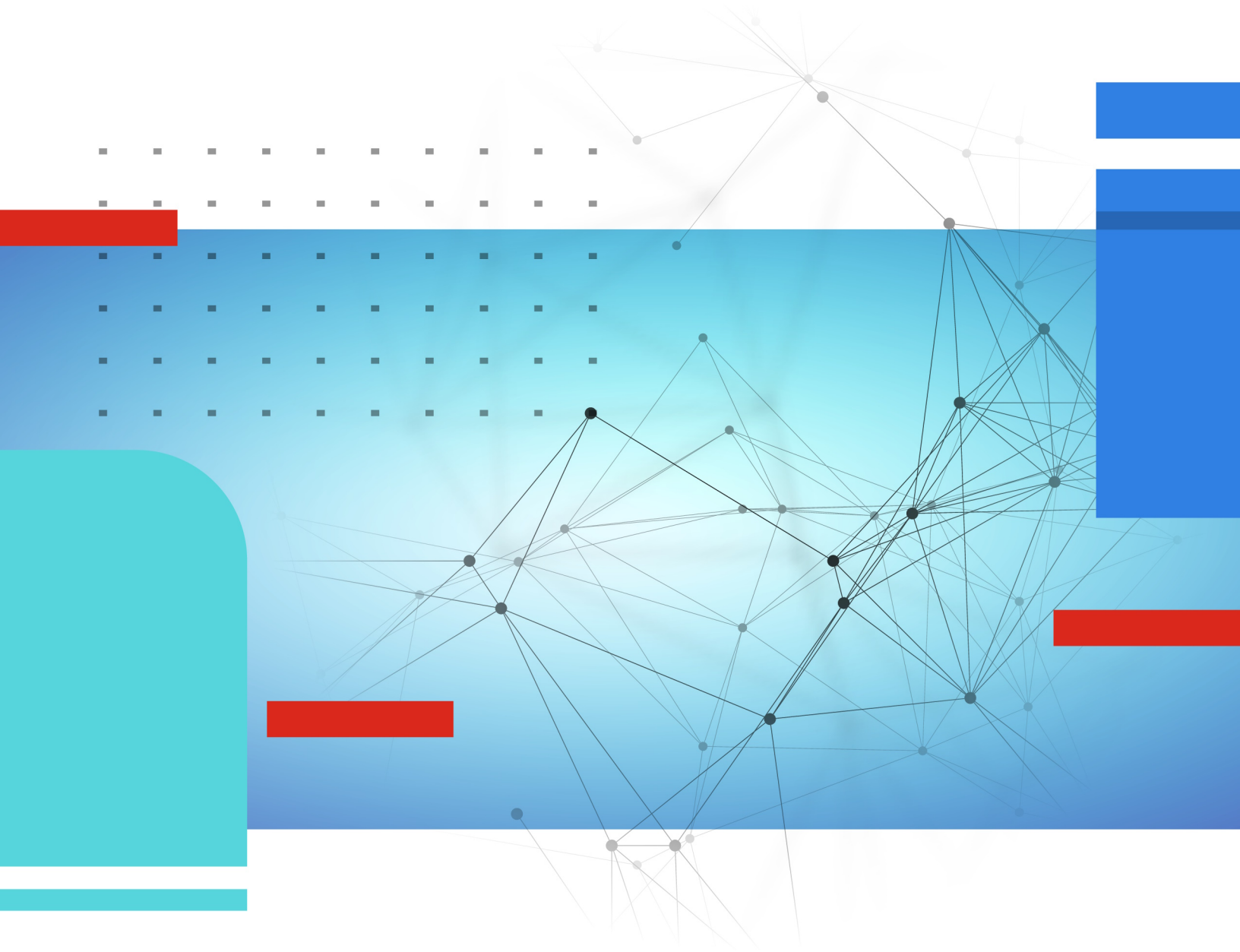
Network Architecture Guide

FortiADC 8.0.0



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

Oct 8, 2025
FortiADC 8.0.0 Network Architecture Guide

TABLE OF CONTENTS

What is FortiADC architecture?

FortiADC is Fortinet's ADC (Application Delivery Controller) solution that provides:

- Application Load Balancing (L2/L4/L7)
- SSL offloading
- Application acceleration
- Web application security (WAF)
- Global Server Load Balancing (GSLB)
- Traffic scripting using HTTP and Stream scripts
- Application authentication

It is designed to ensure that application traffic is fast, secure, and reliably delivered across data centers, cloud, and hybrid environments.

This guide defines how and where you can deploy FortiADC to enhance an organization's application delivery while ensuring application security and performance.

Intended Audience

This guide is primarily for technical professionals, including system architects, security engineers, and DevOps teams who need to understand FortiADC's architecture and how it can be integrated into their organization's network framework. It is particularly beneficial for those in the assessment and planning phase of deploying an application delivery controller.

About This Guide

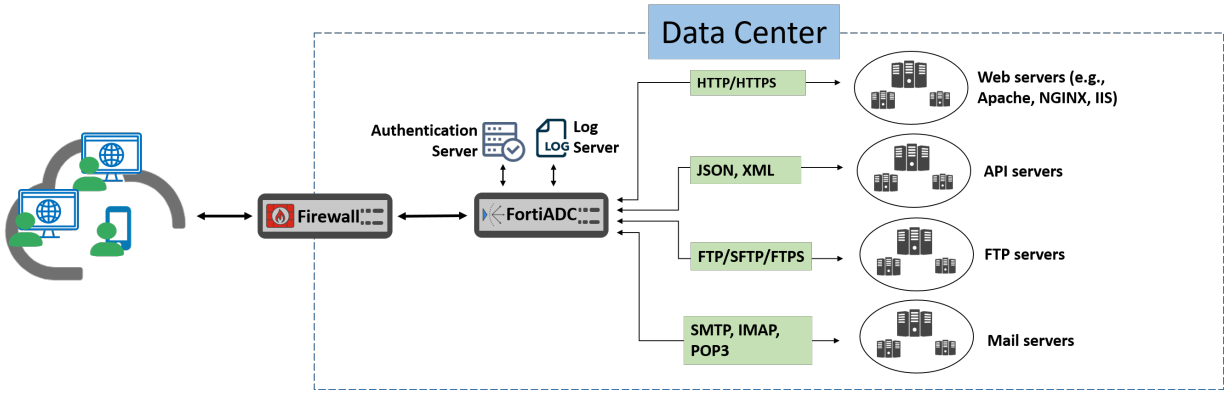
This guide provides a high-level overview of FortiADC's role in different deployment modes and HA modes. It is intended to be used alongside other technical documentation for FortiADC. Where applicable, references to administration guides and technical resources are included to provide deeper insights into each topic.

FortiADC's role and placement in network topology

FortiADC is a specialized network appliance or software that:

- Distributes traffic across multiple back-end application servers (load balancing)
- Accelerates application performance through caching, compression, SSL offloading, and connection multiplexing
- Secures applications with features like Web Application Firewall (WAF), DDoS protection, and IP reputation filtering.
- Monitors health of servers and services to ensure high availability and failover
- Integrates with external authentication servers and IdPs to authenticate users before granting access to protected web applications.

Typical FortiADC Placement in a Network Topology



FortiADC acts as an Application Delivery Controller, and its core role is to efficiently distribute, secure, and accelerate traffic directed to your back-end application servers.

To allow FortiADC to distribute and secure the traffic intended for back-end servers, you usually must install the FortiADC appliance between the back-end servers and all clients that access them, alongside other critical network components such as:

- Routers or switches (to form the network fabric)
- General-purpose firewalls (e.g., FortiGate)
- Authentication servers (e.g., LDAP, RADIUS, SAML, FortiAuth, Azure EntraID)
- Log servers (e.g., FortiAnalyzer, Syslog, SIEM)

This topic includes the following sections:

- Typical FortiADC placement on page 7
  Best practices for positioning FortiADC within your network to ensure optimal protection and performance.
- Integration with other network components on page 8
  Introduction on FortiADC's integration with authentication servers and log servers.

Typical FortiADC placement

- **Behind a Firewall (e.g., FortiGate):**
  FortiADC is typically deployed behind a general-purpose perimeter firewall, such as FortiGate, which is responsible for enforcing comprehensive network-level security—including access control, threat detection, NAT, and traffic inspection between internal networks and the external world.

  However, if deploying a FortiGate or similar firewall is not feasible due to network design constraints, FortiADC can compensate for this gap using its Layer 2 and Layer 4 security capabilities.

  While FortiADC's core strength lies in Layer 7 load balancing and application-aware traffic processing, **it also offers robust Layer 2/4 security features that can enforce essential protections including IP reputation filtering, Geo IP access control, and DDoS mitigation.**

  These features enable FortiADC to act as a lightweight security gateway, especially when deployed in Layer 2 or Layer 4 mode. In such cases, it provides basic perimeter security while still delivering its full suite of traffic distribution and application acceleration functions.
- **In front of devices that enforce SNAT**

  Many of FortiADC's security features rely on knowing the real client IP, such as:
  - Geo IP blocking
  - Rate limiting per client
  - Anomaly detection
  - Period block
  - Session-based behavioral analysis

  When SNAT is used, multiple clients may appear as a single source IP to FortiADC, defeating these protections.

  Therefore, if your network includes any device that enforces SNAT, we strongly recommend deploying FortiADC in front of it.

  However, if this is not feasible within your current network design, you can apply the following workaround: Deploying FortiADC behind a SNAT device on page 8.
- **Throughput considerations**

  The throughput of your FortiADC device should be taken into consideration when you decide how many back-end servers are deployed behind each FortiADC. Selecting a FortiADC model with adequate capacity is crucial to ensure that security inspections do not introduce bottlenecks or latency issues, allowing optimal performance while maintaining strong protection.

  The FortiADC datasheet provides guidance on selecting the appropriate FortiADC model based on the expected total traffic volume forwarded to the back-end servers. See the FortiADC DataSheet for details.
- **Access to the Internet**

  FortiADC relies on FortiGuard Security Services for real-time updates, ensuring that its signature database and threat intelligence feeds stay up to date. These updates enhance protection against evolving web-based threats.

Considerations for closed network environments

In environments with no direct Internet access, such as air-gapped networks, an alternative update mechanism is required:

- **FortiManager:** Acts as a proxy for FortiGuard updates, allowing FortiADC to receive updated security signatures without direct Internet connectivity.
- **Manual Updates:** Security definitions can be manually downloaded from Fortinet's support portal and applied to FortiADC as needed.

Ensuring FortiADC has access to timely updates is crucial for maintaining robust protection against zero-day threats and emerging attack vectors.

Integration with other network components

- **Authentication Server**

  FortiADC supports user authentication through various methods:

  - Remote Authentication Servers: LDAP, RADIUS, NTLM
  - SAML-based Identity Providers (IdPs): Okta, Azure AD, etc.
  - OAuth-based IdPs: Ping Identity, Google, Facebook, and others
  - FortiAuth and FortiToken

  This allows FortiADC to enforce access control policies before granting users access to protected applications.
- **Log Server**

  FortiADC can forward logs to multiple platforms for centralized monitoring and analysis, including:

  - Syslog servers
  - FortiSIEM
  - Splunk
  - Kafka/Elastic
  - Telemetry streaming
  - FortiAnalyzer (for advanced log analytics and reporting)

FortiADC's placement in the network topology may vary depending on the selected deployment mode. Before finalizing the deployment, you should first determine which operation mode best suits your environment. For details, see Choosing the right deployment: Layer 2, Layer 4, and Layer 7 traffic processing on page 12.

Not recommended, but if you must: Handling FortiADC in suboptimal topologies

- ## Deploying FortiADC behind a SNAT device

FortiADC relies on the original client IP address for many of its security functions. Deploying it behind a device that applies Source NAT (SNAT)—such as certain load balancers or firewalls—can obscure the true client IP, reducing the effectiveness of features like rate limiting, geolocation, or IP-based period block.

However, if SNAT is unavoidable, you must configure FortiADC to extract the client's original IP from HTTP headers inserted by the SNAT device in front of it:

1. On FortiADC, go to **Web Application Firewall > WAF Profile**.
2. Enable **Use Original IP**.

FortiADC then inspects the **X-Forwarded-For** header in the incoming request. **If multiple X-Forwarded-For headers are present**—indicating that the request has passed through multiple SNAT devices—**FortiADC extracts the last value**, which corresponds to the rightmost entry in the header sequence. This is assumed to represent the most recent upstream client IP address before reaching FortiADC. Please note that this workaround only applies to L7 deployment for HTTP/HTTPS.

For details, refer to Configuring a WAF Profile.

Traffic types supported by FortiADC

Here's a breakdown of the common types of traffic FortiADC would manage and optimize when handling client requests to back-end application servers. For the full list of protocols supported by FortiADC, see Configuring Application profiles.

FortiADC also supports other TCP and UDP protocols not listed here. However, for those protocols, FortiADC operates at Layer 4 only and cannot perform application-layer parsing or inspection. Application-aware features such as content rewriting, header modification, or protocol-specific security enforcement are not available unless explicitly included in FortiADC's layer 7 application profiles.

Traffic Type	Common Use Cases	FortiADC Functions
Web Traffic (HTTP/HTTPS)	Website and applications	• SSL/TLS offloading • Full application-layer inspection (WAF, IP reputation, AV, DDoS, Bot Mitigation) • HTTP2/HTTP3 support • Content rewriting and URL translation • HTTP compression, caching, and connection reuse • X-Forwarded-For and header manipulation
API Traffic (JSON, XML)	Back-end API calls from SPAs, mobile apps, microservices, or third-party integrations	• Advanced API protection (schema validation, rate limiting, method/path checks) • Layer 7 inspection • Bot/API abuse prevention • Authentication and token handling
DNS Traffic (DNS)	Recursive or authoritative DNS resolution, global DNS load balancing	• Source IP persistence • DNS caching

Traffic Type	Common Use Cases	FortiADC Functions
		• DDoS protection (e.g., query flood)
Email Server Traffic (SMTP)	Sending and receiving emails, load balancing for mail servers	• Protocol-aware load balancing • STARTTLS handling (explicit, implicit) • Command control (EXPN, TURN, VRFY restrictions) • Partial security inspection (IP reputation, AV, DDoS)
File Transfer Traffic (FTP)	Secure file upload/download to app or storage servers	• Protocol inspection with connection timeout handling • SSL offload for FTPS • IP reputation and DDoS defense • Optional client IP preservation
VoIP (SIP)	IP-based voice/video communications (e.g., IP telephony, conferencing, PBXs)	• SIP-aware load balancing • Port persistence & media/control pinning • Optional header manipulation • Partial security inspection (DDoS, IP reputation)
Streaming services (RTSP/RTMP)	Video streaming (e.g., IP cams, media servers) or live video streaming (e.g., social media, video CDN)	• Application-layer session persistence, connection buffering, and load balancing • Source IP preservation and DDoS mitigation support
Database Traffic (MySQL/MSSQL)	Read-write database traffic splitting (e.g., web app backends)	• Read/write-aware load balancing (e.g., SELECT to replicas) • MSSQL login passthrough • Custom profiles per DB account • Automatic query routing
User Authentication Servers (RADIUS/Diameter)	Authentication, authorization, and accounting (AAA) in mobile or enterprise networks	• Protocol-specific AVP rewriting (Origin, Realm, Vendor-ID) • Session persistence • CoA port support (RADIUS) • Connection resilience & idle timeout control
Remote Access (RDP)	Terminal sessions to virtual desktops, servers, or management platforms	• Session-aware load balancing • Client IP preservation or SNAT • Idle/queue/half-closed timeout tuning • IP reputation & Geo-IP enforcement
Financial Protocols (ISO8583)	Transaction processing in banking environments	• Message parsing and length-indicator processing • Binary/ASCII/hex framing support • Low-latency forwarding

Traffic Type	Common Use Cases	FortiADC Functions
		• IP reputation + TCP session control
TCP Traffic	Generic TCP-based applications (e.g., proprietary protocols, streaming, tunneling)	• Layer 4 load balancing based on source/destination IP/port • Session persistence and timeout control • Optional IP reputation and DDoS filtering
UDP Traffic	Lightweight or latency-sensitive apps (e.g., DNS, syslog, RTP, IoT messaging)	• Stateless load balancing with port-based distribution • DDoS protection and rate limiting • Optional Geo IP filtering

Choosing the right deployment: Layer 2, Layer 4, and Layer 7 traffic processing

FortiADC supports multiple deployment modes across different layers of the OSI model–Layer 2, Layer 4, and Layer 7–each suited for specific traffic steering, performance, and security requirements. Selecting the appropriate deployment mode depends on your use case, the application protocols involved, and the level of control or visibility required.

In FortiADC**, the terms Layer 2, Layer 4, and Layer 7 do not just describe physical placement or network topology – they specifically refer to the Virtual Server type you choose during configuration.** That choice determines how deeply FortiADC inspects the packets, what features are available, and how load balancing decisions are made.

This section outlines the capabilities, benefits, and trade-offs of each type.

Layer 2 (Transparent Inline Deployment)

In Layer 2 mode, FortiADC is deployed transparently – physically inserted inline between upstream and downstream network devices such as routers, firewalls, or switches. Traffic flows through FortiADC purely due to its physical cabling, not because the device is the destination of any IP packets. This mode allows FortiADC to preserve the original network topology and perform full-featured load balancing, including Layer 7 parsing, without requiring changes to routing or IP addressing.

Despite the name, "Layer 2" in FortiADC refers to the deployment topology, not to a limitation of inspection depth. FortiADC can still perform application layer traffic processing for HTTP/HTTPS when deployed at layer 2.

- **Layer 7 functionality**

  When HTTP/HTTPS application profile is matched to a Layer 2 virtual server, FortiADC can perform full application-layer (L7) inspection and control, including:

  - SSL/TLS offloading and certificate handling
  - Web Application Firewall (WAF) inspection
  - Content rewriting (e.g., URL, header, cookie)
  - Request routing based on URI, Host, query string, etc.
  - Cookie persistence and session awareness

  This makes Layer 2 deployment fully capable of supporting modern web applications with advanced security and traffic steering needs.

- **Layer 2 functionality**
  For non-HTTP/S protocols (e.g., TCP, UDP, or raw IP protocols like GRE, ESP, OSPF), FortiADC simply acts as a MAC-based traffic distributor, balancing across next-hop devices (routers or L3 firewalls) without back-end server awareness.

Layer 2 Use cases

Layer 2 mode is especially suited for:

- **Environments requiring zero changes** to routing tables, subnets, or IP addressing.
- **Client IP preservation**, as FortiADC does not perform NAT in this mode.
- **Seamless deployment**, acting as a transparent "bump-in-the-wire" load balancer.
- **Bridging layer 3 segments**, when FortiADC is placed inline between routers or switches.
- **HTTP/HTTPS services requiring full Layer 7 processing**, without rearchitecting the network.

For detailed information on Layer 2 deployment, see Layer 2 deployment mode (Inline Transparent Deployment) on page 18.

Watch this video on Layer 2 deployment: FortiADC Layer 2 Deployment - Transparent Inline.

Layer 4 (TCP/UDP-Based Load Balancing)

In Layer 4 deployments, FortiADC is configured with a Virtual IP (VIP) that clients target instead of communicating directly with the real servers. It inspects traffic at the IP and TCP/UDP header level, without delving into application-layer content (e.g., HTTP headers or URIs). Based on this, it uses transport-layer load balancing algorithms—such as Round Robin, Least Connection, or Source IP Hash—to intelligently distribute incoming requests across back-end servers.

Unlike Layer 2 deployments, where FortiADC simply forwards traffic based on MAC/IP, **Layer 4 mode allows FortiADC to manipulate IP-layer information, offering significant architectural flexibility.**

- **SNAT (Source NAT)**: Useful for hiding the original client IP and routing traffic predictably.
- **DNAT (Destination NAT)**: Redirects incoming packets to the chosen server's IP.
- **FullNAT**: Rewrites both source and destination IPs for complete control over packet routing.
- **Direct Routing Mode (DSR)**: Forwards the packet with the source and destination IPs unchanged
- **NAT46**: Translates IPv4 client requests into IPv6 requests to communicate with IPv6-only back-end servers.

This ability to manipulate packets enables FortiADC to operate in multi-subnet, NAT-restricted, or complex routing topologies, where neither clients nor servers need to be directly adjacent to the FortiADC.

In addition to the flexible packet forwarding modes mentioned above, FortiADC also offers the following features at layer 4:

- **Content Routing**: FortiADC can route traffic based on IP address, port number, or protocol
- **Security Protection**: FortiADC provides essential Layer 4 security features, including IP reputation filtering, Geo IP restrictions, and TCP-layer DoS mitigation.

Layer 4 Use cases

- **Multi-site or inter-subnet architectures**:
  - Use FullNAT when clients and servers reside in different network segments.
  - Useful in cloud or hybrid environments where direct routing isn't feasible.
- **High-performance back-end distribution**:
  - Lower CPU overhead than Layer 7, ideal for latency-sensitive services.
- **Basic to moderate security enforcement**:
  - Supports DoS protection, GeoIP filtering, and IP reputation at the IP level.
- **Situations where application-layer decisions aren't necessary**, but back-end visibility and connection control are required.
- **Legacy applications or non-web services** that don't require content-aware routing.

For detailed information on DNAT Mode (Router Mode), FULLNAT Mode (Full Reverse Proxy), DSR (Direct Routing Mode), NAT46 Mode, and Tunneling Mode, see Layer 4 Deployment modes on page 22.

Watch the following videos on Layer 4 deployments:

- FortiADC Layer 4 Deployment - DNAT
- FortiADC Layer 4 Deployment - FULLNAT
- FortiADC Layer 4 Deployment - DSR
- FortiADC Layer 4 Deployment - NAT46
- FortiADC Layer 4 Deployment - Tunneling

Layer 7 (Application Layer / Reverse Proxy Mode)

In Layer 7 deployments, application traffic is still destined to the Virtual IP (VIP) configured on FortiADC, just as in Layer 4. However, unlike Layer 4—which primarily relies on IP and transport-layer information—**Layer 7 mode enables FortiADC to parse and act on application-layer content, such as HTTP headers, SIP Call-IDs, RADIUS attributes, SQL query types, and more.**

This full-proxy architecture provides deep visibility and control over traffic, allowing FortiADC to make context-aware decisions, enforce application-specific security policies, and deliver protocol-optimized performance. It's especially powerful in complex environments where traffic steering, persistence, rewriting, and optimization must be tied to the actual content and semantics of the application protocol—not just source/destination IP and ports.

FortiADC at layer 7 offers the following key features:

- **Application-aware traffic distributing**

  FortiADC parses and interprets protocol-specific data (e.g., SIP headers, MySQL query types, HTTP cookies, DNS queries) to:
  - Perform intelligent request routing
  - Support application-specific persistence mechanisms
  - Enable advanced load balancing algorithms (e.g., URI, HOST, etc)

- **Advanced application security and visibility**
  - Application-layer security inspection, applying Web Application Firewall (WAF) for HTTP-based traffic
  - Protocol-specific validation (e.g., malformed DNS queries can be dropped or passed through)
- **Performance optimization**
  - Offloads compute-intensive SSL decryption from back-end servers (HTTPS, TCPS)
  - Reduces CPU consumption by enabling HTTP to back-end (plaintext) forwarding
  - Offers protocol-aware caching (e.g., DNS caching)
  - Supports content rewriting, compression, and connection reuse

Layer 7 Use cases

- Modern web and API applications needing fine-grained traffic control
- Protocols that benefit from parsing (e.g., VoIP, SQL, DNS)
- Deployments requiring SSL offloading and detailed session control
- Environments where security, observability, and customization are key

For detailed information on Layer 7 deployment, see Layer 7 deployment mode (Reverse Proxy Deployment) on page 39.

Watch this video on Layer 7 deployment: FortiADC Layer 7 Deployment - Reverse Proxy.

Decision Guide

When planning your FortiADC deployment, the first key consideration is whether your environment allows changes to the network configuration—such as modifying routing paths, inserting VIPs, or altering client/server IPs. This foundational choice determines which deployment mode best suits your architecture:

- **If your environment does not permit network-level changes, such as altering IP routes or inserting VIPs, Layer 2 deployment is your only viable option**. It provides transparent inline traffic steering while preserving the existing topology and original IP addresses, making it ideal for drop-in scenarios.
- **If network changes are acceptable**, such as configuring FortiADC as the destination for client traffic via a Virtual IP (VIP), then you can **choose between Layer 4 and Layer 7 based on the level of traffic intelligence and control you require**:
  - **Choose Layer 4 when IP and transport-layer visibility is sufficient**—for example, if you only need to load balance based on source/destination IPs and ports or apply basic security like DoS protection or IP reputation.
  - **Choose Layer 7 when you require granular application-layer awareness**. This enables advanced traffic steering (based on URI, Host, headers, cookies, etc.), deep security inspection (WAF, bot mitigation, API protection), SSL offloading, and full application-aware control.

The following table summarizes the behavior of FortiADC in Layer 2, Layer 4, and Layer 7 deployments across several critical dimensions. This reference helps you determine the right deployment model based on your network environment, visibility requirements, security expectations, and traffic control needs.

Criteria	Layer 2	Layer 4	Layer 7
Deployment Characteristic	FortiADC is **physically inserted** into the network path—wired inline between clients and servers. It receives traffic passively without the need for Virtual IPs (VIPs). Traffic reaches FortiADC because it's on the direct Layer 2 path, not because it's the destination IP.	FortiADC is **logically** deployed using VIPs. Clients connect to the VIP (which resolves to FortiADC), and FortiADC actively terminates or forwards the connections.	FortiADC is **logically** deployed using VIPs. Clients connect to the VIP (which resolves to FortiADC), and FortiADC actively terminates or forwards the connections.
Back-end IP Exposure	✓ Yes. **Clients directly see and connect** to real server IPs.	✗ No. FortiADC acts as the sole ingress point; **back-end IPs are hidden**.	✗ No. Back-end servers are fully abstracted; **clients never see them.**
FortiADC Back-end Awareness	✗ No. FortiADC is unaware of actual back-end server IPs; it distributes traffic based on **next-hop MAC/IP**.	✓ Yes. FortiADC tracks and balances traffic to **specific back-end servers**.	✓ Yes. FortiADC tracks **back-end servers** and can route based on app-layer logic.
Supported protocols	Processes all IP protocols with different parsing logic: • **HTTP/HTTPS:** FortiADC can parse full application-layer content, enabling full-feature support including, SSL offloading, content rewrite, etc. • **Other IP Protocols (e.g., GRE, ESP, ICMP, OSPF)**: FortiADC does not inspect or modify these protocols. It acts as a Layer 2 bridge, forwarding packets based solely on MAC	• TCP • UDP • FTP	• HTTP/HTTPS/HTTP2/HTTP3 (incl. APIs), HTTP Turbo, WebSocket • SIP, RADIUS, Diameter, MySQL, MSSQL, DNS, FTP, RTSP, RTMP, RDP, SMTP, ISO8583

Criteria	Layer 2	Layer 4	Layer 7
	addresses or destination IP		
Load Balancing Basis	**MAC** only (e.g., destination MAC routing)	**IP + TCP/UDP headers** (e.g., source/dest IP and port)	**Full application context**: URI, HOST, etc.
Content Routing	✓ • Supports **application-layer routing** for **HTTP/HTTPS** • Supports **IP and Port-Based** content routing for **TCP/UDP**	✓ Based on **port numbers and IP**	✓ **Application-layer routing** based on URL path, Host header, SNI, session, etc.
Scripting	✗	✗	✓
Persistence Options	Moderate: **Source IP/Port, source IP/Port hash**	Moderate: **Source IP/Port, source IP/Port hash**	Extensive: **Cookie-based**
Security integration	⚠ • Support **WAF and Antivirus** for **HTTP/HTTPS** traffic. • No security features for other traffic	⚠ Basic protections such as **DoS mitigation, GeoIP filtering, and IP reputation.**	✓ • **For HTTP/HTTPS traffic**, support **full security features** including WAF, bot protection, API schema validation, antivirus, DDoS protection, IP reputation, etc. • **For other traffic**, supports **GeoIP, IP reputation, antivirus and DDoS** protection

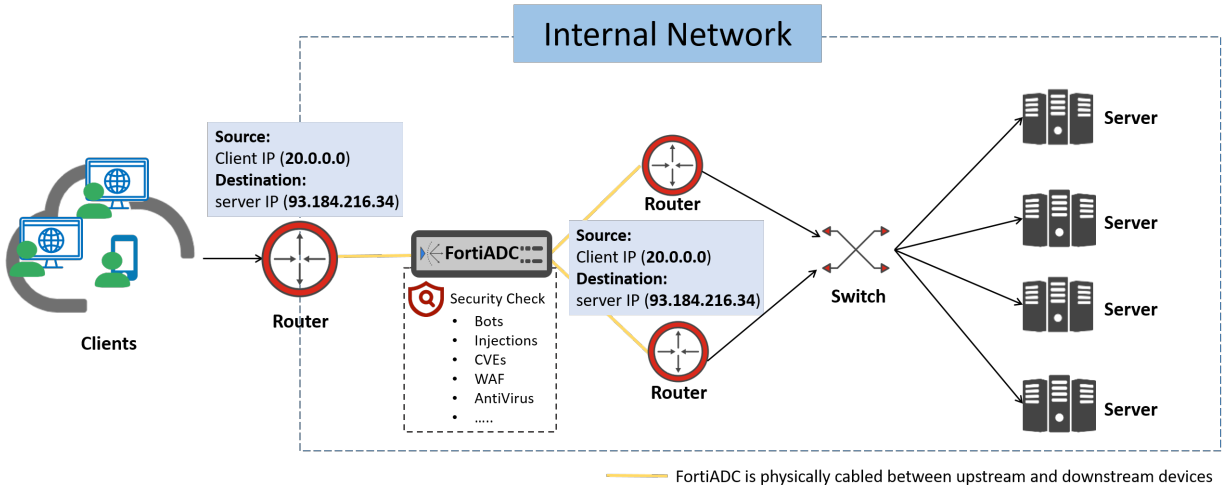
Layer 2 deployment mode (Inline Transparent Deployment)

In Layer 2 deployment, FortiADC operates at the Data Link Layer, delivering traffic to the selected next hop devices by rewriting the destination MAC address while leaving the IP header intact. Although forwarding is performed at Layer 2, FortiADC still uses IP/TCP/UDP information for load-balancing decisions.

Traffic flow of layer 2 deployment

In Layer 2 (Transparent Inline Deployment) mode, FortiADC operates as a bump-in-the-wire and does not require IP reconfiguration. Devices on either side remain unaware of FortiADC's presence. The source and destination IP addresses remain unchanged throughout the path. FortiADC neither performs NAT nor acts as a proxy in Layer 2 mode.

Inline Transparent Deployment (Layer 2)



- **Clients → Router**
  - Clients from the internet initiate requests toward servers inside the data center.
  - The requests first arrive at a router, which is responsible for routing external (internet) traffic into the internal network.
- **Router → FortiADC**
  - The router forwards the traffic to FortiADC, which is placed inline, meaning traffic must pass through FortiADC physically.
  - This is achieved through physical cabling: the router's egress interface is directly connected to one of FortiADC's interfaces (e.g., port1).

- **FortiADC Processing**
  - FortiADC operates transparently at Layer 2. It does not need to know the destination IP address of the packet.
  - **Traffic forwarding:**
    It performs load balancing by selecting the next hop (e.g., router, next router, or back-end servers) based on:
    - Source IP (for source-based content routing, optional)
    - Configured load balancing algorithm (e.g., Round Robin, Source Hash, etc.)
  - **Traffic inspection:**
    If the traffic is HTTP/HTTPS, FortiADC can perform SSL/TLS offloading, WAF, Content rewriting (e.g., URL, header, cookie), and content routing based on URI, Host, query string, etc.
- **FortiADC → Router**
  After processing the packet, FortiADC forwards it to the selected next-hop IP, typically a router connected via a physical port (e.g., port2). Alternatively, FortiADC can forward directly to the destination back-end servers if they are directly connected.
- **Router → Switch →Server Pool**
  - The router then directs the packet to switch, and then reaches one of the back-end servers.
  - The servers are unaware that FortiADC is involved because the deployment is fully transparent (no VIP or SNAT is configured).
- **Return Traffic**
  Response packets from the server follow the reverse path: Server → Switch → Router → FortiADC → Router → Client.
  If the response traffic is HTTP or HTTPS, FortiADC does not merely forward the packets. Instead, it intercepts the response as it passes through (due to its inline placement) and applies configured Layer 7 policies, such as:
  - Web Application Firewall (WAF) rules to detect and block threats.
  - Content rewriting, such as modifying headers or payloads.
  - SSL offloading or re-encryption, depending on how SSL is configured.

Benefits and limitations of Layer 2 deployment

Benefits

- **Topology Preservation:**
  FortiADC can be inserted transparently between routers/switches/servers without changing existing IP addressing or routing logic.
- **Protocol-Agnostic Load Balancing (L2-L4):**
  Supports load balancing based on the IP addresses of next-hop devices and enables content routing based on source IP.
- **Full Layer 7 Inspection for HTTP/HTTPS**
  Even though the deployment is Layer 2, FortiADC can still deeply inspect HTTP/HTTPS traffic. This includes:

  - **SSL/TLS offloading**: Terminate encrypted sessions at FortiADC for inspection.
  - **WAF integration**: Block OWASP Top 10 threats and perform signature-based checks.
  - **Content rewriting**: Modify headers, cookies, or body content in real time.
  - **Load balancing based on application data**: Such as URI, host, headers, cookies, etc.

  These features function identically to Layer 7 deployments, with none of the Layer 7 networking complexity.
- **No VIP Configuration Required:**
  Does not require virtual IPs or DNS changes. It simply passes through traffic, selecting next-hop based on load balancing rules.
- **Quick Deployment:**
  No need for client or server reconfiguration. Just plug between router and switch, configure a few policies, and it's ready.
- **High Performance:**
  Because it operates at lower layers, L2 mode offers minimal latency and high throughput, ideal for high-speed environments.

Limitations of Layer 2 Deployment

- **No back-end server awareness**
  FortiADC cannot perform true server load balancing at this layer. It’s limited to next-hop device load distribution.
- **Inline-Only physical deployment required**
  Because Layer 2 requires inline physical deployment, it lacks routing capabilities and cannot perform cross-subnet or multi-network traffic distribution—unlike Layer 4 or Layer 7, which support packet forwarding across different networks.
- **Limited to simple protocol handling (beyond HTTP/HTTPS)**
  While Layer 2 virtual servers can enable full Layer 7 inspection for HTTP/HTTPS (if configured), non-HTTP protocols (e.g., FTP, SIP, RADIUS) are only processed at Layer 2. It cannot provide Layer 7 content-based features (like SIP Call-ID parsing or content routing based on user attributes) for these protocols. It also doesn't provide any security check for non-HTTP protocols.

Key considerations of network settings in layer 2 deployment

1. Physical cabling

FortiADC must be physically inline between the upstream device (e.g., a router or firewall) and the downstream device (e.g., router, firewall, or servers).

You typically connect:

- Port1 of FortiADC to the upstream router
- Port2 of FortiADC to the downstream router or server(s)

2. No VIP / No routing table required

Unlike Layer 4 or Layer 7 deployments, FortiADC does not require any IP address on the network interfaces (except for management purposes).

- No VIPs are configured.
- Destination IP visibility is not required – FortiADC forwards based on next-hop IP or MAC address.

3. Next-Hop awareness

FortiADC applies load balancing decisions using:

- next hop device's MAC/IP address
- Source IP (Optional. Only required if you apply content routing rules.)

You must configure the next-hop IP addresses as real servers in the FortiADC server pool. These are the devices FortiADC forwards to.

4. Layer 2 virtual server on FortiADC

In **Sever Load Balance > Virtual Server**, define a **Layer 2** type Virtual Server:

- Bind it with a server pool containing the next-hop IPs
- Select an application profile based on the expected traffic type (e.g., HTTP, HTTPS, UDP, TCP). The GUI will automatically present relevant options based on the selected protocol.

When you select an HTTP, or HTTPS application profiles, you will be able to see options like content routing, Content rewriting, WAF profile, etc.

Please note that although the GUI allows you to select an Antivirus profile for non-HTTP protocols, it actually doesn't work. If you attempt to save the configuration with an Antivirus profile selected, the GUI will display an error message.

Refer to Virtual Server and Real Server Pool.

5. HA considerations

In Layer 2 (Inline Transparent) deployments, FortiADC is physically placed in the traffic path between upstream and downstream devices. This inline positioning introduces a single point of failure unless redundancy is built in.

To ensure continuous traffic flow and eliminate downtime risk, it is highly recommended to deploy FortiADC in Active-Passive HA mode. Both appliances must be physically cabled to the same inline points in the network (same router/switch), ensuring seamless takeover.

Layer 4 Deployment modes

Layer 4 mode allows FortiADC to manipulate IP-layer information, offering significant architectural flexibility. FortiADC supports multiple Layer 4 (L4) deployment modes including:

- DNAT Mode (Router Mode)
- FULLNAT Mode (Full Reverse Proxy)
- DSR (Direct Routing Mode)
- NAT46 Mode
- Tunneling Mode

When operating in Layer 4 mode, FortiADC supports:

- IP-based load balancing algorithms (e.g., Round Robin, Least Connection).
- Source and destination IP address modification, allowing FortiADC to rewrite IP headers as needed for routing, NAT, or policy enforcement.
- IP-level security features, including:
  - IP Reputation filtering
  - Geo IP-based access control
  - TCP-layer DoS mitigation

However, due to the lack of visibility into the application payload, Layer 4 deployments do not support:

- Layer-7 based content routing (e.g., URI Hash, Host Hash)
- Web Application Firewall (WAF) protection
- SSL/TLS offloading or inspection
- HTTP header manipulation or rewriting

In the following section, we introduce FortiADC's Layer 4 deployment modes, with a focus on traffic flow, key benefits, and essential network configuration requirements.

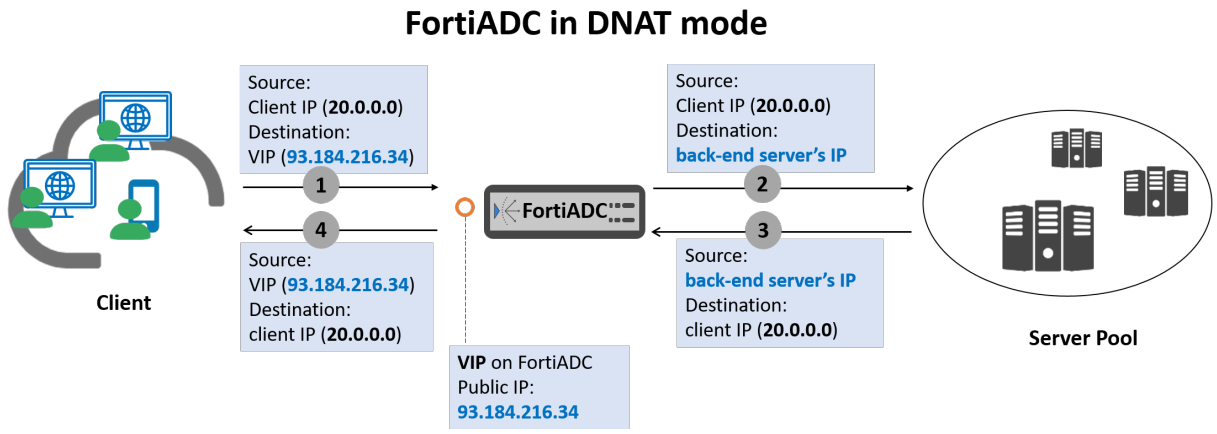
DNAT Mode (Router Mode)

In DNAT mode, FortiADC replaces the destination IP (the VIP) in the client request with the selected back-end server's IP. The source IP remains unchanged.

Traffic flow of DNAT mode

In Destination NAT (DNAT) mode, FortiADC modifies only the destination IP address, preserving the client's source IP.

The following diagram illustrates the traffic flow between the Client, FortiADC, and back-end servers in DNAT mode. Please note that in real-world deployments, additional network devices–such as routers, switches, or firewalls–may exist both in front of and behind FortiADC. However, these components are omitted in the diagram, as they are not directly relevant to the specific packet flow and behavior we aim to explain through FortiADC.



- The response packet from the back-end server is routed back to FortiADC based on the server's routing table, typically via its default gateway.

1. **Client sends a packet to FortiADC's VIP (e.g., 93.184.216.34)**
2. **FortiADC receives the paket and rewrite the destination IP.**
   - FortiADC rewrites the destination IP from VIP to the back-end server's IP (e.g., 192.168.1.20)
   - The source IP (client's IP) is not changed (no SNAT).
   - Packet sent to back-end server:
     - src=Client IP (e.g., 20.0.0.0)
     - dst=server IP (e.g., 192.168.1.20)
3. **Server processes the request and sends the response to Client IP (e.g., 20.0.0.0).**
   - Server sends the packet to its default gateway, which is FortiADC
4. **FortiADC reverses the DNAT, replacing the source IP back to its VIP (e.g., 93.184.216.34).**
   Packet sent to client:
   - src=VIP (e.g., 93.184.216.34)
   - dst=Client IP (e.g., 20.0.0.0)

Benefits and limitations of the DNAT mode

Key benefits

- **Client IP Preservation**: Back-end servers see the original client IP address, as no source NAT is applied. **This is ideal for environments where maintaining accurate client identity is important for logging, auditing, or security policies.**
- **Back-End Server IP Protection**: Clients interact only with the Virtual IP (VIP) configured on FortiADC, not the real IP addresses of the back-end servers. This setup prevents the exposure of internal server IPs to

the Internet, effectively **hiding the actual server infrastructure and reducing the attack surface.**

- **Simpler Server Configuration**: Unlike in Direct Routing (DR) mode, back-end servers do not need to be configured with the VIP. They operate using their own IP addresses, making deployment and maintenance simpler.

Key considerations of network settings in DNAT mode

1. VIP configuration on FortiADC

A publicly accessible Virtual IP (VIP)—which is mapped to your application's domain name in DNS—must be configured on FortiADC to receive incoming client requests.

- This VIP acts as the entry point for external traffic.
- It should be associated with a virtual server and bound to a network interface that can receive requests from the client side (typically the WAN or DMZ interface).

2. Subnet alignment between FortiADC and back-end servers

Ideally, **FortiADC should have at least one network interface in the same subnet as the real IP addresses of the back-end servers**. This allows FortiADC to route traffic directly to the servers without relying on intermediate routers or additional route configurations.

However, if subnet alignment is not feasible in your network design, you can **configure a default gateway (or static routes) on FortiADC that enables it to forward traffic to a next-hop device (e.g., a router)**, which then routes the packets to the back-end servers. Refer to Configuring static routes.

3. Route settings on back-end servers

If FortiADC is routing traffic directly to the back-end servers:

- Each back-end server must have a return route that ensures client-bound (Internet-bound) traffic is sent back through FortiADC.
- This is critical because the client IP is preserved, and return traffic must pass through FortiADC for reverse DNAT.

If there are intermediate routers or switches between the back-end servers and FortiADC:

- Make sure the route table on the back-end servers and those intermediate routers or switches are configured so that the response packet (whose destination IP is the client IP) is forwarded back to FortiADC.
- FortiADC will then perform reverse DNAT, replacing the destination IP (client IP) with the VIP, ensuring session continuity.

4. DNAT setting on FortiADC

On FortiADC, in the **Basic** tab of the **Virtual Server** settings, set the **Type** to **Layer 4** and select **DNAT** as the **Packet Forwarding Method**.

When DNAT mode is enabled:

- FortiADC rewrites the destination IP address of the incoming packet:
  - From the VIP to the selected real server's IP.
  - Source IP (client IP) remains unchanged.
- When the real server replies:
  - FortiADC performs reverse DNAT, replacing the source IP (server IP) back to the VIP, so the client sees the response as coming from the original destination it contacted.

FULLNAT mode (Full Reverse Proxy)

FULLNAT (also known as Full Network Address Translation) is a packet forwarding method where both the source and destination IP addresses of a packet are modified by FortiADC.

Traffic flow of FULLNAT mode

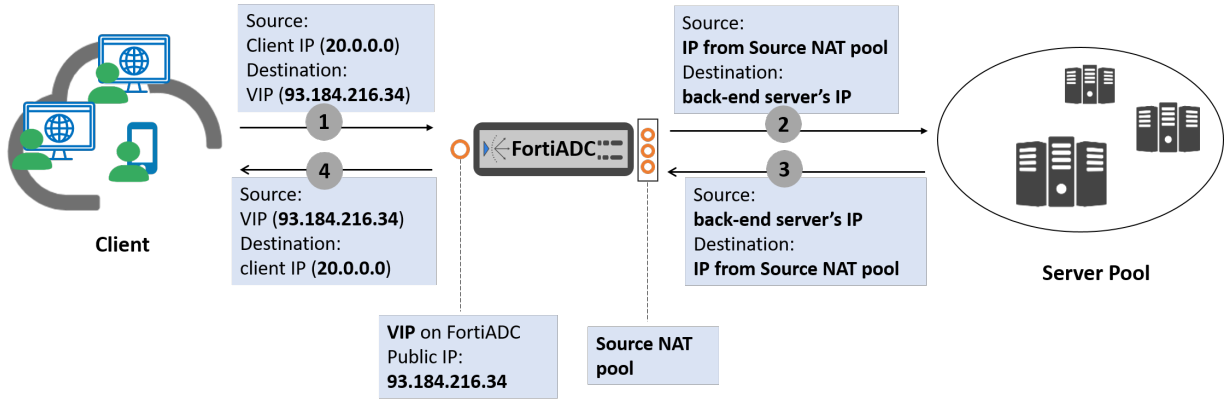
FortiADC performs both Source IP Translation (SNAT) and Destination IP Translation (DNAT) in FULLNAT mode:

- The client's original source IP is replaced with an IP address selected from the FortiADC source NAT pool. This makes the packet appear as if it originates from FortiADC itself (or a NAT pool IP), rather than the real client.
- The destination IP (VIP) is translated to the real IP address of the selected back-end server from the server pool.

This gives FortiADC full control over both directions of the connection and avoids routing complexity.

The following diagram illustrates the traffic flow between the Client, FortiADC, and back-end servers in FULLNAT mode. Please note that in real-world deployments, additional network devices—such as routers, switches, or firewalls—may exist both in front of and behind FortiADC. However, these components are omitted in the diagram, as they are not directly relevant to the specific packet flow and behavior we aim to explain through FortiADC.

FortiADC in FULLNAT mode



Here's how the packet flow works:

1. **Client initiates a request to the VIP residing on FortiADC**
   - src=Client IP (e.g., 20.0.0.0)
   - dst=VIP on FortiADC (e.g., 93.184.216.34)
2. **FortiADC receives the packet, and applies FULLNAT When Forwarding to the Real Server**
   - When forwarding the request to the selected back-end server, FortiADC performs Full NAT by rewriting both source and destination IP addresses:
     - Source IP: FortiADC selects an IP from the configured Source NAT Pool. This pool is a predefined range of IPs (e.g., 192.168.100.10-192.168.100.20) that FortiADC can use to represent the client.
     - Destination IP: The VIP is translated to the real server's IP based on the server pool configuration.
   - The packet that reaches the real server appears as:
     - src = [Selected IP from NAT Pool]
     - dst = [Real Server IP]

   This ensures the back-end server sees the request as coming from a valid, routable IP address that belongs to FortiADC, avoiding routing conflicts or access control issues.
3. **The real server replies to the NAT IP, which it sees as the client.**
4. **FortiADC receives the response, performs reverse NAT:**
   - Source IP is changed back to the VIP
   - Destination IP is changed back to the original client IP
   - The packet sent to client appears as:
     - src=VIP on FortiADC (e.g., 93.184.216.34)
     - dst =Client IP (e.g., 20.0.0.0)

Benefits and limitations of the FULLNAT mode

Key Benefits

- **Simplified routing**: Compared with DNAT mode, FULLNAT mode does not require the back-end server to configure FortiADC as its default gateway.
- **Full path control:** FortiADC handles both directions of traffic.
- **Strong IP hiding:**
  - Client never sees the real server IP.
  - Server never sees the client IP.

Limitations

FULLNAT mode is not suitable for applications that require native client IP visibility, as it replaces the source IP (client IP) with an address from FortiADC's source NAT pool.

However, FortiADC provides a setting that allows it to insert the client IP into the forwarded request. Refer to "Preserving Client IP" in Key considerations of network settings in FULLNAT mode on page 27.

Key considerations of network settings in FULLNAT mode

1. VIP configuration on FortiADC

A publicly accessible Virtual IP (VIP)—which is mapped to your application's domain name in DNS—must be configured on FortiADC to receive incoming client requests.

- This VIP acts as the entry point for external traffic.
- It should be associated with a virtual server and bound to a network interface that can receive requests from the client side (typically the WAN or DMZ interface).

2. Subnet alignment between FortiADC and back-end servers

Ideally, **FortiADC should have at least one network interface in the same subnet as the real IP addresses of the back-end servers**. This allows FortiADC to route traffic directly to the servers without relying on intermediate routers or additional route configurations.

However, if subnet alignment is not feasible in your network design, you can **configure a default gateway (or static routes) on FortiADC that enables it to forward traffic to a next-hop device (e.g., a router)**, which then routes the packets to the back-end servers. Refer to Configuring static routes.

3. Source NAT Pool settings on FortiADC

A Source NAT (SNAT) Pool in FortiADC is a range of IP addresses used to replace the original client IP address when forwarding traffic to the back-end servers.

You can configure Source NAT Pools via the **Source NAT Pool** tab in **Server Load Balance > Virtual Server**. Refer to Using source pools.

4. FULLNAT setting on FortiADC

On FortiADC, in the **Basic** tab of the **Virtual Server** settings, set the **Type** to **Layer 4** and select **FULLNAT** as the **Packet Forwarding Method**.

When FULLNAT mode is enabled, FortiADC:

- Replaces the original client IP address with an IP address selected from a predefined source NAT pool.
- Replaces the VIP (Virtual IP) with the real server IP, so the packet can be routed to the correct back-end server.

5. Preserving Client IP

By default, **FULLNAT mode is not suitable for applications that require the client's original IP address to be visible at the back-end server**. This is because FortiADC replaces the source IP (Client IP) with an IP from its source NAT pool, effectively masking the client IP.

However, when a TCP Application Profile is applied to the virtual server, and the option **Client IP Insertion in TCP Option** is enabled, FortiADC inserts the client's original IP address into the TCP payload of the forwarded request. This allows back-end servers that support parsing such metadata—such as through the PROXY protocol or custom application logic—to extract and log the original client IP. Refer to Configuring Application profiles.

Direct Routing Mode (DSR, Direct Server Return)

The Direct Routing Mode is also known as DSR (Direct Server Return). In this mode, FortiADC receives client requests, selects a real server, and forwards the packet with the source and destination IPs unchanged. The real server responds directly to the client, bypassing FortiADC on the return path.

Traffic flow of Direct Routing (DSR) mode

In DR mode, only the client request goes through FortiADC. The response goes directly from the server to the client, bypassing FortiADC. This is called asymmetric routing, because the request and response take different paths.

In DR mode, the VIP (virtual IP) that lies on FortiADC to receive the client request must also exist on the back-end server, but:

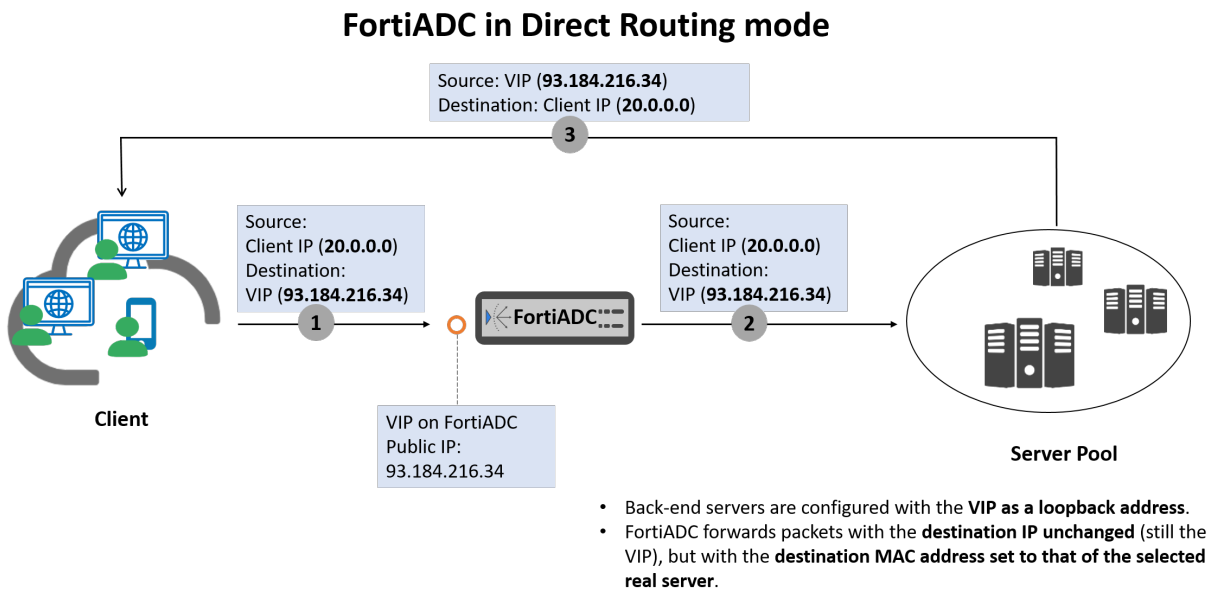
- It should not respond to ARP (Address Resolution Protocol) requests.
- It should be bound as a loopback IP or set with no-arp flag.

This ensures:

- The back-end server accepts packets with the VIP as destination, but
- It does not advertise itself as owning that IP on the network.

The following diagram illustrates the traffic flow between the Client, FortiADC, and back-end servers in Direct Routing (DR) mode. Please note that in real-world deployments, additional network devices—such as routers, switches, or firewalls—may exist both in front of and behind FortiADC. However, these components are omitted

in the diagram, as they are not directly relevant to the specific packet flow and behavior we aim to explain through FortiADC.



1. **Client initiates request to VIP**
   - Client sends a request to the virtual IP (e.g., 93.184.216.34).
   - FortiADC receives the request on its front-end interface configured with the VIP.
2. **FortiADC selects a back-end server**
   - In FortiADC's server pool configuration, each back-end server is identified by its own IP address (e.g., 192.168.1.20), not the VIP.
   - Based on the configured load balancing algorithm and server health status, FortiADC selects a suitable back-end server.
   - FortiADC then sends an ARP request to the selected server's IP to obtain its MAC address.
   - Once resolved, FortiADC forwards the original client packet:
     - Destination IP remains the VIP (93.184.216.34).
     - MAC address is replaced with the selected back-end server's MAC.
3. **back-end server receives the packet and sends the response**
   - The back-end server has the VIP configured as a loopback address (lo:0) with no ARP. This ensures that it does not respond to ARP requests for the VIP (important–only FortiADC should respond to ARP for the VIP).
   - Since it recognizes the VIP as a valid local address, it accepts the packet.
   - It sends response directly to the client, bypassing FortiADC:
     - src = VIP (93.184.216.34)
     - dst = Client_IP (20.0.0.0)

Benefits and limitations of the Direct Routing (DSR) mode

Key Benefits

- Fast – only one hop through FortiADC
- Efficient – FortiADC doesn't become a bottleneck
- Good for high-throughput services (e.g., video streaming, large downloads)

Key considerations of network settings in Direct Routing (DR) mode

1. VIP Configuration on Both FortiADC and Back-end Servers

A publicly accessible Virtual IP (VIP)–which is mapped to your application's domain name in DNS–must be configured on both FortiADC and each back-end server:

- On FortiADC, the VIP (e.g., 93.184.216.34) should be configured in the **Virtual Server** settings and bound to the appropriate network interface that receives client traffic.
- On the back-end servers, the same VIP must be configured as a loopback address (e.g., lo:0) and must be set with no ARP response. This ensures the server can receive packets destined for the VIP, while only FortiADC responds to ARP requests for that IP.

2. Subnet alignment between FortiADC and back-end servers

Ideally, **FortiADC should have at least one network interface in the same subnet as the IP addresses of the back-end servers**. This allows FortiADC to route traffic directly to the servers without relying on intermediate routers or additional route configurations.

However, if subnet alignment is not feasible in your network design, you can **configure a default gateway (or static routes) on FortiADC that enables it to forward traffic to a next-hop device (e.g., a router)**, which then routes the packets to the back-end servers. Refer to Configuring static routes.

3. Direct Routing settings on FortiADC

On FortiADC, in the **Basic** tab of the **Virtual Server** settings, set the **Type** to **Layer 4** and select **Direct Routing** as the **Packet Forwarding Method**.

When set to this mode:

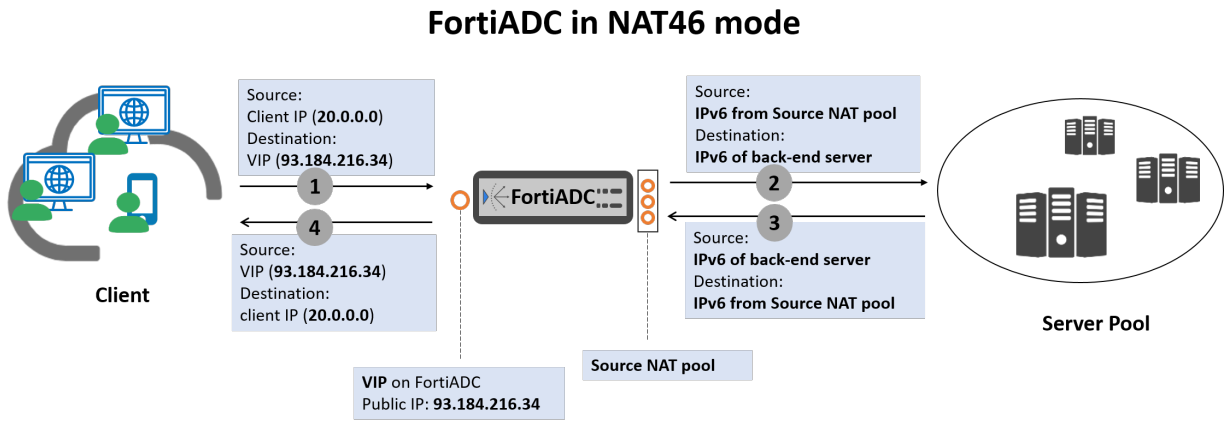
- FortiADC does not modify the source or destination IP address.
- Only the destination MAC address is changed to that of the selected back-end server.

NAT46 mode

NAT46 mode is used when IPv4 clients need to access IPv6 servers. In NAT46 mode, FortiADC serves as a gateway to perform protocol translation between IPv6 and IPv4. This enables legacy IPv4-only clients to access modern IPv6 servers without requiring dual-stack deployment on those servers.

Traffic flow of NAT46 mode

The following diagram illustrates the traffic flow between the Client, FortiADC, and back-end servers in NAT46 mode. Please note that in real-world deployments, additional network devices—such as routers, switches, or firewalls—may exist both in front of and behind FortiADC. However, these components are omitted in the diagram, as they are not directly relevant to the specific packet flow and behavior we aim to explain through FortiADC.



Here's how the packet flow works:

1. Client Sends IPv4 Request

   The client sends a packet to a Virtual IP (VIP) on FortiADC, which is an IPv4 address, publicly reachable, representing the services provided by the application server.

   Example:

   - src = Client IP (e.g., 20.0.0.0)
   - dst = VIP on FortiADC (e.g., 192.168.1.20)
2. FortiADC performs NAT46 translation
   - Source IP is translated: FortiADC replaces the IPv4 client IP with an IPv6 address selected from a NAT46 Source Pool. It also creates a mapping table linking the original IPv4 client IP to the selected IPv6 address for return traffic.
   - Destination IP is translated: FortiADC replaces the VIP (IPv4) with the IPv6 address of the real server (from the server pool).

   FortiADC also adjusts the IP header and protocol fields to conform to IPv6 standards before forwarding the request.

   Resulting packet sent to real server:

- src = 2001:db8:100::1 (from NAT46 pool)
- dst = 2001:db8:200::20 (IPv6 real server)

3. Back-end server sends IPv6 response

   The real server responds to the IPv6 address.

4. FortiADC performs reverse NAT46

   When FortiADC receives the response packet from the back-end server, FortiADC looks up its NAT46 mapping table to determine the original IPv4 client IP that corresponds to the IPv6 destination address.

   In the response it sends to the client:

   - It sets the original IPv4 client IP as the destination IP.
   - It restores the VIP as the source IP.

   Final response packet from FortiADC to the client:

   - src = VIP on FortiADC (e.g., 192.168.1.20)
   - dst = Client IP (e.g., 20.0.0.0)

Benefits and limitations of the NAT46 mode

Key benefits

NAT46 mode offers a clear advantage in dual-stack application deployments, especially when back-end infrastructure supports only IPv6 but clients may be connecting over either IPv4 or IPv6.

- **Seamless IPv4-to-IPv6 Translation**: It enables clients using IPv4 to access services hosted on IPv6-only servers, without requiring any changes on the client side.
- **Simplified Back-end Configuration**: The back-end servers do not need to support or be aware of IPv4. FortiADC handles all the address translation transparently.
- **No DNS Modifications Required**: Clients continue to resolve the domain to an IPv4 VIP. The translation is handled entirely on FortiADC, avoiding DNS-level dual-stack complexity.

Key considerations of network settings in NAT46 mode

1. VIP configuration on FortiADC

A publicly accessible Virtual IP (VIP)—which is mapped to your application's domain name in DNS—must be configured on FortiADC to receive incoming client requests.

- This VIP acts as the entry point for external traffic.
- It should be associated with a virtual server and bound to a network interface that can receive requests from the client side (typically the WAN or DMZ interface).

2. Subnet alignment between FortiADC and back-end servers

Ideally, **FortiADC should have at least one network interface in the same subnet as the real IP addresses of the back-end servers**. This allows FortiADC to route traffic directly to the servers without relying on intermediate routers or additional route configurations.

However, if subnet alignment is not feasible in your network design, you can **configure a default gateway (or static routes) on FortiADC that enables it to forward traffic to a next-hop device (e.g., a router)**, which then routes the packets to the back-end servers. Refer to Configuring static routes.

3. NAT46 setting on FortiADC

On FortiADC, in the **Basic** tab of the **Virtual Server** settings, set the **Type** to **Layer 4** and select **NAT46** as the **Packet Forwarding Method**.

When this mode is enable, FortiADC accept IPv4 client requests to the VIP, and then translates them to IPv6.

4. Source NAT Pool settings on FortiADC

A Source NAT (SNAT) Pool in FortiADC is a range of IP addresses used to replace the original client IP address when forwarding traffic to the back-end servers.

You can configure Source NAT Pools via the **Source NAT Pool** tab in **Server Load Balance > Virtual Server**. Refer to Using source pools.

Tunneling Mode

In the Tunneling mode, FortiADC encapsulates client traffic in a tunnel (IP-in-IP) and sends it to the back-end server. The back-end decapsulates it, processes it, and replies directly to the client. It's useful when real servers are across Layer 3 networks or in cloud/data center environments where direct routing is not feasible. Please note that while powerful for cross-network or hybrid-cloud deployments, tunneling mode is relatively uncommon and may introduce complexity in routing, monitoring, and troubleshooting.

In Tunneling Mode (e.g., using IPIP), the tunnel endpoint that FortiADC forwards the request to is often the public IP of a perimeter device at the remote data center, such as:

- A router
- A switch with tunnel termination capability
- A firewall
- Another FortiADC or FortiGate acting as the tunnel receiver

FortiADC does not see the actual back-end server IP address directly. Instead, it treats the tunnel endpoint as the next hop to reach the real server. Therefore, the “back-end server” defined in FortiADC’s server pool is typically the tunnel termination device at the remote site – not the actual application server receiving the request.

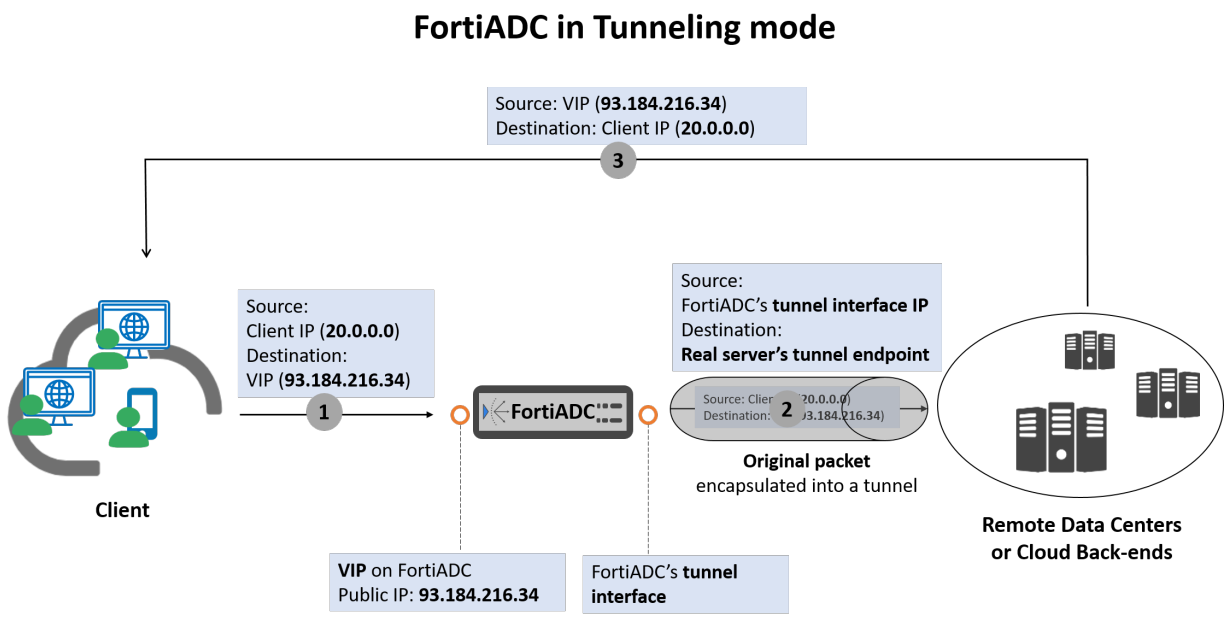
In the following configuration sections, we will continue to use the term “back-end server” for consistency. However, keep in mind that in the context of Tunneling mode, it specifically refers to the perimeter device at the remote site that terminates the tunnel, not necessarily the final application server itself.

Traffic flow of the Tunneling mode

In Tunneling mode:

- FortiADC encapsulates the original client request (e.g., an HTTP, HTTPS, ICA, or TCP packet) within an IPIP tunnel.
- The destination IP of this tunnel is the IP of the tunnel endpoint at the remote data center or cloud (e.g., a router, firewall, or FortiADC/FortiGate instance).
- The payload of the encapsulated packet is preserved – including original source IP (client) and destination IP (VIP) – enabling full transparency at the destination.
- The back-end server configured in the Server Pool refers to the tunnel endpoint IP (not necessarily the app server itself).
- The tunnel endpoint de-capsulates the packet and forwards it internally to the actual application server.

The following diagram illustrates the traffic flow between the Client, FortiADC, and back-end servers in Tunneling mode. Please note that in real-world deployments, additional network devices–such as routers, switches, or firewalls–may exist both in front of and behind FortiADC. However, these components are omitted in the diagram, as they are not directly relevant to the specific packet flow and behavior we aim to explain through FortiADC.



Here's how the packet flow works:

1. **Client initiates a request to FortiADC**
   - A client sends a request to a VIP (Virtual IP) hosted on FortiADC. FortiADC receives the packet on its front-end interface.

     Example packet:
     - src = Client IP (e.g., 20.0.0.0)
     - dst = VIP (e.g., 93.184.216.34)
2. **FortiADC encapsulates the packet and sends it to the back-end server**

- FortiADC selects a real server based on its load balancing algorithm. Instead of modifying the packet headers (as in DNAT or FULLNAT), FortiADC encapsulates the entire original packet into a tunnel and adds a new outer IP header:
  - Outer source IP = FortiADC's outbound interface IP
  - Outer destination IP = Real server's tunnel endpoint
- While the inner payload remains as the original client packet:
  - src = Client IP (e.g., 20.0.0.0)
  - dst = VIP (e.g., 93.184.216.34)

3. **Back-end server sends response**

   The real server receives the tunneled packet destined to the tunnel endpoint.
   - It de-capsulates the packet and sees the original client packet with the VIP as destination. Because it has the VIP configured as a loopback address (with no ARP, similar to Direct Routing mode), packet destined to the VIP can reach the back-end server.
   - The real server sends the response directly back to the client, bypassing FortiADC:
     - src = VIP (e.g., 93.184.216.34)
     - dst = Client IP (e.g., 20.0.0.0)

Benefits and limitations of the Tunneling mode

Key benefits

- **Secure remote deployment with encapsulation**

  Connections between FortiADC and the back-end resources are encapsulated, ensuring secure transmission of traffic across untrusted or public networks.

  This deployment model is particularly well-suited for scenarios where back-end resources are located in distributed, cloud-hosted, or geographically distant environments, and FortiADC must traverse public or external networks to reach them — while still maintaining the integrity, confidentiality, and reliability of the data exchange.
- Allows application servers to see full client details without relying on headers like `X-Forwarded-For`.

Limitations

- Back-end servers must support tunnel de-capsulation.
- Asymmetric routing may require careful firewall and routing rule adjustments.
- Harder to troubleshoot due to encapsulated traffic.

Key considerations of network settings in Tunneling mode

1. VIP Configuration on Both FortiADC and Back-end Servers

A publicly accessible Virtual IP (VIP)—which is mapped to your application's domain name in DNS—must be configured on both FortiADC and each back-end server:

- On FortiADC, the VIP (e.g., 93.184.216.34) should be configured in the **Virtual Server** settings and bound to the appropriate network interface that receives client traffic.
- On the back-end servers, the same VIP must be configured as a loopback address (e.g., lo:0) and must be set with no ARP response. This ensures the server can receive packets destined for the VIP, while only FortiADC responds to ARP requests for that IP.

2. Back-end server's real IP address

In Tunneling Mode, the back-end server IP is typically a publicly routable IP address – or more precisely, the IP of the tunnel endpoint device (e.g., firewall/router) that will receive and de-capsulate the tunneled traffic.

On FortiADC, this IP address must be added to the Server Pool configuration as the "real server" address. FortiADC uses this IP as the destination address of the encapsulated packet when building the IPIP tunnel.

If the tunnel endpoint IP is not on a directly connected subnet, you must configure a default gateway or static routes on FortiADC to ensure it can reach the destination through a next-hop router or firewall. Refer to Configuring static routes.

3. Tunneling mode settings on FortiADC

On FortiADC, in the **Basic** tab of the **Virtual Server** settings, set the **Type** to **Layer 4** and select **Tunneling** as the **Packet Forwarding Method**.

When set to this mode,

- FortiADC encapsulates the original client request (e.g., an HTTP, HTTPS, ICA, or TCP packet) within an IPIP tunnel.
- The destination IP of this tunnel is the IP of the tunnel endpoint at the remote data center or cloud (e.g., a router, firewall, or FortiADC/FortiGate instance).
- The payload of the encapsulated packet is preserved – including original source IP (client) and destination IP (VIP) – enabling full transparency at the destination.
- The back-end server configured in the Server Pool refers to the tunnel endpoint IP (not the actual app server behind it).
- The tunnel endpoint de-capsulates the packet and forwards it internally to the actual application server.

Please note that FortiADC does not require you to manually define an IPIP tunnel interface as you would on a FortiGate or router.

Instead, tunnels are dynamically established per request, using the standard network interface that has IP reachability to the destination IP of the server pool member–which serves as the tunnel endpoint.

Summary

Here's a summary table of FortiADC's Layer 4 deployment modes with their features and differences.

Traffic Flow - Incoming

DNAT	Client ⟶ FortiADC ⟶ Server
FULLNAT	Client ⟶ FortiADC ⟶ Server
Direct Routing	Client ⟶ FortiADC ⟶ Server
NAT46	IPv4 Client ⟶ FortiADC ⟶ IPv6 Server
Tunneling	Client ⟶ FortiADC ⟶ Server
	Traffic Flow - Outgoing
DNAT	Server ⟶ FortiADC ⟶ Client
FULLNAT	Server ⟶ FortiADC ⟶ Client
Direct Routing	Server ⟶ Client
NAT46	IPv6 Server ⟶ FortiADC ⟶ IPv4 Client
Tunneling	Server ⟶ Client
	VIP Handling
DNAT	VIP configured only on FortiADC
FULLNAT	VIP configured only on FortiADC
Direct Routing	VIP configured on both FortiADC and real servers (as loopback, no ARP)
NAT46	VIP configured only on FortiADC
Tunneling	VIP configured on both FortiADC and real servers (as loopback, no ARP)
	Client IP Preservation
DNAT	Yes
FULLNAT	No, but FortiADC supports client IP insertion via TCP Application Profile settings
Direct Routing	Yes
NAT46	No
Tunneling	Yes
	Key Requirements
DNAT	• Server's default gateway is FortiADC
FULLNAT	• Configure source NAT pool on FortiADC • Optional - client IP insertion via TCP Application Profile
Direct Routing	• Servers must support asymmetric routing • VIP set as non-ARP loopback on real servers
NAT46	• NAT46 source pool required

Tunneling	• Servers must support asymmetric routing • Real server (Tunnel destination) usually is the public IP of perimeter device
	Use Cases
DNAT	Standard data center deployments
FULLNAT	Environments needing full control over both directions
Direct Routing	Performance-driven environments needing true client IP
NAT46	Bridging IPv4 clients to IPv6-only back-ends
Tunneling	Multi-site, remote DC, or cloud deployments

Layer 7 deployment mode (Reverse Proxy Deployment)

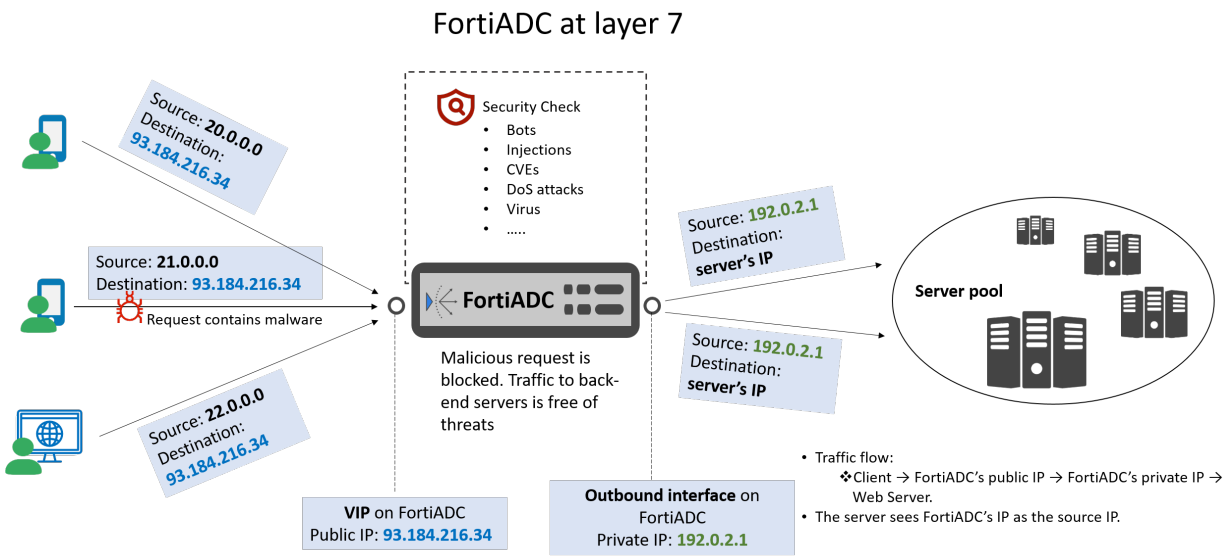
In Layer 7 deployments, application traffic is destined to the Virtual IP (VIP) configured on FortiADC, just as it is in Layer 4. However, unlike Layer 4, FortiADC functions as a full reverse proxy — it terminates the client connection, inspects the application-layer data, and establishes a new connection to the appropriate back-end server.

Layer 7 mode is ideal for application-layer traffic, including:

- **Web applications:** HTTP, HTTPS, WebSocket, and RESTful APIs
- **VoIP:** SIP
- **Media services:** RTSP, RTMP
- **Database protocols:** MySQL, MSSQL,
- **Directory and authentication services:** RADIUS, Diameter
- **Remote desktop access:** RDP
- **DNS-based services:** DNS
- **File transfer:** FTP
- **Email services:** SMTP
- **Transaction systems:** ISO8583

Traffic flow of Layer 7 deployment

In a Layer 7 (Application Layer) deployment, FortiADC operates as a reverse proxy, providing full visibility into and control over **application-layer traffic**. This enables FortiADC to make intelligent, content-aware decisions for advanced traffic management, security enforcement, and load balancing.



Here is the traffic flow and key operations at layer 7:

- **Client initiation**

  A client with IP address 20.0.0.0 initiates a connection to the service hosted behind FortiADC, targeting the virtual server IP (e.g. 93.184.216.34).
- **FortiADC receives and processes the request**

  Operating at Layer 7, FortiADC performs deep application-level inspection of headers and payloads to make intelligent traffic-handling decisions.

  To learn more about FortiADC's Layer 7 capabilities for specific protocols (such as SIP, MSSQL, Diameter, RDP, etc.), refer to Appendix: FortiADC's Layer 7 capabilities by protocol on page 43.
- **Load balancing and connection establishment**

  Based on the policy decisions (e.g., server availability, load balancing algorithm, application attributes), FortiADC selects a back-end server to forward the request to.
- **Source NAT (if configured)**

  When FortiADC initiates a connection to a backend server, the source IP address in the packet is typically the IP address of FortiADC's outgoing interface–for example, 192.0.2.1 in the diagram above. However, if a SNAT (Source NAT) rule is configured, FortiADC replaces the interface IP with an IP address from the defined SNAT source pool.
- **Back-end response handling**

  The back-end server processes the request and sends the response to 192.0.2.1 or the SNAT address. FortiADC then forwards the response back to the original client 20.0.0.0 using the VIP 93.184.216.34.

  If the response traffic is HTTP or HTTPS, FortiADC does not merely forward the packets. Instead, it intercepts the response as it passes through (due to its inline placement) and applies configured Layer 7 policies

Benefits and limitations of layer 7 deployment

Key Benefits

Application-Aware Traffic Handling

FortiADC can inspect the full application payload, allowing it to:

- Route requests based on application content (e.g., URI, headers, cookies).
- Support protocol-specific features for HTTP/S, SIP, DNS, RADIUS, MSSQL, and others.
- Apply intelligent persistence methods based on session cookies, user credentials, or protocol behavior.

Enhanced Security and Policy Enforcement

- Access control based on application-layer attributes (e.g., host headers, paths, user-agent).
- WAF integration for blocking OWASP Top 10 threats.
- Application-layer DoS/DDoS detection.
- Advanced authentication policies for user access control (SAML, LDAP, RADIUS, etc.).

Fine-Grained Load Balancing and Optimization

- Content-based load balancing, routing requests to different server pools based on content.
- Compression, caching, and HTTP/2/3 support to improve performance.
- Header rewriting and application gateway features to adjust and normalize traffic.

Visibility and Logging

- Full logging of application-level transactions, helping with troubleshooting and analytics.
- Detailed statistics per application, URI, or policy match.

Limitations

- Unlike Layer 2 deployments (which can operate transparently in bridge mode), Layer 7 deployment requires network topology changes such as pointing traffic to FortiADC's VIP instead of directly to servers.
- Protocols that do not operate at the application layer (e.g., ICMP, GRE, IPsec, custom TCP) cannot be processed or load balanced at Layer 7. For such traffic, Layer 4 or Layer 2 deployments are more suitable.

Key considerations of network settings in Layer 7 deployment

1. Network Interfaces

Two network interfaces

Ideally, FortiADC is typically configured with two network interfaces:

- One for client-side (WAN) traffic
- One for server-side (LAN) traffic

This separation helps simplify routing, improve performance, and enhance security by logically isolating external and internal traffic flows.

Single Network interface

However, FortiADC can also operate with a single interface, where FortiADC:

- Receives client traffic (e.g., from the internet) on that interface
- Forwards server traffic to a different subnet **via routing**

To make this work:

- You must define a default gateway (or appropriate static routes) on FortiADC
- That gateway/router must be able to route traffic to the back-end servers (e.g., 192.0.2.0/24)

Example setup of single network interface

- **Interface port1:** IP: 20.0.2.1/24. This is the only interface used.
- **VIP configured:** 20.0.2.100 (DNS entry of your app points to this)
- **Back-end servers:** 192.0.2.0/24
- **Default Gateway:** 20.0.2.254 (a router that can reach both 20.0.2.1/24 and 192.0.2.0/24)

FortiADC receives traffic for 20.0.2.100, applies load balancing, and sends requests to 192.0.2.x. The gateway routes it correctly.

Related configuration guides:

- Configuring network interfaces
- Configuring virtual servers
- Configuring static routes

2. SSL/TLS Configuration (If Using TCPS/HTTPS)

FortiADC acts as an SSL proxy when deployed at layer 7. It terminates the TCPS/HTTPS connection from the client and presents a server certificate to prove its authority for your application domain.

After decrypting and inspecting the traffic, FortiADC establishes a new connection to the back-end server, which can be either encrypted (TCPS/HTTPS) or unencrypted (TCP/HTTP), depending on the configuration between FortiADC and the server. This back-end connection is entirely independent of the front-end connection.

Because FortiADC handles the SSL handshake with the client, **you must upload your CA-signed server certificate to FortiADC** so it can present it on behalf of your application and validate the domain's authenticity.

Related configuration guides:

- Importing a local certificate
- SSL offloading
- Configuring client SSL profiles

3. Client IP Preservation

At layer 7 FortiADC terminates the client session and then establishes a new session with the back-end web server. As a result:

- The web server does not see the real IP address of the client.
- Instead, it sees FortiADC's IP address as the source of incoming requests.

Since some web applications need the real client IP (e.g., for rate limiting, logging, or geographical analysis), FortiADC allows you to insert or append the client's original IP into an HTTP header `X-Forwarded-For (XFF)`. For more details, see the description of the "X-Forwarded-For (XFF)" option in Configuring Application profiles.

4. VIP Configuration

Assign the IP address associated with your application domain to FortiADC as the Virtual IP (VIP). When configuring the Virtual Server, enter this IP in the **Address** field.

5. Virtual Server Settings

- **Type:** Select Layer 7.
- **Interface:** Select the interface that receives incoming client traffic (usually WAN-facing).
- **Address:** Enter the VIP (Virtual IP) that matches your application's domain in DNS.
- **Application Profile:** Attach an Application Profile with protocol types.

6. Back-end Server Configuration

- **Firewall Rules:** Allow traffic only from FortiADC's outbound IP (e.g., 192.0.2.1).

7. HA considerations

In Layer 7 (Reverse Proxy) mode, FortiADC acts as the entry point for all traffic destined for your application. If FortiADC fails, the IP address associated with your application's domain becomes unreachable, resulting in service disruption. Therefore, deploying FortiADC in a High Availability (HA) cluster is critical to ensure continuous service delivery, session integrity, and system performance. You can choose from:

- Active-Passive: One node handles all traffic, and the standby takes over upon failure.
- Active-Active: Both nodes process traffic simultaneously.
- VRRP-based HA: Virtual Router Redundancy Protocol to manage failover dynamically.

Appendix: FortiADC's Layer 7 capabilities by protocol

- Layer 7 capabilities for HTTP/HTTPS traffic on page 44
- Layer 7 capabilities for DNS traffic on page 46
- Layer 7 capabilities for MSSQL traffic on page 49
- Layer 7 capabilities for MySQL traffic on page 47

Layer 7 capabilities for HTTP/HTTPS traffic

FortiADC provides full-featured support for HTTP/HTTPS traffic, combining secure protocol handling with advanced application delivery, visibility, and optimization.

Packet Processing

We will introduce FortiADC's HTTP/HTTPS packet processing capabilities from the following perspectives

SSL offloading (If HTTPS is used)

- Terminates incoming HTTPS connections, decrypts the content for inspection and optimization, and optionally re-encrypts when forwarding to the back-end.
- Acts as a secure proxy, presenting server certificates on behalf of applications–centralizing certificate management and simplifying compliance.

HTTP2 and HTTP3 support

Supports next-gen protocols (HTTP/2 and HTTP/3), enabling multiplexing, better latency, and improved mobile performance.

Compression & Decompression

Speeds up content delivery by compressing HTTP payloads (GZIP, Deflate) and decompressing if needed.

Caching

Reduces back-end load and latency by caching static or semi-static content.

Content Rewriting

FortiADC allows you to modify content within HTTP and HTTPS requests/responses using content rewriting policies. This can include:

- **Header Rewriting:** Add, remove, or replace HTTP headers (e.g., Server, Set-Cookie, etc.).
- **URL Rewriting:** Change URL paths or query parameters on-the-fly.
- **Redirects:** Implement conditional HTTP redirects.

Resilient connection handling

- **Multiple timeout settings** manage every stage of HTTPS session lifecycle, from initial connection to request processing, ensuring graceful failover and efficient resource use.
- Requests waiting due to server overload are queued and served or dropped based on **queue timeout settings**.
- Can maintain **session stability** even when one side closes, useful in mobile or distributed environments.
- Supports **HTTP Keepalive** to reuse connections for multiple requests—reducing handshake overhead and improving response time.

Advanced buffer and header tuning

Advanced options like **buffer size and max header count** allow tuning for high-performance or high-concurrency environments.

Client IP preservation

Maintains client IP visibility via:

- Transparent source IP pass-through (when **Client Address** is enabled)
- **X-Forwarded-For** header injection

Content Routing

Routing content based on HTTP Host Header, HTTP Request URL, HTTP Referer Header, Source IP address, SNI.

Load Balancing Methods

Round Robin, Least Connection, URI Hash, Full URI Hash, Host Hash, Host Domain Hash, Dynamic Load

Persistence

Source Address, Source Address Hash, Source Address-Port Hash, HTTP Header Hash, HTTP Request Hash, Cookie Hash, Persistent Cookie, Insert Cookie, Embedded Cookie, Rewrite Cookie, SSL Session ID (HTTPS only), Passive Cookie

Security Check

- Antivirus
- DoS prevention
- IP Reputation
- Geo IP-based access control
- WAF checks including SQL/XSS injection check, signature-based inspection, anomaly detection, bot mitigation, API Protection, etc.

Layer 7 capabilities for DNS traffic

Domain Name System (DNS) is the protocol that translates human-readable domain names (like www.example.com) into IP addresses (like 192.0.2.1) that computers use to locate each other on the internet. It works much like a phonebook for the internet, enabling users to access websites and services by names rather than complex numeric IPs.

FortiADC supports application-aware load balancing and optimization for DNS traffic using its DNS-specific application profiles. The following summarizes FortiADC's DNS-specific capabilities.

Packet Processing

DNS caching

- Granular control over the cache behavior, including **per-record size, total cache size, and the lifespan (TTL) of cached entries**.
- Cached DNS entries can be returned **in full (all records) or rotated using round-robin logic**.

  When a DNS response returns multiple resource records (RRs), such as multiple A or AAAA records for a single domain, most DNS resolvers and clients use the first IP address in the response. By rotating the order (e.g., returns B, C, A → then C, A, B → etc.), FortiADC helps spread the client traffic across all available servers—even if it doesn't perform real load balancing in the traditional sense.

Client address preservation

- FortiADC supports **using the original client IP as the source IP** when connecting to the back-end DNS server. This is useful for server-side logging or policies that depend on original source IP.

Security & Query Handling

FortiADC adds security and robustness to DNS services with features including:

- **Malformed Query Protection**: Automatically drop or forward malformed DNS requests to prevent protocol abuse.

- **UDP-to-TCP Redirection**: Enforce TCP fallback for large or authenticated queries (e.g., DNSSEC), improving reliability and enabling deeper inspection.
- **Query Length Enforcement**: Define maximum allowable query sizes to block overly large or potentially malicious packets.

Content Routing

FortiADC doesn't support content routing for DNS traffic.

Load Balancing Methods

Round Robin, Least Connection

Persistence

FortiADC doesn't support persistence for DNS traffic.

Security Check

DNS Query Flood Protection.

Layer 7 capabilities for MySQL traffic

FortiADC provides application-aware load balancing for MySQL database traffic, supporting both traditional replication and advanced distributed database architectures. It intelligently distributes traffic to MySQL servers based on read/write intent, user, SQL type, and more.

Packet Processing

Single-Primary Mode

FortiADC supports processing traffic by the following rules in Single-Primary Mode.

- **Write operations (INSERT, UPDATE, DELETE) go to a designated primary MySQL server**.
- **Read operations (SELECT) are offloaded to one or more secondary servers**.
- Default behavior is auto-applied, but can be customized with user-defined rules.

Common use cases of Sharding mode include web applications with high read-to-write ratio, E-commerce platforms with real-time inventory reads and transactional writes, etc.

Sharding Mode

In Sharding mode, different subsets of data are processed by different server groups. FortiADC supports setting the following rules in Sharding mode:

- Range-based: routes queries by key ranges.
- Hash-based: uses a hashed key to distribute traffic.

Common use cases of Sharding mode include Large SaaS apps or ERPs needing high concurrency and database scaling, Apps with tenant-based data (multi-group) and heavy transactional loads, etc.

Advanced MySQL-Aware Features

- Granular rules for routing based on:
  - SQL statements (**SELECT, INSERT**, etc.)
  - Database, table, user, Client IP
- Multiple **user-defined MySQL profiles** can be created.
- Uses a valid MySQL user account to parse and inspect SQL requests.

Content Routing

FortiADC doesn't support content routing for MySQL traffic.

Load Balancing Methods

Round Robin, Least Connection

Persistence

FortiADC doesn't support persistence for MySQL traffic.

Security Check

FortiADC doesn't support security check for MySQL traffic.

Layer 7 capabilities for Diameter traffic

Diameter is a protocol used primarily for authentication, authorization, and accounting (AAA) in telecom networks (e.g., LTE, 5G). FortiADC provides application-aware control for Diameter traffic, allowing selective modification and session handling.

- Packet Processing on page 49
- Content Routing on page 49

Packet Processing

Application-Layer Attribute Rewriting

FortiADC allows **modification of critical Diameter AVPs (Attribute-Value Pairs)**, including Origin-Host, Origin-Realm, Vendor-ID, and Product-Name.

Session Handling and Timeout

- FortiADC supports **Idle Timeout** configurable up to 86,400 seconds (24 hours). This controls session persistence.
- FortiADC can **keep the client connection alive** even if the server closes its side. This ensures resilience in mobile core or AAA networks where clients expect persistent sessions.

Security

FortiADC can enable **TLS encryption for Diameter messages** on the client side. This is useful for securing AAA communications in environments that require encrypted Diameter signaling (e.g., Diameter over TLS as per 3GPP standards).

Content Routing

FortiADC doesn't support content routing for Diameter traffic.

Load Balancing Methods

Round Robin

Persistence

Source Address, Diameter Session ID (default)

Security Check

FortiADC doesn't support security check for Diameter traffic.

Layer 7 capabilities for MSSQL traffic

FortiADC supports application-aware load balancing for MSSQL, allowing you to optimize database performance through intelligent traffic distribution.

Packet Processing

FortiADC supports single-primary mode for MSSQL traffic, where:

- **Primary MSSQL server handles all write operations** (INSERT, UPDATE, DELETE).
- **Secondary MSSQL servers handle all read operations** (SELECT).

Content Routing

FortiADC doesn't support content routing for MSSQL traffic.

Load Balancing Methods

Round Robin, Least Connection

Persistence

FortiADC doesn't support persistence for MSSQL traffic.

Security Check

DoS prevention

Layer 7 capabilities for RADIUS traffic

RADIUS (Remote Authentication Dial-In User Service) is a protocol commonly used for authentication, authorization, and accounting (AAA) in network access and policy enforcement. FortiADC offers application-level handling for RADIUS traffic, with security and control enhancements.

Packet Processing

Client Identity Preservation

- FortiADC can **preserve the original source IP** when forwarding requests to the real RADIUS server.
- It can also **maintain the original client-side source port** in upstream connections.

It helps back-end servers maintain accurate client identity for logging or policy application.

Session Timeout Control

FortiADC supports setting **a session lifetime**to ensure efficient resource cleanup for idle or dropped RADIUS sessions.

Dynamic Authorization Support

FortiADC supports **Change of Authorization (CoA)**, allowing real-time policy enforcement and user session control. It's also supported to configure the UDP port (default: 3799) for CoA messages.

Content Routing

FortiADC doesn't support content routing for RADIUS traffic.

Load Balancing Methods

Round Robin

Persistence

RADIUS attribute

Security Check

Geo IP-based access control

Layer 7 capabilities for SIP traffic

SIP (Session Initiation Protocol) is a signaling protocol used to initiate, maintain, and terminate real-time communication sessions. These sessions can include voice, video, messaging, and other multimedia applications over IP networks (like the Internet). Common use cases include VoIP telephony (e.g., Zoom, Skype, Cisco, 3CX), Video conferencing systems, etc.

FortiADC provides application-aware handling of SIP traffic. The following summarizes FortiADC's SIP-specific capabilities.

- Packet Processing on page 52
- Content Routing on page 52

Packet Processing

SIP Traffic Inspection and Handling Capabilities

- **Transport Protocol and Session Awareness**
  - **Supports both TCP and UDP on client and server sides**. This helps bridging the protocol incomplibility between server and client.
  - Implements **CRLF keepalive pings** (especially important for TCP), helping maintain long-lived SIP sessions.
- **Health Checking and Failure Handling**
  When FortiADC detects that a client or server is unreachable, it can **either drop the connection silently or send a SIP response** (e.g., SIP error code). This allows it to act intelligently based on SIP state.

SIP Header Manipulation

FortiADC supports **insertion and deletion of headers in both directions for Request and Response messages**. It's useful in the following scenarios:

- Enforcing header policies (e.g., custom tags, topology hiding)
- Ensuring interworking between mismatched SIP implementations
- Security enhancement by removing sensitive headers

Source and Media Address Management

- **Client address preservation**: FortiADC supports using the original client IP as the source IP when connecting to the back-end SIP server. This is useful for server-side logging or policies that depend on original source IP.
- **Media address rewriting**: FortiADC supports modifying the media IP address to a specified value (e.g., NAT traversal support or symmetric RTP scenarios).
- **X-Forwarded-For**: FortiADC supports inserting the source IP into the SIP request via the X-Forwarded-For header for better traceability and diagnostics.

Size Control

- FortiADC supports **enforcing a size limit to SIP messages**, providing a safeguard against oversized or malformed packets.

Content Routing

FortiADC doesn't support content routing for SIP traffic.

Load Balancing Methods

Round Robin, URI Hash, Full URI Hash

Persistence

Source Address, Source Address Hash, Source Address-Port Hash, SIP Call ID

Security Check

DoS prevention.

Layer 7 capabilities for RTSP and RTMP traffic

RTSP (Real-Time Streaming Protocol) is the protocol used for video streaming, e.g., IP camera feeds, surveillance systems.

RTMP (Real-Time Messaging Protocol) is a streaming protocol used to deliver audio, video, and data. Use cases include live streaming platforms (e.g., YouTube Live, Twitch, Facebook Live), video conferencing, etc.

FortiADC provides application-aware handling of RTSP and RTMP traffic. The following summarizes FortiADC's RTSP/RTMP specific capabilities.

Packet processing

Client IP address preservation (RTSP and RTMP)

FortiADC supports **using the original client IP address as the source IP in connections to the real server**. This is useful for back-end logging, access control, or when the media server performs IP-based session tracking.

Size Control (RTSP only)

FortiADC supports **enforcing a size limit to RTSP Header**, preventing attacks that exploit oversized headers.

Content Routing

FortiADC doesn't support content routing for RTSP and RTMP traffic.

Load Balancing Methods

Round Robin, Least Connection

Persistence

Source Address, Source Address Hash

Security Check

DoS prevention

Layer 7 capabilities for RDP traffic

FortiADC provides application-aware support for Remote Desktop Protocol (RDP) traffic, which is widely used for accessing desktops or servers remotely in enterprise and data center environments.

Packet Processing

Connection management and timeouts

FortiADC allows **fine-tuned control over RDP connection lifecycle** through Client Timeout, Server Timeout, Queue Timeout, and Connect Timeout.

These options help manage load, free up unused resources, and improve user experience by avoiding hanging sessions.

Advanced tuning for high-performance RDP

- Allows **low-level tuning of RDP connection buffers**. It is suitable for high-performance and high-concurrency RDP environments where precise memory and resource optimization is critical.
- FortiADC provides **option to continue serving half-closed connections**, which can help prevent client-side disconnections if the server has silently dropped its end.

Client IP Preservation

FortiADC can **forward the original client IP to the back-end RDP server**, which is critical for accurate session logging or enforcing user-specific policies on the server.

Content Routing

FortiADC doesn't support content routing for RDP traffic.

Load Balancing Methods

Round Robin, Least Connection

Persistence

Source Address, Source Address Hash, Source Address-Port Hash, RDP Cookie

Security Check

IP reputation, Geo IP-based access control, and DoS prevention.

Layer 7 capabilities for FTP traffic

FortiADC provides protocol-aware handling for FTP traffic, helping secure and optimize file transfer sessions across your network.

Packet Processing

Session timeout management

- FortiADC can control **how long an FTP session remains open** when it is idle and the client hasn't formally closed it.
- FortiADC can control **how long to keep a connection alive** after receiving a TCP FIN signal. It's useful for tuning FTP data/control channel closure behavior.

FTP security modes

FortiADC supports three levels of FTP security - **None, Explicit, and Implicit**. These modes allow FortiADC to handle encrypted FTP connections in both active and passive modes, enhancing data privacy without sacrificing flexibility.

Client IP preservation

FortiADC can **forward the original client IP to the back-end FTP server**, which is critical for accurate session logging or enforcing user-specific policies on the server.

Content Routing

FortiADC support source IP-based content routing for FTP traffic.

Load Balancing Methods

Round Robin, Least Connection

Persistence

Source Address, Source Address Hash

Security Check

Antivirus, IP reputation, Geo IP-based access control, and DoS prevention.

Layer 7 capabilities for SMTP traffic

SMTP stands for Simple Mail Transfer Protocol. It is the standard communication protocol used to send emails across the Internet. FortiADC provides application-aware support for SMTP traffic.

Packet Processing

STARTTLS control

FortiADC allows **flexible handling of STARTTLS encryption negotiation**. This is critical for enforcing security policies for SMTP servers.

Client IP preservation

FortiADC can **forward the original client IP to the back-end SMTP server**, which is critical for accurate session logging or enforcing user-specific policies on the server.

Command filtering

FortiADC can **forbid specific SMTP commands** often used for reconnaissance or abuse, including EXPN, VRFY, and TURN. When forbidden, FortiADC blocks these commands from reaching the back-end server, enhancing security posture.

Domain Name matching

You can **specify the domain name to associate with the back-end SMTP server**. If you're using FortiADC to load balance SMTP services for multiple domains, this field helps identify which domain the SMTP Virtual Server is responsible for.

Content Routing

FortiADC doesn't support content routing for SMTP traffic.

Load Balancing Methods

Round Robin, Least Connection

Persistence

Source Address, Source Address Hash

Security Check

Antivirus, Geo IP-based access control, and DoS prevention.

Layer 7 capabilities for ISO8583 traffic

FortiADC supports ISO8583 protocol handling through its customizable TCP-based application profile configuration. ISO8583 is commonly used in financial transaction systems such as ATM networks, POS terminals, and payment gateways.

Packet Processing

Protocol awareness and message parsing

- FortiADC can process ISO 8583 messages encoded in either ASCII (default) or Binary. This ensures compatibility with different implementations of the protocol.
- FortiADC allows you to define how it should parse the message length:
  - **Type**: binary, BCD, decimal-str, or hex-str,
  - **Shift**: Offset (in bytes) to where the length field starts
  - **Size**: Number of bytes to read to calculate length

These settings ensure that FortiADC can accurately extract complete ISO messages from TCP payloads.

- You can configure a fixed header length (before the MTI), and a hexadecimal trailer signature for identifying message boundaries. This is critical for supporting non-standard ISO8583 variations used in banking systems.

Session management and resilience

FortiADC can control how long a session remains open when idle but not formally closed.

Content Routing

FortiADC doesn't support content routing for ISO8583 traffic.

Load Balancing Methods

Round Robin

Persistence

FortiADC doesn't support persistence for ISO8583 traffic.

Security Check

Antivirus, Geo IP-based access control, and DoS prevention.

www.fortinet.com