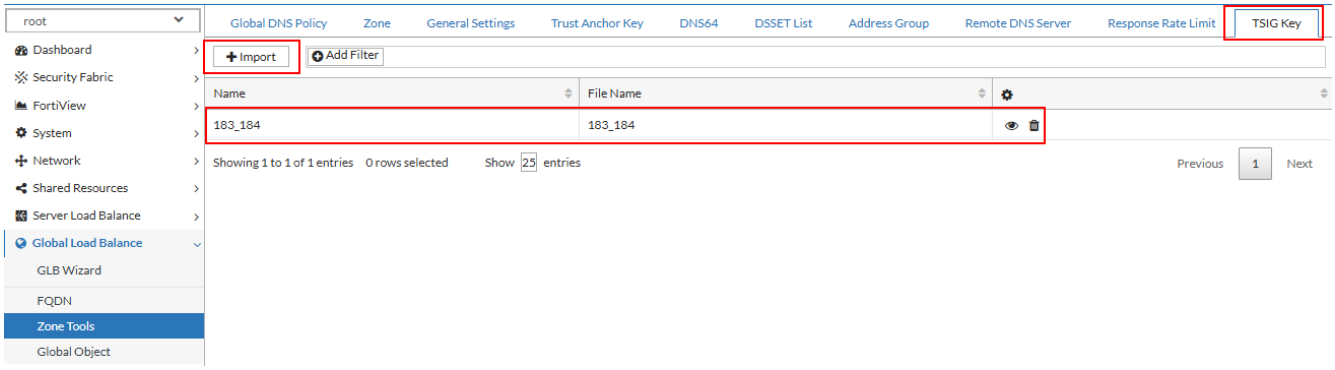
- **Refresh** defines the interval at which secondaries query FortiADC's SOA record to check for changes, giving administrators explicit control over update frequency.

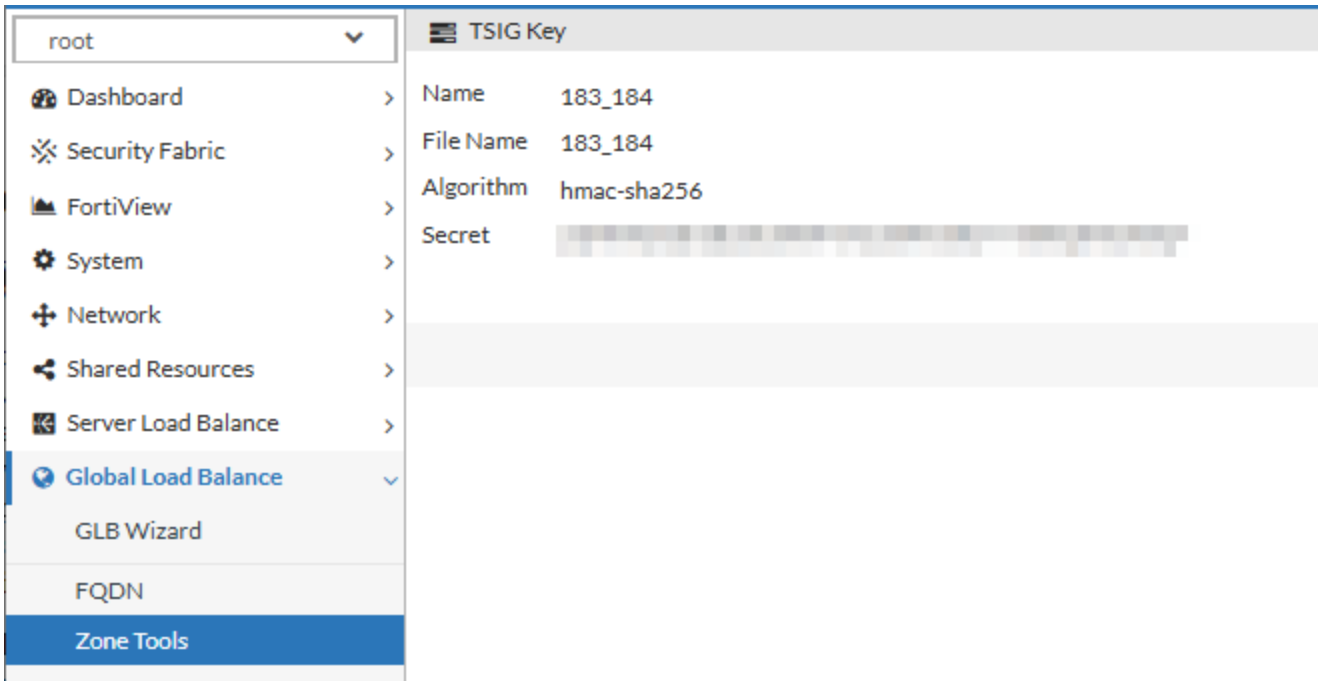
- **Allow Transfer** is now an **Enable/Disable toggle** (enabled by default), replacing the previous Any/None setting. When enabled, FortiADC will respond to zone transfer requests from secondaries.
- **Allow Transfer IP** lets administrators specify which IP addresses are permitted to initiate transfers, providing basic access control.
- **Allow Transfer TSIG Key** allows one or more imported TSIG keys to be assigned, enabling cryptographic authentication of transfer requests. If no key is specified, transfers are validated only against the IP list.

These updates strengthen security and interoperability when FortiADC acts as a Primary server. However, note that FortiADC does not support incremental transfers (IXFR), multiple primaries, or AXFR over TLS.

TSIG Key Management

To support authenticated synchronization, FortiADC introduces a dedicated **TSIG Key** tab under **Global Load Balance > Zone Tools**. This tab lists imported TSIG key files along with their properties, such as filename, description, and algorithm.





TSIG keys play different roles depending on zone type:

- **Primary zones**: TSIG keys can be assigned under Allow Transfer TSIG Key to ensure that only secondaries configured with the same key are able to request zone transfers. This prevents unauthorized servers from replicating FortiADC's zone data.
- **Secondary zones**: TSIG keys can be assigned under TSIG Key to authenticate transfers and NOTIFY messages received from the upstream Primary. This ensures that FortiADC accepts data only from trusted sources.

Key management rules:

- Keys are created automatically when a valid TSIG key file is imported.
- Keys cannot be created or modified manually; the system enforces consistency with the imported file.
- A key in use by any zone cannot be deleted. Attempts to delete such a key return an error.
- Deleting a key removes the underlying file from the system, permanently disabling its use.
- TSIG algorithms supported include HMAC-MD5, HMAC-SHA1, HMAC-SHA224, HMAC-SHA256, HMAC-SHA384, and HMAC-SHA512.

By tying TSIG keys directly to zone configurations, FortiADC ensures that every AXFR transfer and NOTIFY exchange can be cryptographically validated. This prevents spoofed synchronization attempts and strengthens trust relationships between FortiADC and external DNS servers.

User-Defined Certificates and CA Verification for GSLB 8.0.1

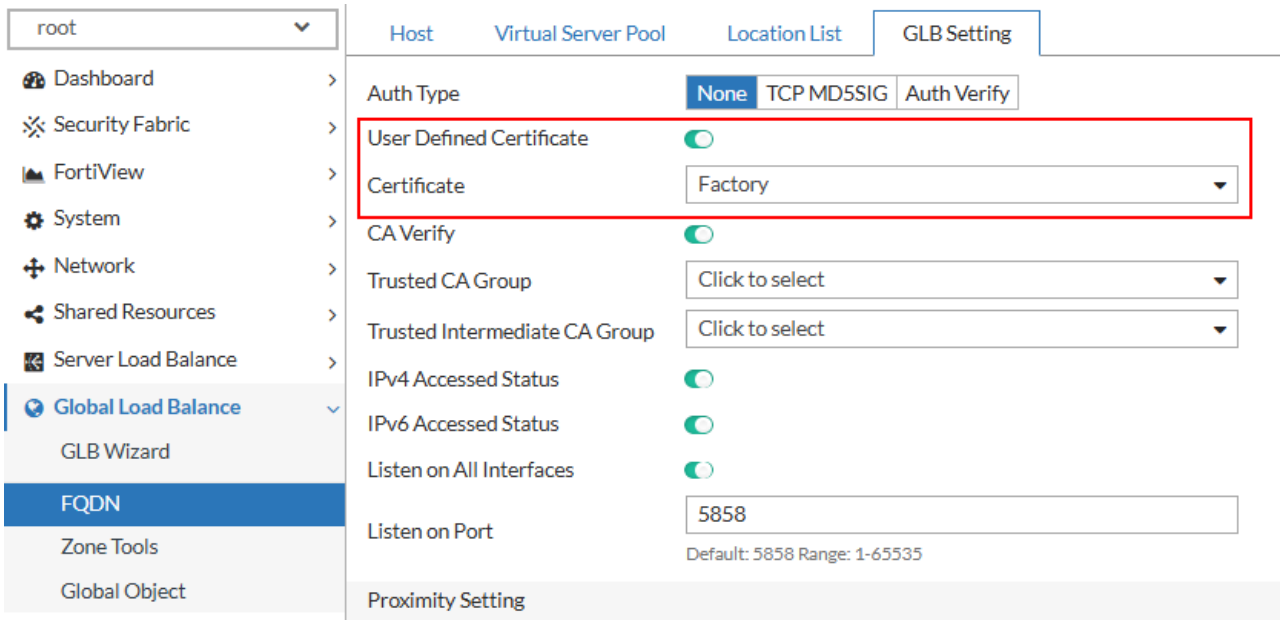
FortiADC now supports user-defined certificates and peer certificate verification for Global Server Load Balancing (GSLB). This enhancement strengthens authentication between GLB and SLB, mitigates man-in-the-middle (MITM) risks, and enables integration with enterprise PKI infrastructures. It also extends cipher suite support to include FIPS-compliant options, ensuring compliance with stricter security requirements.

This information is also available in the FortiADC 8.0.1 Administration Guide and CLI Reference:

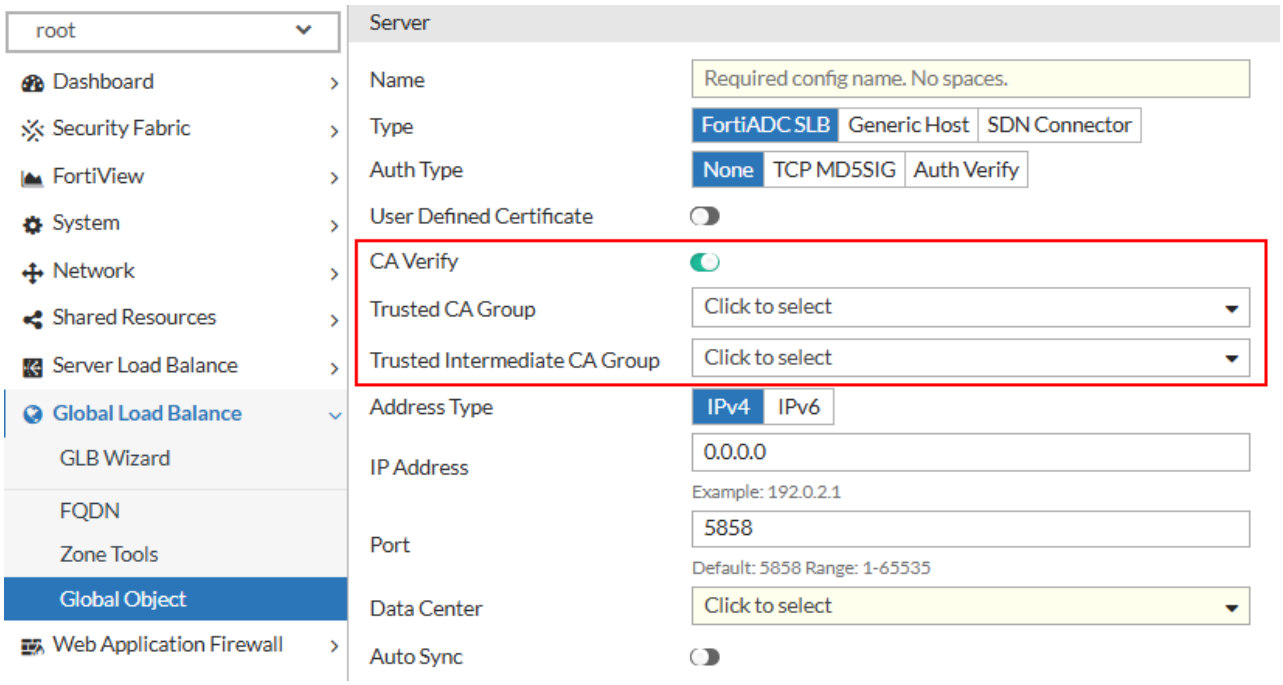
- Configuring GLB settings
- Configuring servers
- config global-load-balance setting
- config global-load-balance servers

Enhancements are introduced in these two **Global Load Balance** configuration areas:

- **FQDN > GLB Setting** –administrators can enable **User Defined Certificate** and select an uploaded certificate for the SLB to present during TLS handshakes with the GLB. By default, FortiADC uses its built-in certificate, but this option allows the use of a certificate signed by a corporate CA or another trusted issuer. In VDOM deployments, each VDOM can only select certificates within its own scope.



- **Global Object > Server** – administrators can enable **CA Verify** and assign trusted CA and intermediate CA groups to validate the SLB's certificate chain. The GLB validates the SLB's presented certificate against the configured CA and intermediate CA groups. Certificate chains are validated according to X.509 path length constraints defined in RFC 5280, with OpenSSL supporting a maximum verification depth of nine levels.



When combined, these options allow the SLB to present a user-defined certificate while the GLB enforces strict certificate validation, providing a complete trust model for secure GSLB communication.

How they work together

The two configurations are independent but complementary. If only **User Defined Certificate** is enabled, the SLB presents the selected certificate, but it will not be verified unless **CA Verify** is also enabled. If only **CA Verify** is enabled, the GLB verifies the default SLB certificate. For secure deployments, both should be enabled so that the SLB presents a user-defined certificate and the GLB validates it against trusted CA groups.

Cipher suite support

This feature extends supported cipher suites for GLB-SLB communication. In addition to the existing defaults (AES256-SHA256, ECDHE-ECDSA-AES256-GCM-SHA384), FIPS-compliant suites such as DHE-RSA-AES128-SHA256 and DHE-RSA-AES128-SHA are supported.

Network Security

The FortiADC 8.0 release includes new features and enhancements in **Network Security**:

CLI Commands to Manage TCP DoS Block List 8.0.1 on page 167

FortiADC introduces two new CLI commands to manage entries in the TCP DoS block list:

- `execute dos get tcp-block-list` displays source IPs currently blocked by a DoS profile, along with source port, destination, and remaining block time.
- `execute dos release tcp-block-list` removes entries from the block list, either by source IP or all at once.

These commands apply specifically to **Layer 4 DoS protections** that use the **Period Block** action, including **TCP access flood protection** and **TCP slow-data attack protection**. When these protections detect excessive or abnormal connection behavior, offending source IPs are temporarily blocked for the configured duration.

Source IP Exception Support for Networking DoS Protections on page 168

FortiADC 8.0.0 introduces a new DoS **Exceptions** configuration feature that enables source IP-based exclusion in Networking-type DoS protection profiles. This enhancement allows administrators to define trusted IPv4 addresses that should bypass specific DoS inspection mechanisms. This feature only supports IPv4 TCP traffic.

CLI Commands to Manage TCP DoS Block List 8.0.1

FortiADC introduces two new CLI commands to manage entries in the TCP DoS block list:

- `execute dos get tcp-block-list` displays source IPs currently blocked by a DoS profile, along with source port, destination, and remaining block time.
- `execute dos release tcp-block-list` removes entries from the block list, either by source IP or all at once.

These commands apply specifically to **Layer 4 DoS protections** that use the **Period Block** action, including **TCP access flood protection** and **TCP slow-data attack protection**. When these protections detect excessive or abnormal connection behavior, offending source IPs are temporarily blocked for the configured duration.

This information is also available in the FortiADC 8.0.1 CLI Reference:

- execute dos get tcp-block-list
- execute dos release tcp-block-list

In earlier releases, administrators could not view or clear blocked entries before the timer expired. The new commands provide real-time visibility and manual control over the TCP DoS block list, allowing administrators to inspect and release blocked IPs as needed.

With this enhancement, administrators can:

- **Inspect** the current block list, including source IP, source port, destination, and time remaining.
- **Release** individual IPs to restore access without waiting for the block period to expire.
- **Clear** the entire block list immediately during troubleshooting or recovery.

Configuration

```
execute dos get tcp-block-list
```

Displays up to 1000 blocked entries with source IP, source port, destination, and time remaining.

```
execute dos release tcp-block-list <source-ip>
```

Removes the specified source IP from the block list.

```
execute dos release tcp-block-list all
```

Clears all entries from the block list.

Source IP Exception Support for Networking DoS Protections

FortiADC 8.0.0 introduces a new DoS **Exceptions** configuration feature that enables source IP-based exclusion in Networking-type DoS protection profiles. This enhancement allows administrators to define trusted IPv4 addresses that should bypass specific DoS inspection mechanisms. This feature only supports IPv4 TCP traffic.

This information is also available in the FortiADC 8.0.0 Administration Guide and CLI Reference:

- Configuring DoS Exceptions
- Configuring an IP Fragmentation Protection policy
- Configuring a TCP SYN Flood Protection policy
- Configuring a TCP Slow Data Flood Protection policy
- Configuring a TCP Connection Access Flood Protection policy
- config security dos exception
- config security dos ip-fragmentation-protection
- config security dos tcp-synflood-protection
- config security dos tcp-slowdata-attack-protection
- config security dos tcp-access-flood-protection

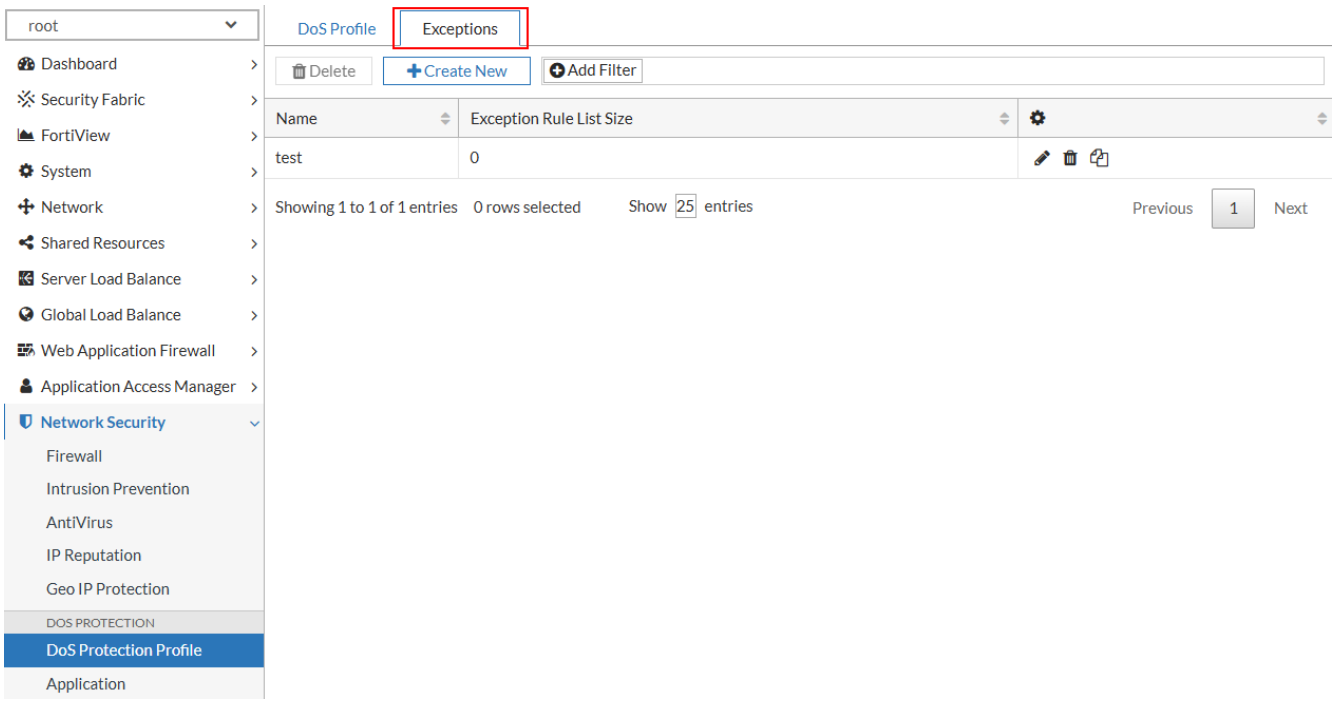
DoS Exceptions can now be applied to the following protection types:

- IP Fragmentation Protection
- TCP SYN Flood Protection
- TCP Slow Data Flood Protection
- TCP Connection Access Flood Protection

During packet inspection, if traffic matches a configured DoS protection criterion, and the packet's source IP also matches an entry in the assigned exception rule, the packet is exempt from DoS inspection and forwarded normally. If there is no match, standard DoS protection logic is applied, and traffic that exceeds the configured thresholds is dropped.

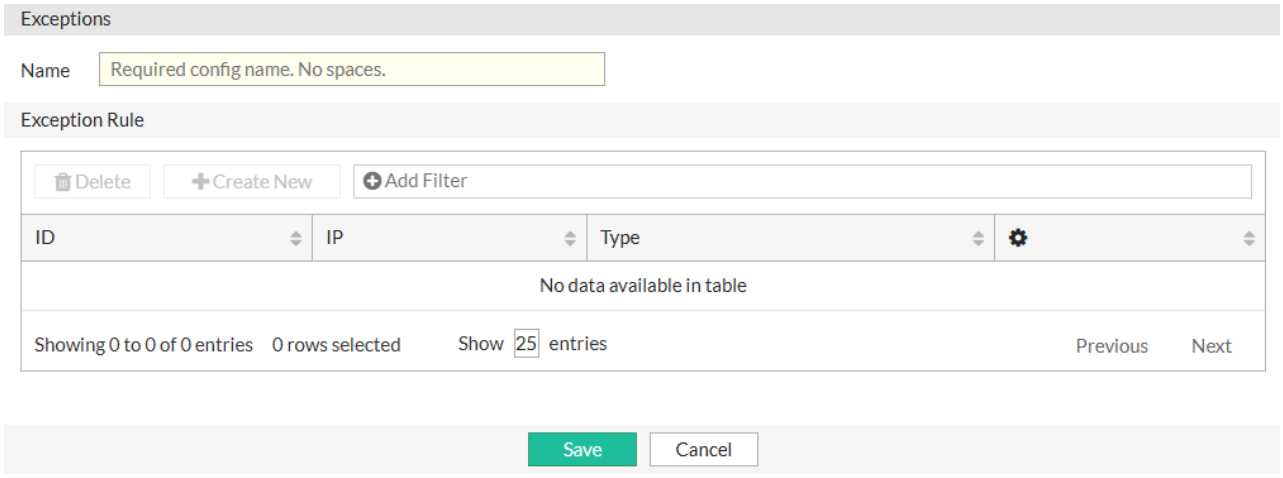
Configuring DoS Exceptions

The new **Exceptions** tab has been added under **Network Security > DoS Protection Profile** in the GUI. From this interface, administrators can define exception rules and assign them to applicable Networking-type DoS protections.



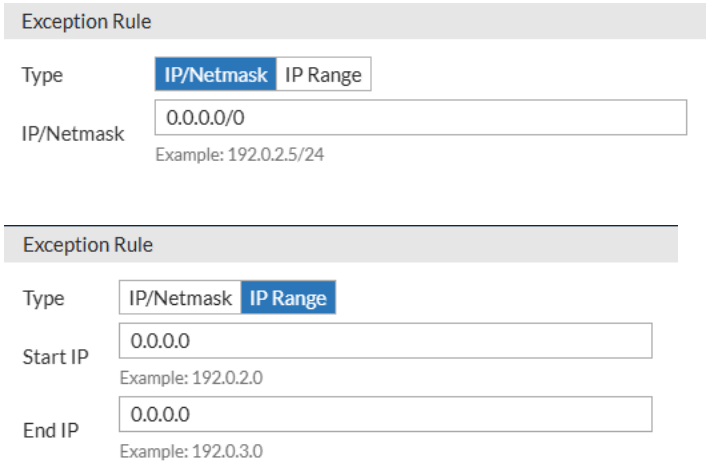
To configure DoS Exceptions from the GUI:

1. Navigate to **Network Security > DoS Protection Profile**.
   The configuration page displays the **DoS Profile** tab.
2. Click the **Exceptions** tab.
3. Click **Create New** to display the configuration editor.



4. In the **Name** field, specify a unique name for the DoS Exceptions configuration object. Valid characters are `A-Z`, `a-z`, `0-9`, `_`, and `-`. No space is allowed.
5. Click **Save**.
   Once the **Exceptions** is created, the **Exception Rule** section becomes configurable.

6. Under the Exception Rule section, click **Create New** to display the configuration editor.



7. Configure the following Exception Rule settings:

Setting	Description
Type	Specifies the format used to define the exception source. • **IP/Netmask** - Defines a network or host using CIDR notation. This is the default option. • **IP Range** - Defines a range of individual IP addresses.
IP/Netmask	The **IP/Netmask** option is available when **Type** is **IP/Netmask**. Defines the exception source using CIDR format. Example: 192.0.2.5/24 Default: 0.0.0.0/0
Start IP	The **Start IP** option is available when **Type** is **IP Range**. Specifies the beginning of the IP address range to exclude from DoS inspection. Example: 192.0.2.0 Default: 0.0.0.0
End IP	The **End IP** option is available when **Type** is **IP Range**. Specifies the end of the IP address range to exclude from DoS inspection. Example: 192.0.3.0 Default: 0.0.0.0

8. Click **Save** to apply the configuration and close the Exception Rule dialog. You can add up to 256 Exception Rules to a single DoS Exception configuration.
9. Click **Save** to apply all changes to the DoS Exceptions configuration and its rule settings.

To configure DoS Exceptions from CLI:

```
config security dos exception
  edit <name>
    config  exception-rule
```

```
      edit <No.>
        set type {ip-netmask|ip-range}
        set ip-network <IPv4/netmask>
        set start-ip <IPv4 address>
        set end-ip <IPv4 address>
      next
    end
  next
end
```

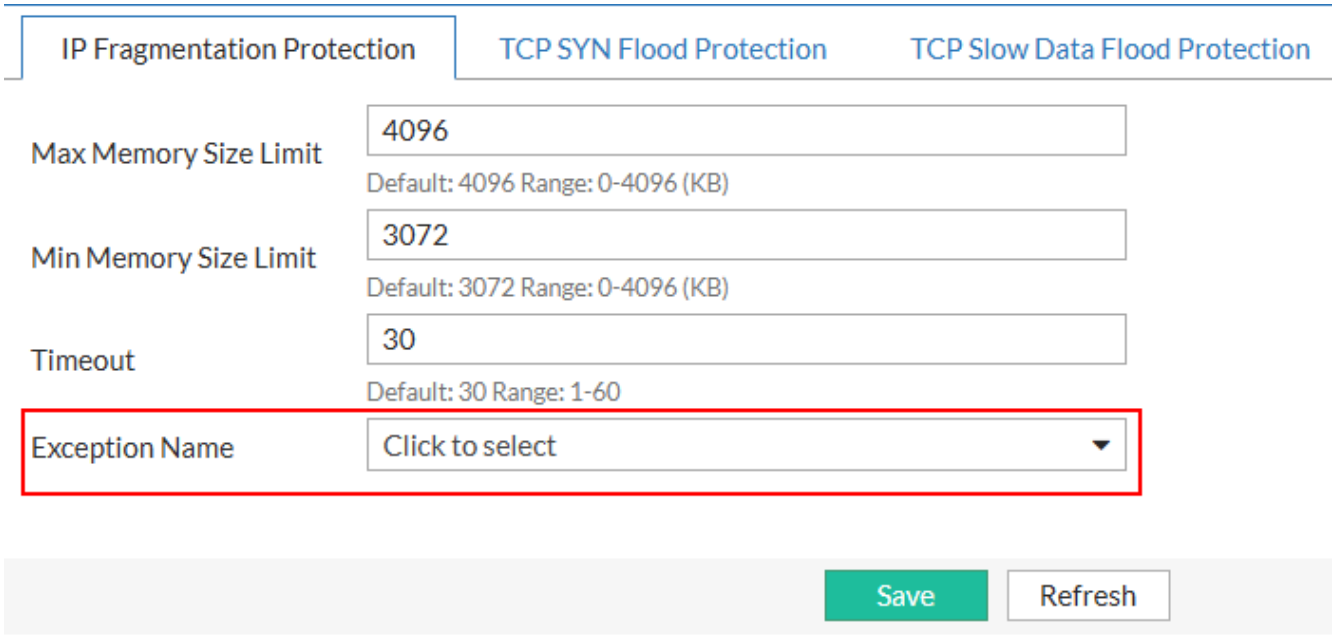
Applying Exceptions to DoS Protection Policies

Once an exception rule is created, it can be assigned directly to a specific DoS protection policy. Within each Networking-type DoS protection configuration, use the Exception Rule field to select the appropriate rule from the drop-down list. This associates the rule with the policy, allowing matching source IPs to bypass inspection.

Only one exception rule can be assigned per policy. To update or remove the association, edit the corresponding DoS protection policy and modify the Exception Rule field as needed.

The following sections explain how exception rules are applied to each supported Networking-type DoS protection policy, with examples to illustrate their behavior.

IP Fragmentation Protection

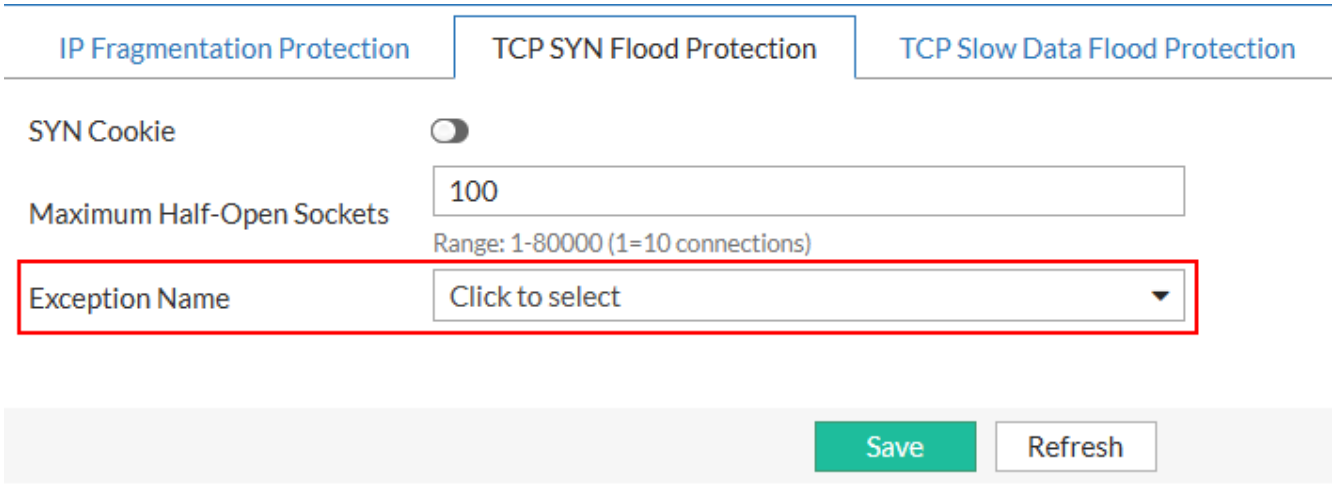


When the memory usage for fragmented packets reaches the configured Max Memory Size limit, FortiADC stops reassembling fragments and drops new fragmented traffic. However, if the source IP of a fragmented packet matches an exception rule, FortiADC continues to accept and forward the packet, bypassing the memory enforcement restriction.

Example:

When Memory Size is set to `0`, FortiADC drops all new fragmented packets by default. However, if a packet's source IP matches an entry in the exception rule—such as `119.1.1.111`—it will still be accepted and reassembled, bypassing the memory restriction.

TCP SYN Flood Protection

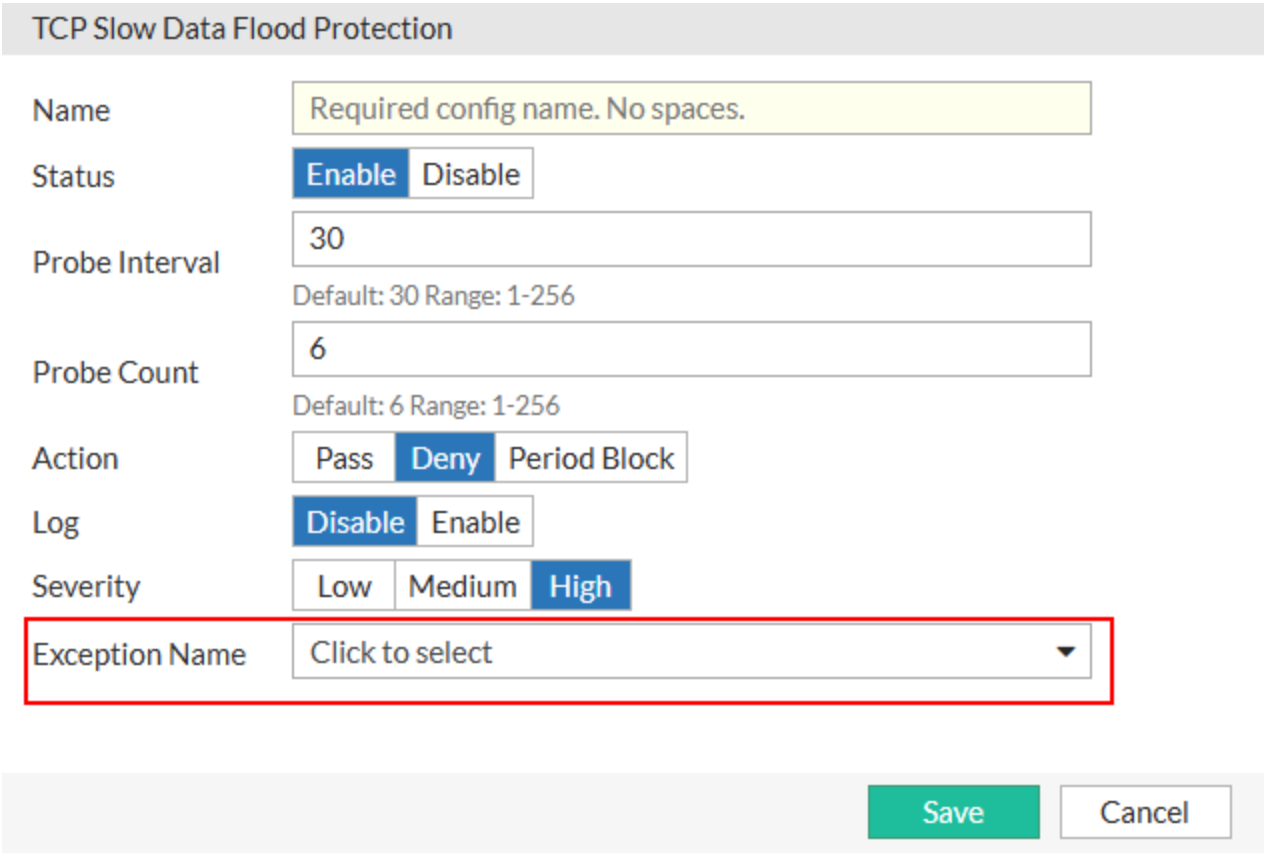


During periods of high SYN packet rates, FortiADC enables SYN Cookie protection to mitigate SYN flood attacks. If the source IP of a new connection matches the configured exception rule, SYN Cookie is not enforced and the connection proceeds normally.

Example:

If the average SYN rate exceeds the **Maximum Half-Open Sockets** threshold, FortiADC applies SYN Cookies to new connections. However, SYN packets from `119.1.1.111` will bypass this mechanism when the IP is part of the exception rule.

TCP Slow Data Flood Protection

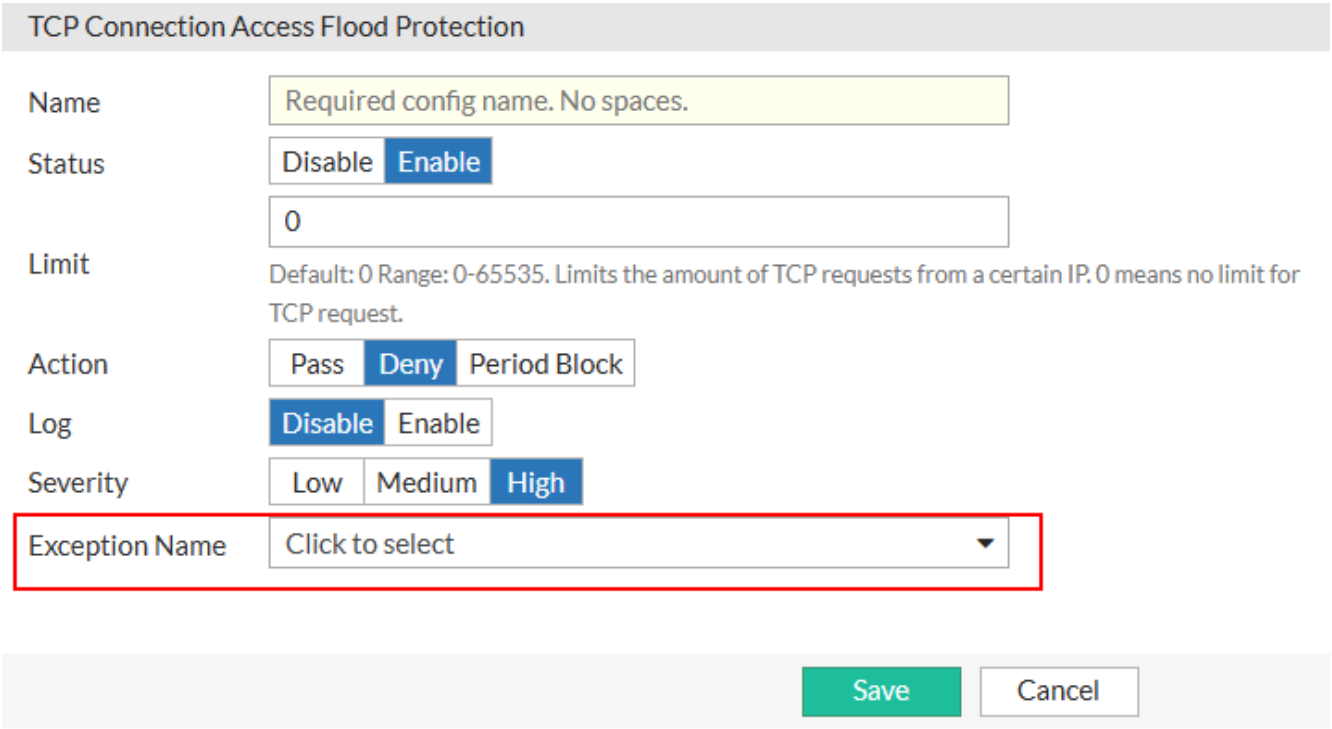


When FortiADC detects TCP sessions exhibiting slow data transfer behavior, it triggers the TCP Slow Data Flood Protection mechanism to prevent resource exhaustion. However, connections from source IPs listed in the exception rule are exempt from this detection and are allowed to proceed uninterrupted.

Example:

A client sending slow data from IP `119.1.1.111` is permitted if the IP is included in the exception rule. If the exception is removed, future slow data packets from this IP will be dropped.

TCP Connection Access Flood Protection



This policy enforces a per-source IP connection limit. When a source IP is included in the exception rule, FortiADC excludes it from enforcement and allows the IP to exceed the defined connection threshold.

Example:

If the concurrent connection limit is reached, traffic from `119.1.1.111` is still accepted as long as it is part of the exception rule. Once the IP is removed, connections are subject to the policy's limits.

Log & Report

The FortiADC 8.0 release includes new features and enhancements in **Log & Report**:

Security Log GUI Redesign and Enhancements 8.0.3 on page 176

Following the updates to the Traffic and Script logs, FortiADC 8.0.3 brings a complete redesign to the **Security Logs**. This update aligns the security interface with the modern, high-performance standard used across the platform while adding specialized investigation tools such as an **Analyze with AI button**, a **Log Details side panel**, and **one-click Policy Exceptions**.

Script Log Enhancement 8.0.3 on page 181

Following the redesign of the Traffic Log interface in version 8.0.1, FortiADC 8.0.3 extends these modern GUI enhancements to the **Script Logs**. This update ensures a consistent user experience across different log types, aligning the Script Log interface with the streamlined, high-performance layout used for traffic analysis.

Traffic Log Enhancement 8.0.1 on page 182

The **Traffic Log** page has been completely redesigned to make log investigation faster, more flexible, and more intuitive. The new interface enables administrators to analyze large datasets without interruption, quickly isolate events using dynamic filters, and customize the log view to focus on the most relevant metrics. These enhancements streamline routine monitoring and accelerate troubleshooting across all traffic log types.

Security Log GUI Redesign and Enhancements 8.0.3

Following the updates to the Traffic and Script logs, FortiADC 8.0.3 brings a complete redesign to the **Security Logs**. This update aligns the security interface with the modern, high-performance standard used across the platform while adding specialized investigation tools such as an **Analyze with AI button**, a **Log Details side panel**, and **one-click Policy Exceptions**.

Standardized Log Experience

The Security Log now shares the same intuitive navigation and analysis features found in other redesigned log modules:

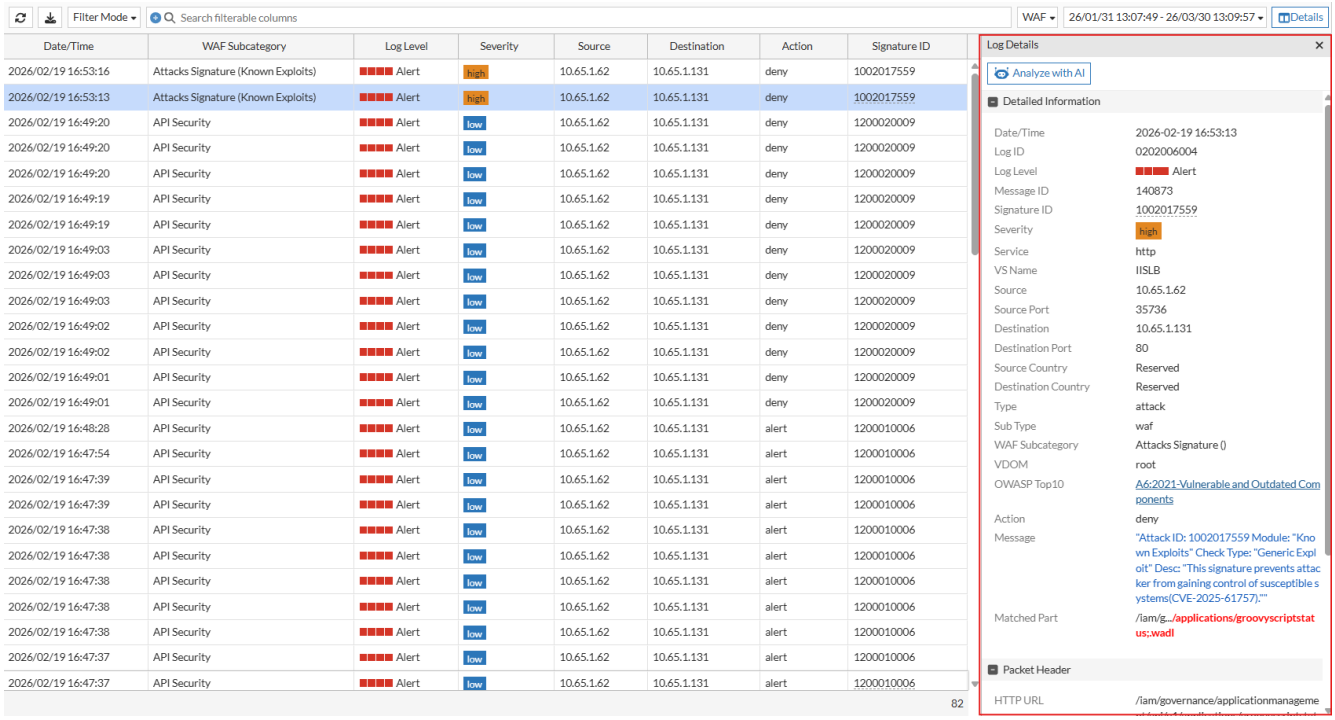
- **Continuous Data Loading**: The log table uses infinite scrolling to automatically load extensive threat datasets without the need for manual page navigation.
- **Unified Search and Dynamic Filtering**: A centralized filter bar supports stacking multiple criteria to isolate specific attack types or sources instantly.
- **Integrated Timestamp Display**: Date and time information are combined into a single column for cleaner chronological analysis.
- **Customizable Table Layout**: Administrators can show, hide, or reorder columns to prioritize the security metrics most relevant to their environment.

Specialized Security Enhancements

Beyond general alignment, the Security Log includes several features specifically designed for threat hunting and rapid response:

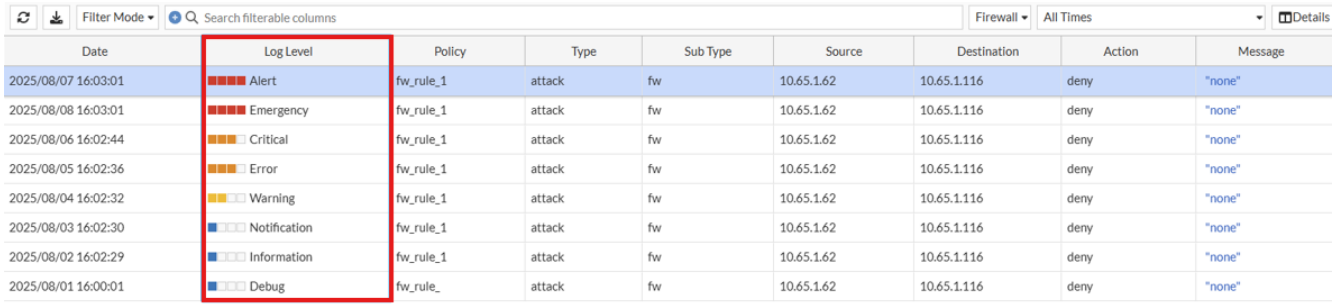
Contextual Log Details Side Panel

You can double-click a log entry or click the **Log Details** button to open an expanded side panel that provides a comprehensive overview of session information and attack specifics. This panel utilizes enhanced color coding to better highlight and differentiate critical log content, such as identifying specific attack signatures and outcomes at a glance.



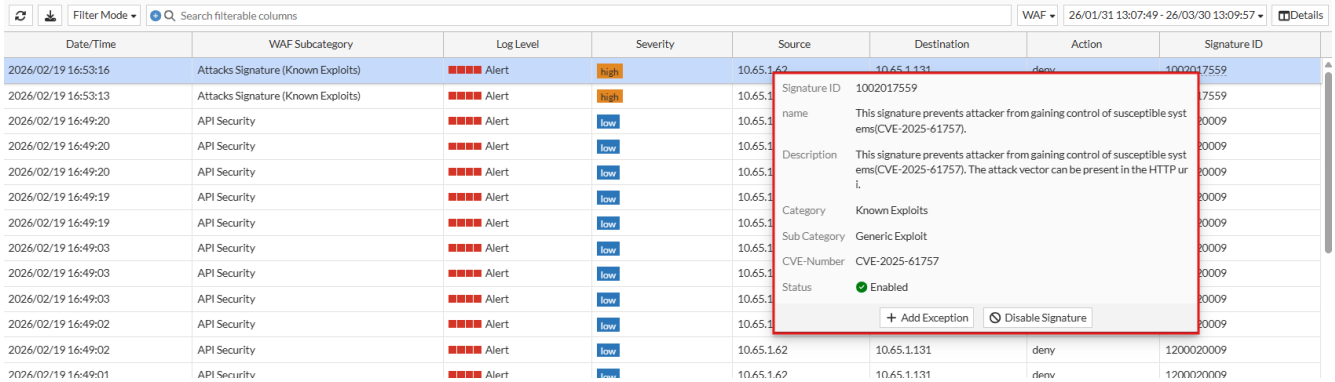
Log Level Visual Indicators

The Log Level column now uses a progress-bar style indicator where different colored blocks represent the severity of the log, allowing for quick visual assessment of threats.



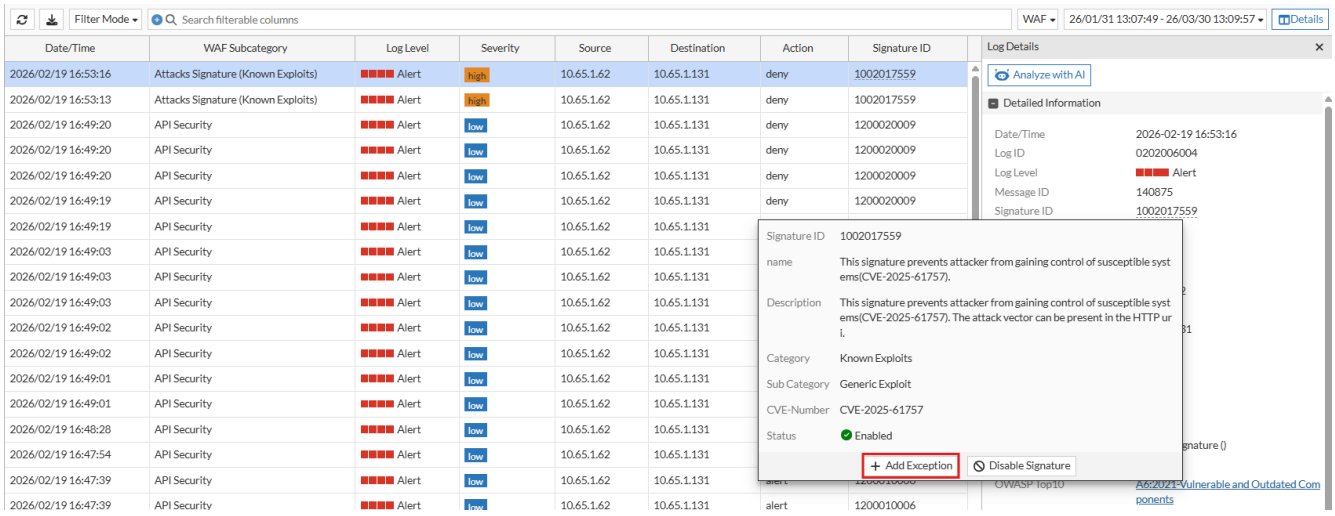
Signature ID Pop-out

You can hover over a Signature ID within the log table or details panel to open a pop-out window for a detailed view of the specific threat signature.



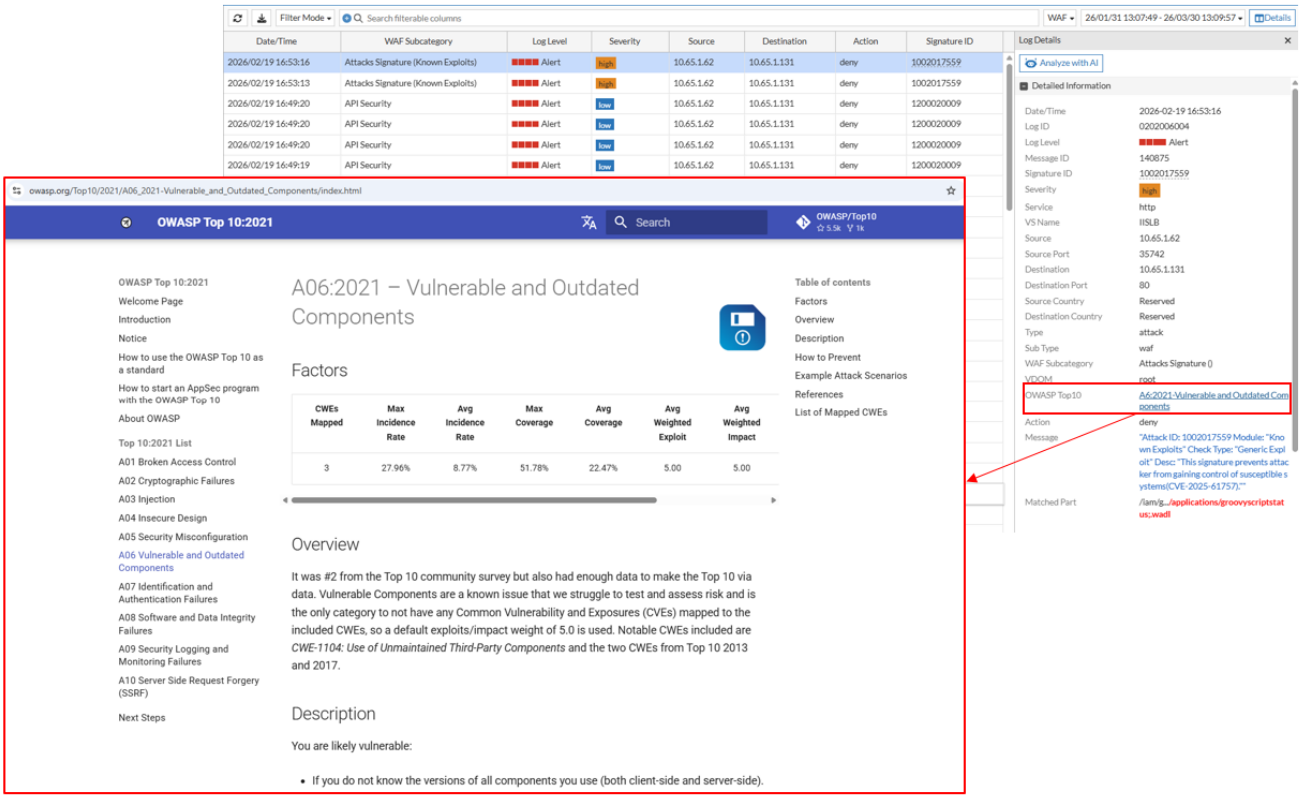
Add Exception Button

Within both the Log Details panel and the Signature ID Pop-out, an "Add Exception" button allows you to instantly create WAF or Signature exceptions to streamline policy tuning and reduce false positives.



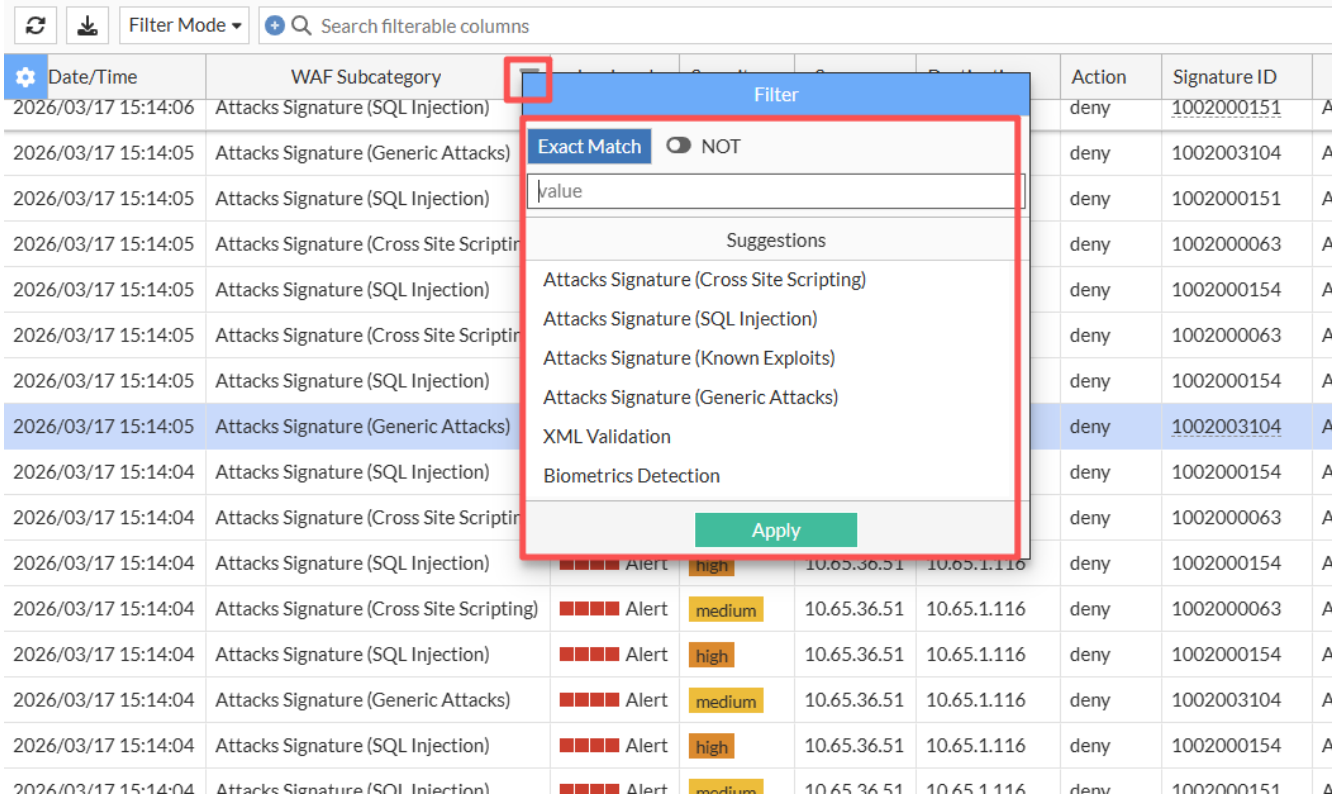
OWASP Top 10 Redirection

Security logs associated with standard vulnerabilities now include links that redirect you to the relevant OWASP Top 10 documentation for deeper context.



Enhanced Column Filtering

The **Configure Table** controls allow for advanced visibility management, including the ability to filter logs based on specialized security columns such as **WAF subcategory**.


Filter Mode
Search filterable columns
Date/Time
WAF Subcategory
Action
Signature ID
Filter
Exact Match
NOT
value
Suggestions
Attacks Signature (Cross Site Scripting)
Attacks Signature (SQL Injection)
Attacks Signature (Known Exploits)
Attacks Signature (Generic Attacks)
XML Validation
Biometrics Detection
Apply
2026/03/17 15:14:06 Attacks Signature (SQL Injection) deny 1002000151
2026/03/17 15:14:05 Attacks Signature (Generic Attacks) deny 1002003104
2026/03/17 15:14:05 Attacks Signature (SQL Injection) deny 1002000151
2026/03/17 15:14:05 Attacks Signature (Cross Site Scriptin deny 1002000063
2026/03/17 15:14:05 Attacks Signature (SQL Injection) deny 1002000154
2026/03/17 15:14:05 Attacks Signature (Cross Site Scriptin deny 1002000063
2026/03/17 15:14:05 Attacks Signature (SQL Injection) deny 1002000154
2026/03/17 15:14:05 Attacks Signature (Generic Attacks) deny 1002003104
2026/03/17 15:14:04 Attacks Signature (SQL Injection) deny 1002000154
2026/03/17 15:14:04 Attacks Signature (Cross Site Scriptin deny 1002000063
2026/03/17 15:14:04 Attacks Signature (SQL Injection) Alert high 10.65.36.51 10.65.1.116 deny 1002000154
2026/03/17 15:14:04 Attacks Signature (Cross Site Scripting) Alert medium 10.65.36.51 10.65.1.116 deny 1002000063
2026/03/17 15:14:04 Attacks Signature (SQL Injection) Alert high 10.65.36.51 10.65.1.116 deny 1002000154
2026/03/17 15:14:04 Attacks Signature (Generic Attacks) Alert medium 10.65.36.51 10.65.1.116 deny 1002003104
2026/03/17 15:14:04 Attacks Signature (SQL Injection) Alert high 10.65.36.51 10.65.1.116 deny 1002000154

Script Log Enhancement 8.0.3

Following the redesign of the Traffic Log interface in version 8.0.1, FortiADC 8.0.3 extends these modern GUI enhancements to the **Script Logs**. This update ensures a consistent user experience across different log types, aligning the Script Log interface with the streamlined, high-performance layout used for traffic analysis.

Standardized Log Experience

The Script Log page now shares the same intuitive navigation and investigation tools recently introduced to the platform:

- **Continuous Data Loading**: The log table now utilizes infinite scrolling, automatically loading additional records as you scroll through large datasets without requiring manual page navigation.
- **Unified Search and Dynamic Filtering**: A new centralized filter bar allows you to stack multiple search criteria to isolate specific script events quickly.
- **Integrated Timestamp Display**: Chronological analysis is simplified by combining date and time information into a single, easy-to-read column.
- **Context-Aware Details Panel**: Double-clicking any script log entry opens an expanded side panel, providing a comprehensive view of the script's execution details and associated metrics.
- **Customizable Table Layout**: You can now show, hide, or reorder columns to focus specifically on the script parameters most relevant to your troubleshooting.
- **Redesigned Action Controls**: Toolbar icons for downloading, refreshing, and configuring the table have been standardized and repositioned for improved accessibility.

These enhancements are part of an ongoing initiative to provide a uniform, efficient workflow for monitoring and troubleshooting across all FortiADC log modules.

Traffic Log Enhancement 8.0.1

The **Traffic Log** page has been completely redesigned to make log investigation faster, more flexible, and more intuitive. The new interface enables administrators to analyze large datasets without interruption, quickly isolate events using dynamic filters, and customize the log view to focus on the most relevant metrics. These enhancements streamline routine monitoring and accelerate troubleshooting across all traffic log types.

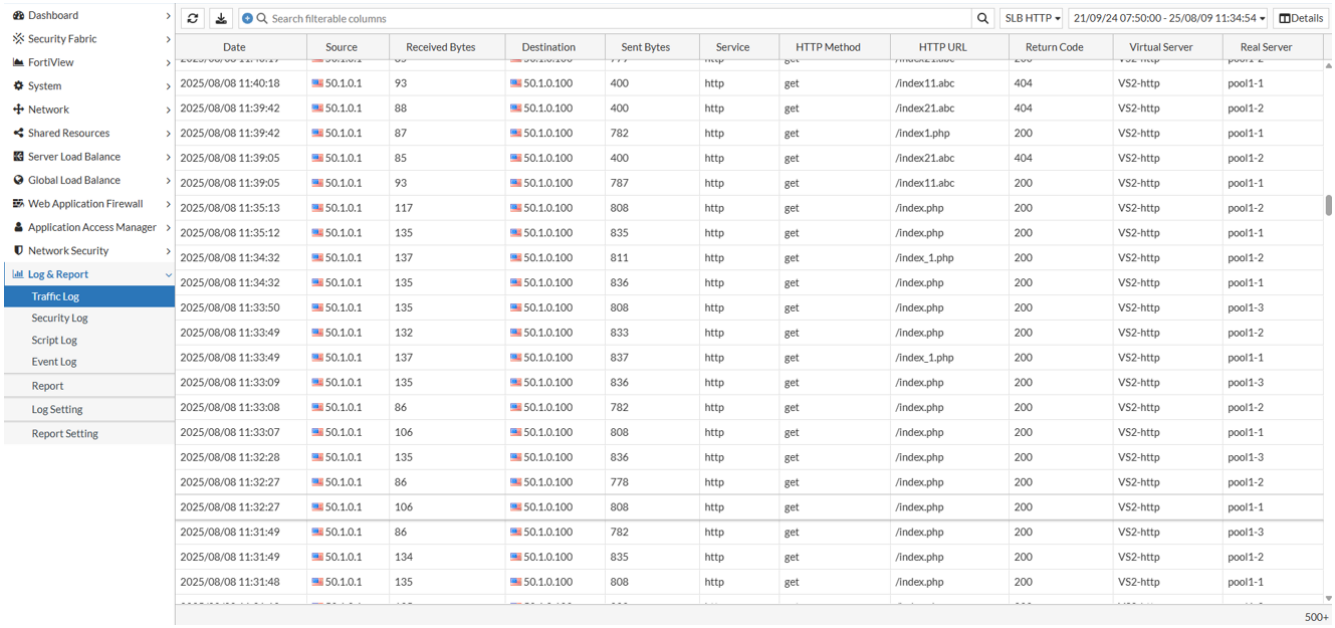
This information is also available in the FortiADC 8.0.1 Administration Guide:

- Using the traffic log

Highlights

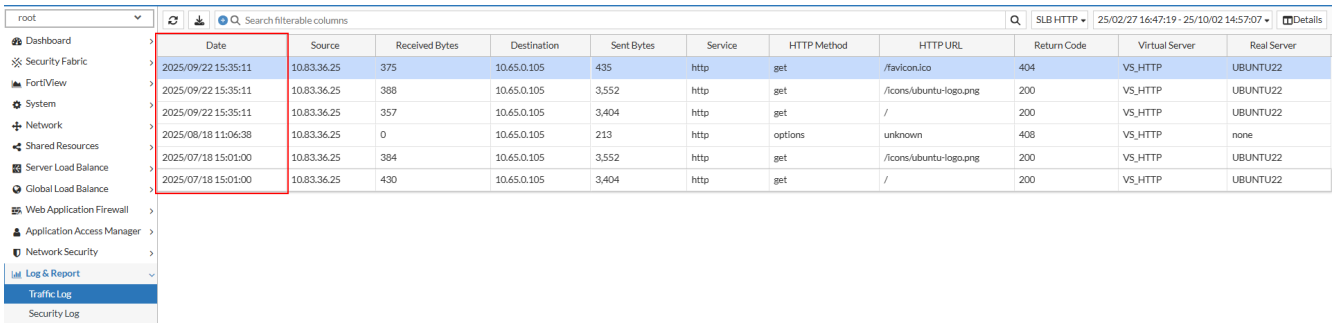
Continuous data loading

The log table now uses infinite scrolling, allowing administrators to browse extensive datasets without page reloads or navigation buttons. When more than 500 entries are available, the table automatically loads additional records as you scroll.



Integrated timestamp display

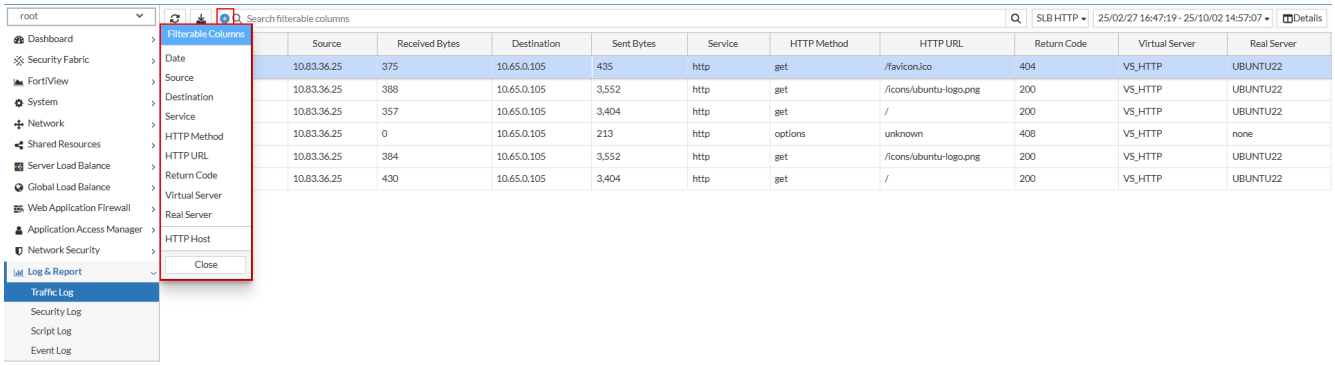
Date and time information are now combined into a single column for cleaner presentation and easier chronological analysis.



Targeted search and dynamic filters

A new unified search and filter bar lets you isolate specific sessions or traffic types instantly.

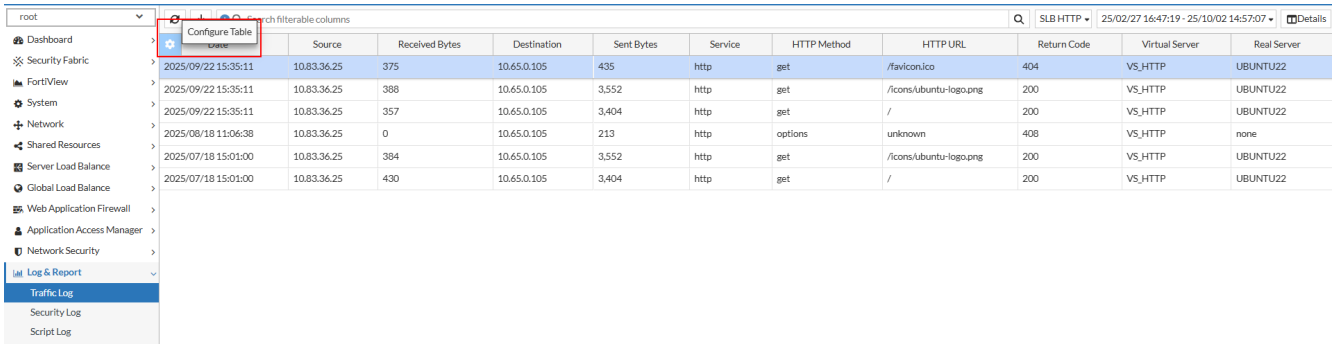
- Filters can include additional, subtype-specific fields such as **HTTP Host** for SLB HTTP logs.
- Multiple filters can be stacked, modified, or cleared without leaving the page.

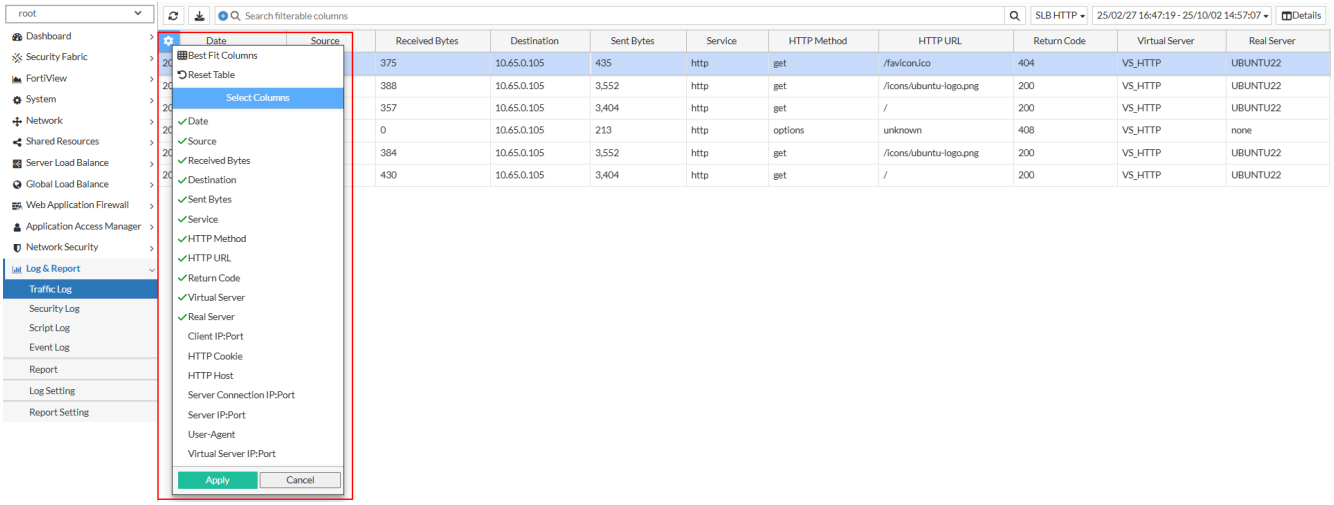


Customizable table layout

The **Configure Table** control lets you show or hide columns and save a focused view that highlights only the metrics relevant to your analysis.

- Use **Reset Table** or **Best Fit Columns** to quickly restore optimal sizing.
- Each traffic log subtype provides its own column set for tailored visibility.

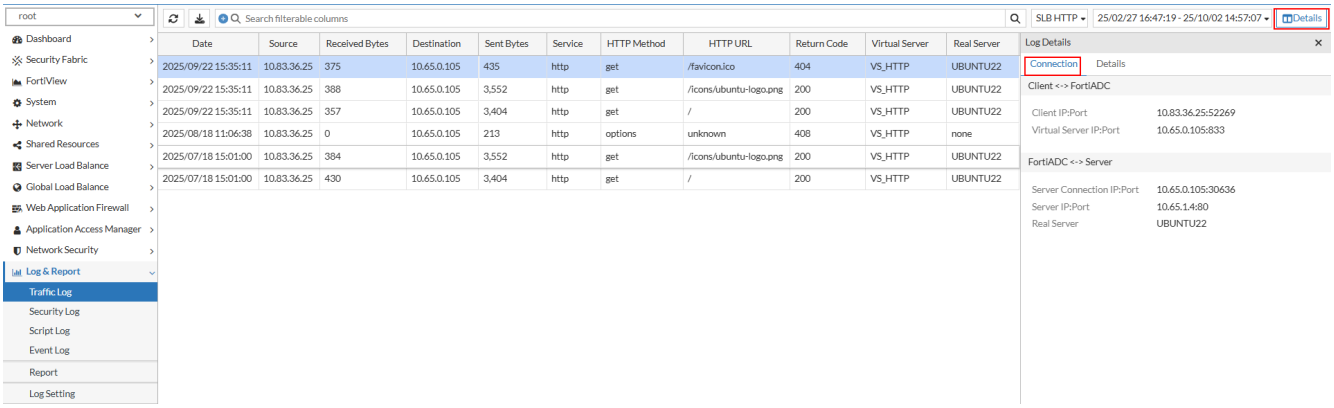




Context-aware log details

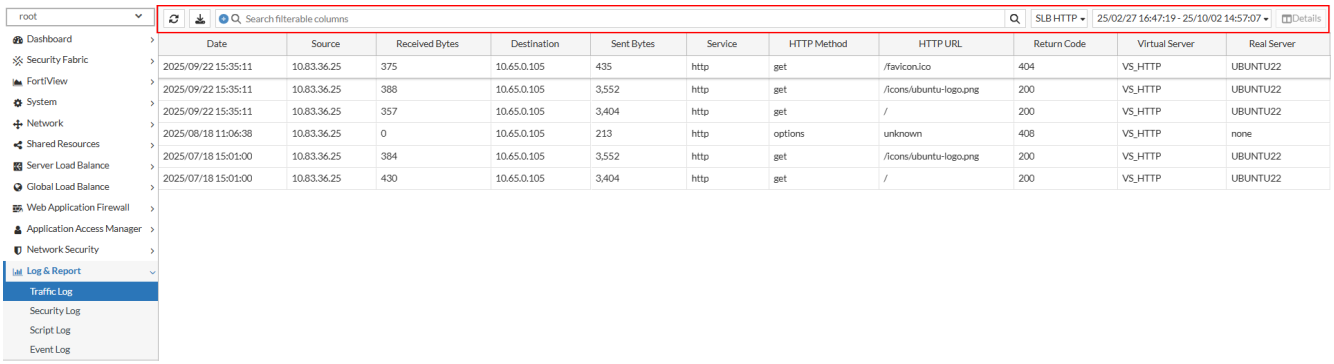
Double-click any entry or select it and click **Details** to open an expanded side panel containing session information and metrics.

- For certain subtypes, a **Connection** tab displays additional per-connection attributes such as source, destination, and session statistics.
- GLB and LLB logs do not include the Connection tab.



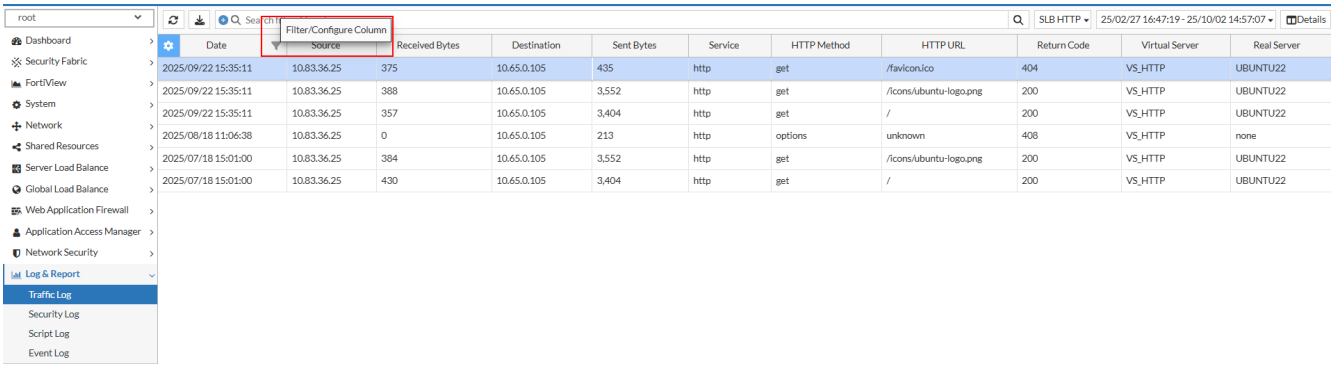
Redesigned action controls

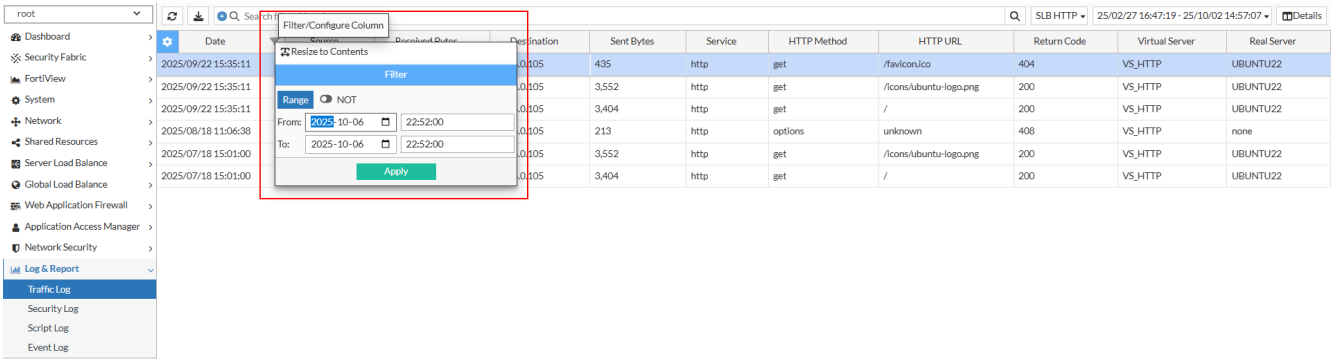
Common actions such as **Download** and **Refresh** have been repositioned and redesigned for improved accessibility and a cleaner interface. Control labels and icons have been standardized across log types for consistent operation.



Enhanced toolbar organization

The new combined **Filter / Configure Columns** control consolidates visibility and filtering options in a single location. This unified toolbar reduces clutter, keeps essential tools within reach, and provides a smoother workflow for viewing, filtering, and exporting logs.





GUI

The FortiADC 8.0 release includes new features and enhancements in the **GUI**:

Enhanced User Interface and Workflow Reorganization 8.0.3 on page 187

You can now navigate a more intuitive and streamlined management interface designed to enhance workflow efficiency and configuration consistency. FortiADC 8.0.3 introduces a large-scale reorganization of the management interface, refining navigation across the **System**, **Server Load Balance**, **Global Load Balance**, and **Web Application Firewall** menus. These updates simplify complex configurations, better align with policy creation flows, and centralize security-first settings for more efficient daily management.

Updated Navigation Menu Structure on page 193

FortiADC8.0.3 reorganizes the GUI navigation layout to improve usability and align feature groupings with functional domains. Key configuration areas—including user authentication modules and DoS protection settings—have been relocated under new parent menus to reflect their expanded scope and associated feature enhancements.

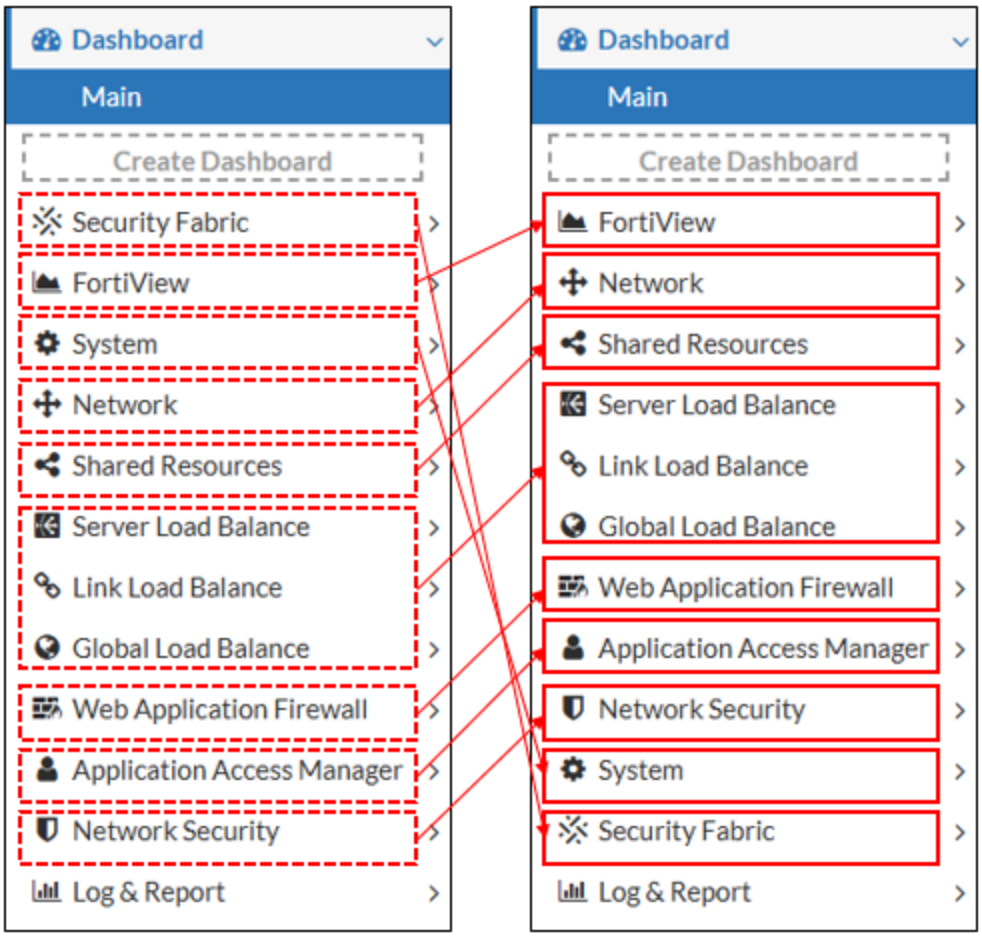
Enhanced User Interface and Workflow Reorganization 8.0.3

You can now navigate a more intuitive and streamlined management interface designed to enhance workflow efficiency and configuration consistency. FortiADC 8.0.3 introduces a large-scale reorganization of the management interface, refining navigation across the **System**, **Server Load Balance**, **Global Load Balance**, and **Web Application Firewall** menus. These updates simplify complex configurations, better align with policy creation flows, and centralize security-first settings for more efficient daily management.

Navigation Menu Reordering

The main navigation menu has been reordered to prioritize high-traffic operational modules and group security and system management at the bottom of the stack. This shift moves **Network** and **Load Balance** functions to the top for faster access, while moving **System** and **Security Fabric** below the primary application delivery modules.

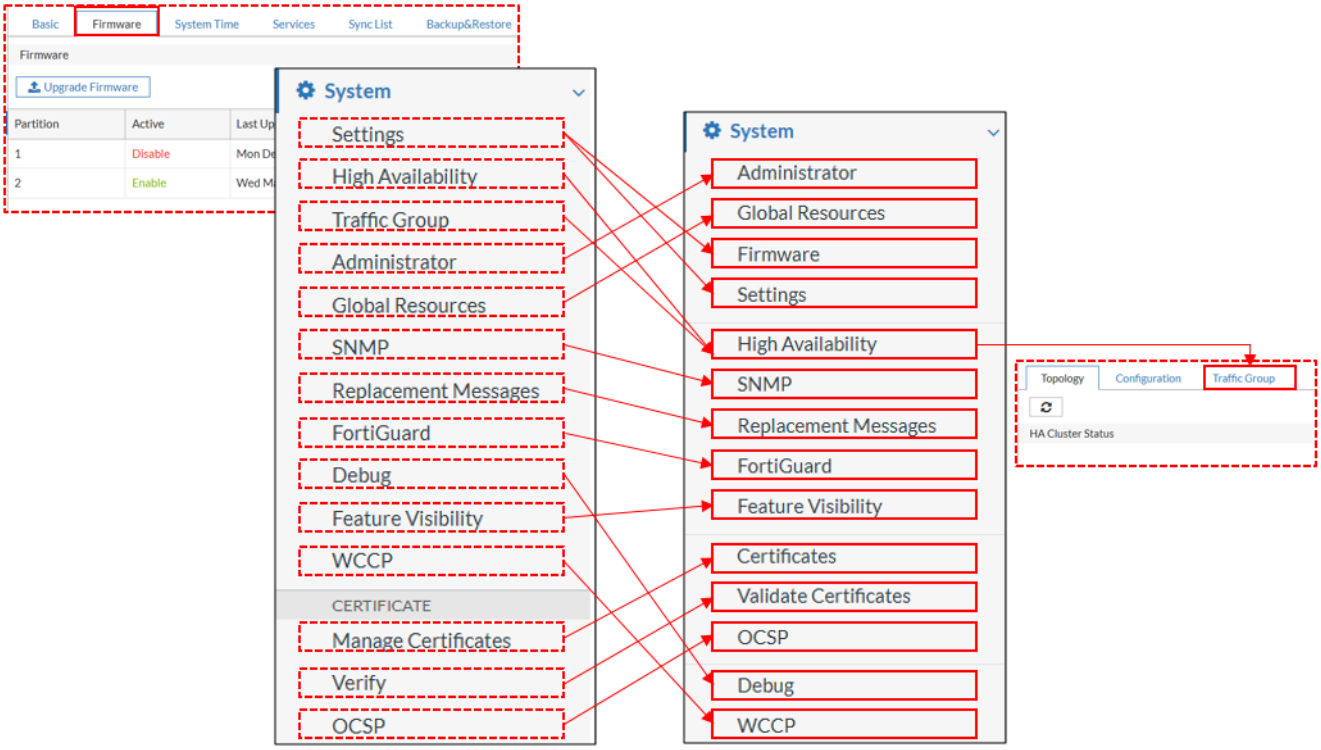
- **Operational Priority**: **Network**, **Shared Resources**, and all **Load Balance** modules are now positioned at the top of the navigation pane.
- **Security Integration**: **Web Application Firewall**, **Application Access Manager**, and **Network Security** follow the core load balancing features.
- **Administrative Foundation**: **System**, **Security Fabric**, and **Log & Report** are now grouped at the base of the menu.



System Menu Enhancements

The **System** menu has been restructured to provide a more professional solution and a consistent user experience.

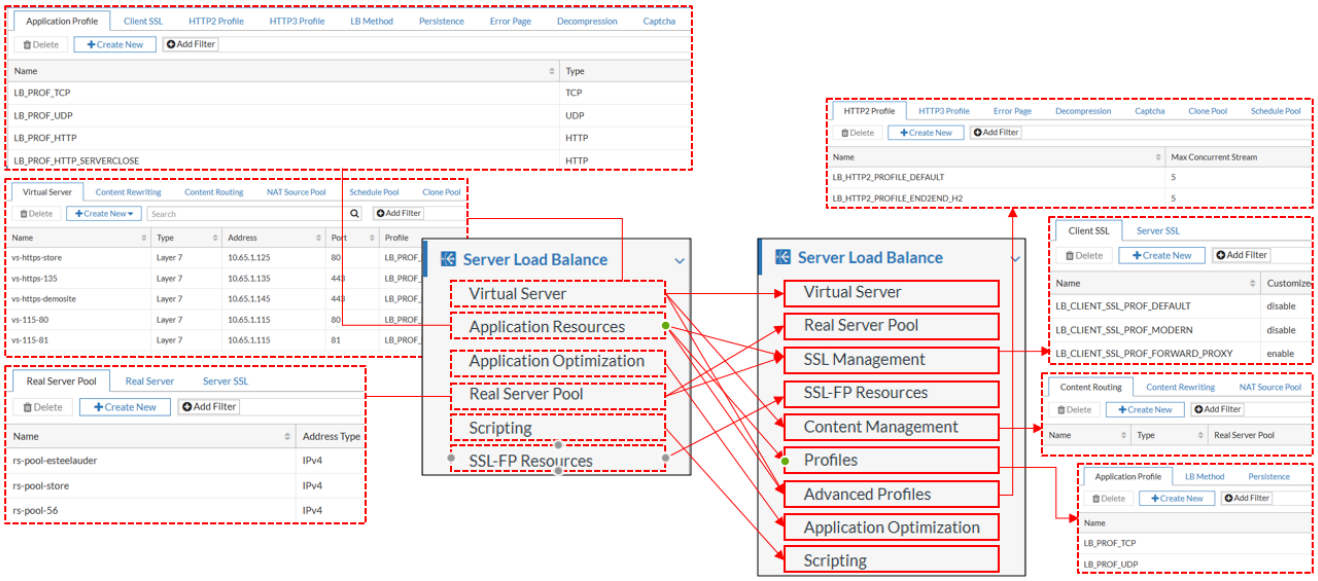
- **Administrators**: Centralizes the management of administrators and access profiles.
- **Firmware**: This section is moved from the Settings tab to its own dedicated menu item for improved visibility.
- **High Availability**: Now includes **Traffic Group** settings to centralize cluster configuration.
- **Certificate Module Renaming**: To improve clarity, "Manage Certificates" is renamed to **Certificate**, and "Verify" is renamed to **Validate Certificates**.



Server Load Balance Optimization

The **Server Load Balance** menu is reorganized to reflect a logical policy flow, moving away from congested menus to a structured application delivery model.

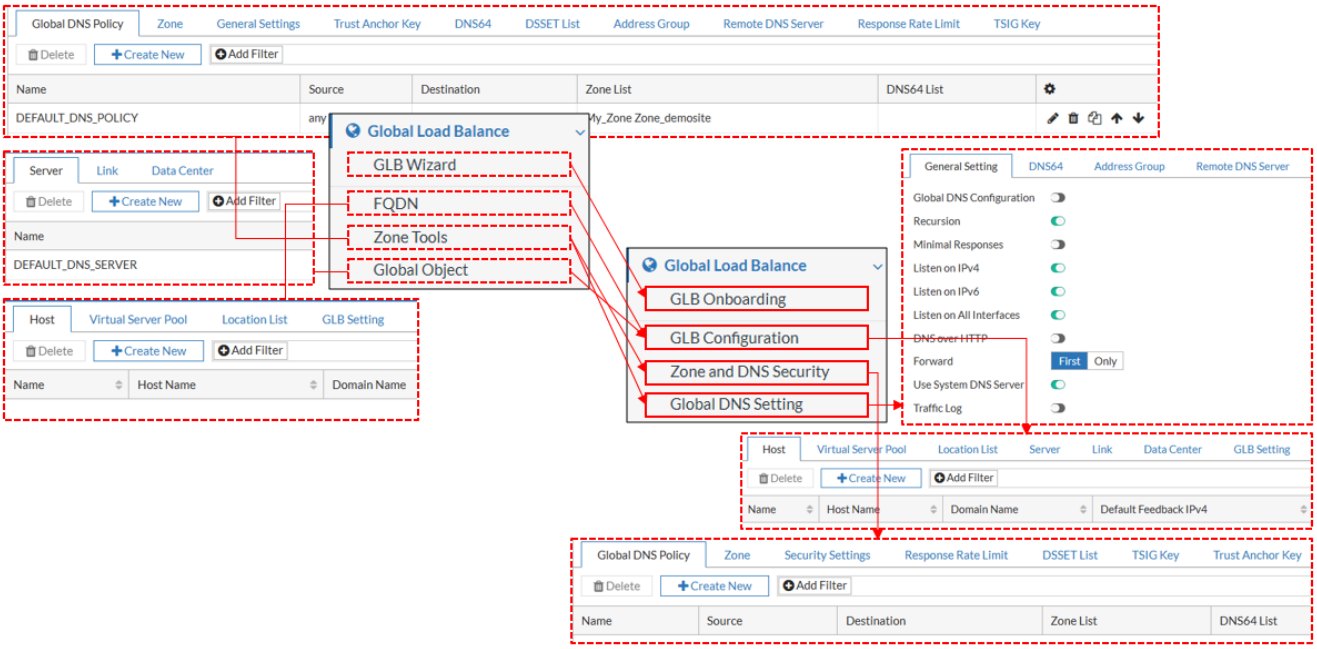
- **Virtual Server**: This module is now standalone, focusing exclusively on the primary virtual server configuration page.
- **Content Management**: A new module that consolidates **Content Routing**, **Content Rewriting**, and **NAT Source Pool** (formerly under Virtual Server).
- **Real Server Pool**: Centralizes **Real Server Pool** and individual **Real Server** settings.
- **SSL Management**: A new module that provides a centralized location for **Client SSL** (moved from Application Resource) and **Server SSL** (moved from Real Server Pool).
- **Profiles**: Groups core traffic management settings, including **Application Profile**, **LB Method**, and **Persistence**.
- **Advanced Profiles**: Consolidates specialized configurations including **HTTP2/HTTP3 Profiles**, **Error Page**, **Decompression**, and **Captcha**. It also includes **Clone Pool** and **Schedule Pool** (moved from Virtual Server).



Global Load Balance Refinement

To address menus with excessive tabs, **Global Load Balance** (GSLB) is now divided into functional groups that improve navigation.

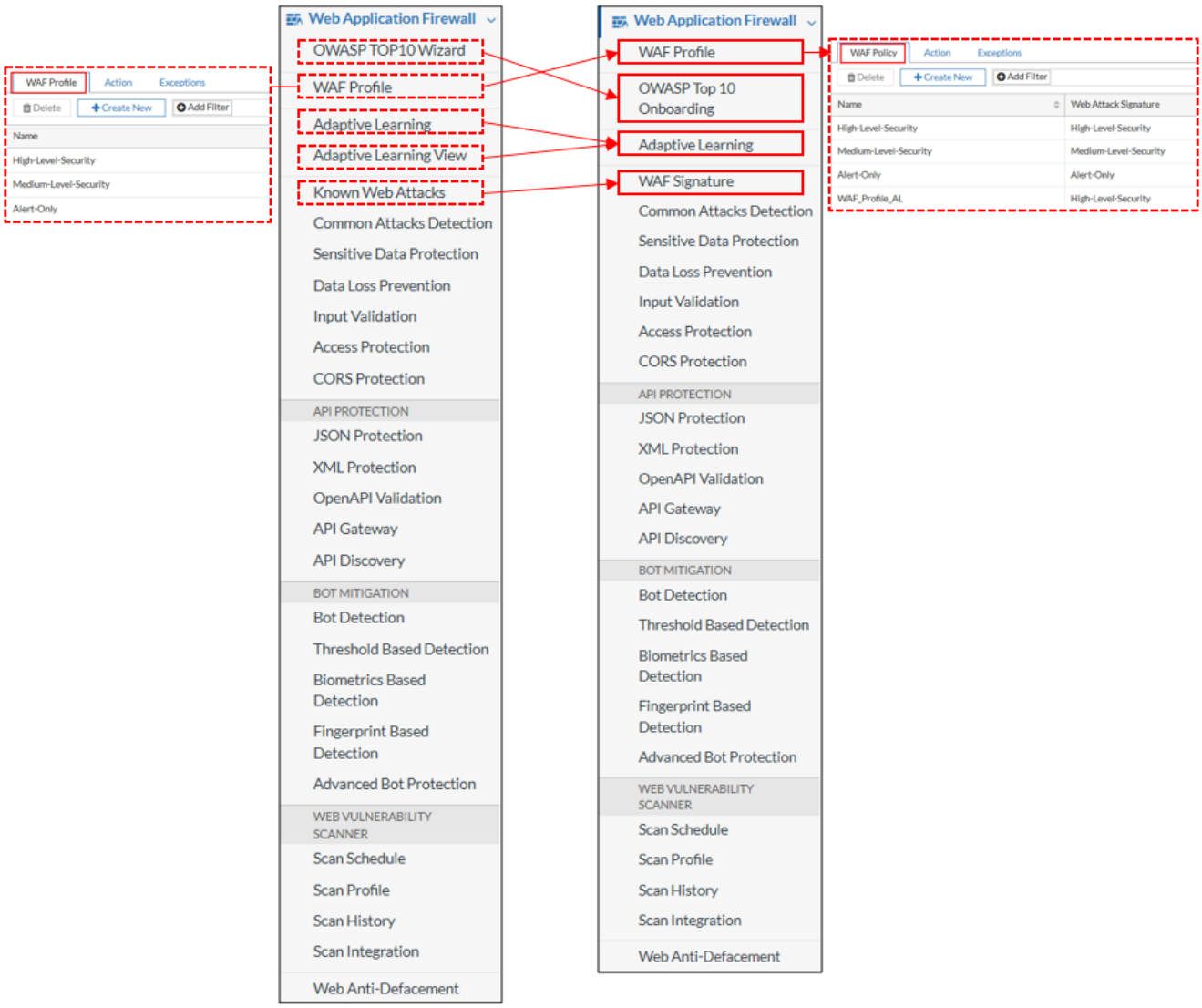
- **GLB Onboarding**: Replaces the former GLB Wizard and includes a new Configuration Overview tab to provide immediate visibility into host and server associations.
- **GLB Configuration**: Consolidates core objects including **Host**, **Virtual Server Pool**, **Location List**, **Server**, **Link**, **Data Center**, and **GLB Setting**.
- **Zone and DNS Security**: A new dedicated module for security-related settings, including **Global DNS Policy**, **Zone**, **Security Settings**, **Response Rate Limit**, **DSSET List**, **TSIG Key**, and **Trust Anchor Key**.
- **Global DNS Setting**: Consolidates general server behavior, including **General Setting**, **DNS64**, **Address Groups** and **Remote DNS Server**.



Web Application Firewall Usability

The **Web Application Firewall** (WAF) layout is refined to improve navigation and align with updated terminology.

- **WAF Policy**: The WAF Profile tab is renamed to **WAF Policy** to better reflect the administrative function.
- **OWASP Top 10 Onboarding**: The "OWASP TOP10 Wizard" is renamed to **OWASP Top 10 Onboarding**.
- **WAF Signature**: The "Known Web Attacks" menu is renamed to **WAF Signature** for technical clarity.
- **Adaptive Learning**: The "Adaptive Learning View" is integrated as a tab within the **Adaptive Learning** menu to reduce menu length.

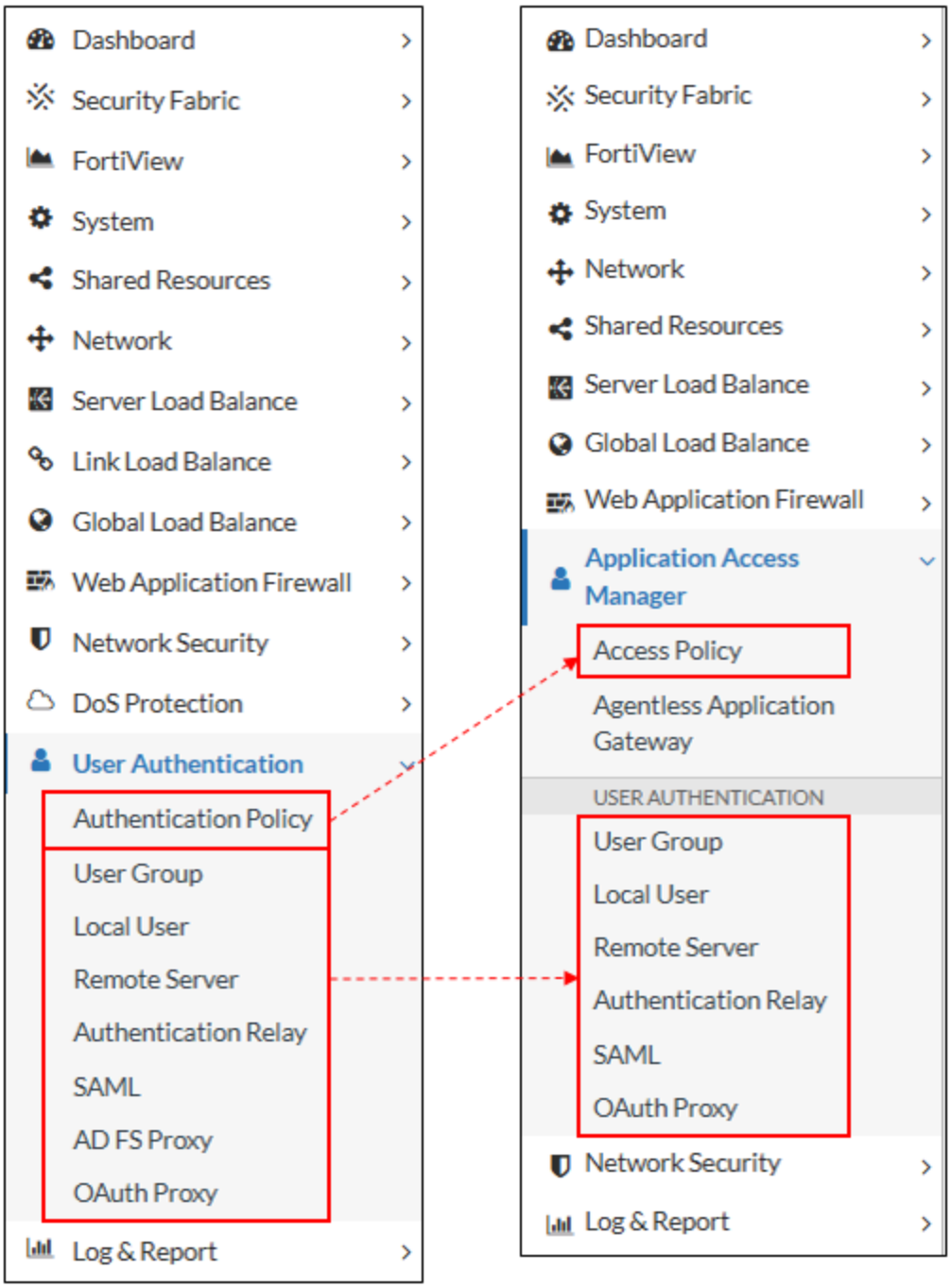

WAF Profile
Action
Exceptions
Delete
Create New
Add Filter
Name
High-Level-Security
Medium-Level-Security
Alert-Only
Web Application Firewall
OWASP TOP10 Wizard
WAF Profile
Adaptive Learning
Adaptive Learning View
Known Web Attacks
Common Attacks Detection
Sensitive Data Protection
Data Loss Prevention
Input Validation
Access Protection
CORS Protection
API PROTECTION
JSON Protection
XML Protection
OpenAPI Validation
API Gateway
API Discovery
BOT MITIGATION
Bot Detection
Threshold Based Detection
Biometrics Based Detection
Fingerprint Based Detection
Advanced Bot Protection
WEB VULNERABILITY SCANNER
Scan Schedule
Scan Profile
Scan History
Scan Integration
Web Anti-Defacement
Web Application Firewall
WAF Profile
OWASP Top 10 Onboarding
Adaptive Learning
WAF Signature
Common Attacks Detection
Sensitive Data Protection
Data Loss Prevention
Input Validation
Access Protection
CORS Protection
API PROTECTION
JSON Protection
XML Protection
OpenAPI Validation
API Gateway
API Discovery
BOT MITIGATION
Bot Detection
Threshold Based Detection
Biometrics Based Detection
Fingerprint Based Detection
Advanced Bot Protection
WEB VULNERABILITY SCANNER
Scan Schedule
Scan Profile
Scan History
Scan Integration
Web Anti-Defacement
WAF Policy
Action
Exceptions
Delete
Create New
Add Filter
Name
Web Attack Signature
High-Level-Security
High-Level-Security
Medium-Level-Security
Medium-Level-Security
Alert-Only
Alert-Only
WAF_Profile_AL
High-Level-Security

Updated Navigation Menu Structure

FortiADC 8.0.3 reorganizes the GUI navigation layout to improve usability and align feature groupings with functional domains. Key configuration areas—including user authentication modules and DoS protection settings—have been relocated under new parent menus to reflect their expanded scope and associated feature enhancements.

Application Access Manager

A new Application Access Manager module has been introduced at the top level of the menu. This consolidates all previously standalone user authentication configuration items—**Access Policy** (formerly Authentication Policy), Local User, Remote User, Authentication Server, SAML, and OAuth—under a unified structure at **Application Access Manager > User Authentication**.

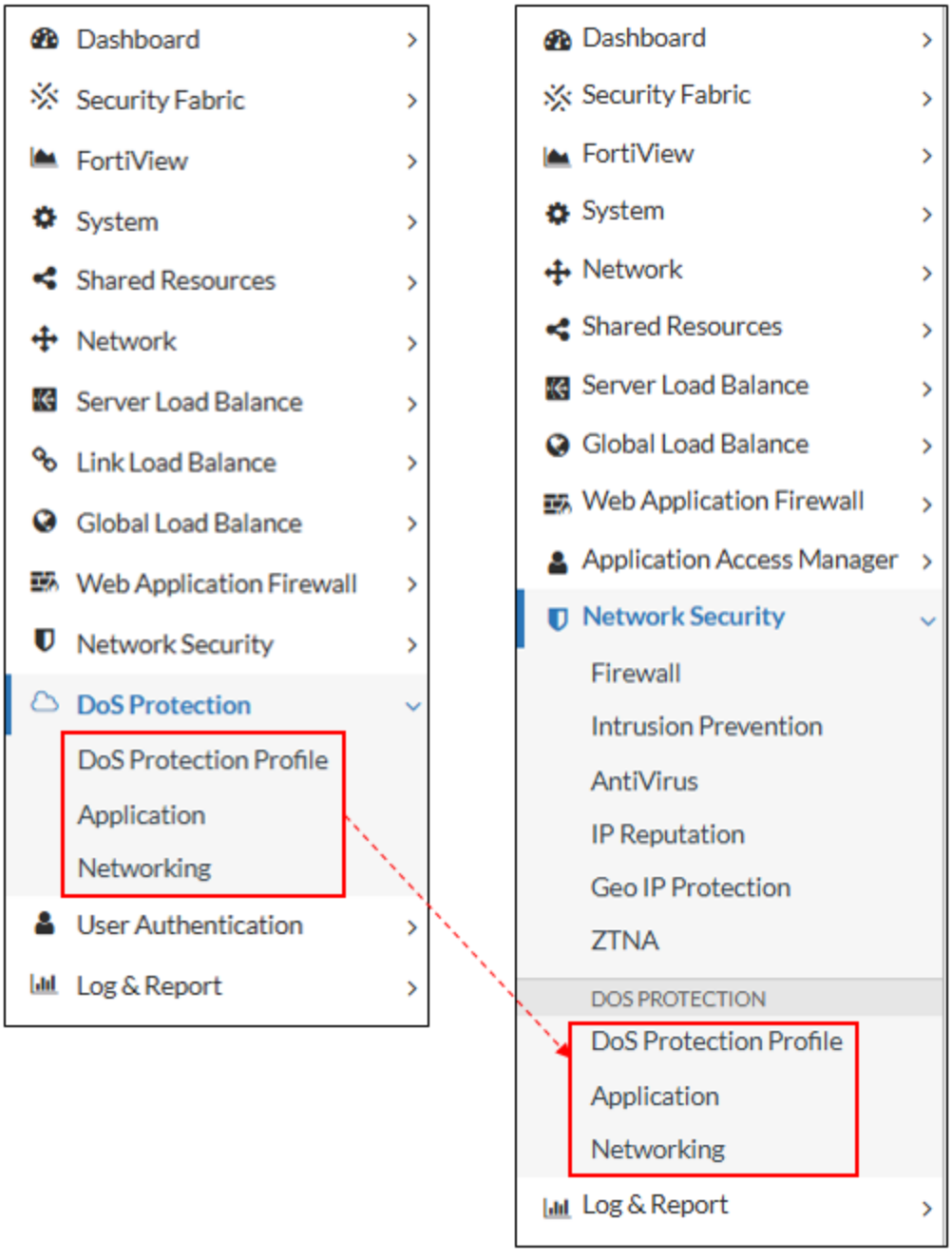


Key changes:

- The legacy Authentication Policy has been replaced by the Access Policy.
- All previously standalone User Authentication configuration pages are now grouped under the Application Access Manager module.
- The new Agentless Application Gateway (AAG) feature is fully integrated within this module.

DoS Protection Moves to Network Security

All DoS protection configurations have been consolidated under **Network Security > DoS Protection**, including the DoS Protection Profile and all DoS-related modules that were previously located under the Application and Networking sections of the GUI.



Key changes:

- The DoS Protection feature set has been reorganized for consistency and improved discoverability.
- The restructured layout makes it easier to manage exception rules and protection profiles from a centralized location.

Platform

The FortiADC 8.0 release includes new features and enhancements to support **Platforms**:

OpenSSL Upgrade to 3.5 8.0.3 on page 198

FortiADC 8.0.3 upgrades the OpenSSL library to version 3.5 to align with the latest security compliance requirements and upstream fixes.

Expanded Local Certificate Group Member Limit 8.0.1 on page 199

FortiADC 8.0.1 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 3.3 8.0.1 on page 200

FortiADC 8.0.1 upgrades the OpenSSL library to version 3.3 to align with the latest security compliance requirements and upstream fixes.

OCI DRCC support 8.0.1 on page 201

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

TPM & Encrypted Data Store Support on page 202

FortiADC now supports Trusted Platform Module (TPM) chips on select hardware platforms, enhancing the security of cryptographic key storage. TPM secures passwords and cryptographic keys by storing and authenticating them using AES-128-CBC encryption. This reduces the risk of tampering and data interception, ensuring private data is securely protected and tied to the hardware device.

Enhanced Azure HA Support with FortiFlex Licensing for Up to 8 Nodes on page 205

FortiADC 8.0.3 introduces support for high availability (HA) deployments of up to eight nodes in Microsoft Azure using FortiFlex licensing. This enhancement removes the previous two-node limitation in Azure HA template deployments and enables greater scalability and flexibility for large-scale cloud deployments.

OpenSSL Upgrade to 3.5 8.0.3

FortiADC 8.0.3 upgrades the OpenSSL library to version 3.5 to align with the latest security compliance requirements and upstream fixes.

Expanded Local Certificate Group Member Limit 8.0.1

FortiADC 8.0.1 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 8.0.1 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 3.3 8.0.1

FortiADC 8.0.1 upgrades the OpenSSL library to version 3.3 to align with the latest security compliance requirements and upstream fixes.

OCI DRCC support 8.0.1

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

TPM & Encrypted Data Store Support

FortiADC now supports Trusted Platform Module (TPM) chips on select hardware platforms, enhancing the security of cryptographic key storage. TPM secures passwords and cryptographic keys by storing and authenticating them using AES-128-CBC encryption. This reduces the risk of tampering and data interception, ensuring private data is securely protected and tied to the hardware device.

With TPM-enabled encryption, FortiADC improves the confidentiality and integrity of private data such as passwords and cryptographic material used in the system configuration. This support extends FortiADC's capabilities in environments that demand strong data protection and hardware-based security assurances.

This information is also available in the FortiADC 8.0.0 CLI Reference:

- config system global
- execute private-encryption-key

Key Features and Benefits

- **Hardware-based Key Protection**: TPM provides secure, tamper-resistant storage for encryption keys, preventing unauthorized access and ensuring that keys remain bound to the physical device.
- **Stronger Encryption**: Private data is encrypted using a randomly generated AES-128-CBC key, rather than a static or user-supplied key.
- **Secure CLI and Config Storage**: Passwords are encrypted when displayed in the CLI and stored securely in the configuration file.
- **Backup Integrity Validation:** Configuration backups include an HMAC of the encryption key. During restoration, the system verifies the key hash to ensure the backup is not used on an unauthorized device.
- **HA Synchronization**: Encryption configuration is synchronized across HA-AP, HA-AA and HA-VRRP deployments.
- **On-Demand Key Regeneration**: Administrators can regenerate the key by disabling and re-enabling the encryption setting.

Supported Hardware Models

TPM-based encrypted key storage is available on the following FortiADC hardware models:

- FortiADC 220F
- FortiADC 320F
- FortiADC 420F
- FortiADC 1200F
- FortiADC 2200F
- FortiADC 4200F

CLI Configuration and HA Behavior

The feature is controlled via a new CLI command:

```
config system global
    set private-data-encryption {enable|disable}
end
```

- **enable**: Generates a random AES-128-CBC encryption key and stores it securely. On TPM-equipped models, the key is stored inside the TPM. On non-TPM systems, it is stored in a file.
- **disable**: Restores the system default encryption key.

TPM is disabled by default.

HA Considerations:

- For **HA-AP** and **HA-AA**, this setting can only be configured on the primary node. The configuration is automatically synced to peer nodes.
- **HA-VRRP** deployments have no such restriction.

Note: Rebooting or upgrading the system does not regenerate the encryption key. To regenerate the key, disable and then re-enable the feature.

Backup and Restore Behavior

When `private-data-encryption` is enabled, the system adds a `private_key` header to the configuration backup file. This value is an HMAC hash of the encryption key.

- During a restore operation, FortiADC compares this value to a local HMAC of the encryption key.
- If the values do not match, the restoration is aborted to prevent misuse on unauthorized systems.

Restoring to a New or Factory-Reset Device:

1. Edit the configuration file:
   a. Set `private_key` header to an empty string.
   b. Disable `private-data-encryption` in `config system global`.
2. After restoration, passwords will be unset.
3. To retain password values, disable encryption before backing up the configuration, then re-enable it after restore.

Key Verification and Troubleshooting

FortiADC introduces two new CLI commands for testing and validating the encryption key:

- `execute private-encryption-key sample`: Generates two sample strings derived from the encryption key.
- `execute private-encryption-key verify`: Accepts the sample strings as input and verifies the integrity of the current encryption key.

Expected Behavior:

- Verification passes if the key remains unchanged.
- If the key has changed (e.g., after regeneration), verification fails as expected.
- If encryption is disabled, attempting to run `sample` will return a prompt indicating the feature is not enabled.

Enhanced Azure HA Support with FortiFlex Licensing for Up to 8 Nodes

FortiADC 8.0.3 introduces support for high availability (HA) deployments of up to eight nodes in Microsoft Azure using FortiFlex licensing. This enhancement removes the previous two-node limitation in Azure HA template deployments and enables greater scalability and flexibility for large-scale cloud deployments.

Key improvements include:

- **Flexible Token Handling**: The updated HA deployment template accepts a comma-separated list of FortiFlex tokens, eliminating hardcoded variables for individual nodes.
- **Automated Token Assignment**: Each VM instance receives its respective FortiFlex token through the Azure customdata field, based on its HA member index (HaInstanceId). The cloudinitd daemon processes this data, extracts the token, and applies the correct license during instance initialization.
- **Improved Template Scalability**: The new design allows the Azure template to scale to eight HA nodes while preserving consistent configuration and licensing behavior.

Limitation: In Azure, the applied FortiFlex license must match or exceed the size of the VM instance. For example, a VM16 license cannot be used on a VM with 32 vCPUs, as this will render the license invalid.

Troubleshooting

The FortiADC 8.0 release includes new features and enhancements to support Troubleshooting:

Enhanced Hardware Diagnostics in CLI on page 207

FortiADC expands its system diagnostics by introducing two new subcommands under the `diagnose hardware` CLI command: `harddisk` and `logdisk`. These enhancements integrate SMART-based health monitoring and reporting for both hard disks and log disks, enabling administrators to identify disk-related issues directly from the CLI.

Enhanced Hardware Diagnostics in CLI

FortiADC expands its system diagnostics by introducing two new subcommands under the `diagnose hardware` CLI command: `harddisk` and `logdisk`. These enhancements integrate SMART-based health monitoring and reporting for both hard disks and log disks, enabling administrators to identify disk-related issues directly from the CLI.

This information is also available in the FortiADC 8.0.0 CLI Reference:

- diagnose hardware harddisk
- diagnose hardware logdisk

SMART (Self-Monitoring, Analysis and Reporting Technology) is a standard feature in most modern HDDs and SSDs. It tracks critical health indicators–such as reallocated sectors, temperature, and I/O error rates–to help detect early signs of hardware failure.

Note: SMART support may vary by platform and disk type. If unsupported, the `harddisk` or `logdisk` commands may return warnings or partial output.

For configuration details, navigate to the following sections:

- Configuring diagnose hardware harddisk on page 207
- Configuring diagnose hardware logdisk on page 209

Configuring `diagnose hardware harddisk`

Use this command to retrieve diagnostic information for hard disks installed in the system. It integrates `smartctl` functionality to expose SMART (Self-Monitoring, Analysis and Reporting Technology) data, allowing administrators to monitor disk health and detect early signs of failure.

Syntax

```
diagnose hardware {get|set} harddisk {attributes|errors|health|info|list}
```

`attributes`	Display SMART attributes such as temperature, power cycle count, reallocated sectors, etc.
`errors`	Show SMART error log entries recorded by the disk controller.
`health`	Check and reports the disk's overall SMART health status.
`info`	Display basic disk identity information, including model, firmware, and capacity.
`list`	List all detected hard disks on the system.

Examples

diagnose hardware get harddisk attributes:

```
(P) FAD22FT221000040 # diagnose hardware get harddisk attributes
===== Disk /dev/sda =====
smartctl 7.1 2019-12-30 r5022 [x86_64-linux-4.14] (local build)
Copyright (C) 2002-19, Bruce Allen, Christian Franke, www.smartmontools.org

=== START OF READ SMART DATA SECTION ===
SMART Attributes Data Structure revision number: 1
Vendor Specific SMART Attributes with Thresholds:
ID# ATTRIBUTE_NAME          FLAG     VALUE WORST THRESH TYPE      UPDATED  WHEN_FAILED RAW_VALUE
  5 Reallocated_Sector_Ct   0x0032   100   100   000    Old_age   Always       -       0
  9 Power_On_Hours          0x0032   100   100   000    Old_age   Always       -       4354
 12 Power_Cycle_Count       0x0032   100   100   000    Old_age   Always       -       43
170 Available_Reservd_Space 0x0033   100   100   010    Pre-fail  Always       -       0
171 Program_Fail_Count      0x0032   100   100   000    Old_age   Always       -       0
172 Erase_Fail_Count        0x0032   100   100   000    Old_age   Always       -       0
174 Unsafe_Shutdown_Count   0x0032   100   100   000    Old_age   Always       -       42
175 Power_Loss_Cap_Test     0x0033   100   100   010    Pre-fail  Always       -       2679 (43 65535)
183 SATA_Downshift_Count    0x0032   100   100   000    Old_age   Always       -       0
184 End-To-End_Error_Count  0x0033   100   100   090    Pre-fail  Always       -       0
187 Uncorrectable_Error_Cnt 0x0032   100   100   000    Old_age   Always       -       0
190 Drive_Temperature       0x0022   077   074   000    Old_age   Always       -       23 (Min/Max 14/26)
192 Unsafe_Shutdown_Count   0x0032   100   100   000    Old_age   Always       -       42
194 Temperature_Celsius     0x0022   100   100   000    Old_age   Always       -       23
197 Pending_Sector_Count    0x0012   100   100   000    Old_age   Always       -       0
199 CRC_Error_Count         0x003e   100   100   000    Old_age   Always       -       0
225 Host_Writes_32MiB       0x0032   100   100   000    Old_age   Always       -       7559
226 Workld_Media_Wear_Indic 0x0032   100   100   000    Old_age   Always       -       20
227 Workld_Host_Reads_Perc  0x0032   100   100   000    Old_age   Always       -       30
228 Workload_Minutes        0x0032   100   100   000    Old_age   Always       -       260863
232 Available_Reservd_Space 0x0033   100   100   010    Pre-fail  Always       -       0
233 Media_Wearout_Indicator 0x0032   100   100   000    Old_age   Always       -       0
234 Thermal_Throttle_Status 0x0032   100   100   000    Old_age   Always       -       0/0
235 Power_Loss_Cap_Test     0x0033   100   100   010    Pre-fail  Always       -       2679 (43 65535)
241 Host_Writes_32MiB       0x0032   100   100   000    Old_age   Always       -       7559
242 Host_Reads_32MiB        0x0032   100   100   000    Old_age   Always       -       3278
243 NAND_Writes_32MiB       0x0032   100   100   000    Old_age   Always       -       25804

===== Disk /dev/sdb =====
smartctl 7.1 2019-12-30 r5022 [x86_64-linux-4.14] (local build)
Copyright (C) 2002-19, Bruce Allen, Christian Franke, www.smartmontools.org

/dev/sdb: Unknown USB bridge [0x125f:0x123c (0x1100)]
Please specify device type with the -d option.

Use smartctl -h to get a usage summary
```

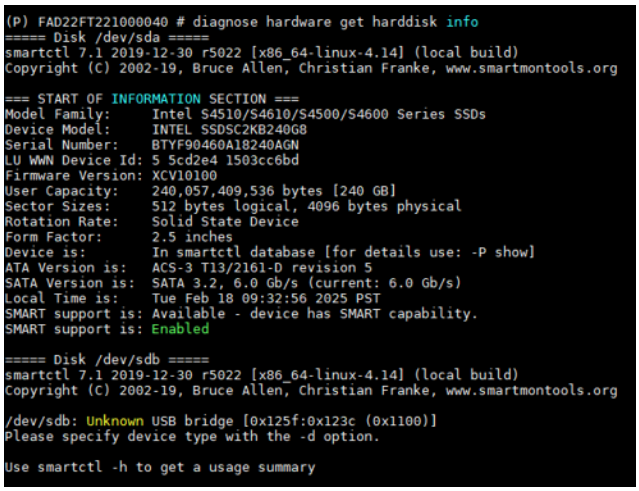
diagnose hardware get harddisk errors:

```
(P) FAD22FT221000040 # diagnose hardware get harddisk errors
===== Disk /dev/sda =====
smartctl 7.1 2019-12-30 r5022 [x86_64-linux-4.14] (local build)
Copyright (C) 2002-19, Bruce Allen, Christian Franke, www.smartmontools.org

=== START OF READ SMART DATA SECTION ===
SMART Error Log Version: 1
No Errors Logged

===== Disk /dev/sdb =====
smartctl 7.1 2019-12-30 r5022 [x86_64-linux-4.14] (local build)
Copyright (C) 2002-19, Bruce Allen, Christian Franke, www.smartmontools.org

/dev/sdb: Unknown USB bridge [0x125f:0x123c (0x1100)]
Please specify device type with the -d option.

Use smartctl -h to get a usage summary
```

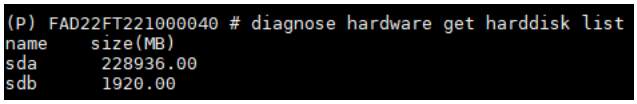
diagnose hardware get harddisk health:

```
(P) FAD22FT221000040 # diagnose hardware get harddisk health
===== Disk /dev/sda =====
smartctl 7.1 2019-12-30 r5022 [x86_64-linux-4.14] (local build)
Copyright (C) 2002-19, Bruce Allen, Christian Franke, www.smartmontools.org

=== START OF READ SMART DATA SECTION ===
SMART overall-health self-assessment test result: PASSED

===== Disk /dev/sdb =====
smartctl 7.1 2019-12-30 r5022 [x86_64-linux-4.14] (local build)
Copyright (C) 2002-19, Bruce Allen, Christian Franke, www.smartmontools.org

/dev/sdb: Unknown USB bridge [0x125f:0x123c (0x1100)]
Please specify device type with the -d option.

Use smartctl -h to get a usage summary
```

diagnose hardware get harddisk info:



diagnose hardware get harddisk list:



Configuring diagnose hardware logdisk

Use this command to retrieve diagnostic information for the log disk used by FortiADC. It reports basic attributes such as device identification, capacity, and status. The output helps verify logging disk availability and detect potential storage issues affecting log integrity or access.

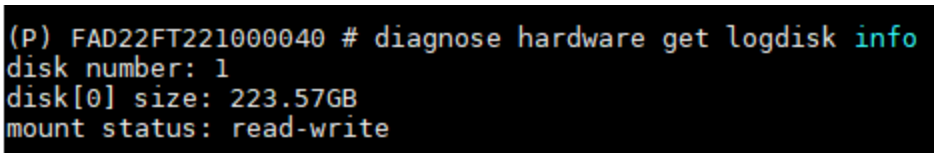
Syntax

```
diagnose hardware {get|set} logdisk info
```

`info`	Display general information about the logging disk used by the system.

Example

diagnose hardware logdisk info:



www.fortinet.com