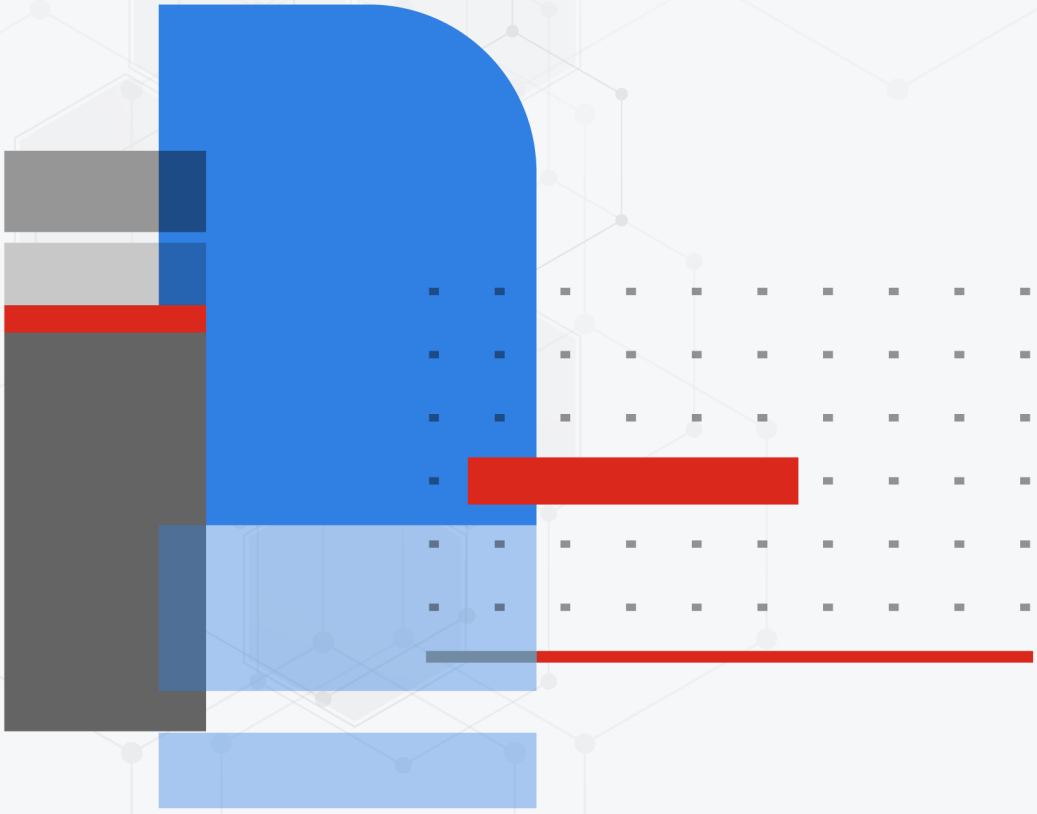
FortiADC on Citrix VDI Deployment Guide



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

May 16, 2025
FortiADC 8.0.0 FortiADC on Citrix VDI Deployment Guide

TABLE OF CONTENTS

Change Log

Date	Change Description
May 16, 2025	FortiADC on Citrix VDI Deployment Guide.

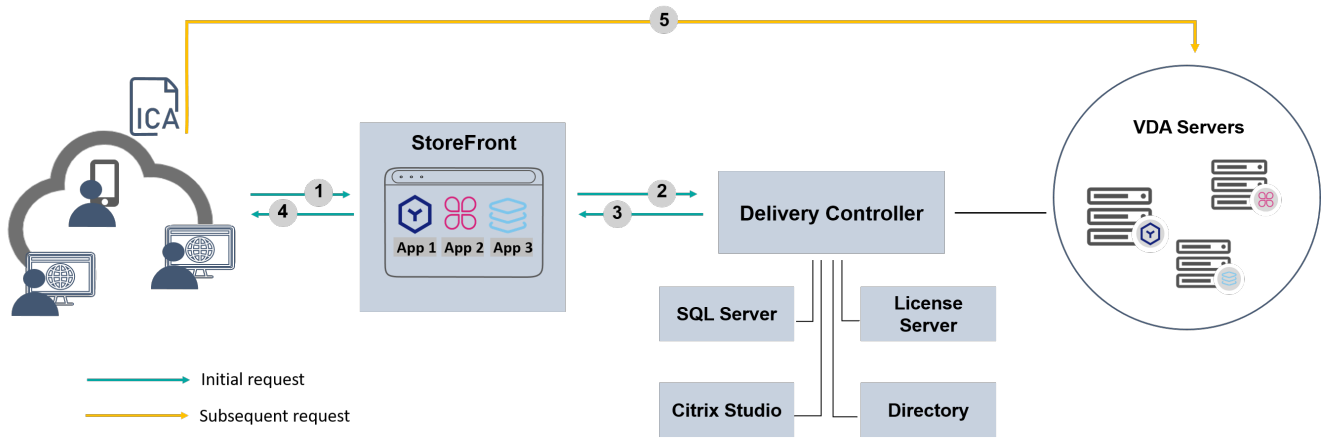
About FortiADC on Citrix VDI

Citrix Workspace is a comprehensive digital workspace solution that provides secure and unified access to applications, desktops, and content (including Citrix DaaS) from anywhere, on any device. Key features include single sign-on (SSO) for streamlined authentication, centralized management for efficient administration, and extensive customization options to optimize the user experience. Citrix Workspace integrates a wide range of services to ensure seamless access to essential tools while maintaining stringent security protocols.

Check the right sidebar or click this link to watch the video introducing FortiADC's deployment in a Citrix VDI environment.

Citrix VDI Components and Traffic flow

Traffic flow of the VDI environment



Initial request

1. **User Accesses StoreFront**
   The user logs in Citrix VDI, opens the StoreFront web portal and selects the desired application or desktop.
2. **StoreFront Communicates with Delivery Controller**
   StoreFront sends the authentication request and resource query to the Delivery Controller, which:
   - Authenticates the user via Active Directory
   - Queries the list of available desktops and applications
   - Selects an appropriate VDA (Virtual Delivery Agent) server for the session

3. **Delivery Controller Responds**
   The Delivery Controller returns the selected VDA server details to StoreFront.
4. **ICA File is Issued**
   StoreFront sends an ICA file to the user's device. This file contains essential metadata needed for the Citrix Workspace App (or Receiver) to establish a session with the appropriate VDA server, such as the VDA Server Address, Connection Type, SSL Settings, and Authentication Tokens.

Subsequent requests

5.User Connects to VDA

Using the ICA file, the user's device initiates a direct connection to the specified VDA server. From this point on, StoreFront and the Delivery Controller are no longer involved in the session.

Components

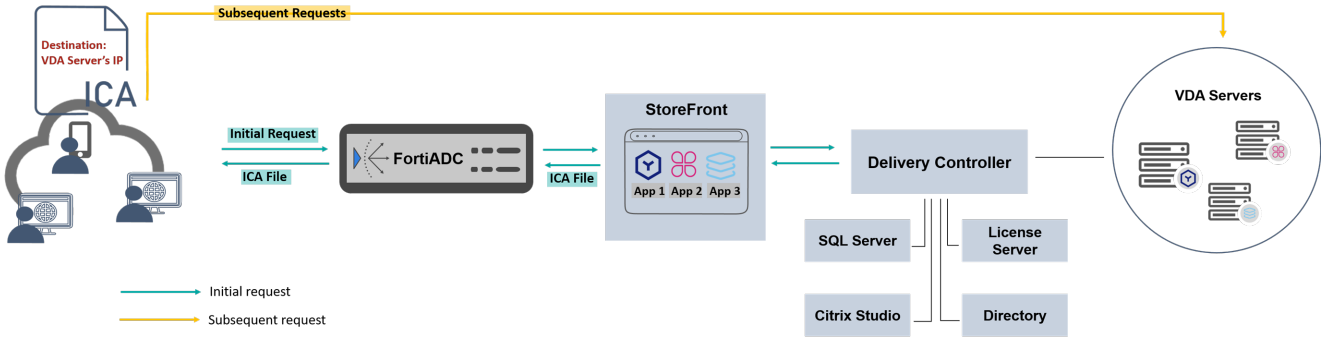
- **User Device**
  The computer, laptop, tablet, or phone that the user uses to access the virtual desktop.
- **StoreFront**
  A web portal where users log in and choose the apps or desktops they want to launch.
- **Delivery Controller**
  The "traffic controller" of Citrix. It decides which virtual desktop or app a user gets and manages the connection between users and servers.
- **ICA File**
  An ICA file is a configuration file used in Citrix VDI to establish a session between the user's device and a VDA (Virtual Delivery Agent) server.
- **Active Directory**
  A user database. It stores usernames, passwords, and group policies. Used for login authentication and access control.
- **Citrix Studio**
  The admin console. IT admins use it to configure and manage all Citrix components, including desktops, apps, and policies.
- **SQL Server**
  Stores the configuration and system data for the Citrix environment, such as site settings, machine catalogs, and delivery groups.
- **License Server**
  Checks if there are enough Citrix licenses when users try to connect. Without valid licenses, users cannot log in.
- **Director**
  A monitoring and support tool. IT staff use it to check system health, troubleshoot issues, and view real-time user sessions.
- **VDA Servers (Virtual Delivery Agent)**
  These are the virtual desktops or app servers the users connect to. They run the actual Windows desktop or published apps.

Network topology of integrating FortiADC with Citrix VDI

When integrating FortiADC into a Citrix Virtual Desktop Infrastructure (VDI), you can choose from two deployment options based on your performance and security requirements.

- **Solution 1** offers a simple and non-intrusive approach that integrates easily into existing network topologies.
- **Solution 2** provides comprehensive, end-to-end load balancing and security for both StoreFront and VDA (Virtual Delivery Agent) server session phases.

Solution 1: FortiADC Processing Traffic to StoreFront Only

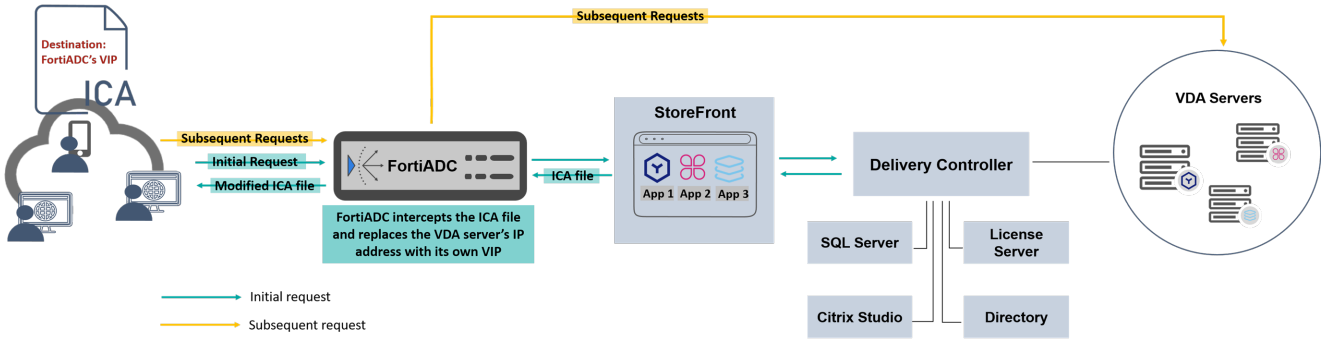


In this deployment, **FortiADC processes only the traffic destined for the StoreFront servers.**

- When a user initially accesses the Citrix VDI environment, the traffic first passes through FortiADC for SSL offloading and security inspection. FortiADC then forwards the traffic to the appropriate StoreFront server.
- After the user is authenticated, an ICA file is returned to the client, containing the connection details for a specific VDA (Virtual Delivery Agent) server. From that point on, FortiADC is no longer involved—the client uses the ICA file to establish a direct connection to the VDA server for session delivery.

For configurations, see Configurations for solution 1: FortiADC processing traffic to StoreFront only on page 18.

Solution 2: FortiADC Processing Traffic to Both StoreFront and VDA Servers



This advanced deployment extends FortiADC's role to include traffic processing **not only during the initial request to StoreFront, but also throughout the subsequent ICA session with VDA servers.**

By default, StoreFront provides an ICA file containing the direct address of the target VDA server. However, in Solution 2, FortiADC can **intercept and rewrite the ICA file to embed routing instructions that redirect ICA traffic back through FortiADC**. This solution delivers end-to-end traffic optimization and security protection, improving performance and user experience in distributed VDI environments.

The key configuration in this deployment is enabling ICA file rewriting on FortiADC using a custom script that replaces the VDA server address with FortiADC's virtual IP address.

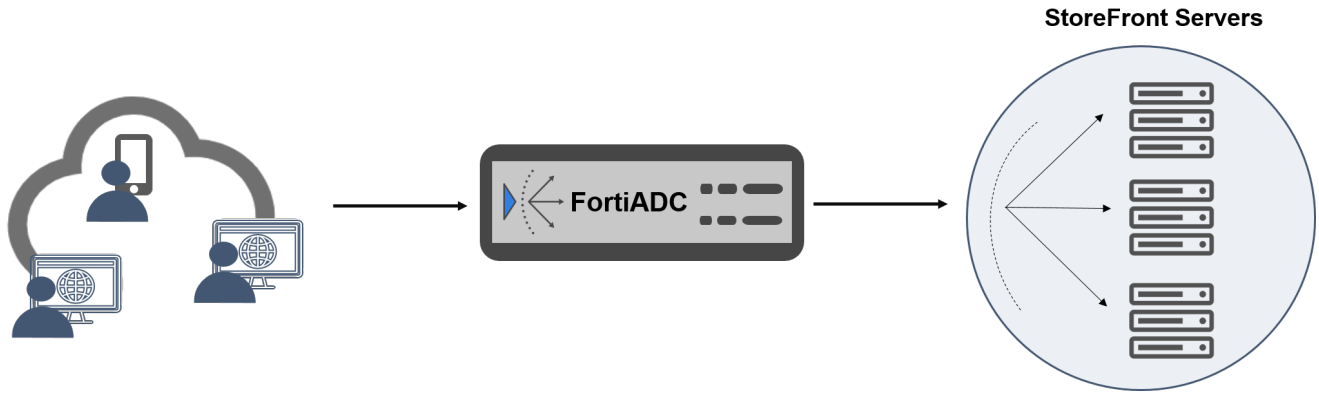
For configurations, see Configurations for solution 2: FortiADC processing traffic to both StoreFront and VDA servers on page 33.

Key benefits by integrating FortiADC with Citrix VDI

The deployment of FortiADC significantly enhances the performance, scalability, and security of a Citrix Virtual Desktop Infrastructure (VDI) by leveraging its advanced features. Here are the key benefits.

Server Load Balancing

FortiADC's server load balancing intelligently distributes client requests across multiple StoreFront servers based on factors such as availability, health status, connection load, and response time.



It supports a variety of load balancing algorithms including Round Robin, Least Connections, and Weighted Load. FortiADC's key load balancing functions include:

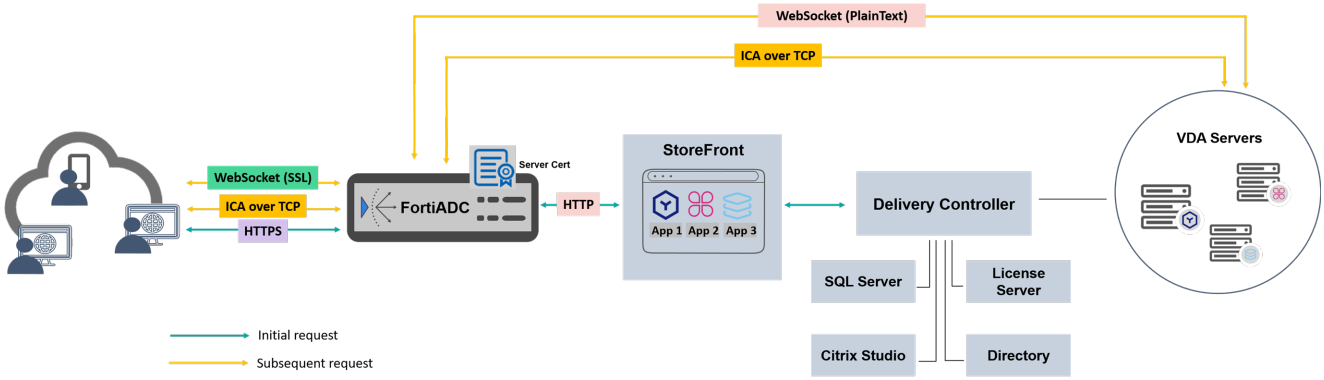
- **Optimized Performance:** Distributes workloads evenly across servers to reduce latency and prevent bottlenecks.
- **Health Monitoring:** Continuously checks the health of StoreFront servers to ensure traffic is only sent to healthy endpoints.
- **Scalability:** Easily accommodates more users by adding StoreFront servers without service disruption.
- **Session Persistence:** Ensures user sessions are consistently directed to the correct StoreFront servers when needed.

For traffic to the VDA servers, Citrix VDI determines which specific VDA server the user should connect to. Therefore, FortiADC does not apply load balancing algorithms proactively. Instead, it functions as a router, forwarding the traffic to the designated VDA server as specified by the Citrix infrastructure (e.g., in the ICA file).

SSL Offloading

FortiADC can act as an SSL proxy, terminating SSL-encrypted traffic and performing SSL offloading by decrypting incoming connections before they reach the backend servers. **By uploading the appropriate server certificates to FortiADC, it can present the certificate to clients and complete the SSL handshake on behalf of the backend servers.**

When establishing connections with backend systems, FortiADC can communicate in plaintext (e.g., HTTP), as the traffic remains within a secure internal network. This offloads the computational burden of SSL encryption and decryption from the backend servers, allowing them to focus on handling user requests and delivering application resources more efficiently.



Designed for high-performance SSL acceleration, FortiADC leverages dedicated hardware and optimized software to process SSL transactions with minimal latency and high throughput, resulting in a smoother and faster user experience.

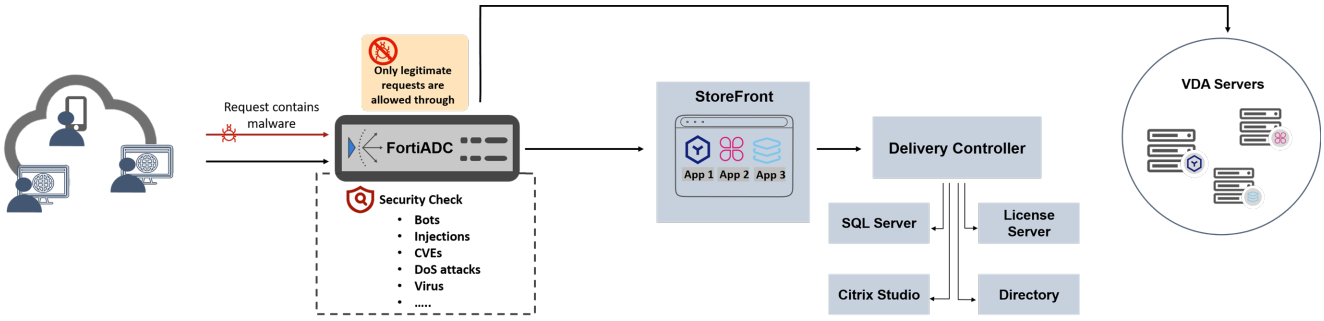
FortiADC supports SSL offloading for both HTTPS and WebSocket traffic, which are commonly used in the following scenarios:

- **Initial access to Citrix VDI**: Clients connect to the StoreFront servers over HTTPS.
- **Subsequent sessions via HTML5 Receiver**: When users access virtual desktops or apps through a browser, the session is established over secure WebSocket (wss://), typically encapsulated in HTTPS traffic over port 443.

However, when users access the Citrix VDI environment through the Citrix Workspace App, subsequent connections to the VDA servers use the ICA protocol over TCP. Because ICA is a proprietary protocol that FortiADC cannot parse, SSL offloading and deep inspection are not supported for this traffic. Nevertheless, FortiADC can still route ICA TCP traffic to the appropriate VDA servers.

Security & Protection

FortiADC provides a comprehensive, multilayered security features that ensure traffic reaching the back-end servers is secure and free from threats.

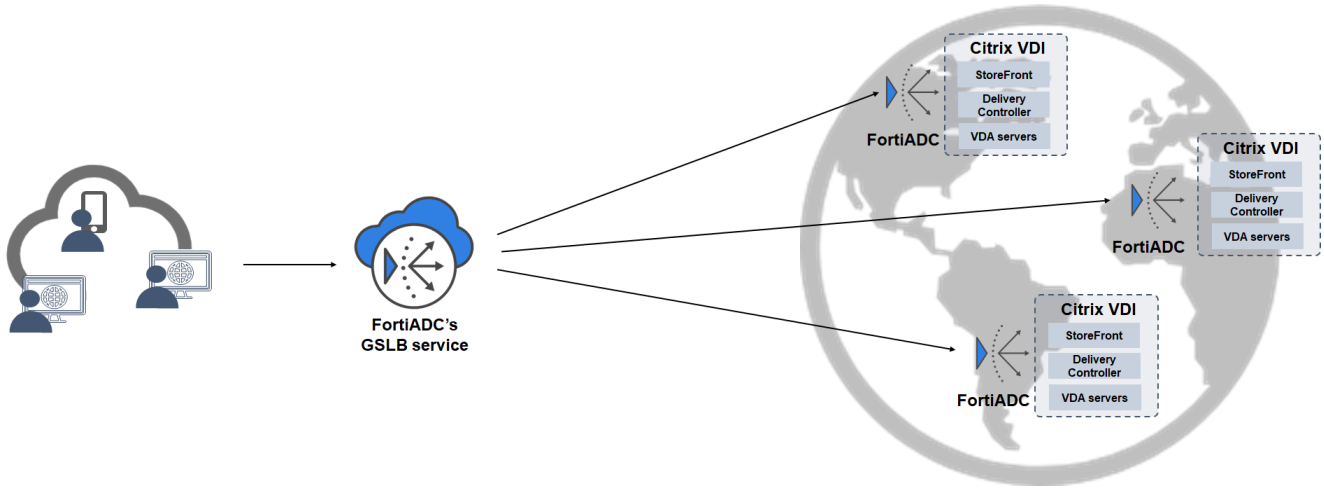


- **Web Application Firewall (WAF)**: Protects against OWASP Top 10 threats like SQL injection and XSS; includes adaptive learning and bot detection.
- **DDoS Protection**: Defends against volumetric and protocol-based attacks.
- **IP Reputation Filtering**: Blocks traffic from known malicious sources.
- **Antivirus & IPS**: Scans for viruses and known intrusions using FortiGuard signatures.
- **HTTP Security Headers**: Enforces policies like HSTS, CSP, and secure cookies.
- **Logging & Visibility**: Provides detailed security event logs and dashboards via FortiView and Threat Analytics.
- **Security Fabric Integration**: Shares telemetry and integrates with FortiGate for unified defense.

Please note that FortiADC can inspect HTTP-based traffic such as HTTPS and WebSocket, but it cannot decrypt, parse, or inspect ICA protocol traffic—even when it is transmitted over TCP.

Global Server Load Balancing (GSLB)

If you have Citrix VDI environments deployed across multiple geographic sites, FortiADC's Global Server Load Balancing (GSLB) service can intelligently route traffic to the most appropriate site based on factors such as geolocation, latency, server health, and load distribution.



GSLB enhances user experience by directing clients to the nearest or least congested site, thereby reducing latency. It also ensures high availability by automatically redirecting traffic to healthy sites in the event of a site failure. By dynamically balancing traffic loads across multiple FortiADC instances, GSLB maintains consistent performance, prevents overload at any single location, and ensures the reliability of your Citrix VDI infrastructure.

Prerequisites

Ensure you have met the following prerequisites before proceeding with the instructions in this guide.

- Citrix VDI installation. For details, see the Citrix Product Documentation.
- Valid FortiADC license.

Assigning internal IP addresses to StoreFront and VDA servers

During the deployment process outlined in the following sections, you will need to assign the client-facing IP address—corresponding to your StoreFront web portal domain—to FortiADC. This ensures that all client requests intended for the web portal are directed to FortiADC first, where SSL offloading and security policies can be applied.

With FortiADC now handling client communication, **key components within the Citrix VDI environment—such as StoreFront and VDA servers—can be deployed in a secure internal subnet**, isolated from the client-facing network. This network segmentation helps reduce the attack surface and enhance overall security.

However, this depends on your specific deployment approach. If FortiADC is configured to handle traffic to StoreFront only, the VDA servers must remain directly reachable by the client.

Enabling loopback communication

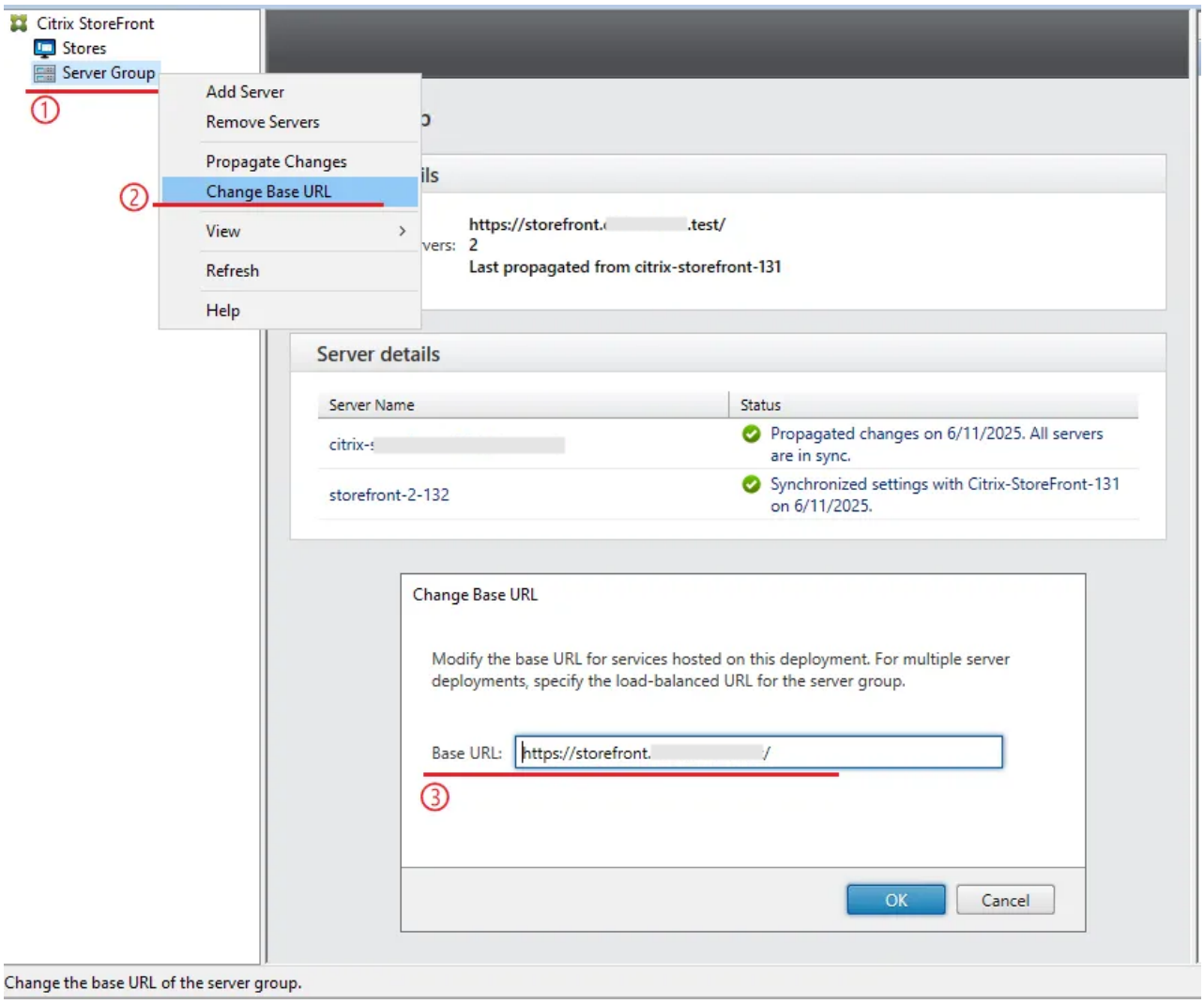
Loopback communication allows a StoreFront server to send HTTP(S) requests to itself using its Base URL (e.g., https://storefront.company.com)—typically for operations like:

- Authenticating the user
- Enumerating available apps and desktops

By default, these internal requests use the same Base URL that clients use. In a load-balanced environment (e.g., with FortiADC), this URL resolves to the VIP (Virtual IP) of FortiADC. When a StoreFront server sends a self-request via the Base URL, it will first reaches FortiADC and FortiADC then may route that request to a different StoreFront server in the server pool. This breaks session consistency and potentially causes authentication or enumeration failures—especially during explicit login workflows.

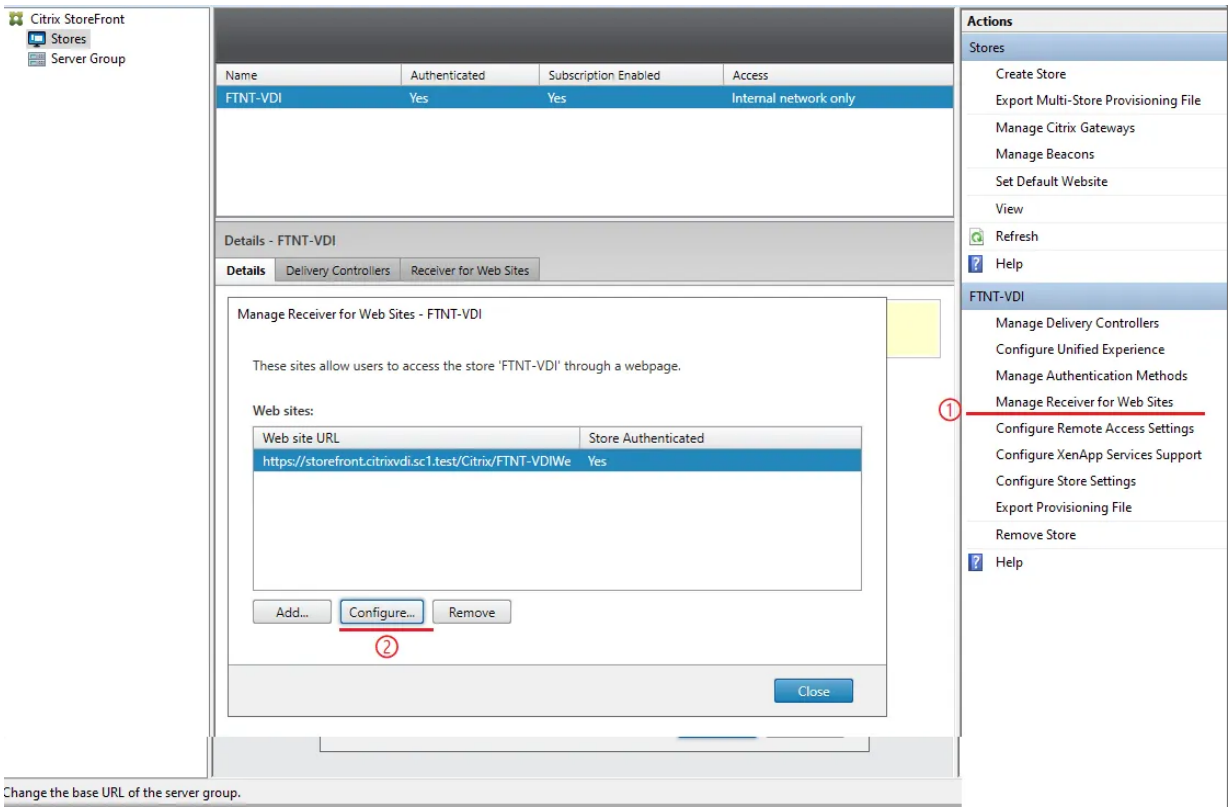
To avoid this issue, perform the following in Citrix VDI:

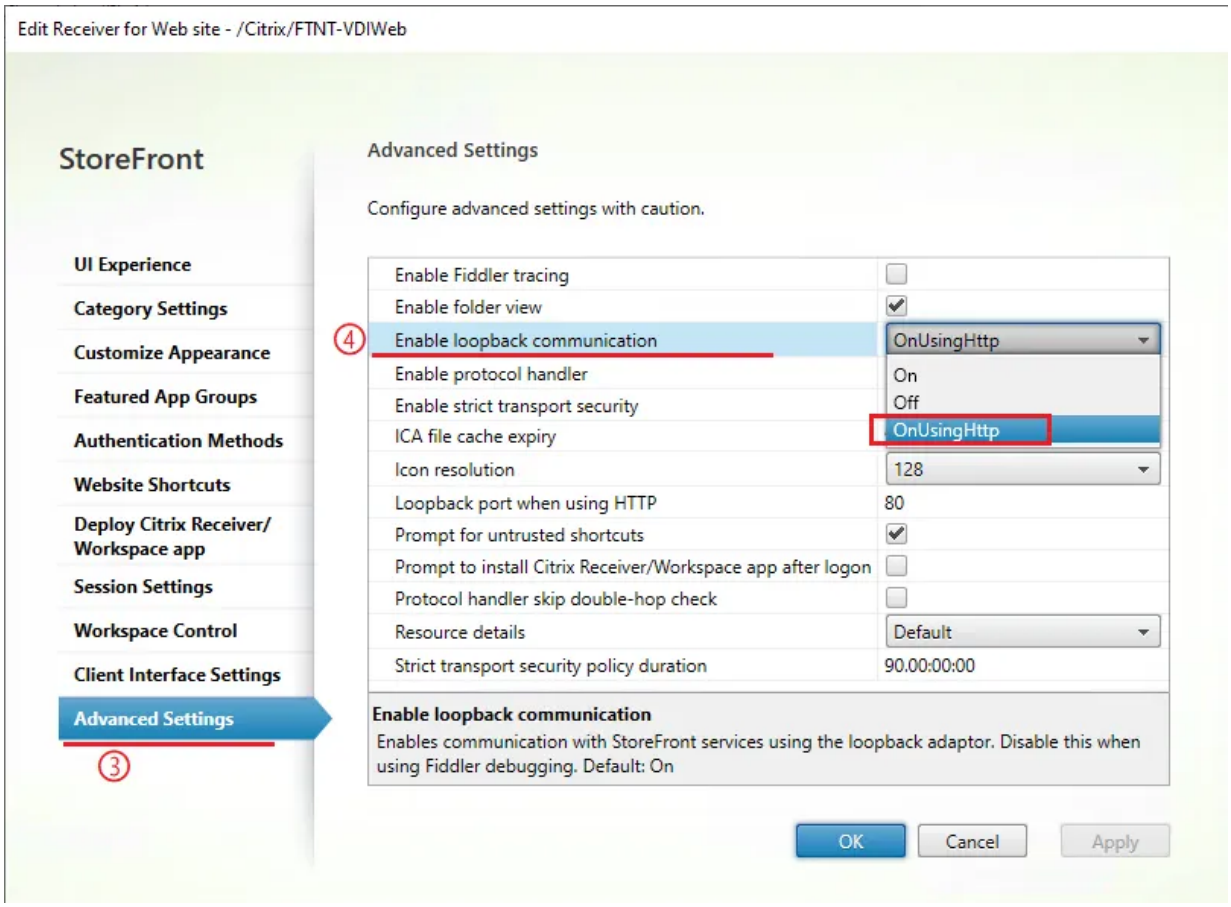
1. Ensure that the StoreFront Base URL is correctly set and begins with https://, matching the domain of the StoreFront web portal (e.g., https://storefront.company.com).



2. For the **Enable loopback communication**, set it to **OnUsingHTTP**. It tells the StoreFront servers to:
   a. For self-requests, go directly to itself via loopback.
   b. Use plain HTTP instead of HTTPS.
   In this way, the loopback communication avoids resolving to the Base URL, because it uses HTTP instead of HTTPS. As a result, the request bypasses FortiADC and communicates directly with the StoreFront server

itself through the loopback interface.


Citrix StoreFront
Stores
Server Group
Name
Authenticated
Subscription Enabled
Access
FTNT-VDI
Yes
Yes
Internal network only
Details - FTNT-VDI
Details
Delivery Controllers
Receiver for Web Sites
Manage Receiver for Web Sites - FTNT-VDI
These sites allow users to access the store 'FTNT-VDI' through a webpage.
Web sites:
Web site URL
Store Authenticated
https://storefront.citrixvdi.sc1.test/Citrix/FTNT-VDIWe
Yes
Add...
Configure...
Remove
Close
Actions
Stores
Create Store
Export Multi-Store Provisioning File
Manage Citrix Gateways
Manage Beacons
Set Default Website
View
Refresh
Help
FTNT-VDI
Manage Delivery Controllers
Configure Unified Experience
Manage Authentication Methods
Manage Receiver for Web Sites
Configure Remote Access Settings
Configure XenApp Services Support
Configure Store Settings
Export Provisioning File
Remove Store
Help
①
②
Change the base URL of the server group.


Edit Receiver for Web site - /Citrix/FTNT-VDIWeb
StoreFront
UI Experience
Category Settings
Customize Appearance
Featured App Groups
Authentication Methods
Website Shortcuts
Deploy Citrix Receiver/ Workspace app
Session Settings
Workspace Control
Client Interface Settings
Advanced Settings
3
Advanced Settings
Configure advanced settings with caution.
Enable Fiddler tracing
Enable folder view
4
Enable loopback communication
OnUsingHttp
Enable protocol handler
On
Enable strict transport security
Off
ICA file cache expiry
OnUsingHttp
Icon resolution
128
Loopback port when using HTTP
80
Prompt for untrusted shortcuts
Prompt to install Citrix Receiver/Workspace app after logon
Protocol handler skip double-hop check
Resource details
Default
Strict transport security policy duration
90.00:00:00
Enable loopback communication
Enables communication with StoreFront services using the loopback adaptor. Disable this when using Fiddler debugging. Default: On
OK
Cancel
Apply

Supported versions

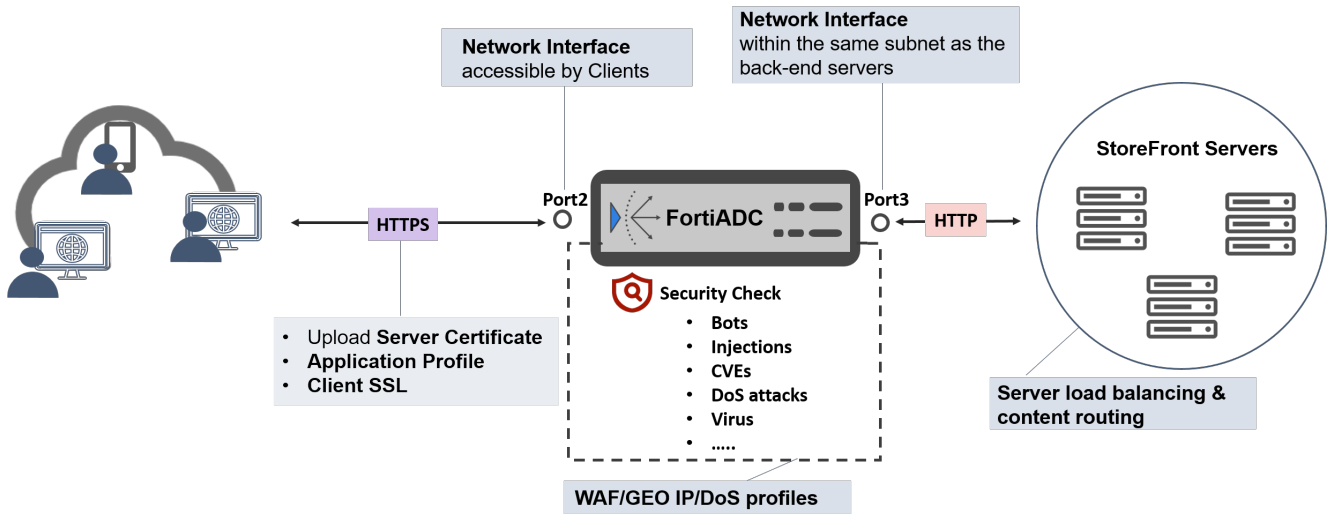
Product	Version
FortiADC	All versions
Citrix DaaS (formerly Citrix Virtual Apps and Desktops service), and Citrix StoreFront	All versions

Configurations for solution 1: FortiADC processing traffic to StoreFront only

In this solution, FortiADC handles traffic exclusively for StoreFront servers. When a user initially accesses the Citrix VDI environment, the traffic first passes through FortiADC for SSL offloading and security inspection. FortiADC then forwards the traffic to the appropriate StoreFront server.

The following diagram illustrates the traffic flow between the client, FortiADC, and StoreFront servers, along with the main configuration steps corresponding to each component's role in the process.

This diagram focuses on configuration details. To view a broader overview of the traffic flow—including Citrix VDI components such as the Delivery Controller—please refer to the section Network topology of integrating FortiADC with Citrix VDI on page 7.



Configuring network interfaces for the incoming and outgoing traffic

When FortiADC is integrated into a Citrix VDI environment, it takes over the traffic destined for the StoreFront servers. All client traffic is routed through FortiADC before reaching the backend StoreFront servers. To enable this, FortiADC must have its network interfaces properly configured to receive incoming client connections and forward traffic to the backend servers.

Configuration Steps

- **Configure a Network Interface for the incoming traffic from the client**

  To enable FortiADC to handle traffic destined for StoreFront servers (and for VDA servers in Solution 2), you need to configure a physical or VLAN interface on FortiADC. This interface will later be bound to a virtual server (see Configuring a virtual server to link incoming and outgoing traffic on page 30).

  This network interface should be:

  - **Publicly accessible if users connect over the Internet, or**
  - **Within the same internal network as the client, if users access StoreFront internally.**

  It is recommended to assign the interface an IP range, such as 10.1.0.140/24. This allows you to:

  - Assign any IP within that range to a virtual server , and
  - Bind multiple virtual servers to the same network interface.

  For example, you can bind both the StoreFront virtual server and the VDA virtual server to this single interface using different IPs from the same subnet.

  Once configured, any client traffic addressed to the virtual server IP will enter FortiADC through this interface, enabling it to process the request (e.g., SSL offloading, load balancing) and forward it to the appropriate backend server.

- **Configure a Network Interface for outgoing traffic to back-end servers**

  To enable FortiADC to distribute traffic to the backend servers, you need to **configure a physical or VLAN interface on FortiADC that resides in the same subnet as the StoreFront and VDA servers**.

  However, if FortiADC is only used to process traffic for the StoreFront servers, then this interface only needs to reach the StoreFront subnet—connectivity to the VDA servers is not required.

  This interface handles outbound traffic to the backend servers and does not need to be explicitly bound to a virtual server or other configuration objects. FortiADC automatically selects the appropriate interface based on the destination IP address of the outgoing packet, using the one that resides in the same subnet as the target server.

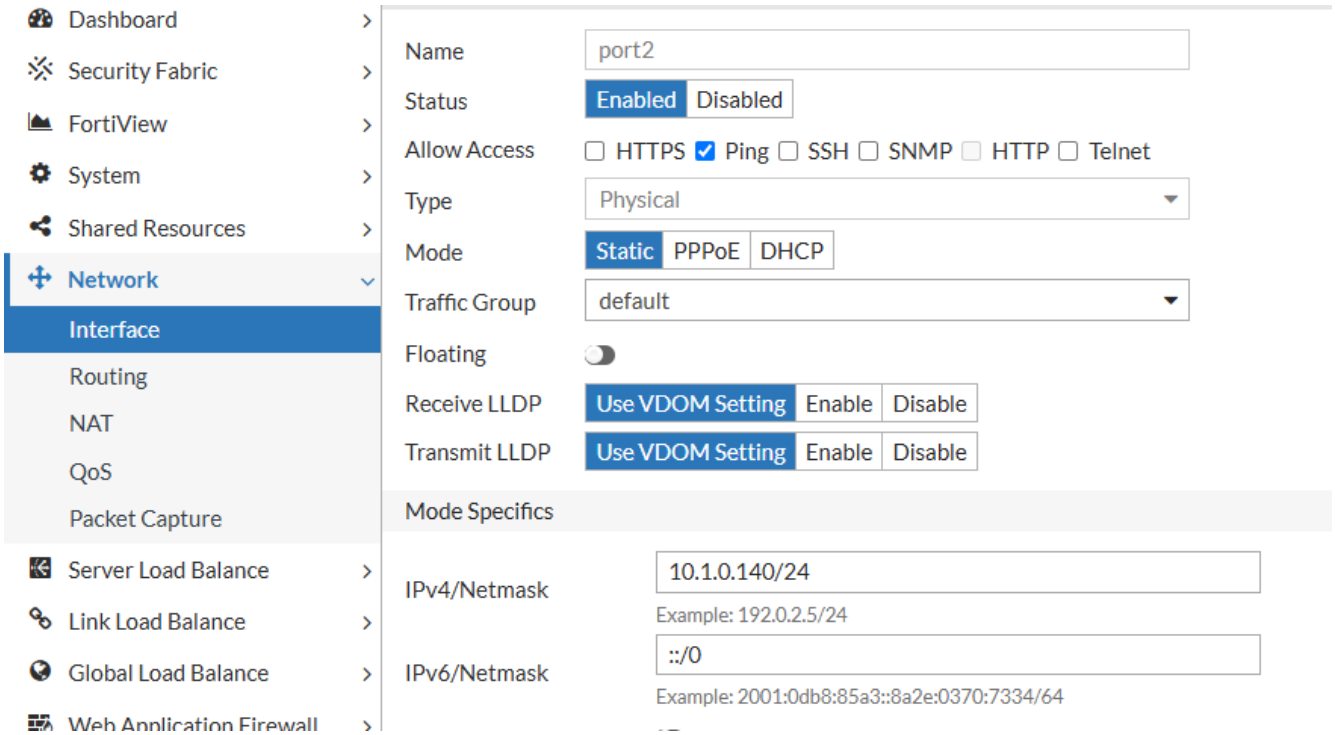
To configure a Network Interface

1. Go to **Networking > Interface** to display the configuration page.
2. Double-click the row for port2, for example, to display the configuration editor.
3. Enter the IP address and other interface settings and save the configuration.

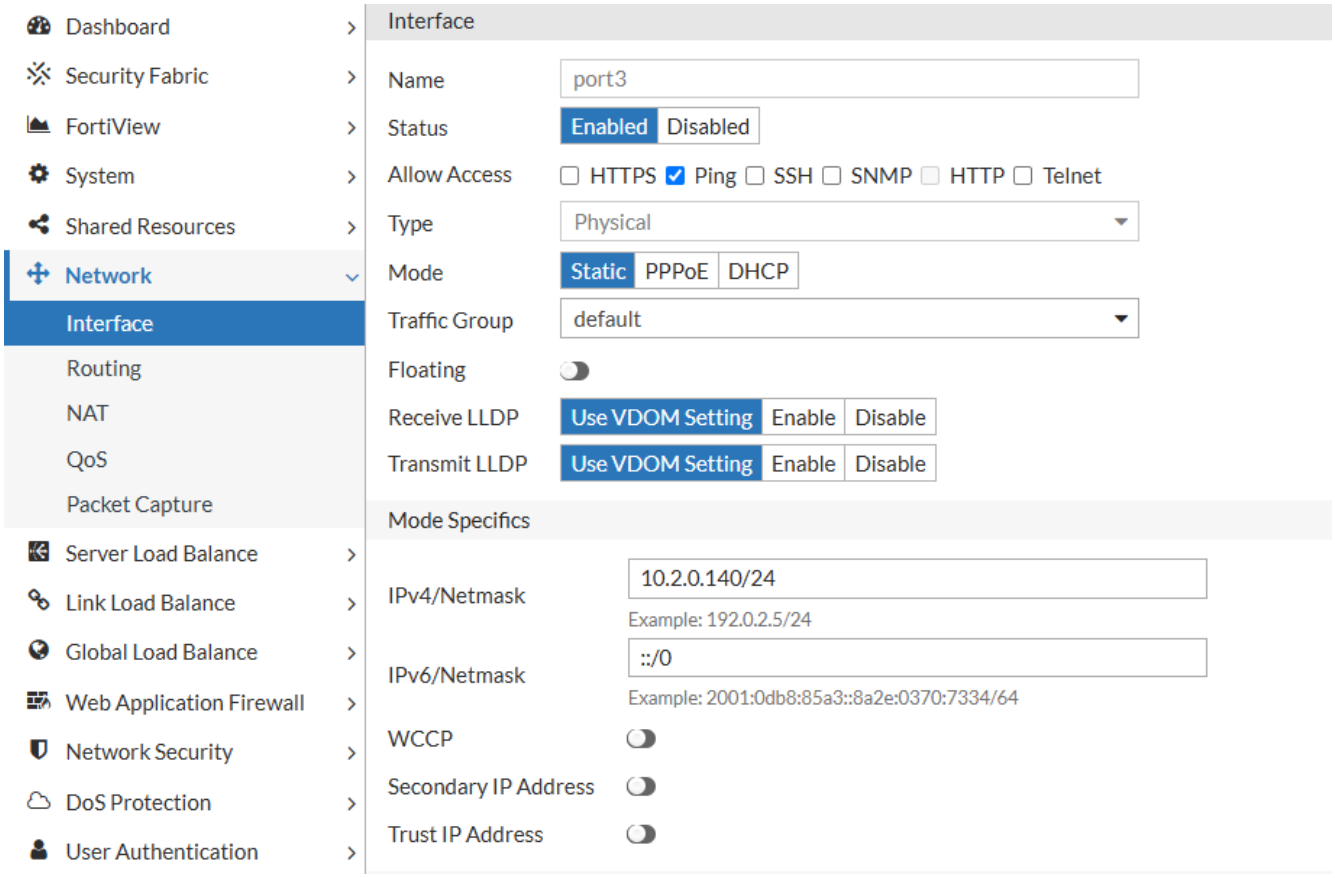
For detailed information, see Configuring network interfaces.

In this example, we have configured Port 2 for the incoming traffic and Port 3 for the outgoing traffic, and created a route entry to reach the gateway used by the back-end servers.

Port 2 settings



Port 3 settings



Configuring an application profile to specify the supported protocols

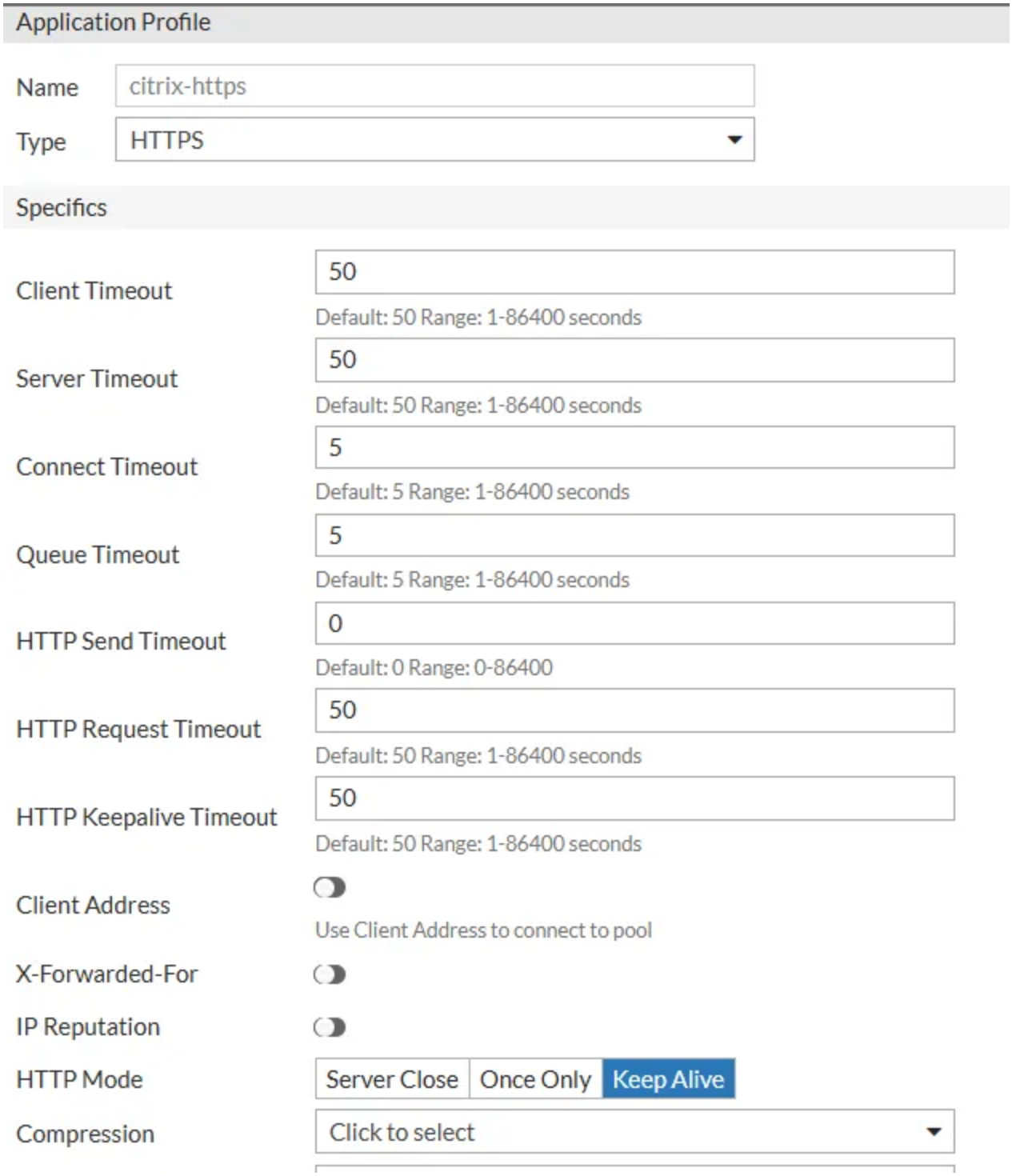
When FortiADC handles SSL connections with clients on behalf of the StoreFront servers, the protocol used is HTTPS. This is configured through an **Application Profile**.

While you may use the predefined HTTPS profile, in this example we will create a custom HTTPS Application Profile. This approach allows flexibility for advanced configurations—such as referencing a Decompression Rule and a Geo IP Protection Rule, which will be added in later steps.

Application Profile for HTTPS

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. Enter a unique name for the **Application Profile** and select the **HTTPS Type**.



Note: For details of each configuration parameter, see the FortiADC Administration Guide for Configuring Application profiles.

4. Save the configuration.

Configuring server certificate and Client SSL for the SSL handshake between FortiADC and the clients

When FortiADC is integrated into a Citrix VDI environment, it takes over the traffic destined for the StoreFront servers. As part of the SSL offloading process, FortiADC performs the SSL handshake with the clients, decrypting incoming connections before they reach the StoreFront servers.

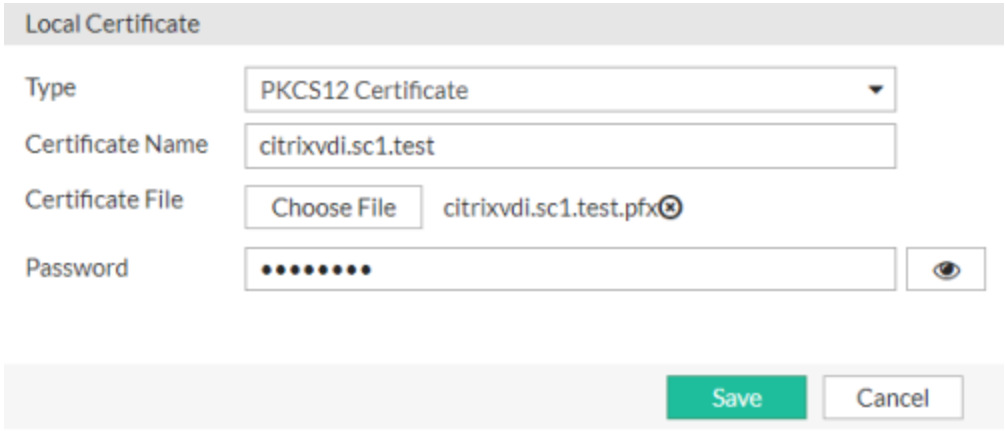
To enable this functionality, FortiADC must be configured with the appropriate server certificate so it can securely represent the StoreFront servers during the handshake.

Configuration Steps

- **Import the StoreFront Server Certificate to FortiADC**
  Upload the StoreFront's server certificate and private key to FortiADC. This allows FortiADC to authenticate itself to clients and complete the SSL handshake on behalf of the StoreFront servers.
- **Configure a Client SSL Profile**
  Create or modify a Client SSL profile to reference the uploaded certificate. In this profile, you can define the supported SSL/TLS versions and cipher suites used during the encryption and decryption process.

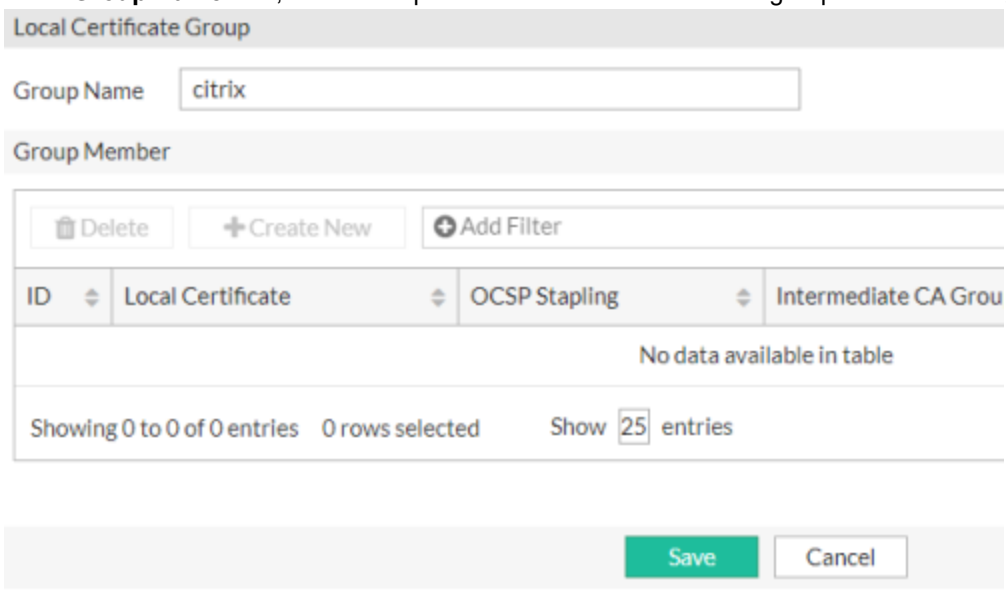
To import the server certificate:

1. Navigate to **System > Manage Certificates**. Click the **Local Certificate** tab.
2. Click **Import** to display the configuration editor.
3. In the Local Certificate configuration editor, select the certificate type that suits your certificate file type and configure the associated settings. Save the configuration.
   In the example below, the **Type** is **PKCS12 Certificate**.
   It is recommended to deploy a wildcard certificate, such as *.storefront.com. This approach supports seamless scalability by allowing new StoreFront servers to be added without the need to replace or update the certificate. For example, FortiADC can use the same wildcard certificate to authorize itself on behalf of domains like a.storefront.com, b.storefront.com, and other subdomains under storefront.com



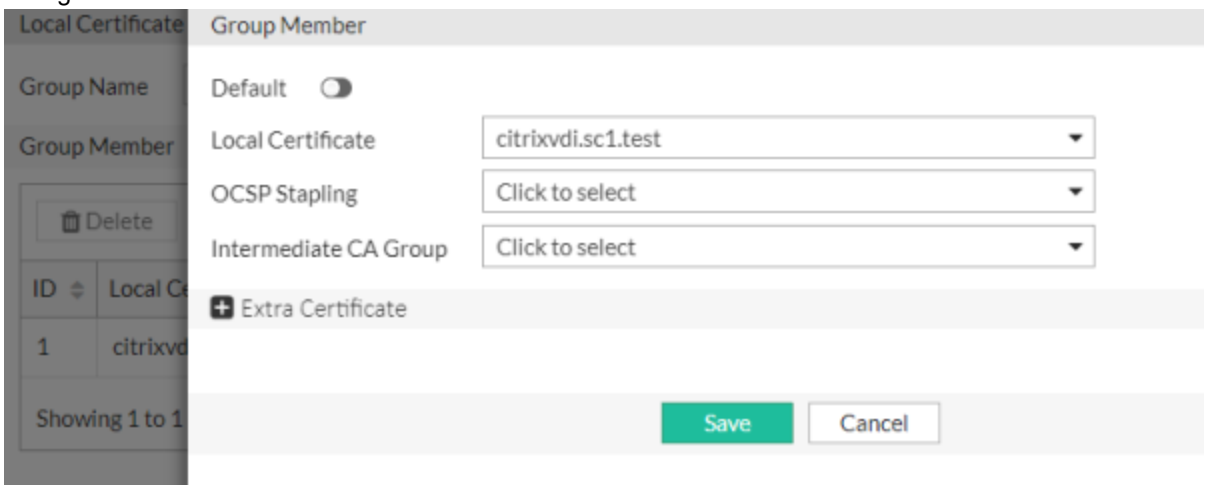
   Note: For details of each configuration parameter, see the FortiADC Administration Guide for Importing a local certificate.
4. Go to the **Local Certificate Group** tab.
5. Click **Create New** to display the configuration editor.

6. In the **Group Name** field, enter a unique name for the local certificate group. Save the configuration.



After the local certificate group is saved initially, the Group Member section becomes available to configure.

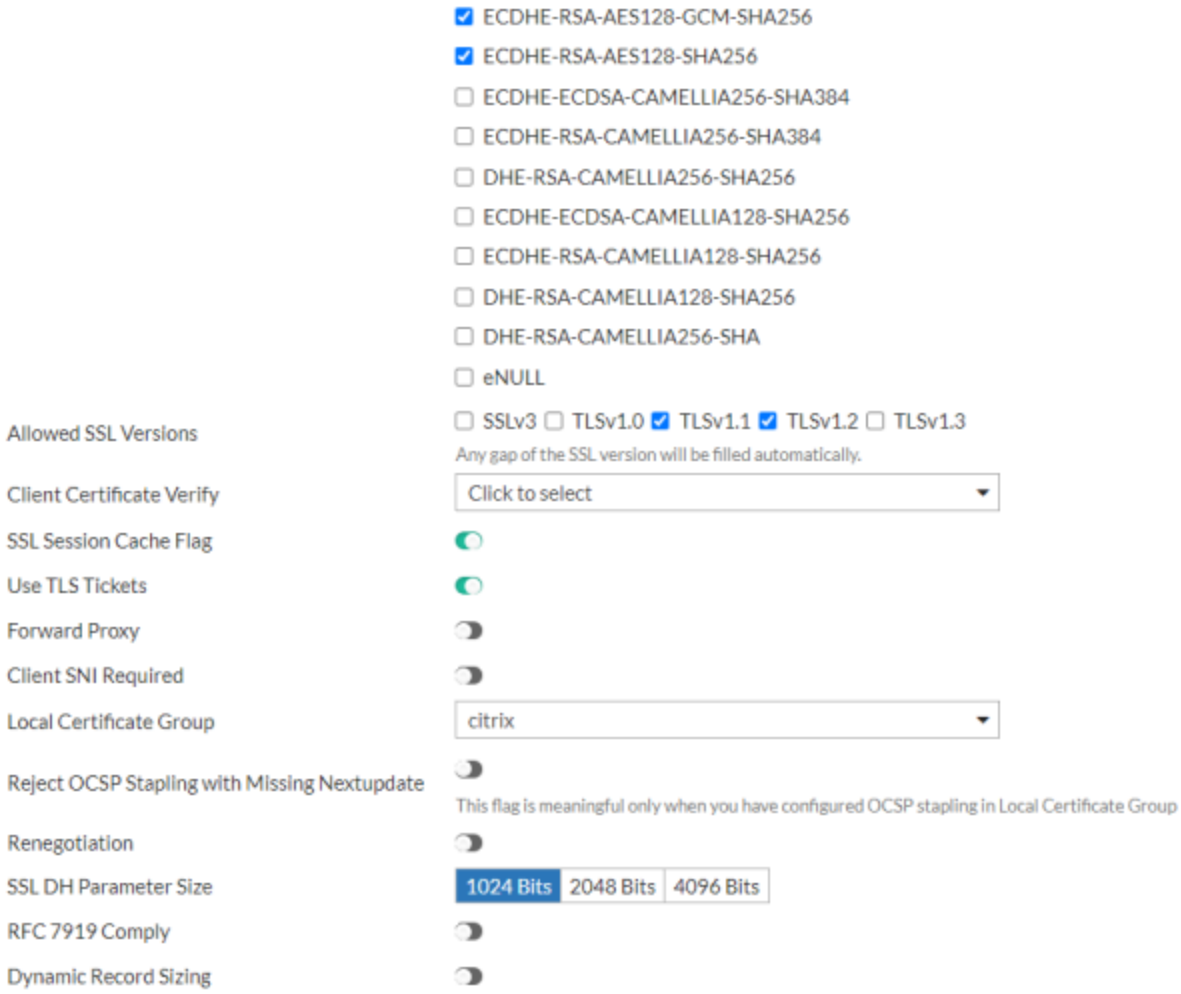
7. Under the **Group Member** section, click **Create New** to display the configuration editor.
8. In the **Local Certificate** field, select the local certificate previously imported from the drop-down menu. Save the configuration.



9. Save the Local Certificate Group configuration to commit the changes for the Group Member.

To configure Client SSL Profile :

1. Navigate to **Server Load Balance > Application Resources**. Click the **Client SSL** tab.
2. Click **Create New** to display the configuration editor.
3. Enter a unique name for the Client SSL Profile and select the **Local Certificate Group** previously created as part of the initial configuration setup. For details, see Importing the Local Certificate.



Note: For details of each configuration parameter, see the FortiADC Administration Guide for Configuring client SSL profiles.

4. Save the configuration.

Configuring a sever pool for StoreFront servers

In this section, configure a Real Server Pool that consists of the Citrix StoreFront servers.

Configuration Steps

- **Creating a Health Check profile**
  This enables FortiADC to perform automated health checks on StoreFront servers. Only servers that pass the health check will receive traffic, ensuring that requests are forwarded exclusively to operational and responsive servers.
- **Creating Real Servers**
  In this step, you define each StoreFront server by specifying its IP address. These entries represent the actual backend servers that FortiADC will forward traffic to.

- **Creating Real Server Pool**

  In this step, you group the StoreFront servers into a server pool and associate it with the health check profile. FortiADC applies the configured load balancing algorithm to distribute traffic across the servers in the pool, ensuring even load distribution and optimal performance.

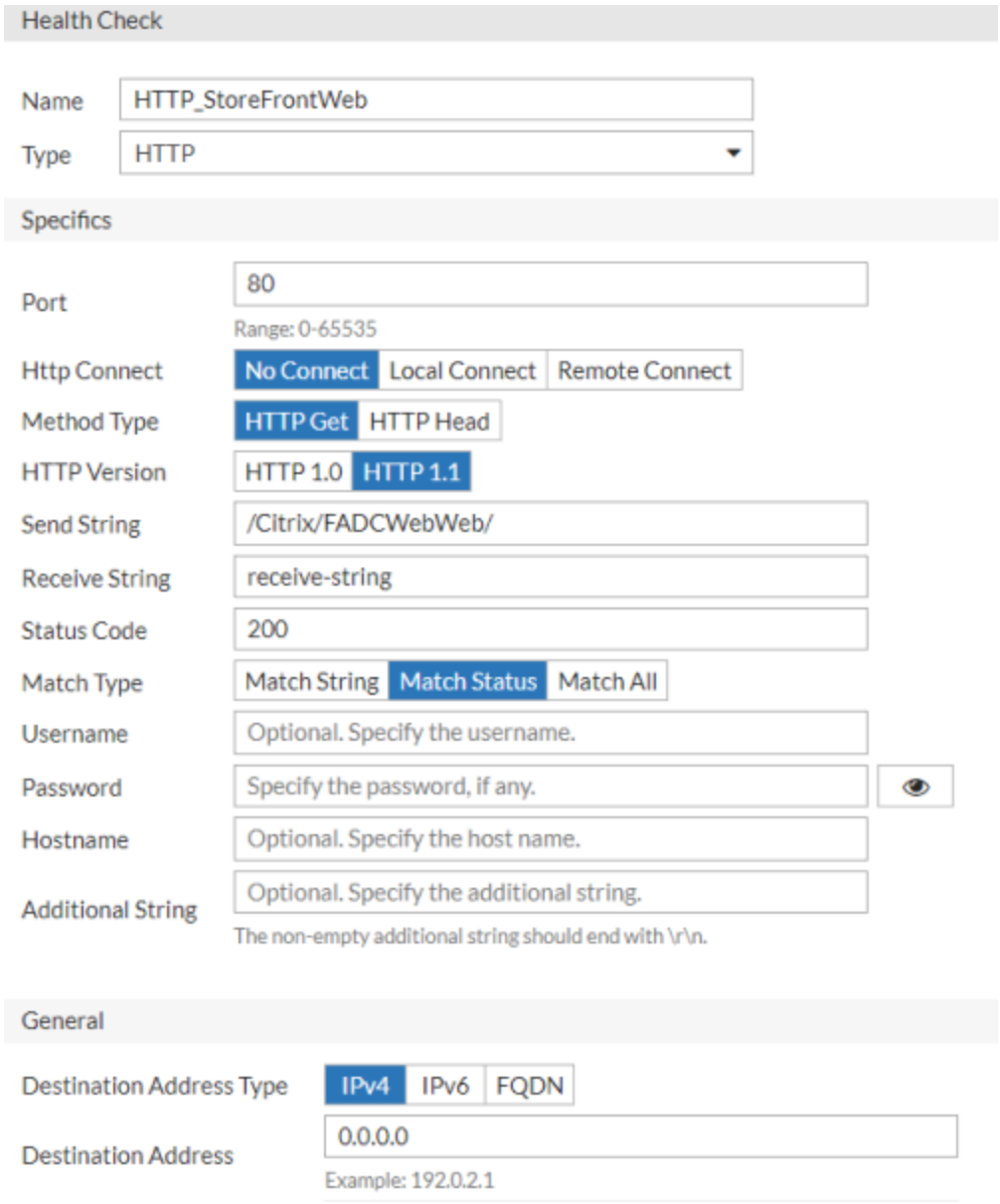
  When adding StoreFront servers to the server pool, specify port 80 to receive traffic from FortiADC.

  This is because FortiADC performs SSL offloading—it terminates the HTTPS connection from the client, decrypts the traffic, and then typically forwards it to the backend using HTTP. Since the communication between FortiADC and StoreFront servers takes place within a secure internal network, using HTTP on port 80 reduces CPU load on the backend servers without compromising security.

To configure a health check profile:

1. Navigate to **Shared Resources > Health Check**.
   The configuration page displays the **Health Check** tab.
2. Click **Create New** to display the configuration editor.
3. Configure the following key settings:

Setting	Guideline
Name	Enter a unique name for the Health Check profile.
Type	Select **HTTP** from the drop-down menu.
Port	The default port number of Citrix StoreFront is 80.
Method Type	Select **HTTP Get**.
Send String	Enter the URL on the Citrix StoreFront server. This is typically `/citrix/{YourStoreName}`.
Match Type	Select **Match Status**.



Note: The configuration parameters detailed in this example applies exclusively to StoreFront servers that operate on the HTTP protocol. FortiADC's Health Check functionality encompasses multiple protocols, including ICMP, TCP, HTTP/HTTPS, DNS, and email protocols. For details, see the FortiADC Administration Guide for Configuring health checks.

4. Save the configuration.

To create a real server:

1. Navigate to **Server Load Balance > Real Server Pool**. Click the **Real Server** tab.
2. Click **Create New** to display the configuration editor.
3. Enter a unique name for the real server. Then, in the **Address** field, enter the IP address of the Citrix StoreFront server.
   Note: For details of each configuration parameter, see the FortiADC Administration Guide for Configuring real

servers.

4. Save the configuration.
5. Repeat the steps to add more real server configurations for each Citrix StoreFront server.

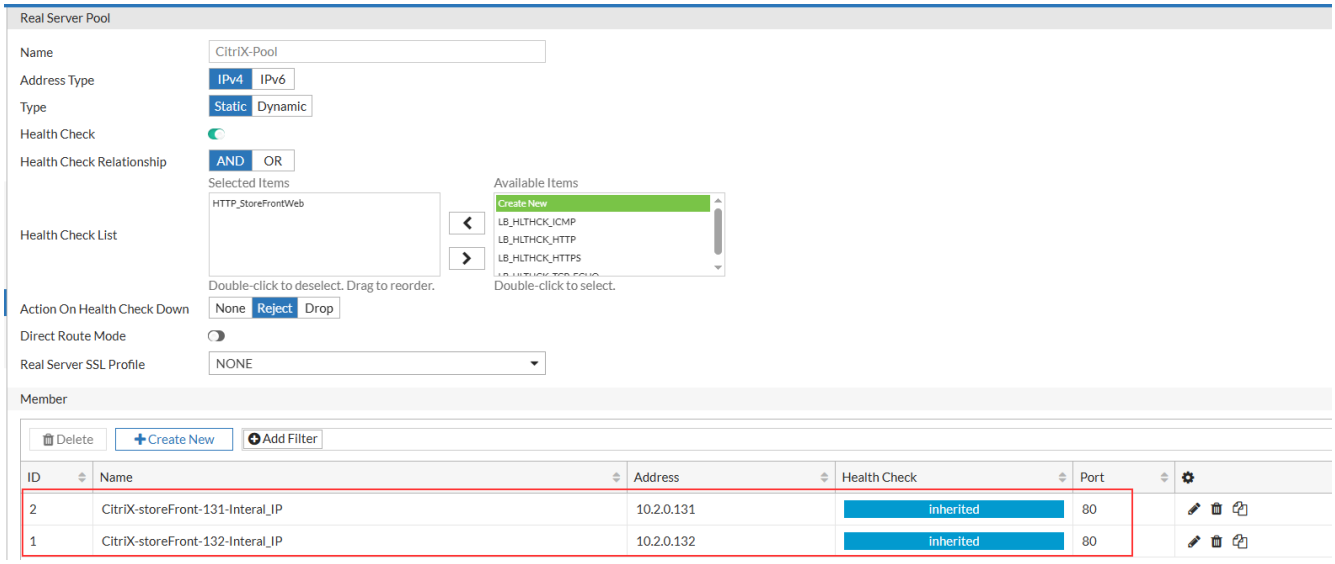
To create a real server pool:

1. Navigate to **Server Load Balance > Real Server Pool**.
   The configuration page displays the **Real Server Pool** tab.
2. Click **Create New** to display the configuration editor.
3. Configure the following key settings:

Setting	Guideline
Name	Enter a unique name for the Real Server Pool.
Health Check	Enable Health Check.
Health Check List	Select the Health Check profile configured as part of the initial configuration setup. For details, see Configuring Health Check.

   Note: For details of each configuration parameter, see the FortiADC Administration Guide for Using real server pools.
4. Save the configuration. After the real server pool configuration is saved initially, the Member section becomes available to configure.
5. Under the **Member** section, click **Create New** to display the configuration editor.
6. Configure Real Server Pool Members for each Real Server configurations previously created for each Citrix StoreFront server. For details, see Real Server configuration.
7. Save the Real Server Pool configuration to commit the changes for the Real Server Members.

StoreFront Server Pool Example



Configuring a virtual server to link incoming and outgoing traffic

In FortiADC, a virtual server serves as the central component that links front-end and back-end configurations. Within the virtual server, you specify the Virtual IP (VIP) for incoming client traffic and reference all previously configured elements—such as the network interface, application profile, real server pool, routing rule, and optional WAF profiles.

This setup enables FortiADC to properly process and forward traffic between the client, FortiADC, and the StoreFront servers, ensuring a complete and functional data path for Citrix VDI access.

To create a virtual server:

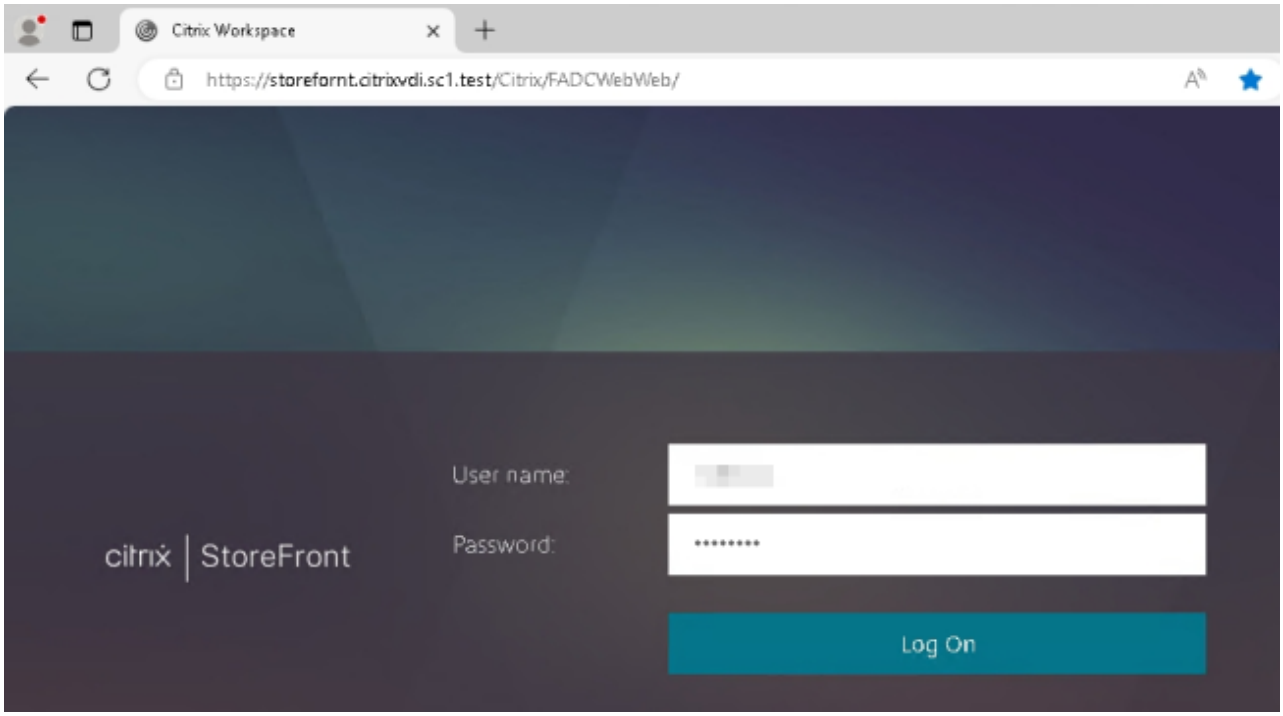
1. Navigate to **Server Load Balance > Virtual Server**.
   The configuration page displays the **Virtual Server** tab.
2. Click **Create New** and select the **Advanced Mode** to display the configuration editor.
3. Configure the following key settings:

Setting	Guideline
Name	Enter a unique name for the Virtual Server.
Type	Select **Layer 7**.
Address	Enter the IP address for the Virtual Server.
Port	Enter **443** for the HTTPS common port
Interface	Select the interface to which you bind the VS.
Profile	Select the Application Profile configuration previously created. For details, see Configuring an application profile to specify the supported protocols on page 22.
Client SSL Profile	Select the Client SSL Profile configuration previously created. For details, see Client SSL Profile configuration.
Persistence	Select the **LB_PERSIS_HASH_SRC_ADDR** profile.
Real Server Pool	Select the Real Server Pool configuration previously created. For details, see Real Server Pool configuration.
WAF Profile	Select **High-Level-Security** (the recommended predefined WAF Profile). For details, see Configuring the WAF Profile on page 59.

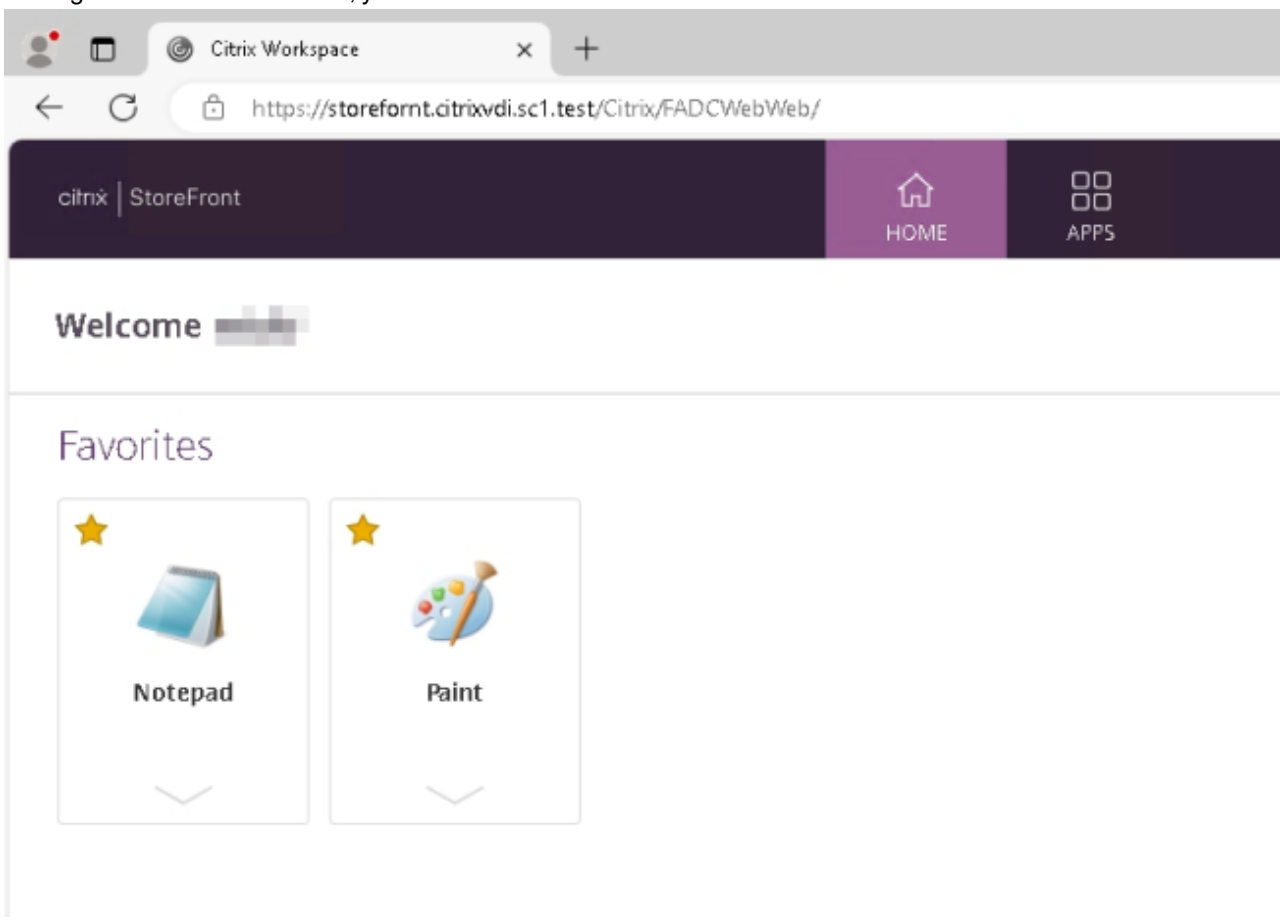
   Note: For details of each configuration parameter, see the FortiADC Administration Guide for Configuring virtual servers.
4. Save the configuration.

Verifying the HTTPS connection of the virtual server to the Citrix StoreFront

1. Open a web browser and enter the URL for accessing your virtual server and Citrix StoreFront service.
2. From the Citrix StoreFront webpage, login with the User name and Password.



3. Through an HTTPS connection, you should be able to access the Citrix StoreFront.



Configurations for solution 2: FortiADC processing traffic to both StoreFront and VDA servers

This advanced deployment extends FortiADC's role to include traffic processing **not only during the initial request to StoreFront, but also throughout the subsequent ICA session with VDA servers.**

By default, after the user is authenticated, an ICA file is returned by StoreFront to the client, containing the connection details for a specific VDA server. However, in Solution 2, FortiADC can **intercept and rewrite the ICA file to embed routing instructions that direct the ICA traffic through FortiADC first, then to the VDA servers**.

This solution delivers end-to-end traffic optimization and security protection, improving performance and user experience in distributed VDI environments.

The key configuration in this deployment is enabling ICA file rewriting on FortiADC using a custom script that replaces the VDA server address with FortiADC's virtual IP address.

For Solution 2, you must first complete the configuration steps outlined in Solution 1. Then, follow the additional steps listed in the sections below to establish communication between the client, FortiADC, and VDA servers.

The diagrams below focus on configuration details of the traffic between clients, FortiADC, and the VDA servers. To view a broader overview of the traffic flow—including Citrix VDI components such as StoreFront and Delivery Controller—please refer to the section Network topology of integrating FortiADC with Citrix VDI on page 7.

Configuration overview

There are two common communication scenarios between the client, FortiADC, and VDA servers, **depending on how the user accesses the Citrix VDI environment.** Each scenario involves different protocols and corresponding FortiADC configurations.

Scenario 1: User Accessing VDI via HTML5 Receiver (Browser-Based Access)

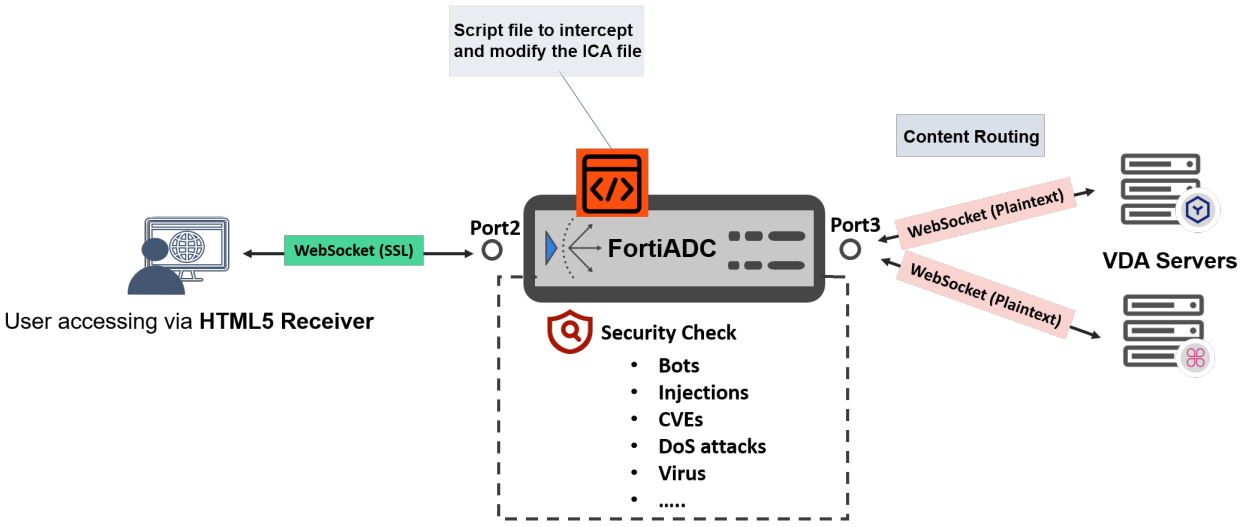
When a user accesses the VDI environment using a modern browser, the session is launched directly in the browser through the **HTML5 Receiver**, rather than using a Citrix Workspace App to open the ICA file.

The communication between the client and the VDA server occurs over secure WebSocket (wss://), typically encapsulated in HTTPS traffic over port 443.

In this scenario, FortiADC can perform SSL offloading and Layer 7 inspection for these connections because it is HTTP-based.

The diagram below outlines the key FortiADC configurations required to handle the WebSocket traffic between the client and the VDA servers.

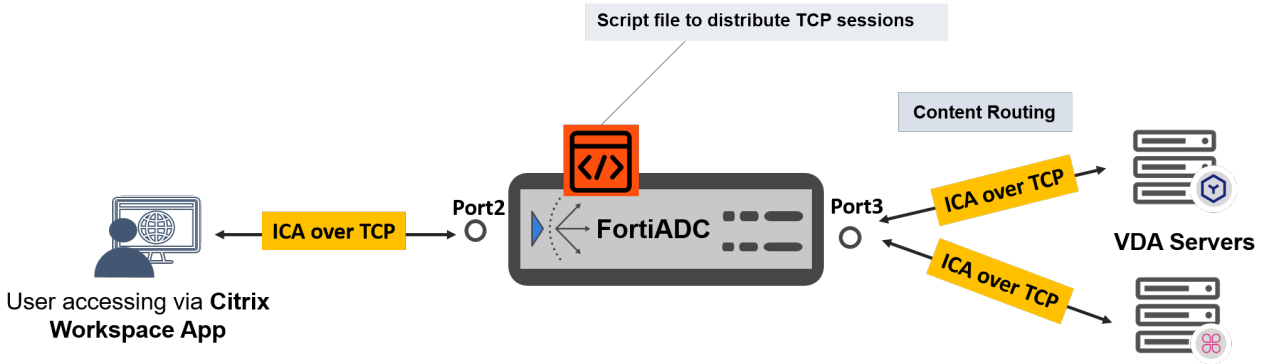
For configuration details, see Configuring FortiADC to handle the WebSocket traffic from HTML5 Receivers on page 34.



Scenario 2: User Accessing VDI via Citrix Workspace App

When users access the VDI environment using the Citrix Workspace App, their connections to the VDA servers use the ICA protocol over TCP. Since ICA is a proprietary protocol, FortiADC cannot decrypt, parse, or inspect ICA traffic, and therefore cannot perform Layer 7 security functions on it.

However, **FortiADC can still play a valuable role by routing the traffic to the corresponding VDA servers**. The diagram below outlines the key FortiADC configurations required to handle ICA over TCP traffic.



For configuration details, see Configuring FortiADC to handle ICA TCP traffic from Citrix Workspace App on page 48.

Configuring FortiADC to handle the WebSocket traffic from HTML5 Receivers

FortiADC can process the WebSocket traffic using the same core components configured in **Solution 1** (which handles StoreFront traffic), with the following additional steps:

- **Configure a Decompression Rule**

  Set up a rule to decompress HTTP responses from the URI that StoreFront uses to deliver ICA files. This enables FortiADC to intercept and rewrite ICA file content.

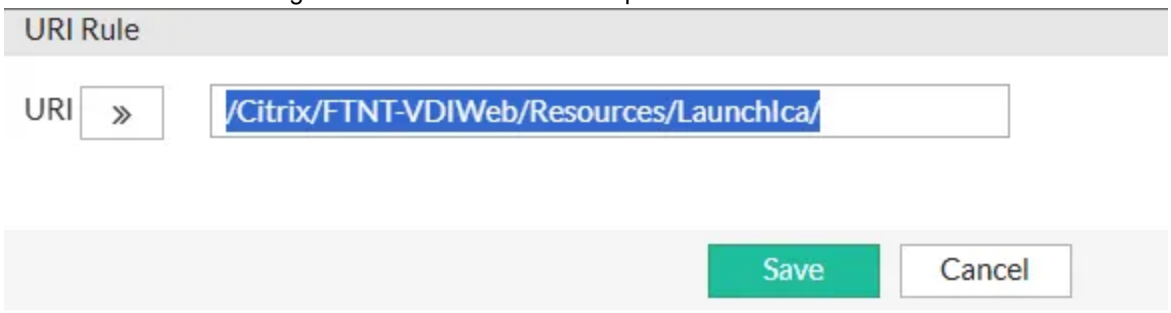
- **Edit the Application Profile**
  Add the decompression rule to the existing HTTPS Application Profile.
- **Add an HTTP Script**
  Create and attach a script that rewrites the ICA file, replacing the original VDA server address with FortiADC's VIP. This ensures that subsequent WebSocket traffic is routed through FortiADC.
- **Add VDA server pools**
  Create a separate server pool for each VDA server
- **Content Routing Rules**
  Create content routing rules to forward traffic to the StoreFront and VDA servers based on the content of **HTTP Host Header**.
- **Edit the Virtual Server**
  Update the existing virtual server to reference:
  - The newly added content routing rules
  - The HTTP script used to rewrite ICA files
- **Enable WebSocket in Citrix VDI**
  Ensure that the relevant WebSocket settings are enabled on the Citrix VDI side.

Configuring the decompression rule

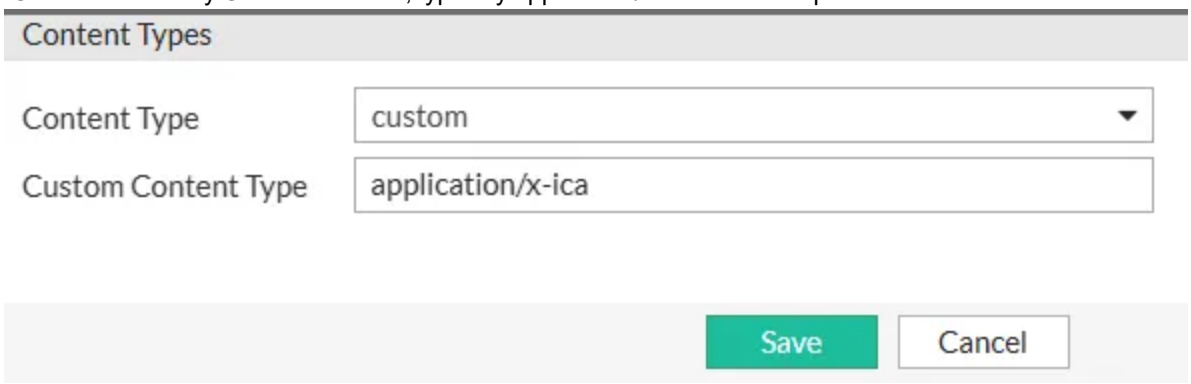
When the ICA file is sent from the StoreFront servers, it's a compression file transferred in HTTP. To be able to parse this file, FortiADC needs to decompress it first.

To configure a decompression rule:

1. Click **Server Load Balance > Application Resources**.
2. Click the **Decompression** tab.
3. Click **Create New** to display the configuration editor.
4. Under the **URI Rule** section, click **Create New** to configure URI members to match the exact URI used by Citrix StoreFront when delivering ICA files to clients. For example:



5. Under the **Content Types** section, click Create New to configure a Content Types to match the MIME type of the ICA file returned by Citrix StoreFront, typically application/x-ica. For example:



6. Save the configuration.

For configuration details, see Configuring decompression rules.

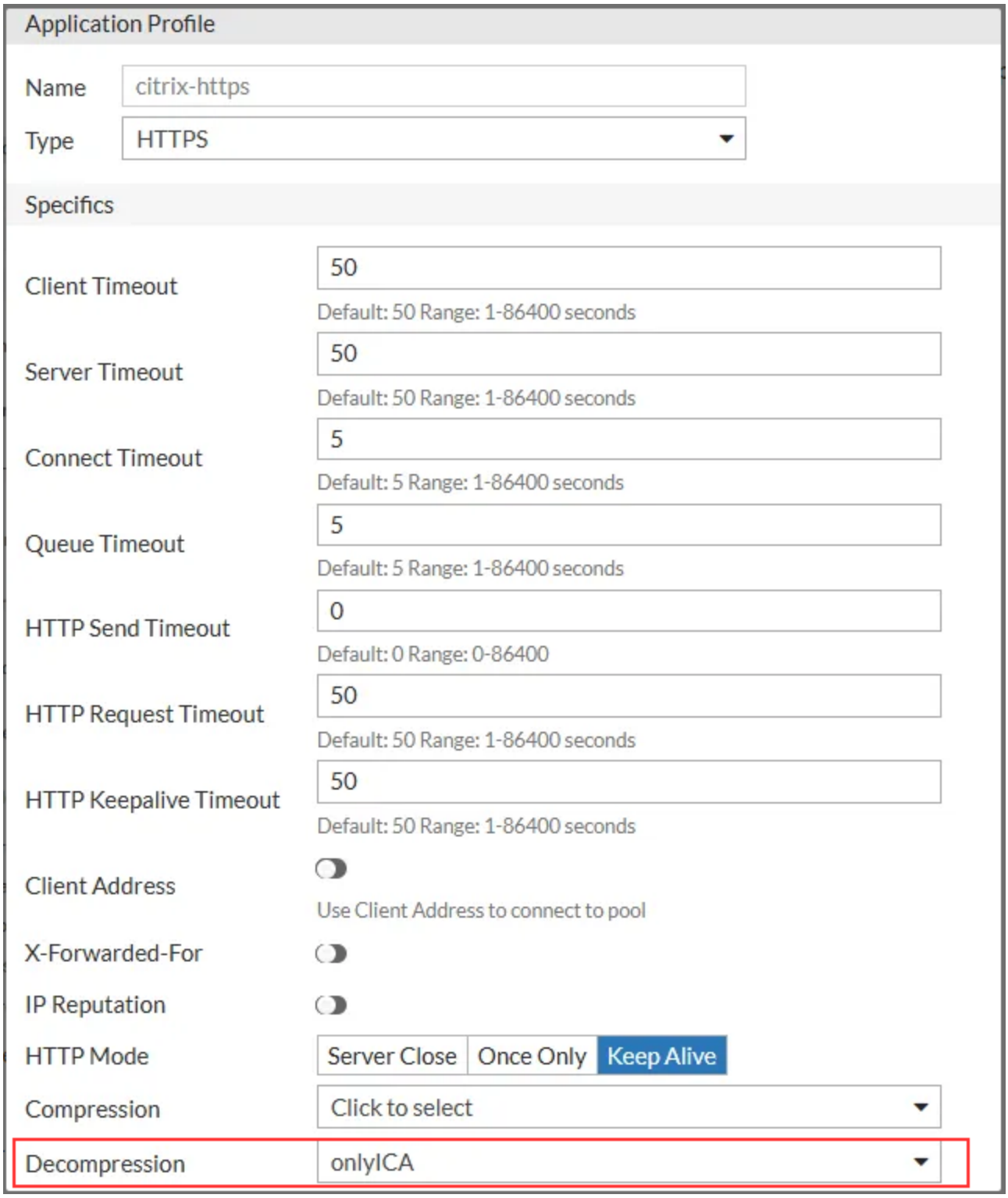
Referencing the decompression rule in the HTTPS Application Profile

The WebSocket traffic shares the same virtual server as the HTTPS traffic to StoreFront, and therefore uses the same Application Profile.

You need to locate the Application Profile you have created in Solution 1 (Configuring an application profile to specify the supported protocols on page 22), and add a reference to the Decompression Rule you just created.

This is necessary so FortiADC can decompress the StoreFront responses (e.g., ICA file delivery) and apply additional processing—such as content modification—required for WebSocket-based VDI sessions.

This is an example of an Application Profile that includes a reference to the Decompression Rule.


Application Profile
Name
citrix-https
Type
HTTPS
Specifics
Client Timeout
50
Default: 50 Range: 1-86400 seconds
Server Timeout
50
Default: 50 Range: 1-86400 seconds
Connect Timeout
5
Default: 5 Range: 1-86400 seconds
Queue Timeout
5
Default: 5 Range: 1-86400 seconds
HTTP Send Timeout
0
Default: 0 Range: 0-86400
HTTP Request Timeout
50
Default: 50 Range: 1-86400 seconds
HTTP Keepalive Timeout
50
Default: 50 Range: 1-86400 seconds
Client Address
Use Client Address to connect to pool
X-Forwarded-For
IP Reputation
HTTP Mode
Server Close
Once Only
Keep Alive
Compression
Click to select
Decompression
onlyICA

Compiling an HTTP script to modify the ICA file

After FortiADC intercept the ICA file from the response from the StoreFront servers, it uses a script to modify ICA file content dynamically, such as:

- Replacing internal IP addresses with the Virtual Server IP (`ica_vs_ip`)
- Updating the FQDN to match the StoreFront domain for HTML5 users
- Adjusting the ICA file based on client source or other policies(Enable SSL)

Add the script in **Server Load Balancing > Scripting > HTTP**.

Below is a sample script tailored for the network topology used in this guide. It may not match your specific deployment environment. If that's the case, you will need to customize or write your own script in Lua to suit your topology and traffic routing requirements.

Strongly recommend you check Reference: Network topology used in the examples of Solution 2 on page 54 to understand this script in the context of network topology.

In this example script, we use the following parameters:

- **ica_vs_ip**

  This is the IP address of the virtual server for handling TCP traffic.

  This parameter is for routing TCP traffic to VDA servers. We will introduce it in Configuring FortiADC to handle ICA TCP traffic from Citrix Workspace App on page 48.
- **IP and host mapping**
  - **10.106.216.135/10.106.216.136**

    Example: 10.106.216.135, 10.106.216.136

    These are the internal IP addresses of your VDA servers that the ICA file originally points to.
  - **proxy_host**

    Format: "*StoreFront_Domain:Port*", e.g., `storefront.citrixvdi.sc1.test:12005`

    This value will replace the VDA server IP in the ICA file, redirecting traffic through FortiADC.

    For example, ICA files originally directing traffic to 10.106.216.135 will now route to `storefront.citrixvdi.sc1.test:12005`.
- **wr_resource_urls**

  This parameter defines the HTTP URL paths where Citrix clients request ICA files. It tells FortiADC which HTTP response to inspect and possibly modify.

  The `%` symbol is used to escape special characters (such as the hyphen `-`) in Lua pattern syntax.

```
when RULE_INIT {
    -- Modify these to fit your setup and environment
    local ica_vs_ip = "10.1.0.200"
    -- You can delete or add more of these

    mapping = {
    -- The FQDN of your StoreFront is "storefront.citrixvdi.sc1.test"
        ["10.106.216.135"] = { vs_ip = ica_vs_ip, proxy_host =
"storefront.citrixvdi.sc1.test:12005" },
        ["10.106.216.136"] = { vs_ip = ica_vs_ip, proxy_host =
"storefront.citrixvdi.sc1.test:12006" }
    }
```

```
    -- Note: In Lua patterns, '^' means start-of-string, and '%' is used to escape special
characters (like '-')
    wr_resource_urls = {
        "^/Citrix/FTNT%-VDIWeb/Resources/LaunchIca/",
        "^/Citrix/FTNT%-VDI/clientAssistant/getIcaFile"
    }
    -- Don't modify anything from here
    collect_response = false
    enable_ssl = false
}

when HTTP_REQUEST {
    local uri = HTTP:uri_get()
    collect_response = false
    enable_ssl = false
    for i = 1, #wr_resource_urls do
        if uri:find(wr_resource_urls[i]) then
            collect_response = true
            debug("Matched ICA Launch URI: %s\n", uri)
            if i==1 then
                enable_ssl = true
            end
            break
        end
    end
}

when HTTP_RESPONSE {
    if collect_response then
        HTTP:collect({})
    end
}

when HTTP_DATA_RESPONSE {
    if not collect_response then return end

    local t = { operation = "size" }
    if HTTP:payload(t) == 0 then return end

    t = { operation = "content" }
    local body = HTTP:payload(t)
    if not body then return end

    local modified_lines, line_count = {}, 0
    local function append_line(line)
        line_count = line_count + 1
        modified_lines[line_count] = line
    end

    local last_matched = nil

    local pending_ssl_line = false

    for line in string.gmatch(body, "[^\r\n]+") do
        local matched_vda = false
```

```
        -- Try matching VDA IPs first
        for vda_ip, map in pairs(mapping) do
            if line:find(vda_ip, 1, true) then
                line = line:gsub(vda_ip, map.vs_ip)
                last_matched = map
                matched_vda = true
                debug("Replaced VDA IP %s → %s\n", vda_ip, map.vs_ip)
                break
            end
        end

        -- Handle SSLEnable=Off (defer injection)
        if enable_ssl and line == "SSLEnable=Off" then
            pending_ssl_line = true

        elseif line:match("^CGPAddress=%*:2598") and last_matched then
            -- Also patch CGP port based on proxy_host
            local port = last_matched.proxy_host:match(":(%d+)$") or "2598"
            append_line("CGPAddress=*:" .. port)
            debug("Rewrote CGPAddress to port %s\n", port)
            pending_ssl_line = nil

        else
            -- Inject SSLProxy if pending and match is ready
            if pending_ssl_line and last_matched then
                append_line("SSLEnable=On")
                append_line("SSLProxyHost=" .. last_matched.proxy_host)
                debug("Injected SSLProxyHost=%s\n", last_matched.proxy_host)
                pending_ssl_line = nil
            end

            append_line(line)
        end
    end

    local new_body = ""
    for i = 1, line_count do
        new_body = new_body .. modified_lines[i] .. "\n"
    end

    t = { operation = "set", data = new_body }
    local ret = HTTP:payload(t)
    if ret then
        debug("ICA content modified successfully.\n")
    else
        debug("Failed to update ICA content.\n")
    end
}
```

Configuring VDA server pools

For VDA servers, FortiADC does not perform load balancing in the traditional sense. Instead, **it simply routes traffic to the correct VDA server based on predefined criteria—typically the destination port.**

To enable this behavior, you need to create a separate server pool for each VDA server, even if each pool contains only one server. Then, you define Content Routing Rules that match the port number in the client request and route the traffic to the corresponding server pool.

In effect, this setup allows FortiADC to map each port to a specific VDA server, ensuring that user sessions are consistently routed to the intended destination.

Here we assume we have two VDA servers, IP addresses are 10.2.0.135 and 10.2.0.136 respectively.

Server Pool Name	Server Name	IP address	Port
VDA-135	VDA-win2022-135-internal-IP	10.2.0.135	8008 *Port 8008 is the default port Citrix uses on the VDA side to receive WebSocket ICA traffic.
VDA-136	VDA-win2022-136-internal-IP	10.2.0.136	8008

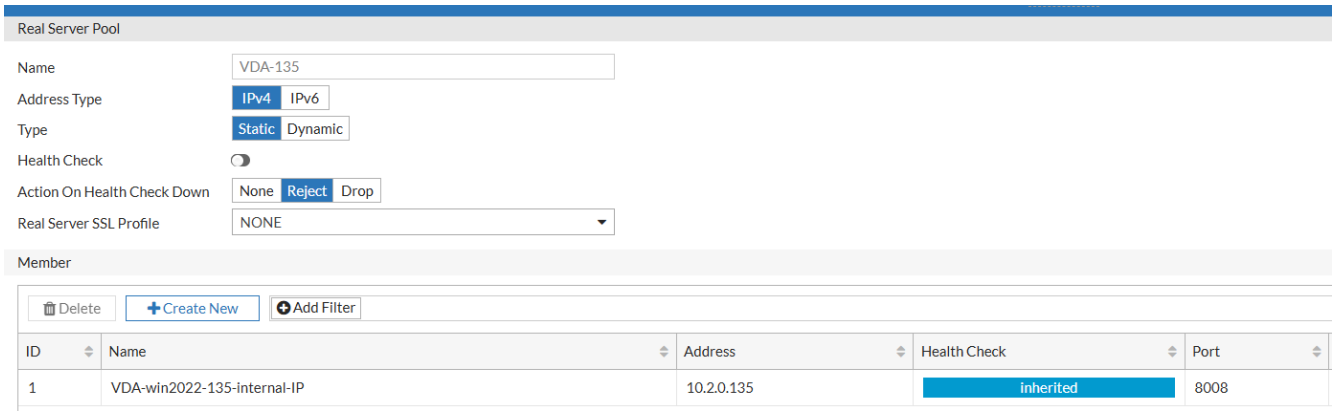
To create a real server:

1. Navigate to **Server Load Balance > Real Server Pool**. Click the **Real Server** tab.
2. Click **Create New** to display the configuration editor. For details of each configuration parameter, see the FortiADC Administration Guide for Configuring real servers.
3. Save the configuration.
4. Repeat the steps to add more real server configurations for each VDA server.

To create a real server pool:

1. Navigate to **Server Load Balance > Real Server Pool**.
   The configuration page displays the **Real Server Pool** tab.
2. Click **Create New** to display the configuration editor. For details of each configuration parameter, see the FortiADC Administration Guide for Using real server pools.
3. Save the configuration. After the real server pool configuration is saved initially, the Member section becomes available to configure.
4. Under the **Member** section, click **Create New** to display the configuration editor. For details, see Real Server configuration.
5. Save the configuration.

VDA Server Pool Example



Configuring content routing rules to route traffic to corresponding server pools

Content Routing (Layer 7 Routing) enables FortiADC to make traffic forwarding decisions based on application-layer details such as the HTTP Host header, URL path, or custom headers.

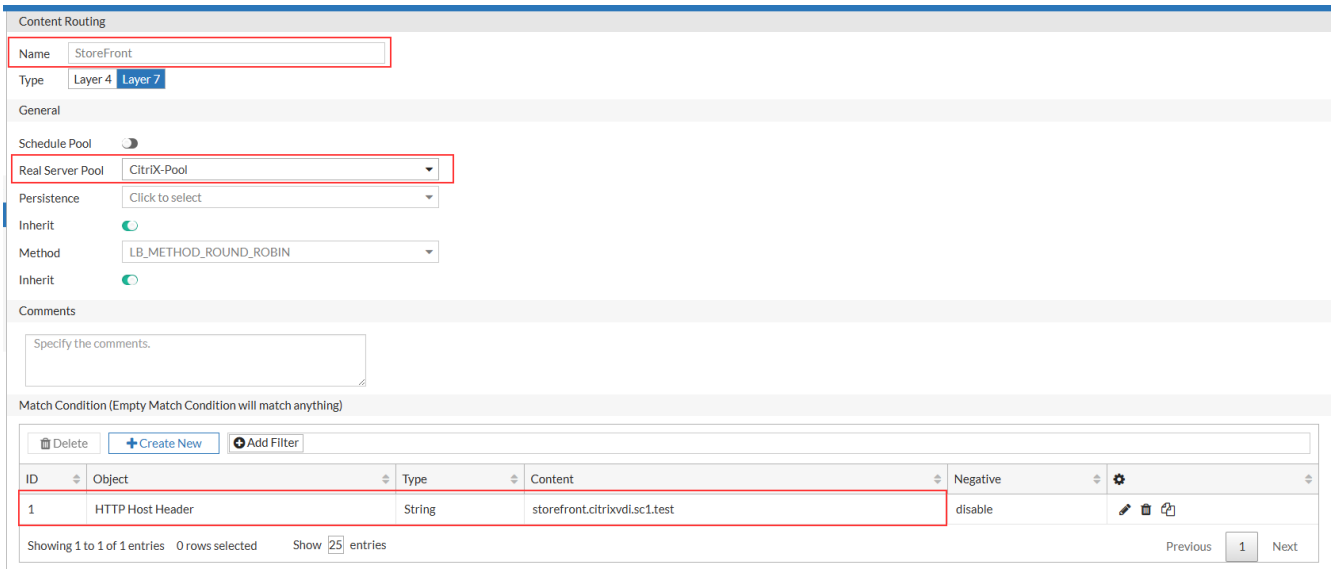
In this configuration, we create content routing rules because the traffic destined to the IP address 10.1.0.140 should be forwarded to both StoreFront and VDA servers. Since these types of traffic use different Host headers, content routing is necessary to properly distinguish and route them to their respective server pools. We will create three content routing rules respectively for the StoreFront server pool, and the two VDA server pools we have created in the previous step.

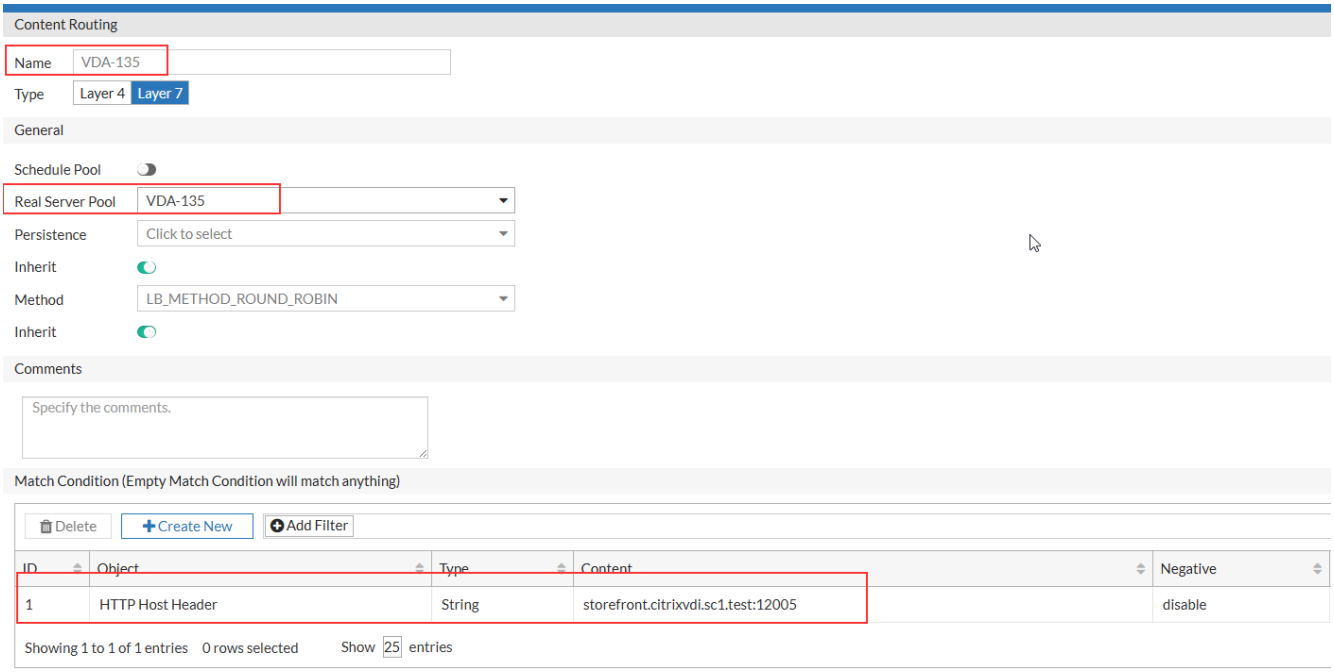
Content Routing Rule Name	Match Object	Match Content	Route to Server Pool
VDA-135	**HTTP Host Header**	storefront.citrixvdi.sc1.test:12005	VDA-135
VDA-136	**HTTP Host Header**	storefront.citrixvdi.sc1.test:12006	VDA-136
StoreFront	**HTTP Host Header**	storefront.citrixvdi.sc1.test	CitriX-Pool (the StoreFront Server pool)

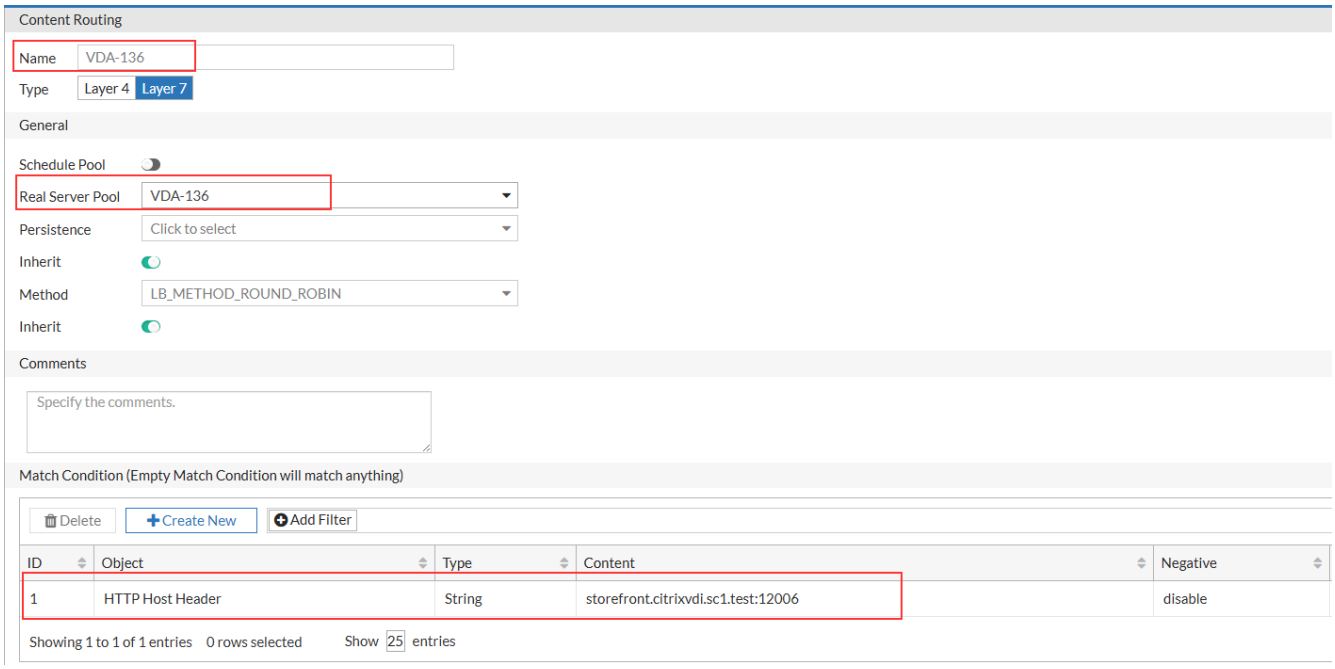
We recommend referring to the Reference: Network topology used in the examples of Solution 2 on page 54 to understand the overall traffic flow design, which will help you configure the content routing rules more effectively.

To create content routing rule:

1. Go to **Server Load Balance > Virtual Server**.
2. Click the **Content Routing** tab.
3. Click **Create New** to display the configuration editor. For configuration details, see Configuring content routes.
4. Save the configuration.
   Once saved, the new Content Routing configuration will be listed in the Content Routing page.


Content Routing
Name StoreFront
Type Layer 4 Layer 7
General
Schedule Pool
Real Server Pool CitriX-Pool
Persistence Click to select
Inherit
Method LB_METHOD_ROUND_ROBIN
Inherit
Comments
Specify the comments.
Match Condition (Empty Match Condition will match anything)
Delete
Create New
Add Filter
ID Object Type Content Negative
1 HTTP Host Header String storefront.citrixvdi.sc1.test disable
Showing 1 to 1 of 1 entries 0 rows selected Show 25 entries
Previous 1 Next


Content Routing
Name VDA-135
Type Layer 4 Layer 7
General
Schedule Pool
Real Server Pool VDA-135
Persistence Click to select
Inherit
Method LB_METHOD_ROUND_ROBIN
Inherit
Comments
Specify the comments.
Match Condition (Empty Match Condition will match anything)
Delete
Create New
Add Filter
ID Object Type Content Negative
1 HTTP Host Header String storefront.citrixvdi.sc1.test:12005 disable
Showing 1 to 1 of 1 entries 0 rows selected Show 25 entries



Editing the previously created virtual server to reference the new settings

Since both the HTTPS traffic to StoreFront servers and the WebSocket traffic to VDA servers use the same protocol (HTTPS) and arrive through the same Virtual IP (VIP) and network interface, they can be handled by the same virtual server in FortiADC.

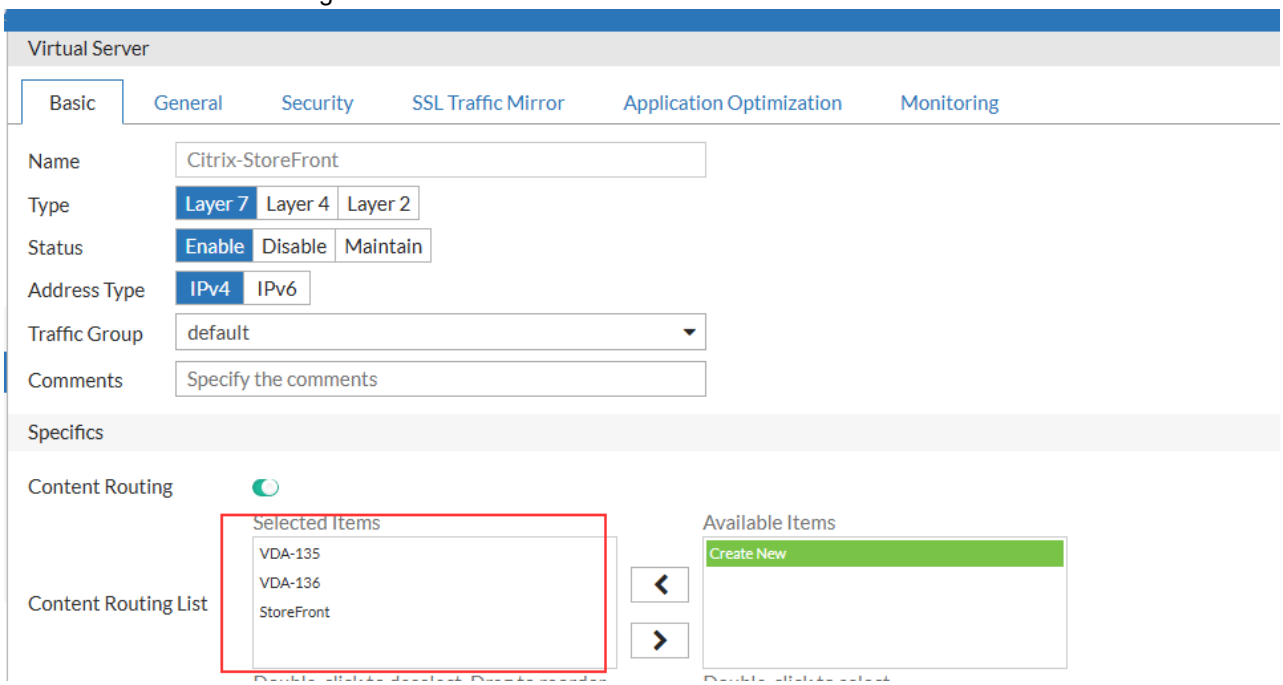
Find the virtual server you previously created for the HTTPS traffic to StoreFront (Configuring a virtual server to link incoming and outgoing traffic on page 30), and update it to reference:

- The newly added content routing rules that direct traffic to VDA and StoreFront Servers.
- The HTTP script used to rewrite ICA files so that client requests to the VDA servers are routed through FortiADC.
- Specify the port numbers FortiADC will use to receive traffic for VDA servers. These must match the port mappings defined in your ICA rewrite script (See Compiling an HTTP script to modify the ICA file on page 38).

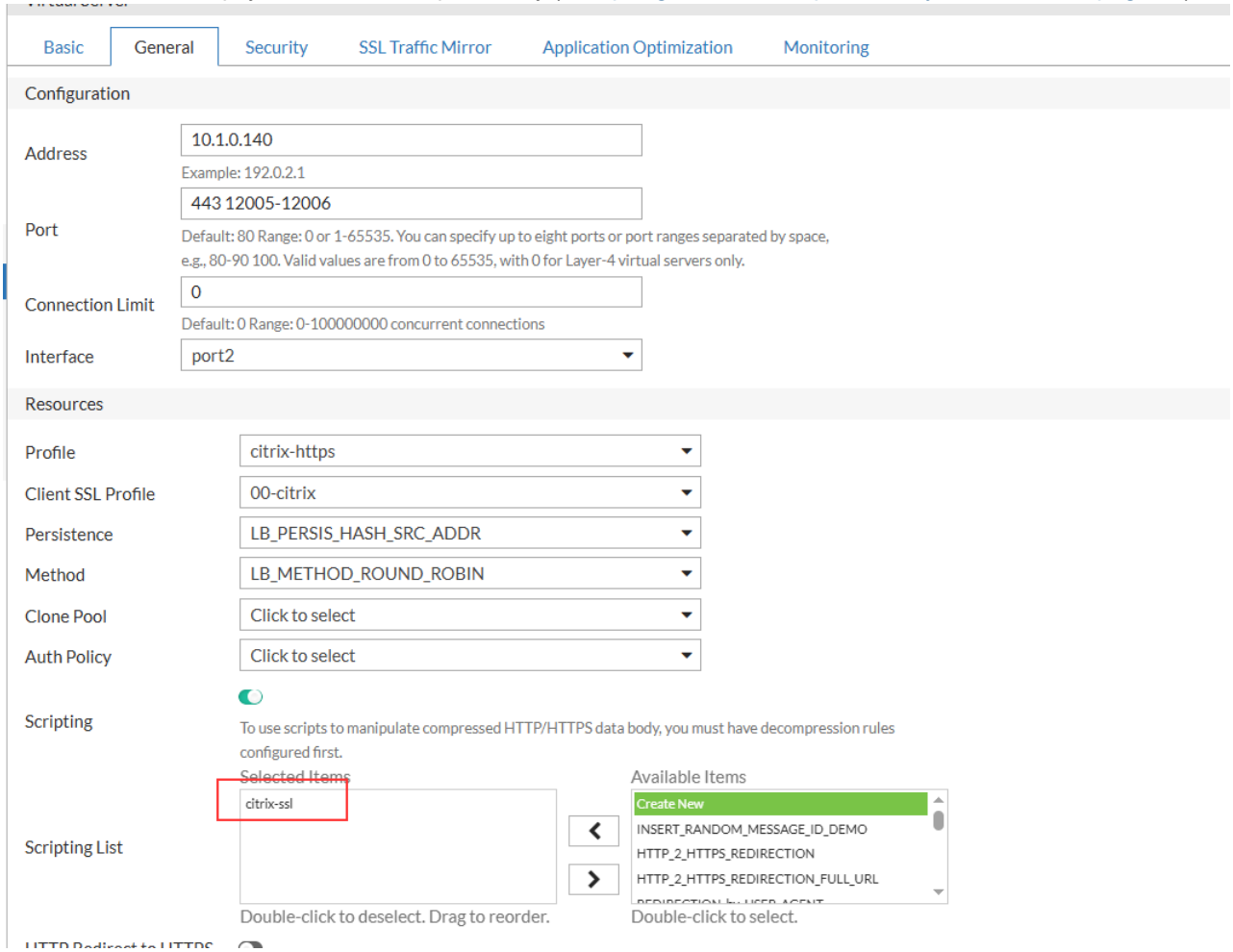
To edit the virtual server:

1. Navigate to **Server Load Balance > Virtual Server**.
   The configuration page displays the **Virtual Server** tab.
2. Find the virtual server you have created for the StoreFront traffic, and click **Edit**.

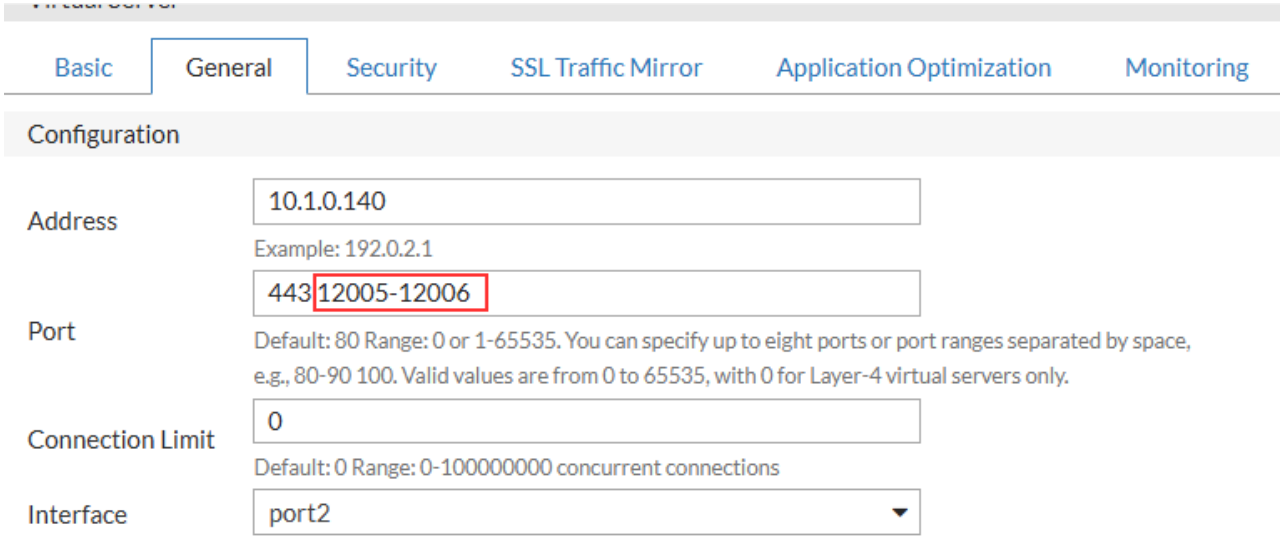
3. Reference the content routing rules.



4. Select the HTTP script you have created previously (Compiling an HTTP script to modify the ICA file on page 38).



5. Set the port numbers FortiADC will use to receive traffic for VDA servers. Strongly recommend you check Reference: Network topology used in the examples of Solution 2 on page 54 to understand how to set Port numbers to allow traffic successfully goes through.

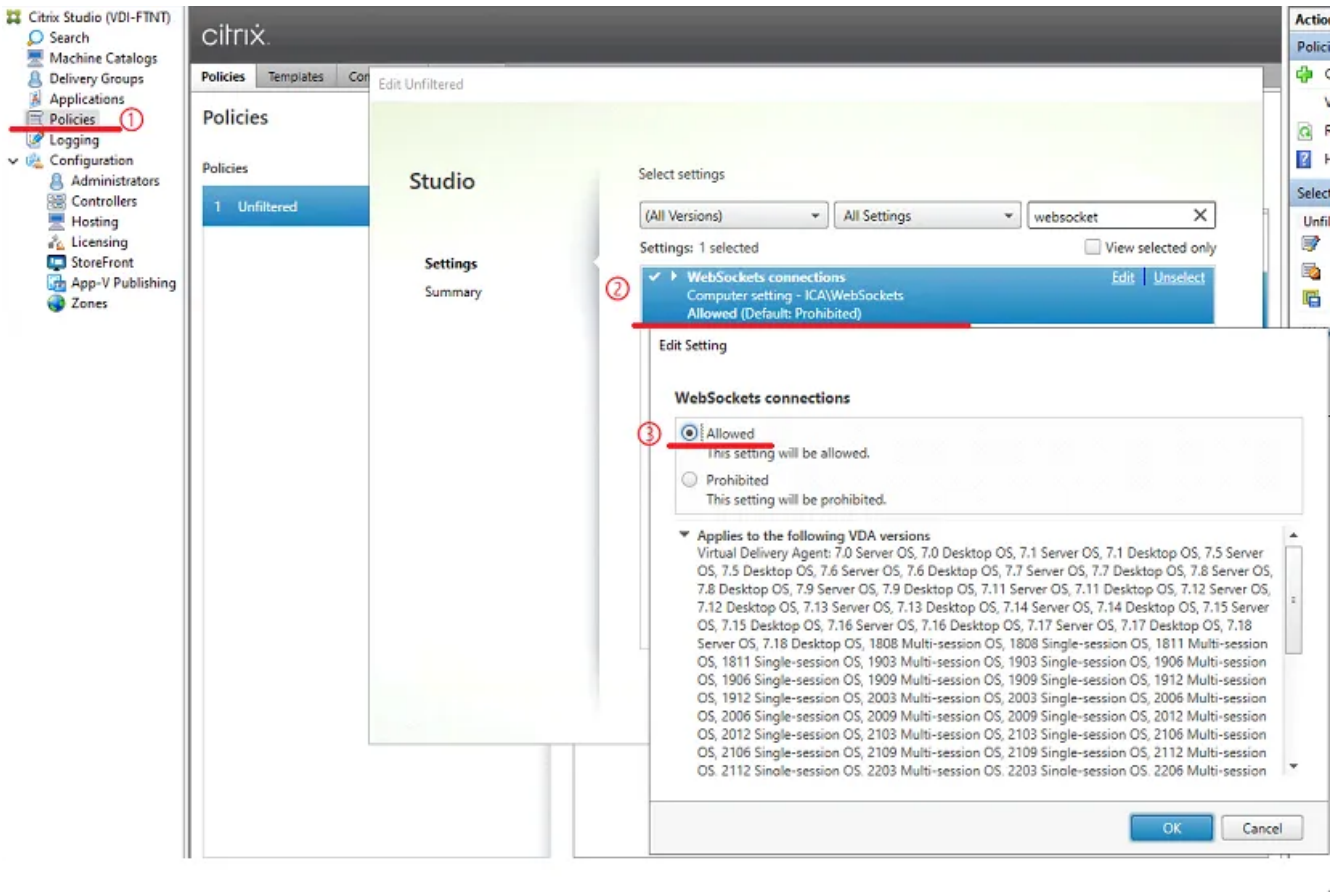


6. Save the configuration.

For details of each configuration parameter, see the FortiADC Administration Guide for Configuring virtual servers.

Enabling WebSocket protocol in Citrix VDI

To use WebSocket protocol in a Citrix VDI environment, ensure that the relevant WebSocket settings are enabled on the Citrix VDI side, as WebSocket support must be configured end-to-end for successful communication.

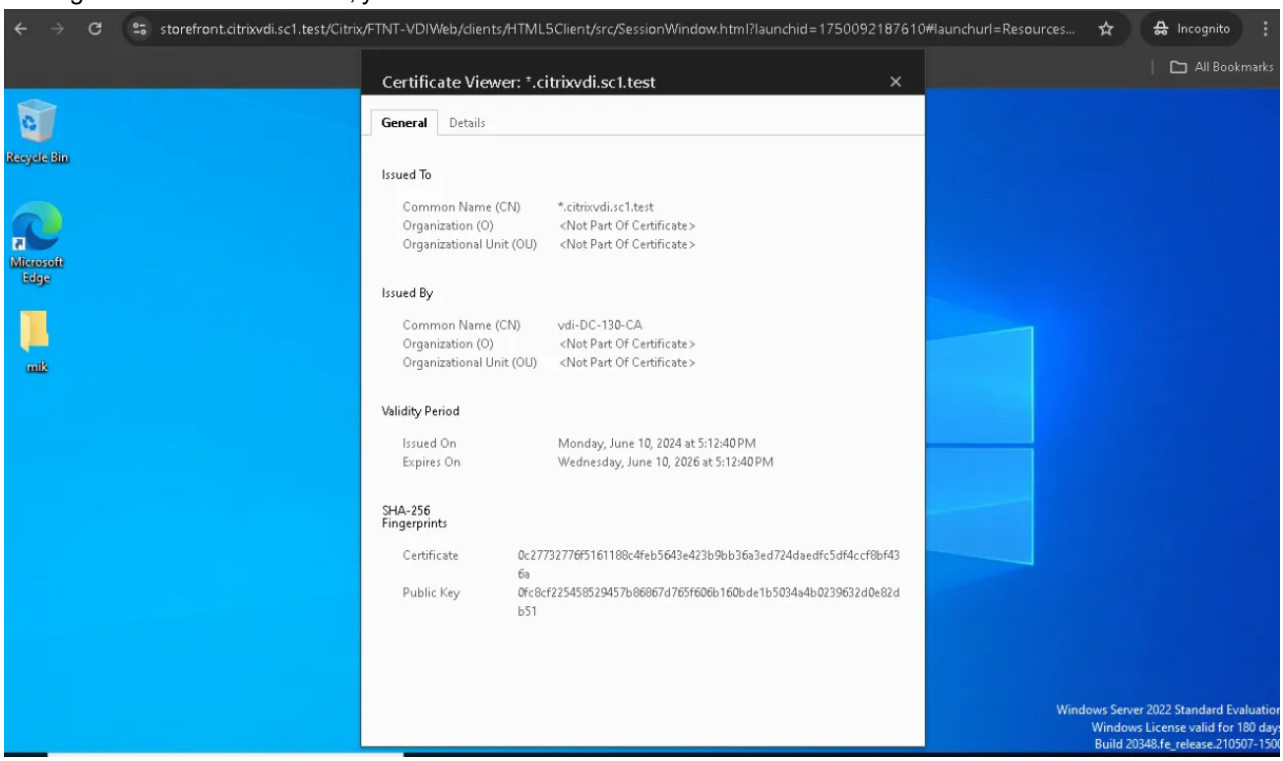


Verifying the WebSocket connection to the VDAs (via HTML5 Receivers)

This section verifies that the FortiADC configuration allows Citrix WorkSpace HTML5 Receivers users to successfully establish WebSocket SSL connections to VDA servers.

1. Open a web browser and enter the URL for accessing your virtual server and Citrix StoreFront service.
2. From the Citrix StoreFront webpage, login with the User name and Password.
3. Click on any available application or desktop to initiate a connection request to the VDA server.

4. Through an HTTPS connection, you should be able to access the Citrix VDAs server.



Configuring FortiADC to handle ICA TCP traffic from Citrix Workspace App

When user accesses VDA servers via Citrix Workspace App, the traffic uses ICA over TCP.

To support ICA traffic over TCP, FortiADC requires the following configurations:

- **Add a TCP Stream Script**
  Configure a TCP stream script to control how TCP sessions are distributed to the VDA servers.
- **Create a server pool**
  Create a server pool to include the VDA servers.
- **Create a Virtual Server for ICA TCP**
  Reference the default layer 7 TCP Application Profile "LB_PROF_L7_TCP", VDA server pool, and the TCP stream script. The ICA TCP virtual server uses the same network interface as the HTTPS/WebSocket virtual server but is assigned a different IP address.

Configuring a sever pool for VDA servers

In this section, you would typically configure a Real Server Pool that includes the VDA servers.

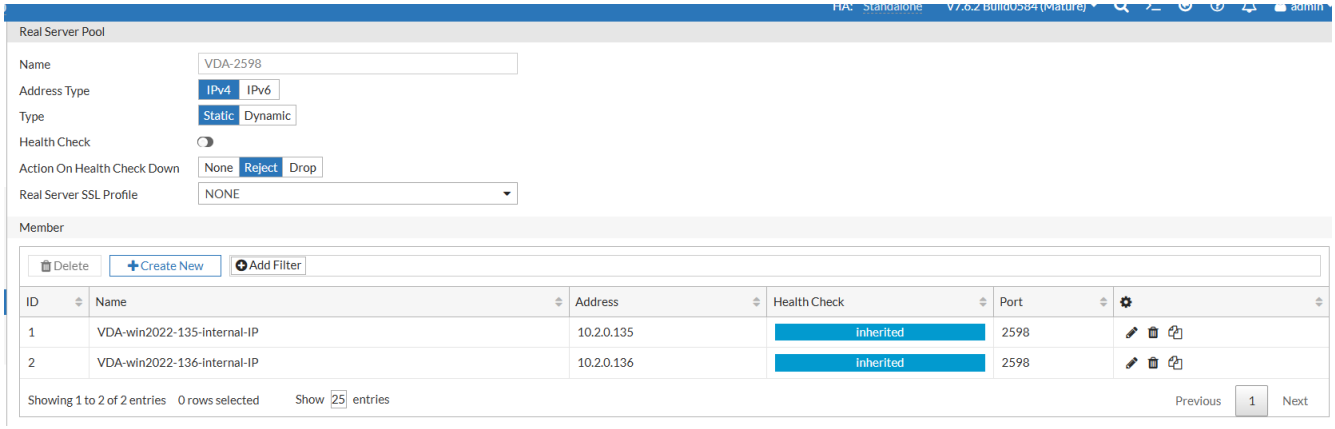
Because both ICA TCP traffic (from Citrix Workspace App) and WebSocket-based ICA traffic (from HTML5 Receiver) are directed to the same set of VDA servers, here we can re-use the VDA servers you have created for the in earlier

steps for WebSocket traffic. In the VDA server pool settings, specify the TCP port number used for ICA traffic. By default, this port is **2598** in Citrix VDI environments.

To create a real server pool:

1. Navigate to **Server Load Balance > Real Server Pool**.
   The configuration page displays the **Real Server Pool** tab.
2. Click **Create New** to display the configuration editor. For details of each configuration parameter, see the FortiADC Administration Guide for Using real server pools.
3. Save the configuration. After the real server pool configuration is saved initially, the Member section becomes available to configure.
4. Under the **Member** section, click **Create New** to display the configuration editor.
5. Add the VDA servers as Real Server Pool Members.
6. Save the Real Server Pool configuration to commit the changes for the Real Server Members.

TCP Server Pool Example



Compiling a streaming script for traffic routing

Add a Stream Script by navigating to **Server Load Balancing > Scripting > Stream**.

Below is a sample TCP script based on the network topology used in the examples throughout this guide.

We strongly recommend reviewing the Reference: Network topology used in the examples of Solution 2 on page 54 to understand the script in its proper context. This will help you adapt the logic to fit your own network deployment accurately.

```
when STREAM_REQUEST_DATA {
    local port=IP:local_port()
       if (port == 12005) then
        LB:upstream("VDA-win2022-135-internal-IP")
       elseif (port == 12006) then
        LB:upstream("VDA-win2022-136-internal-IP")
    end
}
```

Configuring a virtual server to link the incoming and outgoing TCP traffic

In FortiADC, a virtual server serves as the central component that links front-end and back-end configurations. Within the virtual server, you specify the Virtual IP (VIP), network interface, and port number for incoming client traffic and reference the previously configured real server pool and the script.

This setup enables FortiADC to properly route ICA TCP traffic between the client, FortiADC, and the VDA servers, ensuring a complete and functional data path for Citrix VDI access.

Strongly recommend you check Reference: Network topology used in the examples of Solution 2 on page 54 to understand how to set the IP and Port numbers of the TCP virtual server.

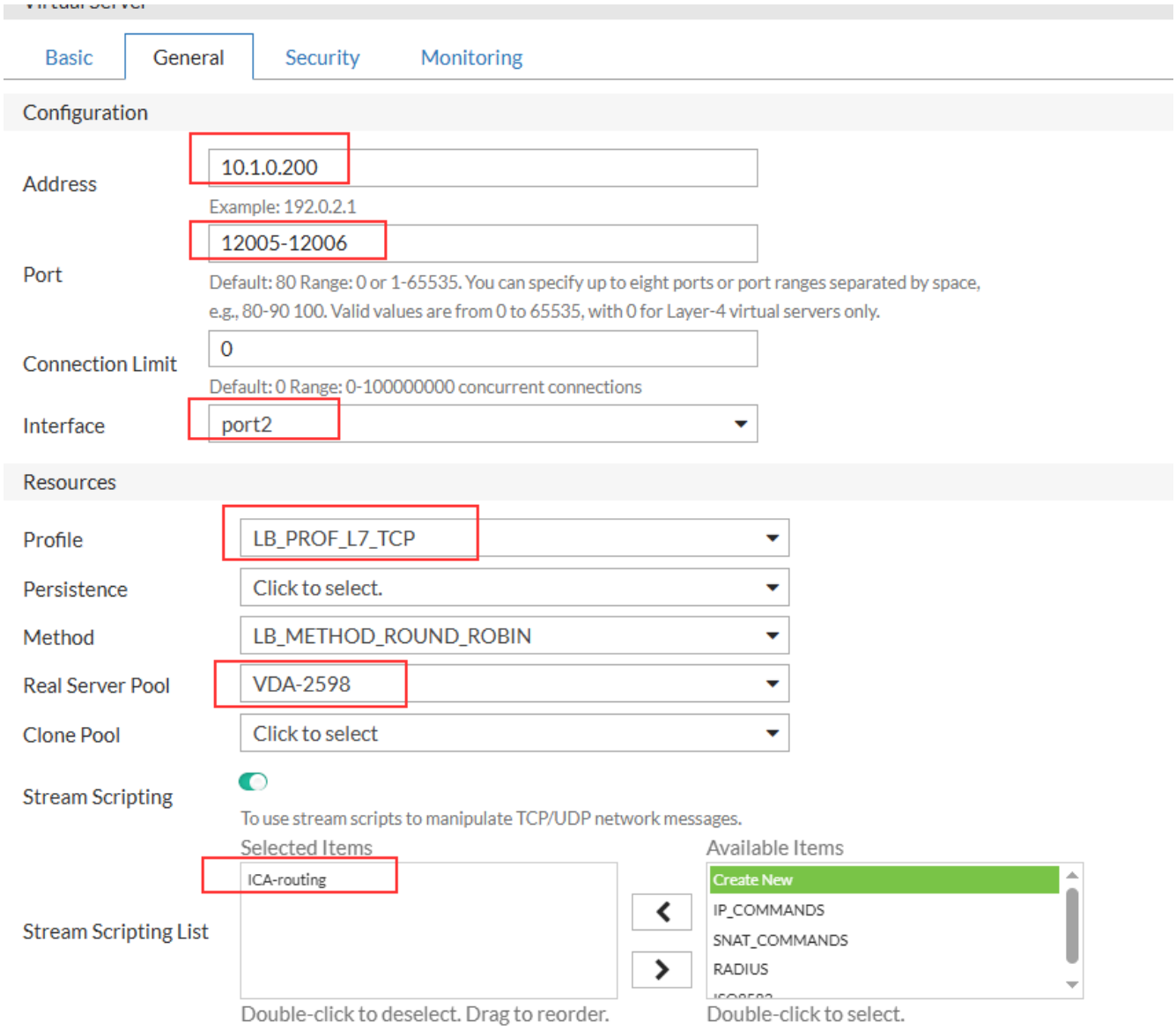
To create a virtual server:

1. Navigate to **Server Load Balance > Virtual Server**.
   The configuration page displays the **Virtual Server** tab.
2. Click **Create New** and select the **Advanced Mode** to display the configuration editor.
3. Configure the following key settings:

Setting	Guideline
Name	Enter a unique name for the Virtual Server.
Type	Select **Layer 7**.
Address	TCP traffic to FortiADC is destined to this IP address. In this example, set it to 10.1.0.200.
Port	Set the port numbers FortiADC will use to receive traffic for VDA servers. You have specified it in the Stream Script: Compiling a streaming script for traffic routing on page 49. In this example, these port numbers are 12005 and 12006.
Interface	Select the same network interface that is used by the HTTPS/WebSocket virtual server.
Profile	Select the default Application Profile **LB_PROF_L7_TCP**.
Real Server Pool	Select the Server Pool previously created. For details, see Real Server Pool configuration.
Stream Scripting	Enable.
Stream Scripting List	Select the Stream Script you previously created.

   Note: For details of each configuration parameter, see the FortiADC Administration Guide for Configuring virtual servers.
4. Save the configuration.

TCP Traffic Virtual Server Example

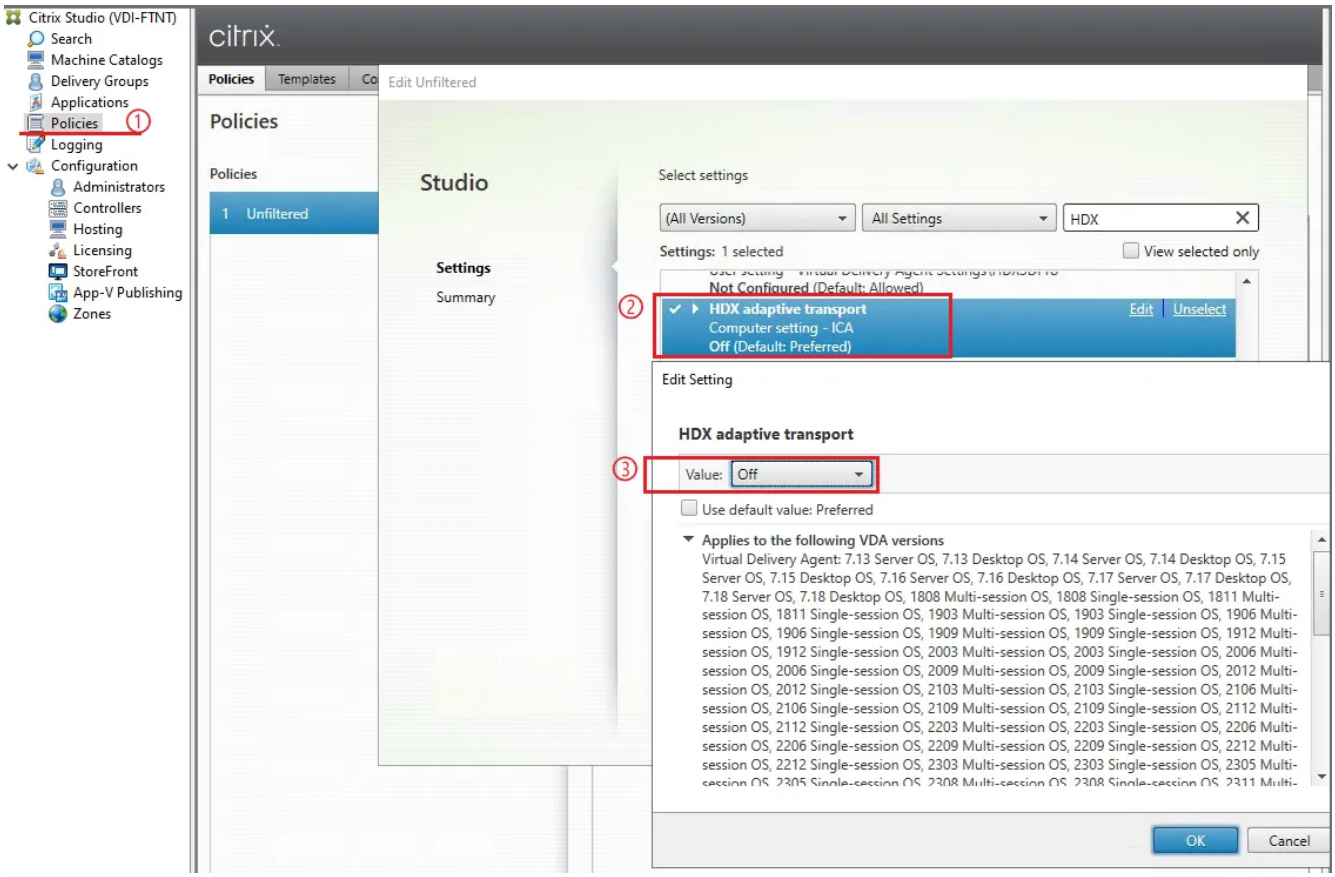


Configuring HDX settings in Citrix VDI

HDX Adaptive Transport is a Citrix feature that dynamically selects the best transport protocol (TCP or UDP) for delivering ICA traffic between the Citrix client and the VDA.

If you deploy FortiADC with a TCP-only virtual server for ICA traffic:

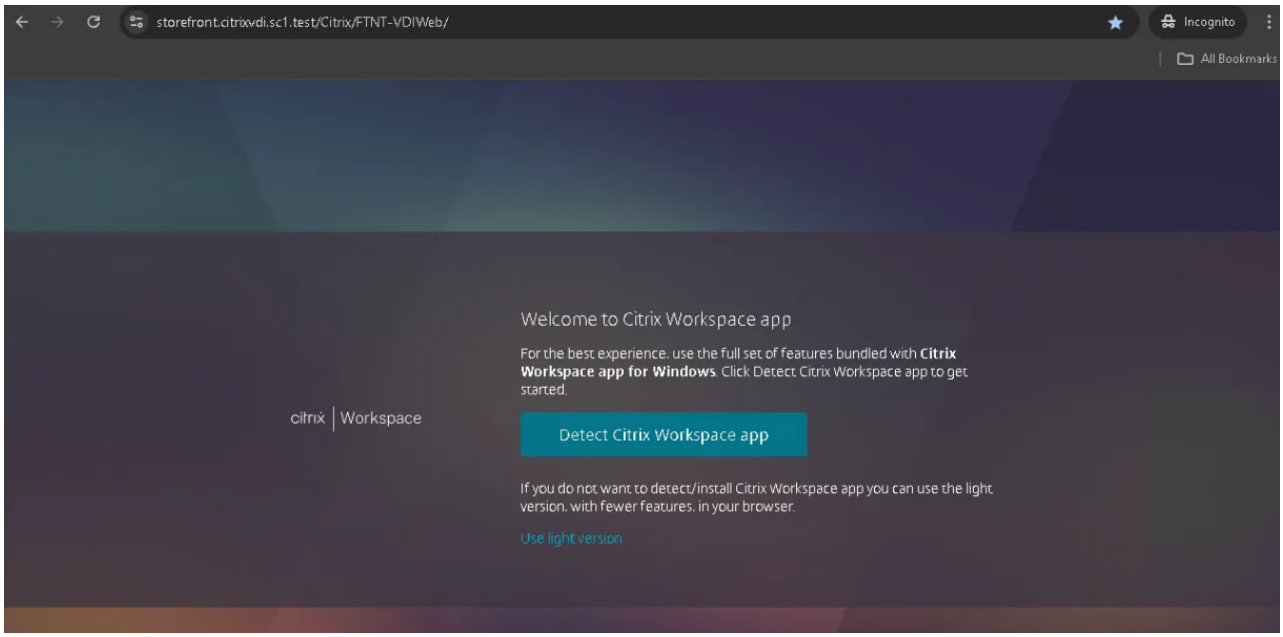
- You must disable Adaptive Transport, or set it to "Off", so ICA stays on TCP.
- Otherwise, the client may try to use UDP, which FortiADC won't handle or route.



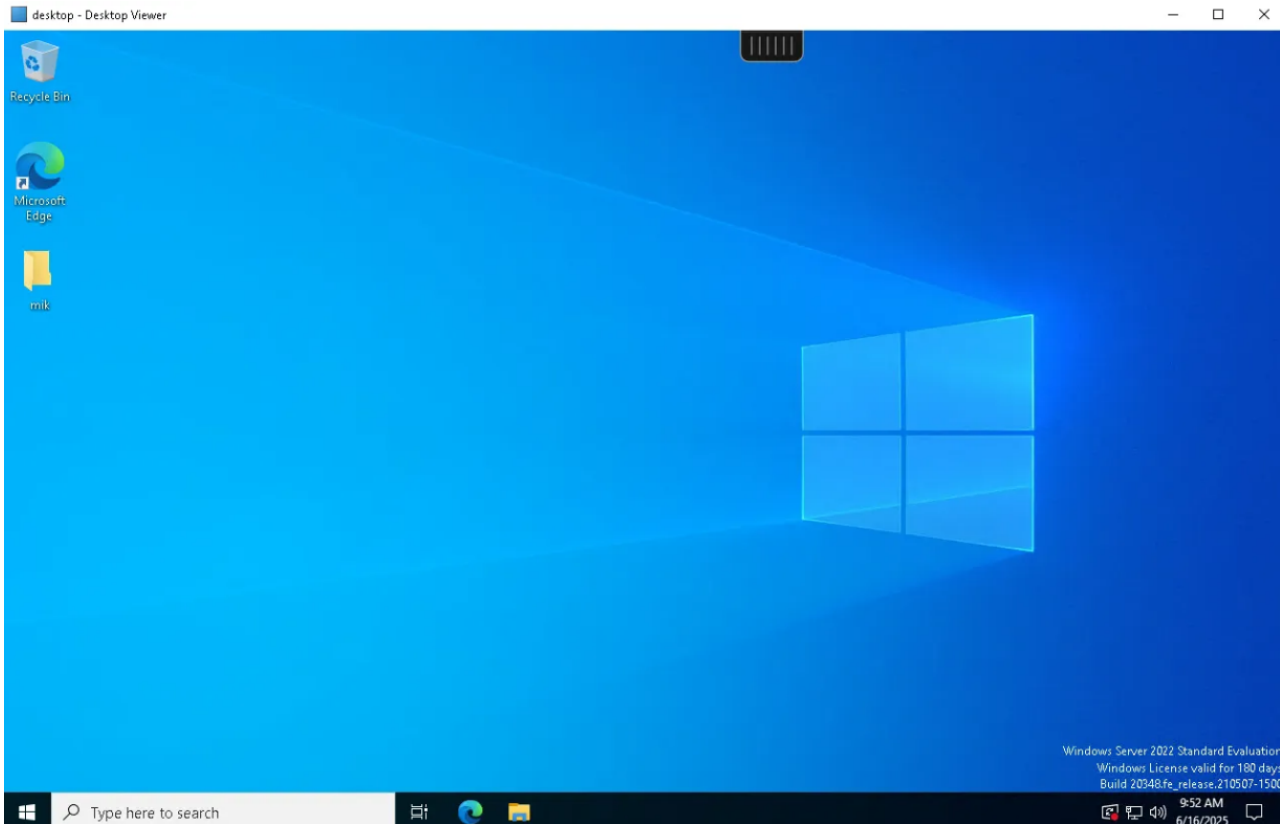
Verifying the TCP connection to the VDAs (via Citrix WorkSpace)

This section verifies that the FortiADC configuration allows Citrix WorkSpace App users to successfully establish ICA/HDX connections to the backend VDA servers over TCP.

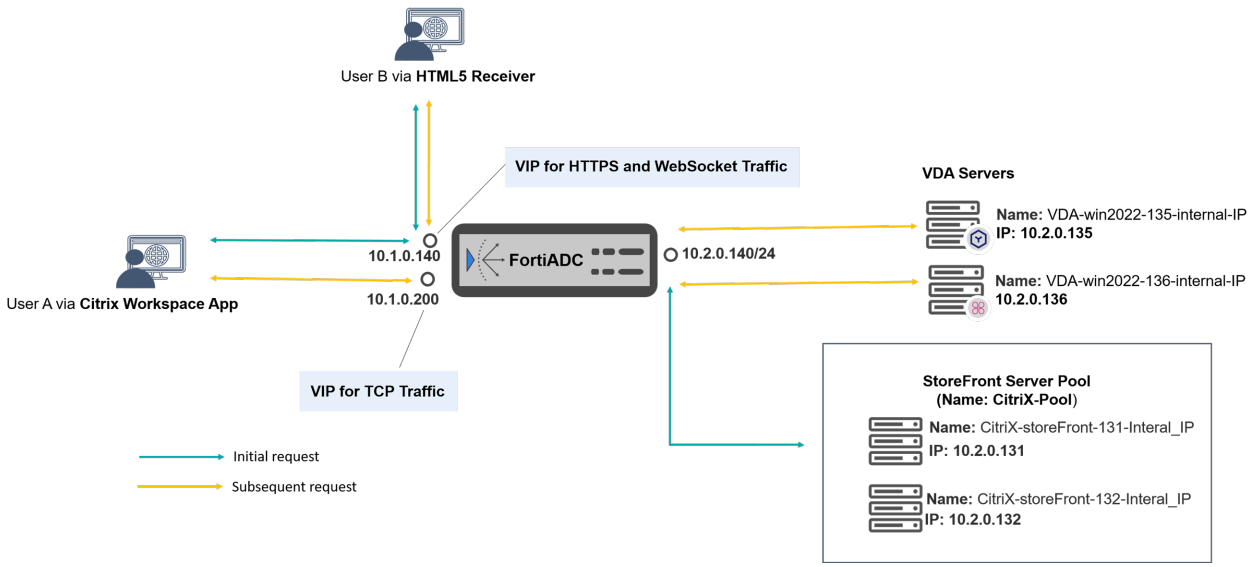
1. Open a web browser and enter the URL for accessing your virtual server and Citrix StoreFront service and click “Detect Citrix Workspace app”



2. From the Citrix StoreFront webpage, login with the User name and Password.
3. Select and open any available application or desktop to Launch the Citrix Workspace application and initiate the ICA connection.
4. FortiADC will be able to use the port number to determine the correct VDA server and forward the connection accordingly.



Reference: Network topology used in the examples of Solution 2



In this guide, **separate IP addresses** are used on FortiADC to handle requests from the **HTML5 Receiver and the WorkSpace App**. This impacts how you configure virtual servers, server pools, and IP mappings within the HTTP and Stream Scripts.

We recommend referring to this topic as you go through the configuration steps—it offers essential context to help you understand the relationships between virtual IPs, ports, and real server IPs, as well as the logic behind the HTTP and Stream scripts. This foundational understanding will enable you to effectively adapt the configuration to your specific network deployment.

DNS record:

- **Domain:** storefront.citrixvdi.sc1.test
- **IP:**10.1.0.140

HTTPS/WebSocket Traffic

HTTPS traffic refers to the **initial client requests to the StoreFront servers**, while WebSocket traffic represents the **subsequent connections to the VDA servers** for users accessing Citrix VDI through the **HTML5 Receiver**.

In the example of this guide, both types of traffic enter FortiADC through the same IP address but use different port numbers. They are handled by the same virtual server on FortiADC.

In this guide, we use the following configurations in:

- Configurations for solution 1: FortiADC processing traffic to StoreFront only on page 18
- Configuring FortiADC to handle the WebSocket traffic from HTML5 Receivers on page 34

Virtual server on FortiADC	• **IP:** 10.1.0.140

	• **Port numbers:**
	  • **443:** for the **HTTPS** connections to **StoreFront server pool**
	  • **12005:** for the **WebSocket** connections to VDA server **VDA-win2022-135-internal-IP**
	  • **12006:** for the **WebSocket** connections to VDA server **VDA-win2022-136-internal-IP**
	• **Interface: Port 2 (IP: 10.1.0.140/24)**
	• **Content routing:**
	  • Requests with hostname "**storefront.citrixvdi.sc1.test**" is routed to **StoreFront Server Pool**.
	  • Requests with hostname "**storefront.citrixvdi.sc1.test:12005**" is routed to VDA: 10.2.0.135.
	  • Requests with hostname "**storefront.citrixvdi.sc1.test:12006**" is routed to VDA: 10.2.0.136.

StoreFront Server pool added on FortiADC	**Name: CitriX-Pool**
	• Server 1
	  • IP: **10.2.0.131**
	  • Listening port: **80**
	• Server 2
	  • IP: **10.2.0.132**
	  • Listening port: **80**

VDA server pools added on FortiADC	• Server Pool **VDA-135** contains one VDA server as below:
	  • Name: **VDA-win2022-135-internal-IP**
	  • IP: **10.2.0.135**
	  • Listening port: **8008**
	• Server Pool **VDA-136** contains one VDA server as below
	  • Name: **VDA-win2022-136-internal-IP**
	  • IP: **10.2.0.136**
	  • Listening port: **8008**

Content Routing Rules	• Requests contains the HTTP Host Header "**storefront.citrixvdi.sc1.test:12005**" are routed to server pool "**VDA-135**". • Requests contains the HTTP Host Header "**storefront.citrixvdi.sc1.test:12006**" are routed to server pool "**VDA-136**". • Requests contains the HTTP Host Header "**storefront.citrixvdi.sc1.test**" are routed to server pool "**CitriX-Pool**".

TCP Traffic

TCP traffic refers to the subsequent connections to the VDA servers for users accessing Citrix VDI through **Citrix Workspace App**.

In the example provided in this guide, **TCP traffic enters FortiADC through a different IP address** than the one used for WebSocket traffic. Both traffic types are routed to the same set of VDA servers.

In this guide, we use the following configurations in Configuring FortiADC to handle ICA TCP traffic from Citrix Workspace App on page 48.

Virtual server on FortiADC	• IP: **10.1.0.200**
	• Interface: **Port 2 (IP: 10.1.0.140/24)**
	• Port numbers:
	  • **12005:** for the connections to VDA server **VDA-win2022-135-internal-IP**
	  • **12006:** for the connections to VDA server **VDA-win2022-136-internal-IP**

VDA server pool added on FortiADC	Name: **VDA-2598**
	• Server 1:
	  • Name: **VDA-win2022-135-internal-IP**
	  • IP: **10.2.0.135**
	  • Listening port: **2598**
	• Server 2:
	  • Name: **VDA-win2022-136-internal-IP**
	  • IP: **10.2.0.136**
	  • Listening port: **2598**
	Traffic distribution to the two servers follows the rules defined in the Stream Script.

HTTP Script

In the HTTP Script used to modify the ICA file, through the following code:

```
when RULE_INIT {
    -- Modify these to fit your setup and environment
    local ica_vs_ip = "10.1.0.200"
    -- You can delete or add more of these

    mapping = {
    -- The FQDN of your StoreFront is "storefront.citrixvdi.sc1.test"
        ["10.106.216.135"] = { vs_ip = ica_vs_ip, proxy_host =
"storefront.citrixvdi.sc1.test:12005" },
        ["10.106.216.136"] = { vs_ip = ica_vs_ip, proxy_host =
"storefront.citrixvdi.sc1.test:12006" }
```

We define:

- **For WebSocket Traffic:**
  - The original destination 10.106.216.135 in user's request should be replaced with storefront.citrixvdi.sc1.test:12005
  - The original destination 10.106.216.136 in user's request should be replaced with storefront.citrixvdi.sc1.test:12006

  The domain storefront.citrixvdi.sc1.test is resolved via DNS to **10.1.0.140**, which is the IP address assigned to FortiADC's HTTPS/WebSocket virtual server. This virtual server is configured to listen on port 12005 and 12006,

enabling it to receive and handle WebSocket traffic directed to:

- storefront.citrixvdi.sc1.test:12005
- storefront.citrixvdi.sc1.test:12006

As a result, user requests using the WebSocket protocol can successfully reach FortiADC.

- **For TCP traffic:**
  - The original destination 10.106.216.135 in user's request should be replaced with 10.1.0.200:12005.
  - The original destination 10.106.216.136 in user's request should be replaced with 10.1.0.200:12006.

  We use 10.1.0.200 as the IP address of the TCP virtual server on FortiADC, and this virtual server is configured to listen on port 12005 and 12006, enabling it to receive and handle TCP traffic directed to:
  - 10.1.0.200:12005
  - 10.1.0.200:12006

This HTTP Script is added through **Server Load Balancing > Scripting > HTTP**, and referenced in the HTTPS/WebSocket virtual server.

For the full script, see: Compiling an HTTP script to modify the ICA file on page 38

If network topology described in this code doesn't fit your specific deployment environment, you will need to customize or write your own script in Lua to suit your topology and traffic routing requirements.

Stream Script

In the HTTP script, we defined redirection rules as follows:

- Traffic originally destined for **10.106.216.135** is rewritten to **10.1.0.200:12005**
- Traffic originally destined for **10.106.216.136** is rewritten to **10.1.0.200:12006**

Here, **10.1.0.200** is the IP address of FortiADC's TCP virtual server. When FortiADC receives packets on ports 12005 or 12006, it needs to determine which backend VDA server to forward them to.

This is where the Stream Script comes in. It inspects the destination port of incoming packets and explicitly maps them to the appropriate VDA server:

```
when STREAM_REQUEST_DATA {
    local port=IP:local_port()
       if (port == 12005) then
        LB:upstream("VDA-win2022-135-internal-IP")
       elseif (port == 12006) then
        LB:upstream("VDA-win2022-136-internal-IP")
    end
}
```

This logic ensures:

- Packets arriving on port **12005** are forwarded to **VDA-win2022-135-internal-IP** (Corresponds to **10.106.216.135** according to the Real Server settings we have configured in TCP Traffic on page 55)
- Packets arriving on port **12006** are forwarded to **VDA-win2022-136-internal-IP** (Corresponds to **10.106.216.136** according to the Real Server settings we have configured in TCP Traffic on page 55)

As a result, FortiADC successfully routes TCP traffic—originally intended for 10.106.216.135 and 10.106.216.136—to the correct VDA servers based on port numbers.

Configurations for security checks

FortiADC provides a comprehensive, multilayered security features that ensure traffic reaching the back-end servers is secure and free from threats. In Citrix VDI environment, ForiADC can perform security inspection for HTTPS and WebSocket traffic.

Configure the following profiles and then reference them in a virtual server:

- Configuring the WAF Profile on page 59
- Geo IP Protection configuration on page 66
- DoS Protection configuration on page 68

Please note that for Citrix VDI environment, FortiADC can't perform security check for ICA TCP traffic.

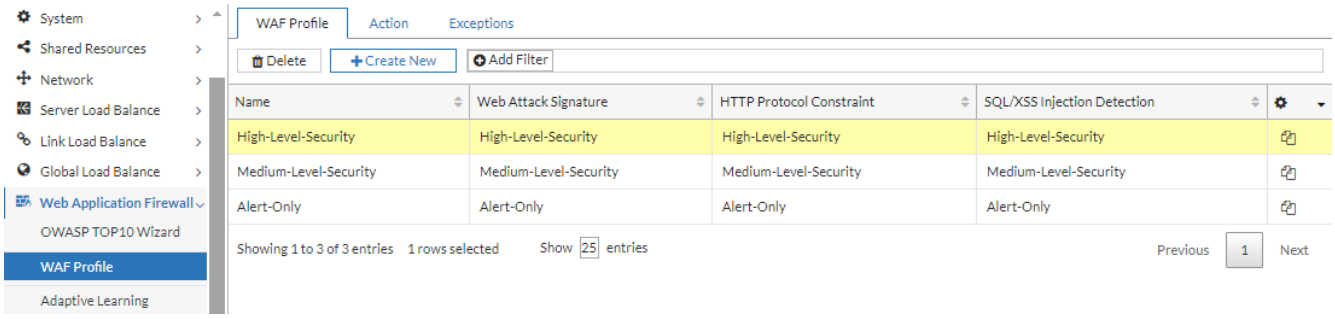
Configuring the WAF Profile

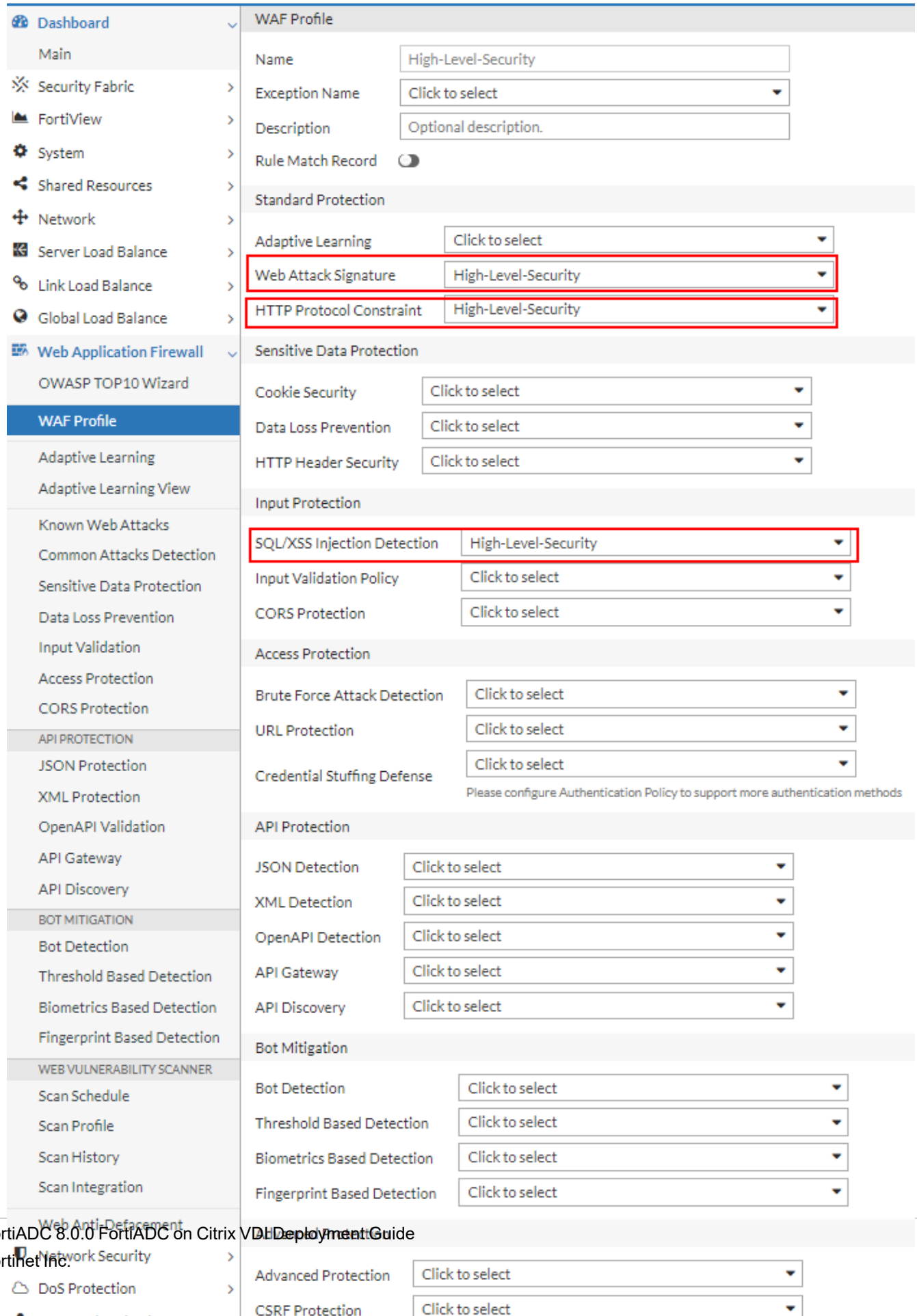
In this section, you can configure a custom WAF profile or choose from the predefined profiles provided by FortiADC. Custom profiles allow for granular control over security policies, while predefined profiles offer standardized protection settings tailored for common use cases. For details, see the FortiADC Administration Guide on Configuring a WAF Profile.

In deploying FortiADC within a Citrix VDI environment, we recommend utilizing the **High-Level-Security** predefined WAF Profile. This profile is engineered to deliver robust, enterprise-grade protection by enabling a comprehensive suite of advanced security features. It includes an extensive library of Web Attack Signatures, stringent HTTP Protocol Constraints, and precise SQL/XSS Injection Detection mechanisms. These collectively fortify the environment against sophisticated and evolving web-based threats, ensuring unparalleled security for your virtual desktop infrastructure.

WAF Profile: High-Level-Security

The following describes in detail each of the WAF policy settings applied in the predefined WAF Profile named "High-Level-Security": Web Attack Signature, HTTP Protocol Constraint, and SQL/XSS Injection Detection.




Dashboard
Main
Security Fabric
FortiView
System
Shared Resources
Network
Server Load Balance
Link Load Balance
Global Load Balance
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
WEB VULNERABILITY SCANNER
Scan Schedule
Scan Profile
Scan History
Scan Integration
Web Anti-Defacement
Network Security
DoS Protection
WAF Profile
Name
High-Level-Security
Exception Name
Click to select
Description
Optional description.
Rule Match Record
Standard Protection
Adaptive Learning
Click to select
Web Attack Signature
High-Level-Security
HTTP Protocol Constraint
High-Level-Security
Sensitive Data Protection
Cookie Security
Click to select
Data Loss Prevention
Click to select
HTTP Header Security
Click to select
Input Protection
SQL/XSS Injection Detection
High-Level-Security
Input Validation Policy
Click to select
CORS Protection
Click to select
Access Protection
Brute Force Attack Detection
Click to select
URL Protection
Click to select
Credential Stuffing Defense
Click to select
Please configure Authentication Policy to support more authentication methods
API Protection
JSON Detection
Click to select
XML Detection
Click to select
OpenAPI Detection
Click to select
API Gateway
Click to select
API Discovery
Click to select
Bot Mitigation
Bot Detection
Click to select
Threshold Based Detection
Click to select
Biometrics Based Detection
Click to select
Fingerprint Based Detection
Click to select
Advanced Protection
Click to select
CSRF Protection
Click to select

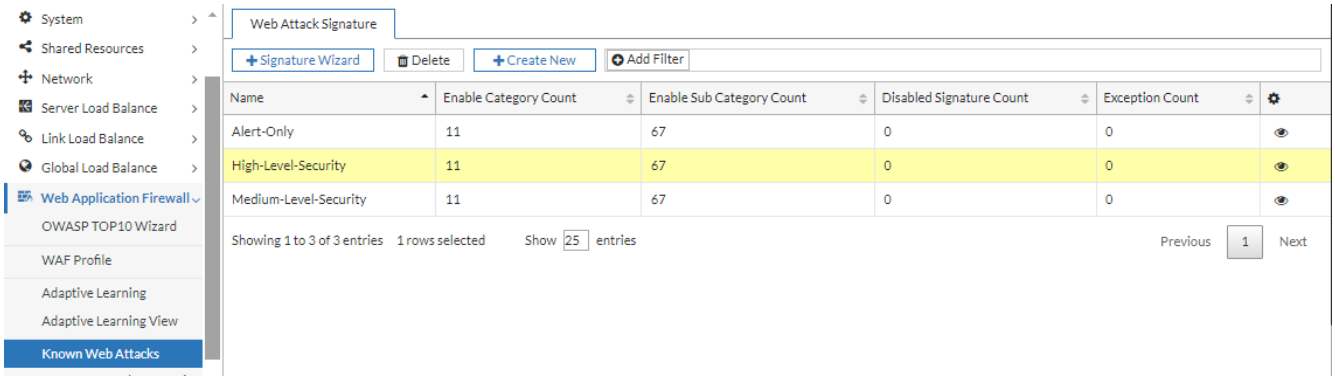
Web Attack Signature policy: High-Level-Security

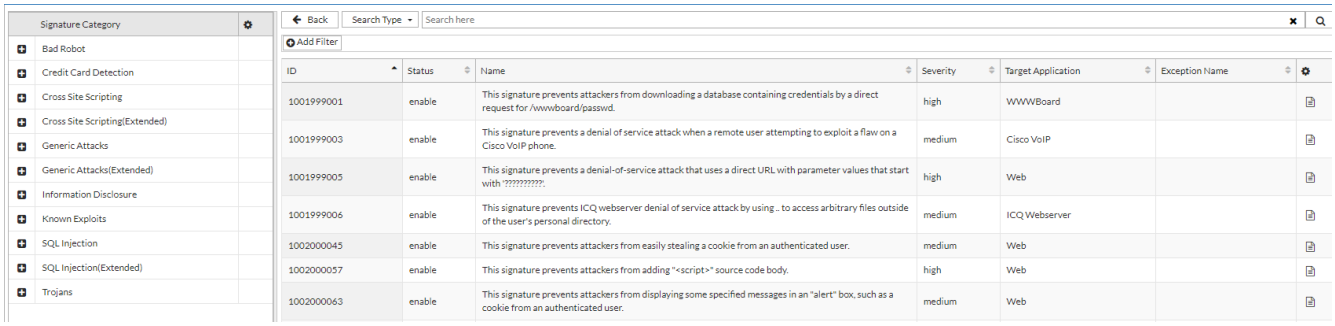
The FortiGuard Web Attack Signature service provides a database of attack signatures that is updated periodically to protect against new kinds of attacks.

The predefined Web Attack Signature policy "High-Level-Security" is applied in the predefined WAF Profile also named "High-Level-Security". This predefined Web Attack Signature policy contains the following settings for the class of scanpoints, the action when traffic matches the signatures, and the Web Attack Signature categories and subcategories of threats that are detected by the signatures.

Scanpoint	Description	Status
HTTP header	Scans traffic against HTTP header signatures.	Enabled
HTTP Request Body	Scans traffic against HTTP request body signatures.	Enabled
HTTP Response Body	Scans traffic against HTTP response body signatures.	Disabled

Event Severity	Action
High	Deny
Medium	Deny
Low	Alert





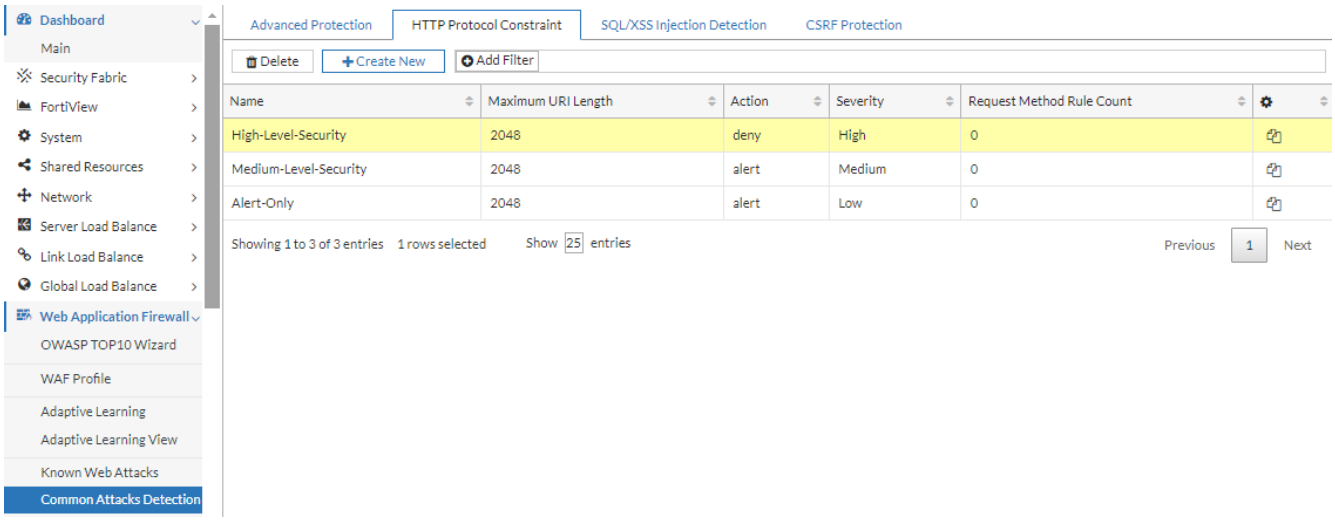
For more information, see the FortiADC Administration Guide on Configuring a Web Attack Signature policy.

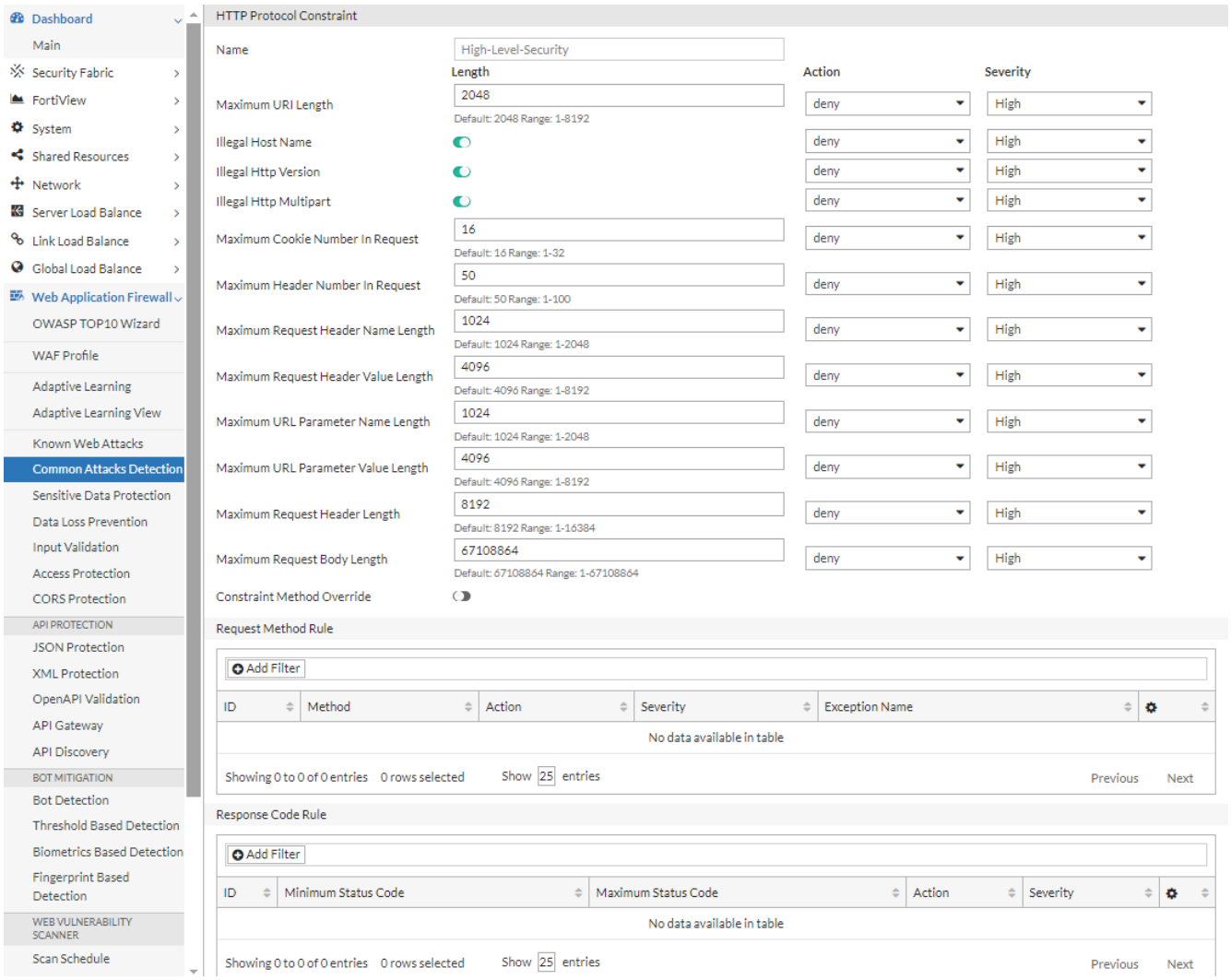
HTTP Protocol Constraint policy: High-Level-Security

As part of Common Attacks Detection, the HTTP Protocol Constraint policy includes the following rules:

- HTTP request parameters—Limit the length of URIs, headers, and body to prevent several types of attacks, such as buffer overflow and denial of service.
- HTTP request methods—Restrict HTTP methods allowed in HTTP requests. For example, do not allow the PUT method in HTTP requests to prevent attackers from uploading malicious files.
- HTTP response codes—Drop response traffic containing HTTP response codes that might contain information attackers can use to craft attacks. For example, some HTTP response codes include fingerprint data like web server version, database version, OS, and so on.

The predefined HTTP Protocol Constraint policy "High-Level-Security" is applied in the predefined WAF Profile also named "High-Level-Security", with the following policy settings:





For more information, see the FortiADC Administration Guide on Configuring an HTTP Protocol Constraint policy.

SQL/XSS Injection Detection policy: High-Level-Security

SQL/XSS Injection Detection policies are designed to identify and prevent SQL injection and cross-site scripting (XSS) attacks. These types of attacks occur when user-supplied data is improperly processed by an interpreter. In an SQL injection attack, malicious HTTP requests are crafted to manipulate SQL queries, allowing unauthorized access to a web application's database. XSS attacks, on the other hand, exploit vulnerabilities that enable the execution of malicious scripts within a user's web browser.

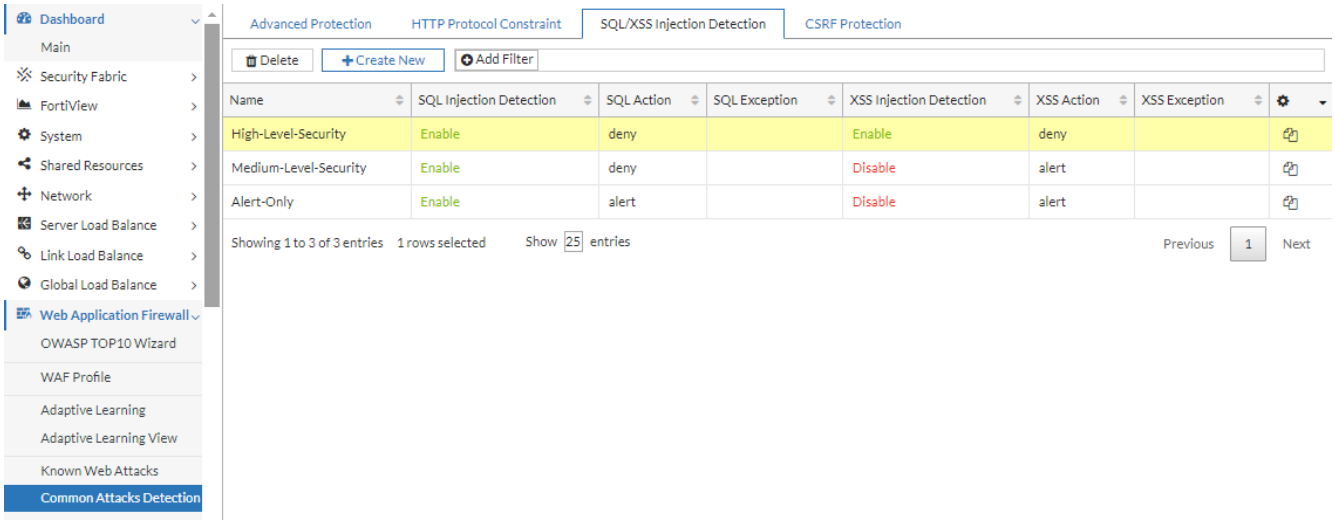
Unlike traditional signature-based detection methods, the Web Application Firewall (WAF) SQL and XSS injection detector module employs lexical analysis to identify these threats. This approach is both complementary and faster, offering enhanced detection capabilities.

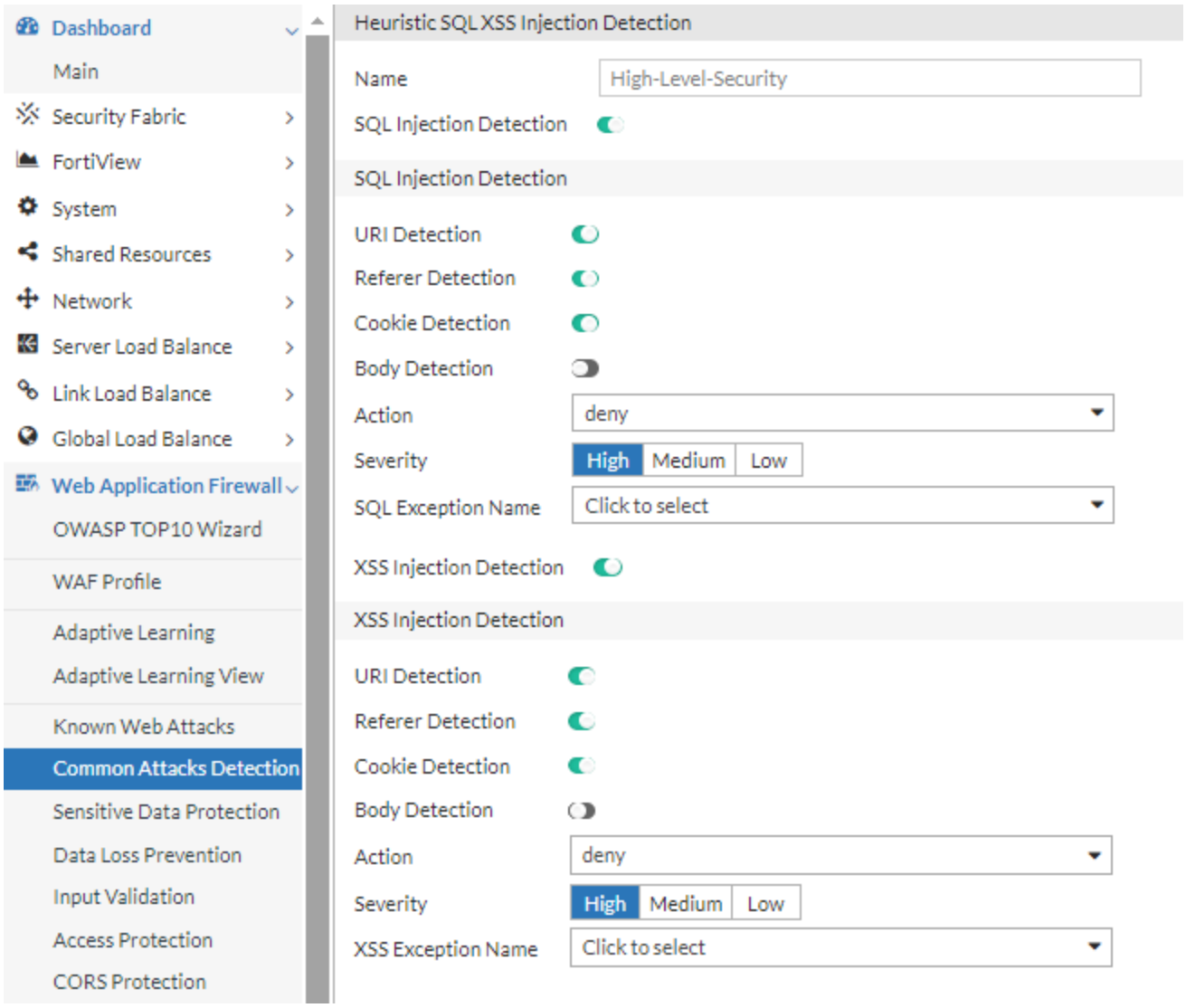
The policy configuration allows for enabling or disabling scanpoints, defining actions when traffic matches specific signatures, and setting the severity of events.

You can enable detection in the following scanpoints:

- SQL Injection: URI—Analyzes content in the URI.
- SQL Injection: Referer—Analyzes content in the HTTP Referer header.
- SQL Injection: Cookie—Analyzes content in the HTTP Cookie header.
- SQL Injection: Body—Analyzes content in the HTTP request body.
- XSS Injection: URI—Analyzes content in the URI.
- XSS Detection: Referer—Analyzes content in the HTTP Referer header.
- XSS Detection: Cookie—Analyzes content in the HTTP Cookie header.
- XSS Detection: Body—Analyzes content in the HTTP request body.

The predefined SQL/XSS Injection Detection policy "High-Level-Security" is applied in the predefined WAF Profile also named "High-Level-Security", with the following policy settings:



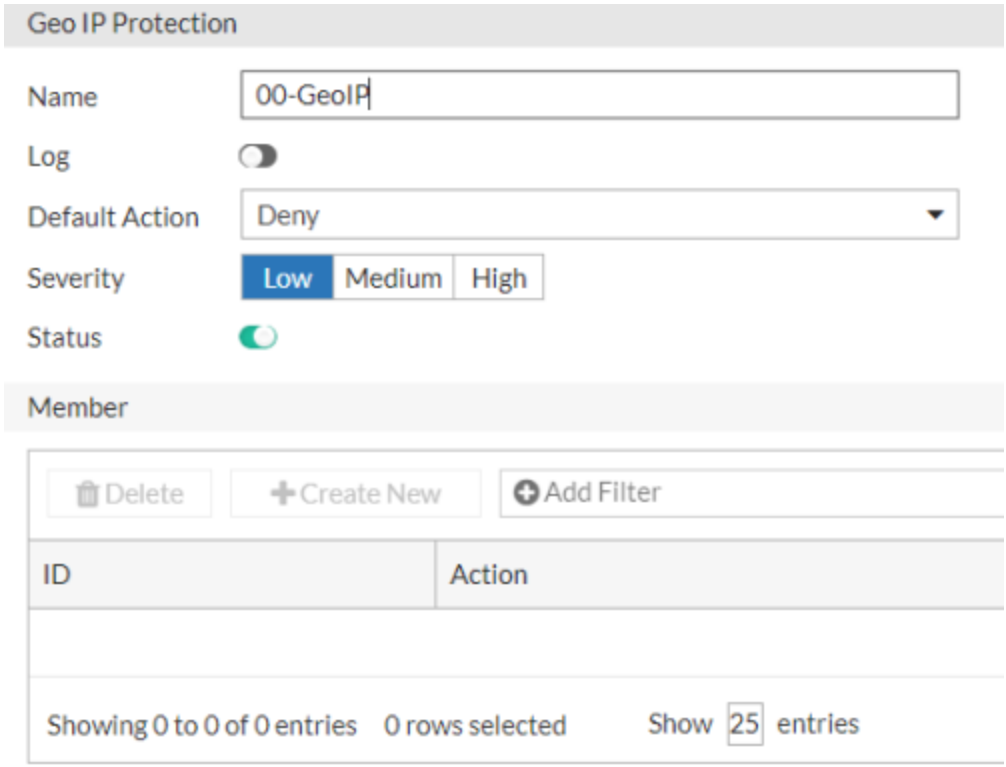

Dashboard
Main
Security Fabric
FortiView
System
Shared Resources
Network
Server Load Balance
Link Load Balance
Global Load Balance
Web Application Firewall
OWASP TOP10 Wizard
WAF Profile
Adaptive Learning
Adaptive Learning View
Known Web Attacks
Common Attacks Detection
Sensitive Data Protection
Data Loss Prevention
Input Validation
Access Protection
CORS Protection
Heuristic SQL XSS Injection Detection
Name
High-Level-Security
SQL Injection Detection
SQL Injection Detection
URI Detection
Referer Detection
Cookie Detection
Body Detection
Action
deny
Severity
High
Medium
Low
SQL Exception Name
Click to select
XSS Injection Detection
XSS Injection Detection
URI Detection
Referer Detection
Cookie Detection
Body Detection
Action
deny
Severity
High
Medium
Low
XSS Exception Name
Click to select

Geo IP Protection configuration

Optionally, you can configure a Geo IP block or allow policy to take action based on the geographic location of incoming requests.

Geo IP protection enhances security by controlling access based on the geographic location of incoming requests. It allows for blocking or allowing traffic from specific regions, which mitigates threats by filtering out potentially malicious traffic, ensures compliance with regional data protection regulations, and optimizes performance by managing legitimate access. By implementing Geo IP policies, FortiADC can enforce geographic access rules, thereby strengthening the security and efficiency of the Citrix infrastructure.

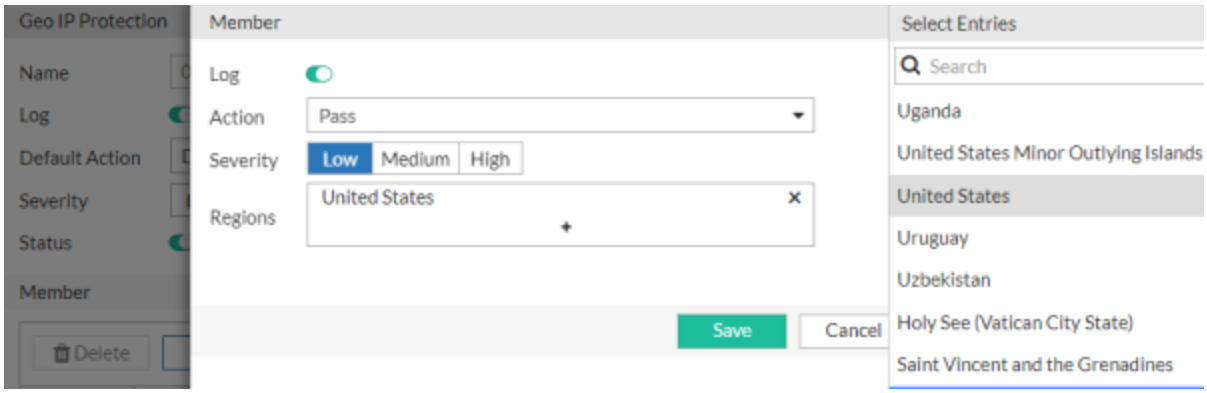
1. Navigate to **Network Security > Geo IP Protection**.
   The configuration page displays the **Geo IP Protection** tab.
2. Click **Create New** to display the configuration editor.
3. In the **Name** field, enter a unique name for the Geo IP profile. Save the configuration.



   After the Geo IP Protection configuration is saved initially, the Member section becomes available to configure.
   Note: For details of each configuration parameter, see the FortiADC Administration Guide for Geo IP Protection.
4. Under the **Member** section, click **Create New** to display the configuration editor.
5. Configure the following key settings:

Setting	Guideline
Log	Enable Log.

Setting	Guideline
Action	Select **Pass** from the drop-down menu to allow only the regions of the country to access the VS.
Regions	Click **+** and select the Country from the right panel.



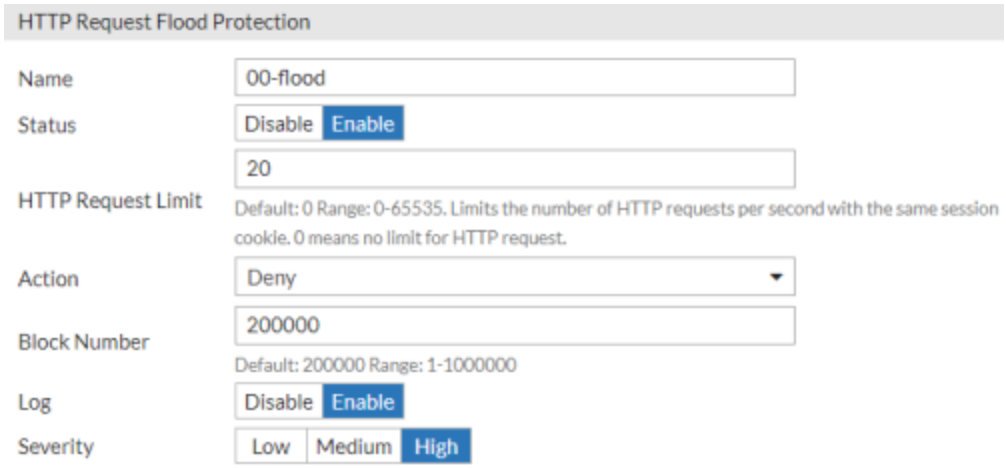
6. Save the Member configuration and then save the Geo IP Protection configuration to submit the Member changes.
7. Go to the **Application Profile** tab.
8. Locate the Application Profile previously created (and currently attached to the virtual server configuration) and click the (edit icon) to display the configuration editor.
9. In the Geo IP Blocklist field, select the newly created Geo IP Protection configuration to bind it to the Application Profile.
10. Save the configuration.

DoS Protection configuration

Optionally, you can configure a DDoS Protection to detect and mitigate against Denial of Service (DoS) attacks by analyzing incoming network traffic for anomalous patterns that typically indicate an attack. For Citrix StoreFront servers, this translates to robust protection against DoS attacks, ensuring that legitimate users can access services without interruption. This proactive defense mechanism maintains the performance and security of the server, safeguarding it from potential disruptions and ensuring continuous service availability.

1. Navigate to **DoS Protection > Application**. Click the **HTTP Request Flood** tab.
2. Click **Create New** to display the configuration editor.
3. Configure the following key settings:

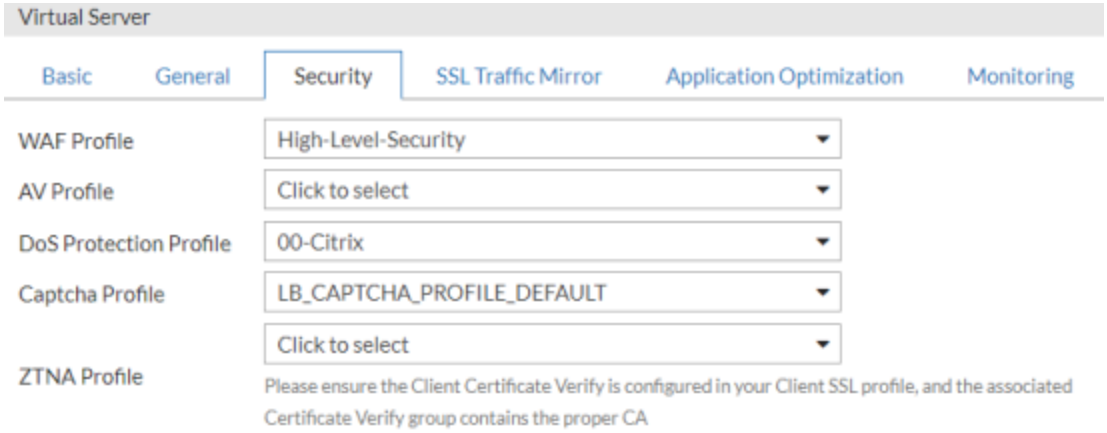
Setting	Guideline
Name	Enter a unique name for the HTTP Request Flood profile.
HTTP Request Limit	Enter a value to limit the number allowable HTTP requests per second with the same session cookie.
Log	Enable Log.



Note: For details of each configuration parameter, see the FortiADC Administration Guide for Configuring an HTTP request flood policy.

4. Save the configuration.
5. Go to **Server Load Balance > Virtual Server**.
   The configuration page displays the **Virtual Server** tab.
6. Locate the Virtual Server configuration previously created (which is currently used by the Citrix service) and click the (edit icon) to display the configuration editor.

7. Click the **Security** tab. In the **DoS Protection Profile** field, select the newly created HTTP Request Flood Protection profile.



8. Save the configuration.

Troubleshooting

To troubleshoot issues with Citrix connections, you can use the following tools:

- Use the FortiADC console to print diagnostic messages when traffic is incoming. For details, see Printing diagnostic messages on page 70.
- Enable the **Traffic Log** feature in the FortiADC GUI to observe each connection's URL and response results. For details, see Enabling FortiADC Traffic Log on page 70.
- Utilize Citrix Windows Event Logging to review Citrix service logs. For details, see Using Citrix Windows Event Logging on page 70.

Printing diagnostic messages

Set up the diagnose debug print out level in the FortiADC console.

1. Connect your management computer to the FortiADC.
2. Enable the diagnose debug output for httproxy and ssl-of-httproxy.

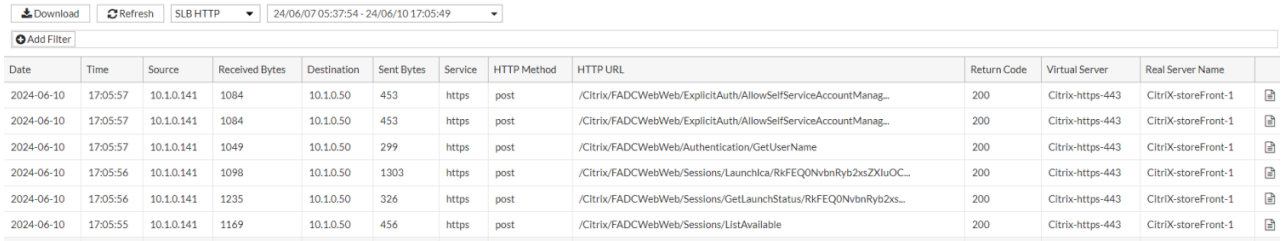
```
FortiADC-VM #  diagnose debug module httproxy all
FortiADC-VM #  diagnose debug module ssl-of-httproxy all
FortiADC-VM #  diagnose debug enable
```

You should see the related Citrix HTTP traffic information printed.

Enabling FortiADC Traffic Log

Enable the Traffic Log in the Virtual Server configuration to view the VS traffic activity from FortiADC Log & Report.

1. Navigate to **Server Load Balance > Virtual Server**.
   The configuration page displays the **Virtual Server** tab.
2. Locate the Virtual Server configuration that is currently used by the Citrix service and click the (edit icon) to display the configuration editor.
3. Click the **Monitoring** tab and enable **Traffic Log**. Save the configuration.
4. Go to **Log & Report > Traffic Log**.
5. From the top navigation, select **SLB HTTP** from the drop-down menu to view the traffic log.

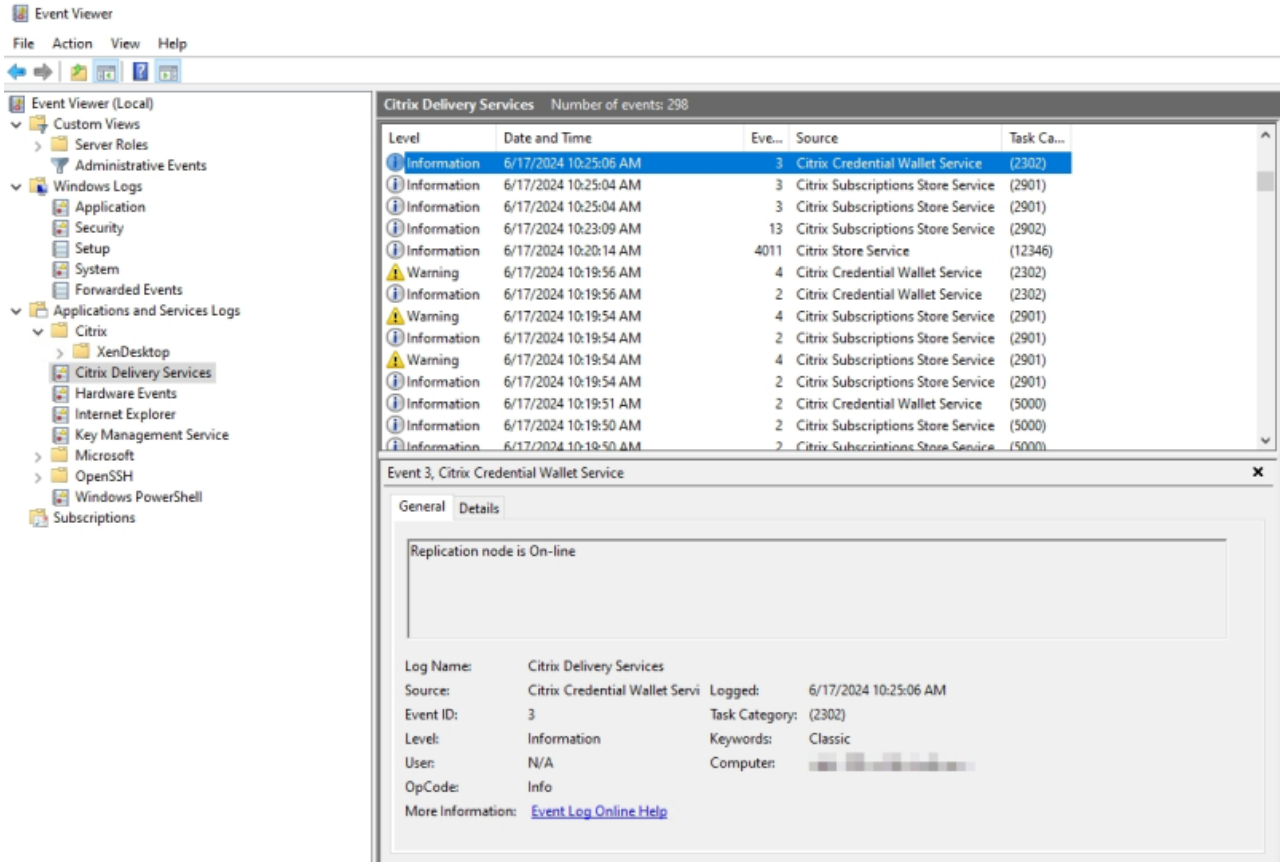

Download | Refresh | SLB HTTP | 24/06/07 05:37:54 - 24/06/10 17:05:49
Add Filter

Date	Time	Source	Received Bytes	Destination	Sent Bytes	Service	HTTP Method	HTTP URL	Return Code	Virtual Server	Real Server Name
2024-06-10	17:05:57	10.1.0.141	1084	10.1.0.50	453	https	post	/Citrix/FADCWebWeb/ExplicitAuth/AllowSelfServiceAccountManag...	200	Citrix-https-443	CitriX-storeFront-1
2024-06-10	17:05:57	10.1.0.141	1084	10.1.0.50	453	https	post	/Citrix/FADCWebWeb/ExplicitAuth/AllowSelfServiceAccountManag...	200	Citrix-https-443	CitriX-storeFront-1
2024-06-10	17:05:57	10.1.0.141	1049	10.1.0.50	299	https	post	/Citrix/FADCWebWeb/Authentication/GetUserName	200	Citrix-https-443	CitriX-storeFront-1
2024-06-10	17:05:56	10.1.0.141	1098	10.1.0.50	1303	https	post	/Citrix/FADCWebWeb/Sessions/Launchlca/RkFEQ0NvbnRyb2xsZXIuOC...	200	Citrix-https-443	CitriX-storeFront-1
2024-06-10	17:05:56	10.1.0.141	1235	10.1.0.50	326	https	post	/Citrix/FADCWebWeb/Sessions/GetLaunchStatus/RkFEQ0NvbnRyb2xs...	200	Citrix-https-443	CitriX-storeFront-1
2024-06-10	17:05:55	10.1.0.141	1169	10.1.0.50	456	https	post	/Citrix/FADCWebWeb/Sessions/ListAvailable	200	Citrix-https-443	CitriX-storeFront-1

Using Citrix Windows Event Logging

Review Citrix service logs through Citrix Windows Event Logging.

1. Access the **Citrix Studio** server (Windows server) and launch **Event Viewer**.
2. From the left panel menu, go to **Applications and Services Logs** then select **Citrix**.



www.fortinet.com