Deployment Guide

Deploying VMware Horizon View 7 with A10 Thunder ADC (Application Delivery Controller)

Table of Contents

Disclaimer

This document does not create any express or implied warranty about A10 Networks or about its products or services, including but not limited to fitness for a particular use and noninfringement. A10 Networks has made reasonable efforts to verify that the information contained herein is accurate, but A10 Networks assumes no responsibility for its use. All information is provided "as-is." The product specifications and features described in this publication are based on the latest information available; however, specifications are subject to change without notice, and certain features may not be available upon initial product release. Contact A10 Networks for current information regarding its products or services. A10 Networks' products and services are subject to A10 Networks' standard terms and conditions.

Overview

This deployment guide contains configuration procedures for the A10 Networks® Thunder® ADC series line of high-performance Application Delivery Controller (ADC) to support VMware Horizon View 7.x deployment.

VMware Horizon View is a virtual desktop infrastructure (VDI) solution that simplifies IT manageability and control while delivering the highest fidelity end-user experience across devices and networks. For more information on VMware Horizon View 7.x, visit: https://my.vmware.com/web/vmware/evalcenter?p=horizon-7.

Deploying A10 ADC for VMware Horizon View

The A10 Thunder® Application Delivery Controller (ADC) is built upon A10 Networks' Advanced Core Operating System (ACOS®) platform and works seamlessly with any business application to ensure highly available, fast, secure, and consistent application delivery in any physical or cloud data centers. Deploying the A10 Thunder ADC solution for enterprise business applications such as VMware Horizon View, enables IT operations to enjoy reliable application services while strengthening high availability and maximizing elasticity and performance for business-critical applications.

This guide provides the deployment topology and design and detailed configuration steps of Thunder ADC when load balancing the VMware Horizon View 7.x.

The Thunder Series fully supports VMware Horizon View and provides the following benefits:

- Load balancing and high availability of VMware Connection Servers
- Usage of VMware Connection Servers in private networks (not directly reachable from outside)
- Offloading/relaxing of CPU-intensive SSL/TLS functionalities from VMware Connection Servers to Thunder ADC.

For example, stronger TLS ciphers (e.g., ECC and PFS) can be used on users and Thunder ADC (front end), while weaker ciphers such as RSA can be used on the backend between Thunder ADC and VMware Connection Servers to reduce CPU load due to complicated TLS transaction.

***Note**: It is possible for operators to deploy SSL Offload by disabling SSL/TLS connection on the Connection Servers (refer to KB), however it is not recommended. Please design and deploy carefully based on organization's security policy and environmental requirements.*

Deployment Prerequisites

When deploying Thunder ADC (Application Delivery Controller) with VMware Horizon View, the following are prerequisites and assumptions:

- Users have some basic configuration familiarity with both the A10 ADC and VMware Horizon View.
- The various VMware Horizon View servers are already installed and in good working order.
- The A10 Application Delivery Controller (ADC) is running ACOS Release 4.1.4-GR1-P3 or higher (tested with vThunder with ACOS 5.1.0, hosted on VMware ESXi hypervisor)
- Product and version tested:
  - vThunder ADC running ACOS version 5.1.0
  - Hypervisor: ESXi 6.7
  - VMware Horizon View 7.12

***Note**: While the A10 vThunder ADC is referenced throughout this guide, the A10 Thunder ADC hardware appliance can be used as well.*

Deployment Architecture

Topology

Figure 1 shows the VMware Horizon View deployment topology used for this guide. The Thunder ADC (Application Delivery Controller) is deployed as a proxy for authentication (1st phase connection) and HTTPS traffic destined to the Connection Servers for both internal and tunneled connection/external clients, and provides load balancing for virtual desktop connections for tunneled connection (external) clients using multiple protocols such as Blast Extreme, PC-over-IP, RDP and HTTPS/ HTML-based Blast Extreme.

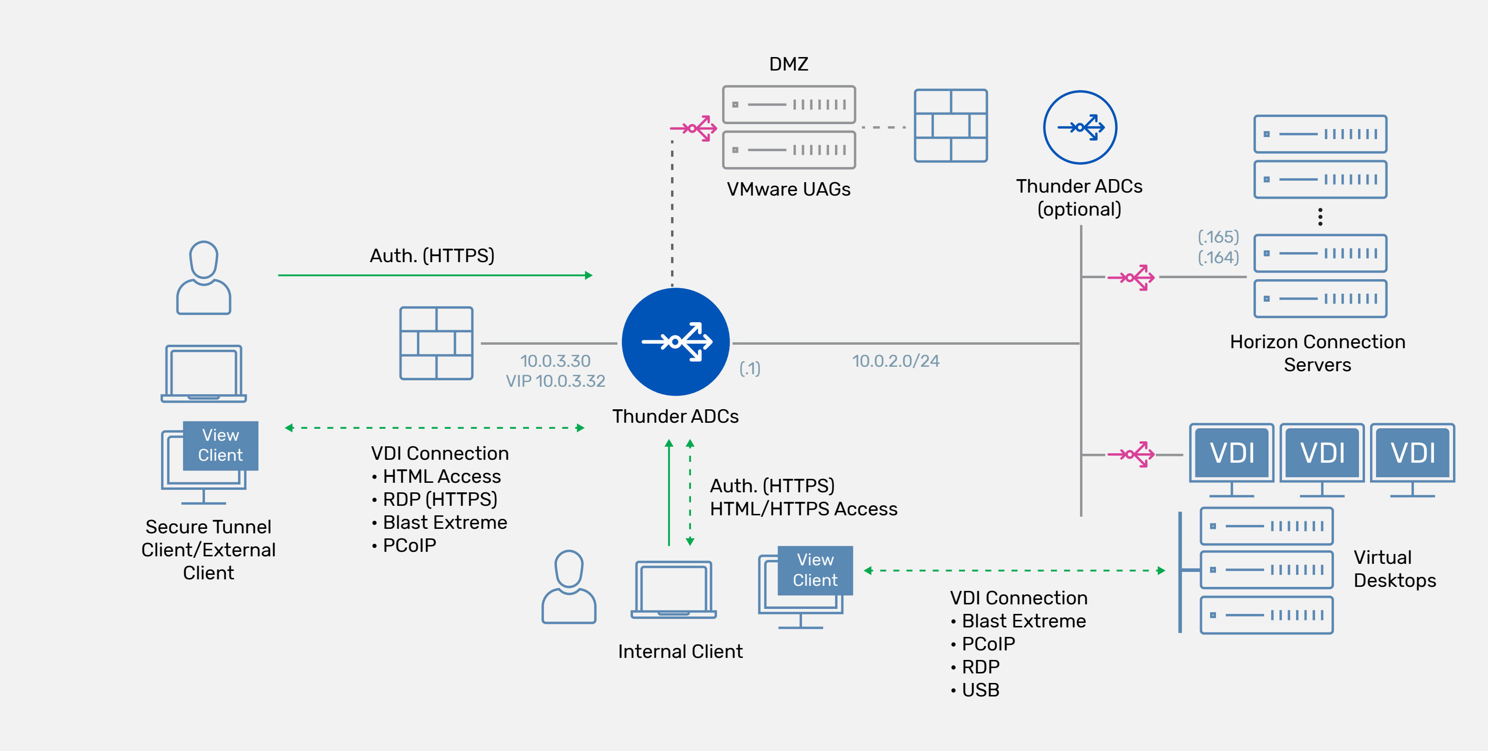


Figure 1: Horizon View topology

Horizon Protocols

Horizon View uses several different protocols and two phases to establish the desktop connection. When Horizon View clients connect to their virtual desktops, the first phase of the connection is **always** a connection to the primary XML-API protocol over HTTPS (443), which provides authentication, authorization, and session management. After the successful authentication from the first phase, the second phase follows based on Connection Server settings, client location, and protocols to be used.

The following are protocols used for VMware Horizon View when clients access their virtualized desktops:

- For HTTPS (HTML5) access: TCP 443
- For RDP access: TCP 3389
- For PC-over-IP (PCoIP) access: TCP/UDP 4172
- For Blast Extreme access: TCP 8443
- For HTTPS (Browser-based) Blast Extreme access: TCP 8443

Internal/Direct Connection

When **“Use secure tunnel connection to desktop**” is disabled (unchecked) and referred to as a “direct (or internal) connection,” the first phase connection is for authentication from the Horizon Client to the Connection Server over HTTPS (TCP 443). In the second phase, protocol session (Blast Extreme, RDP, PCoIP or HTTP HTML5 access) will then connect directly from the Horizon Client to the Horizon Agent virtual desktop.

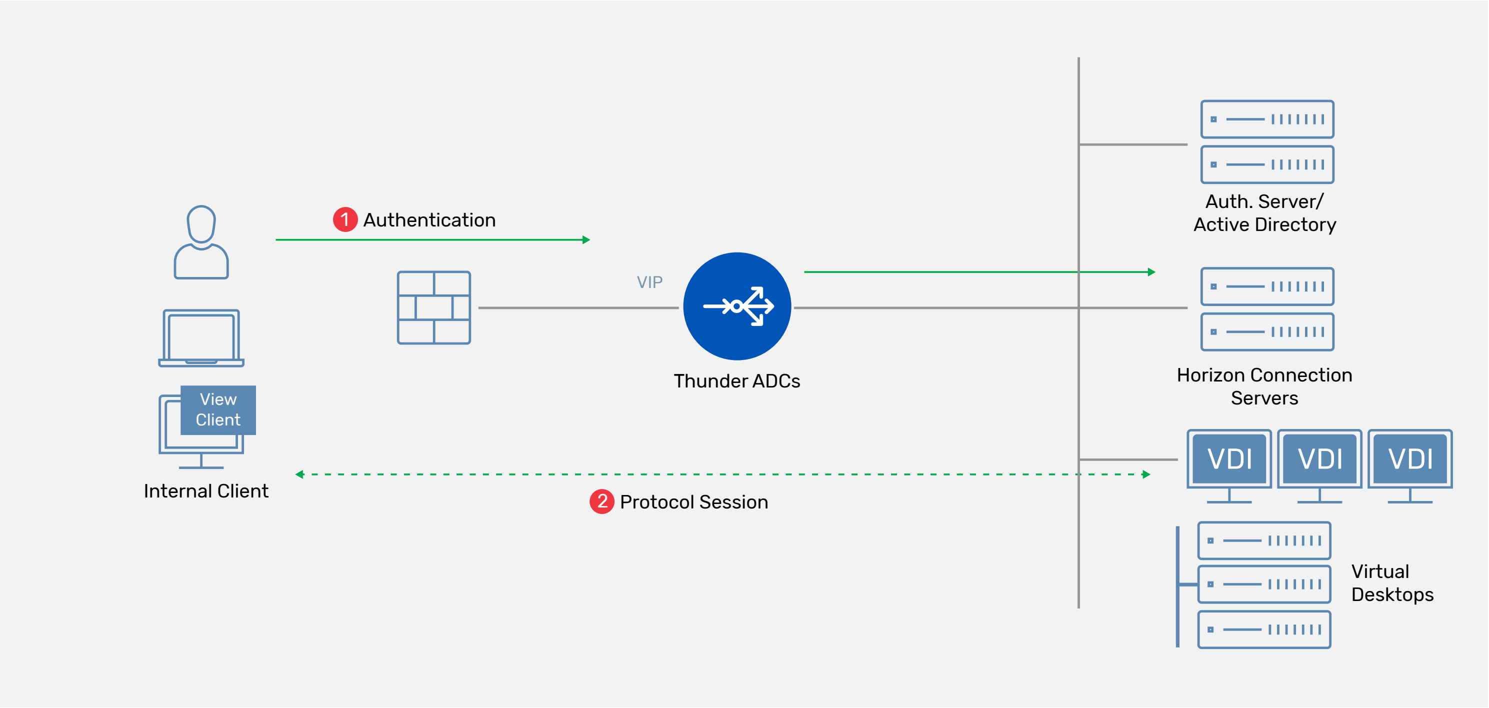


Figure 2: Internal connection

In deployments with the Thunder ADC for load balancing for Connection Servers, the traffic flow is as follows:

1. The client device makes a connection to the virtual server (VIP) address on the ADC for the first phase. The ADC then establishes a new connection to the Horizon Connection Servers for authentication.
2. Once authenticated and desktop availability are determined, the protocol session (regardless of Blast Extreme, RDP, PCoIP or HTTP HTML5 access) from Horizon Client will be sent directly to the assigned virtual desktop on Horizon Agents.

__Note__: Depending on the configuration, the protocol session can be routed, but not proxied, through the Thunder ADC.

__Note__: If the client uses Web Browser to access a virtual desktop, HTML Access (TCP 8443) goes to the Blast Secure Gateway on the Horizon Connection Server via Thunder ADC.

__Note__: Depending on the network configuration, direct server return (DSR) deployment can be used with Thunder ADC when it is deployed with one-arm mode. That way, the return traffic from the Horizon Connection Servers are directly routed back to the client without passing through Thunder ADC.

Tunneled Connection (secure tunnel enabled)

When the "**secure tunnel**" is enabled, the Horizon Client first-phase connection is for authentication from the Horizon Client to the Connection Server over HTTPS (TCP 443). The second-phase connection goes through the Horizon View Connection Server via Thunder ADC when users connect to a virtual desktop. The protocols session using either RDP, Blast, PCoIP or HTTPS HTML5 connect through the connection server using secure tunnel to the connection server.

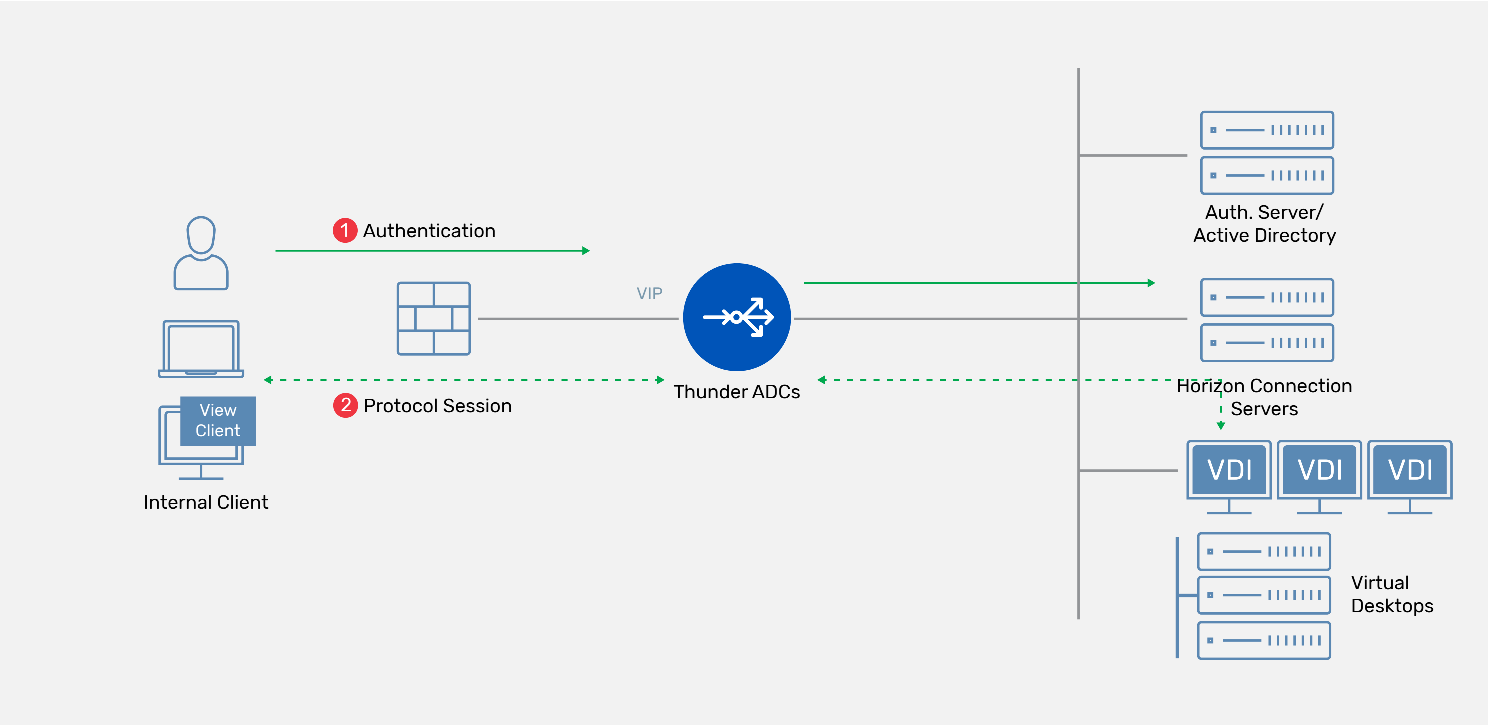


Figure 3: Secure tunneled connection

For deployments with the Thunder ADC as a reverse proxy and load balancer for Horizon Connection Servers, the traffic of secure tunneled connection is as follows:

1. The client device makes an HTTPS connection to the virtual server (VIP) address on the Thunder ADC for the first phase. The ADC then establishes a new connection to the Horizon Connection Servers after successful authentication.
2. Once authenticated and desktop availability is determined, the protocol session (regardless of Blast Extreme, RDP, PCoIP or HTTP HTML5 access) from Horizon Client or Web Browser will be sent to the virtual server (VIP) address on the Thunder ADC that provides full proxy to the assigned virtual desktop on Horizon Agents.

__Note__: DSR deployment is also supported in case Thunder ADC is deployed with one-arm mode and return traffic needs to be directly routed back to the client.

__Note__: For more details on the Horizon View 7 protocols and connection type, refer to https://techzone.vmware.com/resource/network-ports-vmware-horizon-7 and VMware Horizon 7.x Reference Architecture guides.

Deployment with Unified Access Gateways for Horizon View

The VMware Unified Access Gateway (formerly called Access Point) is a platform that provides secure edge services and access to defined resources that reside in the internal network. This allows authorized, external users to access internally located virtual desktop infrastructure resources in a secure manner. The Unified Access Gateway is usually deployed in the DMZ and provides authentication (1st phase connection) and protocol session forwarding (2nd phase). From ADC's deployment perspective, protocol usage and traffic flow are similar to the one with Secured Tunneled Connection.

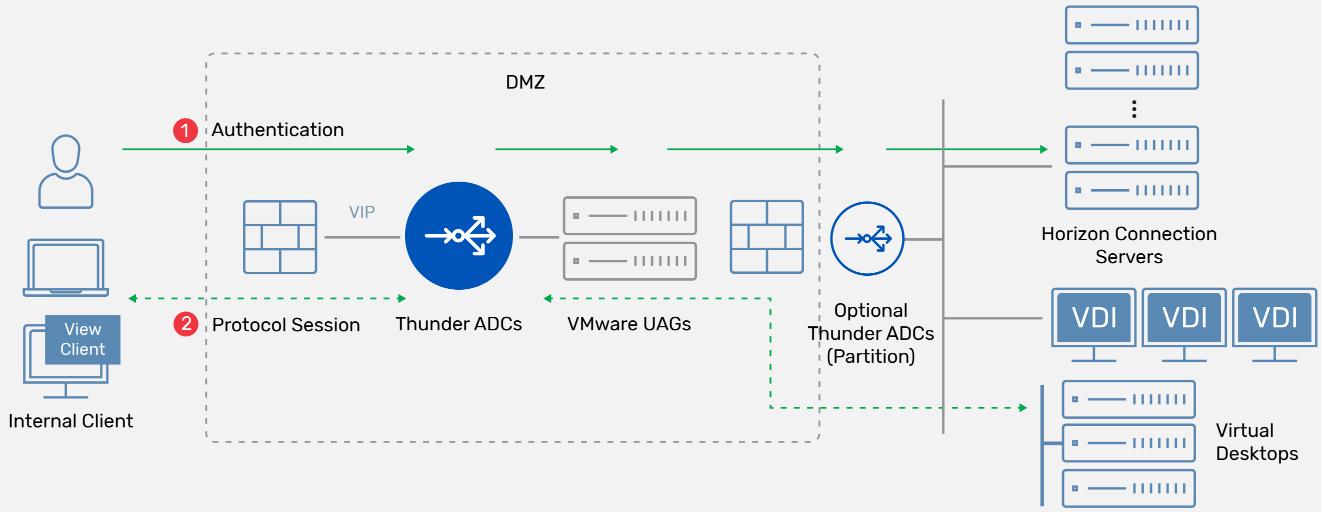


Figure 4: ADC deployment with UAG

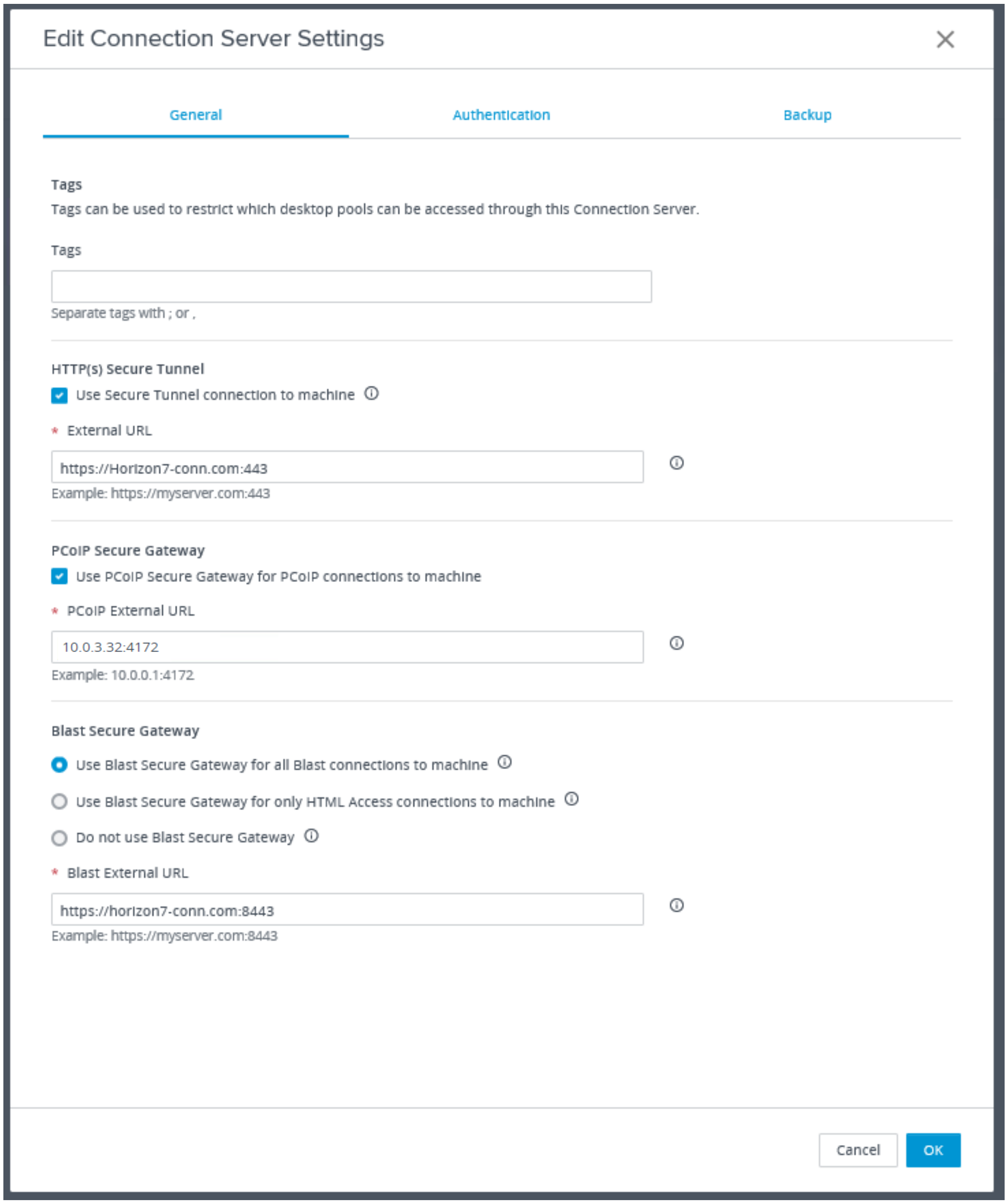
***Note**: For more details of Unified Access Gateway deployment, refer to https://docs.vmware.com/en/Unified-Access-Gateway/index.html.*

***Note**: This document mainly provides configuration based on the Horizon View deployment with Connection Servers. The deployment with UAG is out scope. However, most of the configuration can be referenced by replacing Connection Server with UAG.*

VMware Horizon View Administration Configuration

To direct Horizon View Client through Thunder VIP for various access protocols

1. Log on to VMware Horizon View Administrator.
2. Navigate to **View Configuration > Servers > View Connection Servers**.
3. Change the **External URL** for Secure Tunnel to the "Thunder ADC's VIP address" or "FQDN."
4. Edit Secure Gateway options as appropriate for your environment.
    a. Enable PCoIP Secure
    b. Blast Secure Gateway
5. Repeat the steps above for each Horizon View Connection Server.



Tags

Tags can be used to restrict which desktop pools can be accessed through this Connection Server.

Tags

Separate tags with ; or ,

HTTP(s) Secure Tunnel

☑ Use Secure Tunnel connection to machine ⓘ

* External URL

https://Horizon7-conn.com:443 ⓘ

Example: https://myserver.com:443

PCoIP Secure Gateway

☑ Use PCoIP Secure Gateway for PCoIP connections to machine

* PCoIP External URL

10.0.3.32:4172 ⓘ

Example: 10.0.0.1:4172

Blast Secure Gateway

● Use Blast Secure Gateway for all Blast connections to machine ⓘ

○ Use Blast Secure Gateway for only HTML Access connections to machine ⓘ

○ Do not use Blast Secure Gateway ⓘ

* Blast External URL

https://horizon7-conn.com:8443 ⓘ

Example: https://myserver.com:8443

Figure 5: Horizon View Connection Server settings

Thunder ADC Configuration for Horizon View Connection Servers

Service Port Mapping

The following table shows the protocols and port mappings for Horizon View deployment. As described in the previous Deployment Architecture section, Thunder ADC provides reverse proxy for authentication and any deployment and desktop connection in tunneled connection and external connection.

Port	Horizon View Protocol	VIP Type	Remarks
Port 443	HTTPS for Auth (1st phase) HTTPS Secure Tunnel (RDP) HTML via Web Browser	HTTPS (L7 SLB)	Choose either one. This guide uses HTTPS VIP for port 443.
		TCP (L4 SLB)	
Port 4172	PCoIP (Secure Tunnel)	TCP & UDP	
Port 8443	Blast Extreme (Secure Tunnel) HTML Blast (Secure Tunnel)	TCP	
Port 80	N/A	HTTP (no service associated)	To redirect HTTP request to HTTPS (port 443)

The configuration steps overview on the Thunder ADC is as follows.

1. Access Thunder ADC.
2. Configure Connection Servers as real (backend) servers.
3. Define application layer Health Monitor for Horizon View service.
4. Create ADC service templates including session persistence and source NAT rules.
5. Create Service Group.
6. Create Virtual Server (VIP).

Accessing Thunder ADC

To access Thunder ADC from a Command Line Interface (CLI) or Graphical User Interface (GUI), follow the steps below.

CLI – The CLI is a text-based interface in which you type commands on a command line. You can access the CLI directly through the (serial) console or over the network using either of the following protocols:

- Secure protocol – Secure Shell (SSH) version 2
- Unsecure protocol – Telnet (if enabled)

GUI – This is a web-based interface in which you click buttons, menus and other graphical icons to access the configuration or management pages. From these pages, you can type or select values to configure or manage the device. You can access the GUI using the following protocol:

- Secure protocol – Hypertext Transfer Protocol over Secure Socket Layer (HTTPS)

***Note**: HTTP requests are redirected to HTTPS by default on Thunder ADC.*

Default Access Information:

- Default username: “admin”
- Default password: “a10”
- Default IP address of the device: “172.31.31.31”

***Note**: For detailed information on how to access the Thunder ADC device, refer to the System Configuration and Administration Guide in the A10 Networks support portal.*

Create Servers for Horizon Connection Servers

To configure Horizon View Connection Servers as real servers, follow the steps below.

Via Web GUI

1. Navigate to A**DC > SLB > Servers** and click **Create**.
2. Enter Connection Server information.
   - Name: hv-conn-svr1
   - Host: 10.0.2.164
   - Health Monitor: ping (or blank= Default)
3. On the **Port** section, add service ports by clicking **Create**.
   - Port or Port Range: 443
   - Protocol: TCP
   - Repeat this to add additional ports and protocols to support various connection types.
     - 4172 TCP and UDP
     - 8443 TCP
4. **Clone** or repeat this step for other Horizon Connection Servers.

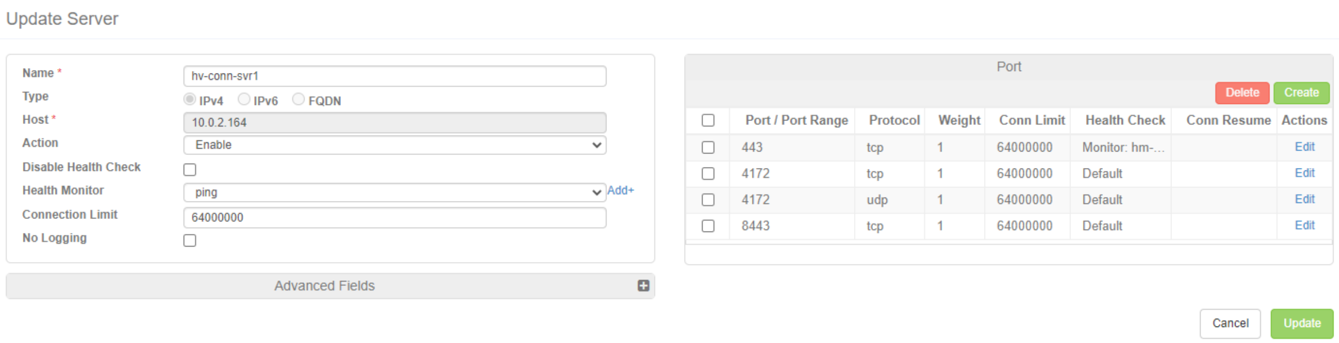
***Note**: Configure additional ports/protocols to support the various connection capabilities*

Via CLI

```
ADC(config)#slb server hv-conn-svr1 10.0.2.164
ADC(config-real server)#health-check ping
ADC(config-real server)#port 443 tcp
ADC(config-real server-node port)#port 4172 tcp
ADC(config-real server-node port)#port 4172 udp
ADC(config-real server-node port)#port 8443 tcp

ADC(config)#slb server hv-conn-svr2 10.0.2.165
ADC(config-real server)#health-check ping
ADC(config-real server)#port 443 tcp
ADC(config-real server-node port)#port 4172 tcp
ADC(config-real server-node port)#port 4172 udp
ADC(config-real server-node port)#port 8443 tcp
```

Repeat this step for other Horizon Connection Servers.



Update Server

Name *	hv-conn-svr1
Type	● IPv4 ○ IPv6 ○ FQDN
Host *	10.0.2.164
Action	Enable
Disable Health Check	□
Health Monitor	ping
Connection Limit	64000000
No Logging	□

□	Port / Port Range	Protocol	Weight	Conn Limit	Health Check	Conn Resume	Actions
□	443	tcp	1	64000000	Monitor: hm-...		Edit
□	4172	tcp	1	64000000	Default		Edit
□	4172	udp	1	64000000	Default		Edit
□	8443	tcp	1	64000000	Default		Edit

Figure 6: Adding Connection Servers as real server

***Note**: By default, ping is used for server health check. Please make sure the firewall setting of the Horizon Connection Server authorizes ping (ICMP echo request) from the Thunder ADC. Otherwise, use the other health monitor method based on the deployment environment.*

***Note**: If Thunder ADC is deployed with SSL Offload for VMware Horizon View, port 80 TCP needs to be added to the port instead of port 443 TCP. Refer to the KB for SSL Offload configuration on Connection Server configuration.*

Create Application Health Monitor for Horizon View Connection Service

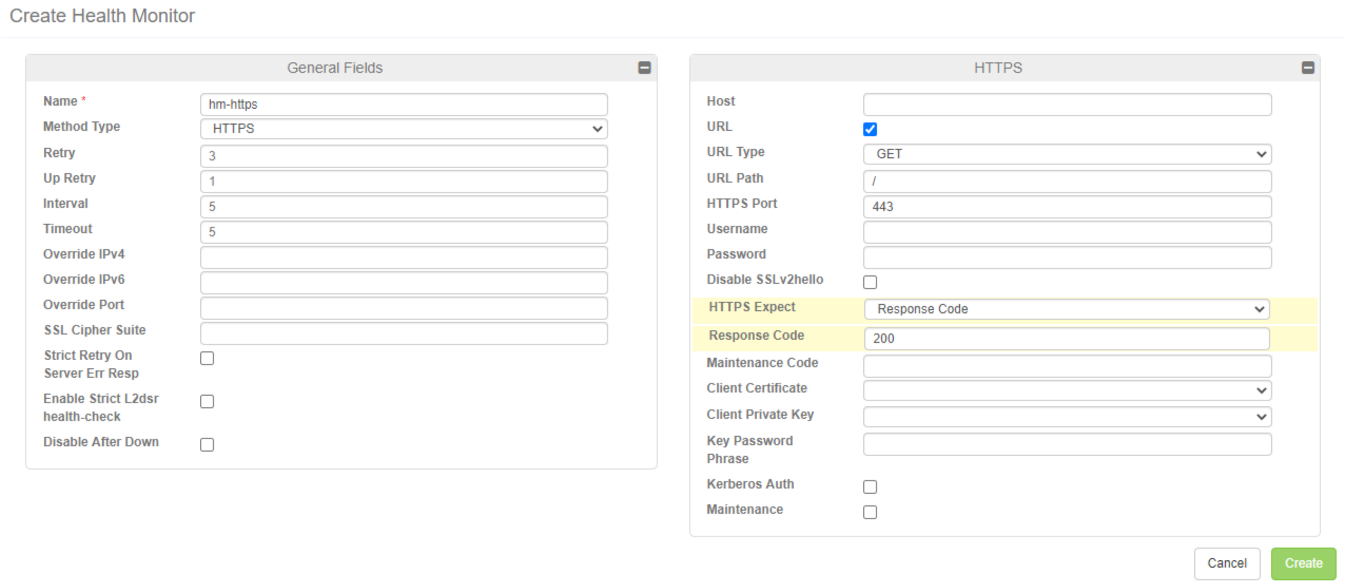
To create an application-level health monitor template for the service availability check of the Horizon View server and service, follow the steps below. This example uses an HTTPS-based health monitor.

Via Web GUI

1. Navigate **ADC > Health Monitors** and click **Create**.
1. Enter the health monitor template information.
   - Name: hm-https
   - Method Type: HTTPS
   - Check URL
   - Specify URL Type and Path: “GET /”
   - Specify HTTP Expected Response: “Response Code 200”

Via CLI

```
ADC(config)#health monitor hm-https
ADC(config-health:monitor)# method https port 443
expect response-code 200 url GET /
```



Create Health Monitor

General Fields

Name *	hm-https
Method Type	HTTPS
Retry	3
Up Retry	1
Interval	5
Timeout	5
Override IPv4	
Override IPv6	
Override Port	
SSL Cipher Suite	
Strict Retry On Server Err Resp	□
Enable Strict L2dsr health-check	□
Disable After Down	□

HTTPS

Host	
URL	☑
URL Type	GET
URL Path	/
HTTPS Port	443
Username	
Password	
Disable SSLv2hello	□
HTTPS Expect	Response Code
Response Code	200
Maintenance Code	
Client Certificate	
Client Private Key	
Key Password Phrase	
Kerberos Auth	□
Maintenance	□

Create ADC Optimization Feature Templates

This section describes the configuration of ADC optimization features for Horizon View deployment. This guide includes the following optimization features with the recommended options.

- General ADC Optimization Features
  - **Client persistence:** It is crucial for ADC to persist the client connection request to the same server. There are a couple of options for maintaining client session persistency with the Connection Server and the assigned VDI. This guide uses source IP persistent, which uses the source address to direct all subsequent requests from a given client to the same server.
  - **Source NAT:** Source NAT (SNAT) is required when your network topology is based on “one-arm” deployment and/or when you want to ensure that response traffic from the servers is directed to the ADC.

    Note: *When source NAT is enabled on the ADC, the request received by the real server does not have the client’s IP address. If required, use the “client-ip insertion” option with the HTTP template so that ADC inserts the client’s IP address into the HTTP header.*

    Note: *If you want to deploy using DSR (direct server return), skip this configuration.*

  - **HTTP to HTTPS redirection:** ADC will securely redirect a client connection request to HTTPS (port 443) URL in case the original request uses HTTP (port 80).
- Application Layer (L7) Specific Optimization Features

Note*: Ignore these configurations if port 443 service is configured as TCP port VIP instead of HTTPS.*

  - **TCP connection reuse**: For 1st phase connection or HTTPS/HTML-based connection on port 443.
  - **RAM cache**: ADC caches HTTP contents (RFC 2616 compliant) from the server response and uses the cached data to respond back to the client request. This can improve response time and reduce server load and utilization associated with subsequent transactions.
  - **HTTP compression**: Compression reduces the amount of bandwidth required to send content to clients. The content types (e.g., pdf and ppt) that should be compressed can be specified while enabling the option.

Via Web GUI	Via CLI

Source IP Persistence Template
1. Navigate **ADC > Templates > Persistence**, click **Create** and select **Persist Source IP**.
2. Enter the source IP based persistence information
a. Name: src_ip_persist
b. Match Type: Server
Note*: In the case when VDI clients are behind the same NAT/ proxy and share the same IP address (inline NAT device) that hides their IP addresses, source IP persistence may not be effective from a load balancing point of view.*
Note*: Refer to Appendix B if you need to use a cookie (JSESSIONID) assigned by Horizon Connection Server for session persistence.*
	`ADC(config)#slb template persist source-ip src_ip_persist`
	`ADC(config-slb)# match-type server`

Source NAT Template 1. Navigate ADC > IP Source NAT > IPv4 Pools and click Create. 2. Enter the IPv4 Pools information. a. Name: hv-nat-pool b. Start Address: 10.0.2.12 c. End Address: 10.0.2.12 d. Netmask” /32	`ADC(config)# ip nat pool hv-nat-pool 10.0.2.12 10.0.2.12 netmask /32`

HTTPS Redirection
1. Navigate **ADC > Templates > L7 Protocols**, click **Create** and select **HTTP**.
2. Enter the HTTP Template information
a. Name: hv-http-redirect
b. Under Redirect Section
• Redirect: Port
• Use HTTPS: checked
• Port: 443
	`ADC(config)# slb template http hv-http-redirect`
	`ADC(config-http)# redirect secure port 443`

TCP Connection Reuse

1. Navigate **ADC > Templates > Application**, click **Create** and select **Connection Re-use**.
2. Enter the TCP Connection Reuse Template information.
   a. Name: hv-conn-reuse

```
ADC(config)#slb template connection-reuse hv-conn-reuse
```

RAM Cache

1. Navigate **ADC > Templates > Application**, click **Create** and select **RAM Caching**.
2. Enter the RAC Caching Template information.
   a. Name: hv-caching

```
ADC(config)#slb template cache hv-cache
```

HTTP Compression

1. Navigate **ADC > Templates > L7 Protocols**, click **Create** and select **HTTP**.
2. Enter the HTTP Template information.
   a. Name: hv-compression
   b. Check Enable under the Compression section.

```
ADC(config)#slb template http hv-compression
ADC(config-http)#compression enable
```

Create SSL Templates

It is then assumed that the TLS/SSL connection request over HTTPS from clients are terminated on the Thunder ADC using HTTPS VIP. Therefore, a proper TLS/SSL certificate for Horizon View service should be imported in advance and associated with the client-SSL template.

***Note**: For testing purposes, you can create and use a self-signed TLS/SSL certificate on the Thunder ADC.*

It is common practice to use TLS/SSL communication again between Thunder ADC and Connection servers unless you have a specific reason to use SSL offload deployment where you use HTTPS on the front end (i.e., between client and ADC) and HTTP (clear text) on the back end (i.e., between ADC and servers). In TLS re-encryption deployment use case, you can use stronger TLS ciphers (e.g., ECC, PFS or longer key length) on the front end while lighter ciphers such as RSA can be used on the back end. This helps the VMware server reduce CPU load while providing end-to-end encrypted communication.

On Thunder ADC, the client-ssl template is used for front-end TLS/SSL configuration, and the server-ssl template is used for the back end.

***Note**: If you are deploying Thunder ADC with Layer 4 load balancing for port 443 (HTTPS/HTML) connection instead of HTTPS, ignore this section.*

Client SSL Template

To configure the client-ssl template for TLS/SSL communication specification, follow the steps below.

Via Web GUI

Create SSL Cipher Template

1. Navigate **ADC > Templates > SSL**, click **Create** and select **SSL Cipher**.
2. Enter the SSL Cipher information.
   a. Name: hv-client-ciphers
   b. Add ciphers you want to use.

Note*: Stronger ciphers are preferred for front-end communication. Refer to CLI command configuration for a sample cipher suites list.*

Via CLI

```
ADC(config)#slb template cipher hv-client-ciphers
ADC(config-cipher)#TLS1_DHE_RSA_AES_256_GCM_SHA384
ADC(config-cipher)#TLS1_DHE_RSA_AES_128_GCM_SHA256
ADC(config-cipher)#TLS1_ECDHE_RSA_AES_128_SHA priority 10
ADC(config-cipher)#TLS1_ECDHE_RSA_AES_256_SHA priority 10
ADC(config-cipher)#TLS1_ECDHE_RSA_AES_128_SHA256 priority 10
ADC(config-cipher)#TLS1_ECDHE_RSA_AES_128_GCM_SHA256 priority 10
ADC(config-cipher)#TLS1_ECDHE_ECDSA_AES_256_SHA priority 10
ADC(config-cipher)#TLS1_ECDHE_ECDSA_AES_128_SHA256 priority 10
ADC(config-cipher)#TLS1_ECDHE_ECDSA_AES_128_GCM_SHA256 priority 10
ADC(config-cipher)#TLS1_ECDHE_ECDSA_AES_128_SHA priority 10
```

Via Web GUI

Create Client-SSL Template

1. Navigate **ADC > Templates > SSL**, click Create and select **Client SSL**.
2. Enter the client SSL specification.
   a. Name: hv-ssl-client
   b. Server Certificate: *<<your certificate>>*
   c. Server Private Key: *<<your key>>*
   d. Version: TLSv1.2
   e. Downloadable Version: TLS 1.1
   f. Reject Client Requests for SSLv3
   g. Enable TLS Alert Logging

Via CLI

```
ADC(config)#slb template client-ssl hv-ssl-client
ADC(config-client ssl)#cert <<your certificate>>
ADC(config-client ssl)#key <<your key>>
ADC(config-client ssl)#template cipher hv-client-ciphers
ADC(config-client ssl)#ssl-false-start-disable
ADC(config-client ssl)#disable-sslv3
ADC(config-client ssl)#version 33 32
```

Server SSL Template

To configure the client-ssl template for TLS/SSL communication specification, follow the steps below.

Via Web GUI

Create Server-SSL Template

1. Navigate **ADC > Templates > SSL**, click **Create** and select Server **SSL**.
2. Enter the server side SSL information
   a. Name: hv-ssl-server

Via CLI

```
ADC(config)#slb template server-ssl hv-ssl-server
```

Create Service Groups for Horizon Connection Service

To create a service group for the Horizon View Connection Servers, follow the steps below. Create additional protocols/ports to support the various connection service types depending on your deployment architecture. For example, RDP, PCoIP, and Extreme Blast.

Via Web GUI

1. Navigate **ADC > SLB > Service Group**, and click **Create**.
2. Enter the Service Group information for port 443.
   a. Name: sg-hv-443
   b. Protocol: TCP
   c. Algorithm: Least Connection
   d. Health Monitor: hm-https
3. In the **Member** section, **Create** and add each View Connection Server with TCP 443.
   a. Creation Type: Existing Server
   b. Server: hv-conn-svr1
   c. Port: 443
   d. Add other Horizon Connection Servers (e.g. hv-conn-svr2)
4. Repeat this step for other service ports
   a. Blast Extreme 8443 TCP
   b. PCoIP 4172 TCP
   c. PCoIP 4172 UDP

Via CLI

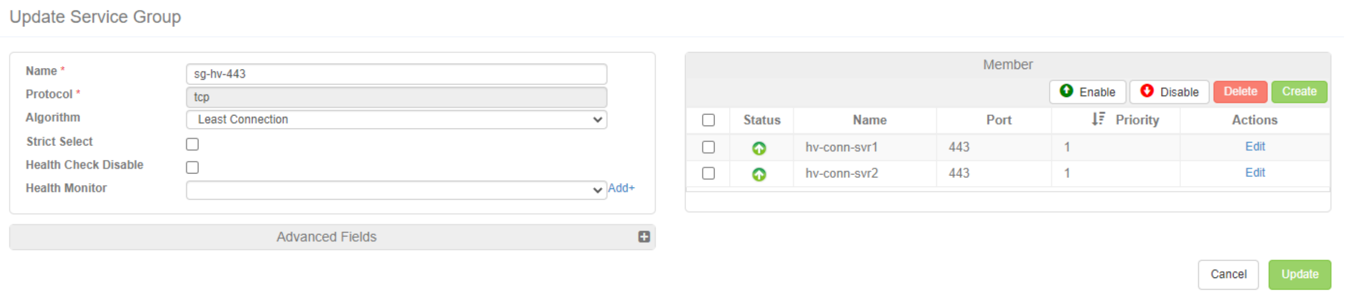
```
ADC(config)#slb service-group sg-hv-443_https tcp
ADC(config-slb svc group)#method least-connection
ADC(config-slb svc group)# health-check hm-https
ADC(config-slb svc group)#member hv-conn-svr1 443
ADC(config-slb svc group)#member hv-conn-svr2 443

ADC(config)#slb service-group sg-hv-4172_pcoip
tcp
ADC(config-slb svc group)#method least-connection
ADC(config-slb svc group)#member hv-conn-svr1
4172
ADC(config-slb svc group)#member hv-conn-svr2
4172

ADC(config)#slb service-group sg-hv-8443_blast
tcp
ADC(config-slb svc group)#method least-connection
ADC(config-slb svc group)#member hv-conn-svr1
8443
ADC(config-slb svc group)#member hv-conn-svr2
8443

ADC(config)#slb service-group sg-hv-4172_pcoip-
udp udp
ADC(config-slb svc group)#method least-connection
ADC(config-slb svc group)#member hv-conn-svr1
4172
ADC(config-slb svc group)#member hv-conn-svr2
4172
```

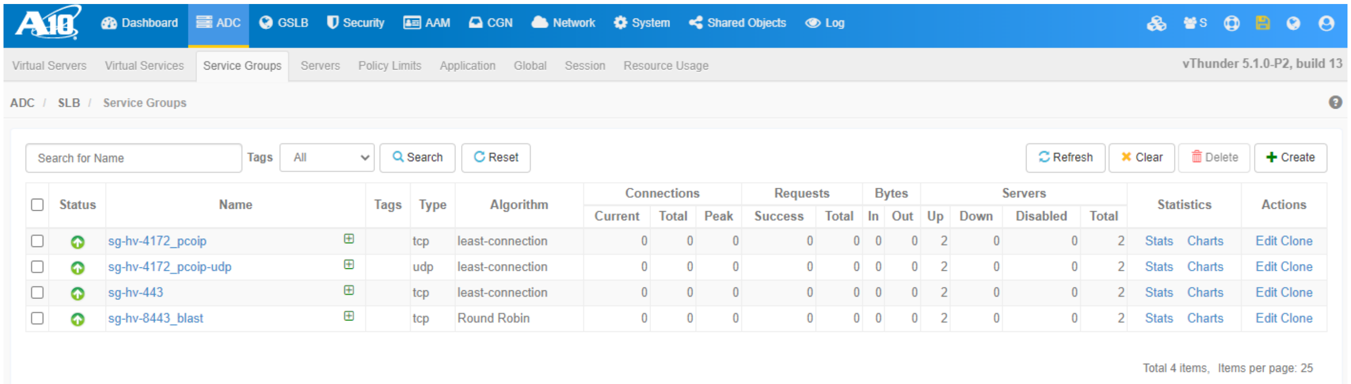
***Note**: If Thunder ADC is deployed with SSL Offload for VMware Horizon View, create service group for port 80 TCP. Refer to the KB for SSL Offload configuration on Connection Server configuration.*



Name *	sg-hv-443
Protocol *	tcp
Algorithm	Least Connection
Strict Select	☐
Health Check Disable	☐
Health Monitor	Add+

☐	Status	Name	Port	Priority	Actions
☐	●	hv-conn-svr1	443	1	Edit
☐	●	hv-conn-svr2	443	1	Edit

Figure 7: Creating Service Group for a service port 443



☐	Status	Name	Tags	Type	Algorithm	Connections			Requests		Bytes		Servers				Statistics	Actions
						Current	Total	Peak	Success	Total	In	Out	Up	Down	Disabled	Total		
☐	●	sg-hv-4172_pcoip		tcp	least-connection	0	0	0	0	0	0	0	2	0	0	2	Stats Charts	Edit Clone
☐	●	sg-hv-4172_pcoip-udp		udp	least-connection	0	0	0	0	0	0	0	2	0	0	2	Stats Charts	Edit Clone
☐	●	sg-hv-443		tcp	least-connection	0	0	0	0	0	0	0	2	0	0	2	Stats Charts	Edit Clone
☐	●	sg-hv-8443_blast		tcp	Round Robin	0	0	0	0	0	0	0	2	0	0	2	Stats Charts	Edit Clone

Total 4 items, Items per page: 25

Figure 8: Service Groups for Horizon View services

Create Virtual Server (VIP) for Horizon View Service

To create the virtual server (or VIP), which is the proxied IP address that end-users will use to access Horizon View, follow the steps below.

Via Web GUI

1. Navigate **ADC > SLB > Virtual Servers**, click **Create**.
2. Enter the Virtual Server information.
   - Name: vip-hv
   - IP Address: 10.0.3.32
   - Netmask: /32
3. In the **Virtual Port** section, click **Create** to add service port 443 HTTPS for Horizon View service.
   - Name: vmware_view_vport_443
   - Protocol: HTTPS
   - Port or Port Range: 443
   - Service Group: sg-hv-443
   - Source NAT Pool: hv-nat-pool
   - Persistent Type: source-ip with src_ip_persist template
   - HTTP template: hv-compression
   - RAM cache template: hv-cache
   - SSL template: [Client] hv-ssl-client, [Server] hv-ssl-server

4. Add service port 80 HTTP
   - Name: vmware_view_vport_80_redirect
   - Protocol: HTTP
   - Port or Port Range: 80
   - Service Group: sg-hv-443
   - HTTP Template: hv-http-redirect

   ***Note**: The "Redirect to HTTPS" feature is available on HTTP port under "Advanced Fields." This can be used instead of HTTP template.*

5. Add following service ports with associating Source NAT profile and the persist template.
   - Virtual ports

   a. Blast Extreme: 8443 TCP

   b. PCoIP: 4172 TCP

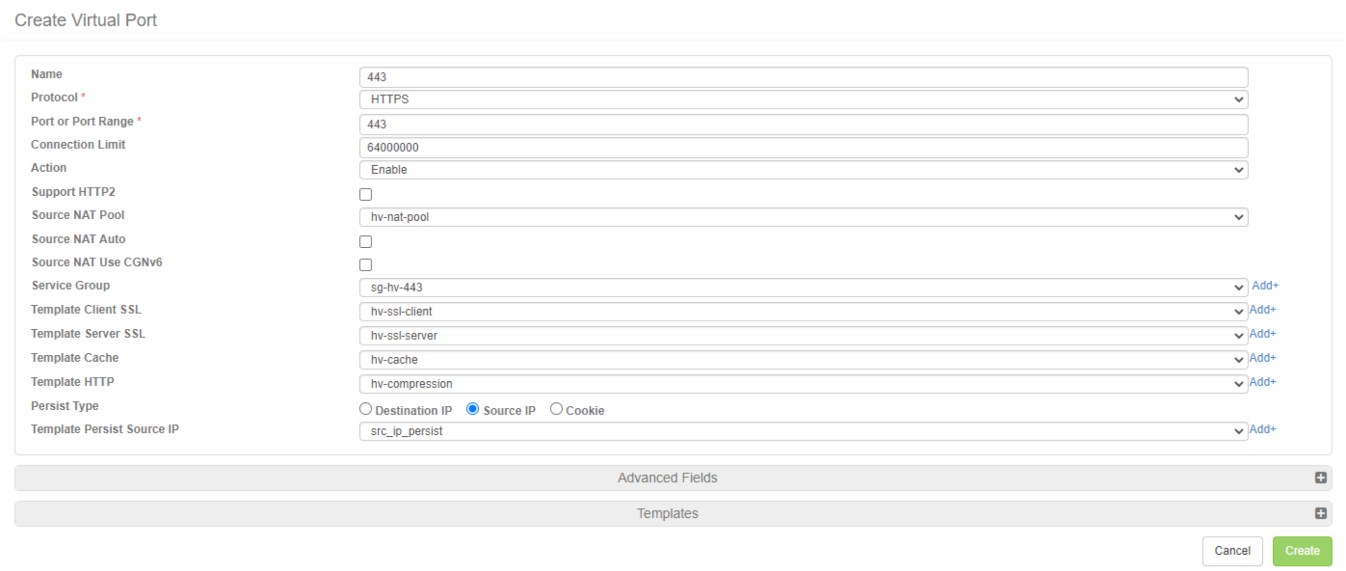
   c. PCoIP: 4172 UDP

   - Source NAT Pool: hv-nat-pool
   - Persistent Template: source-ip with src_ip_persist template

   ***Note**: If your deployment is with DSR (direct server return), make sure to exclude Source NAT configuration from each port.*

Via CLI

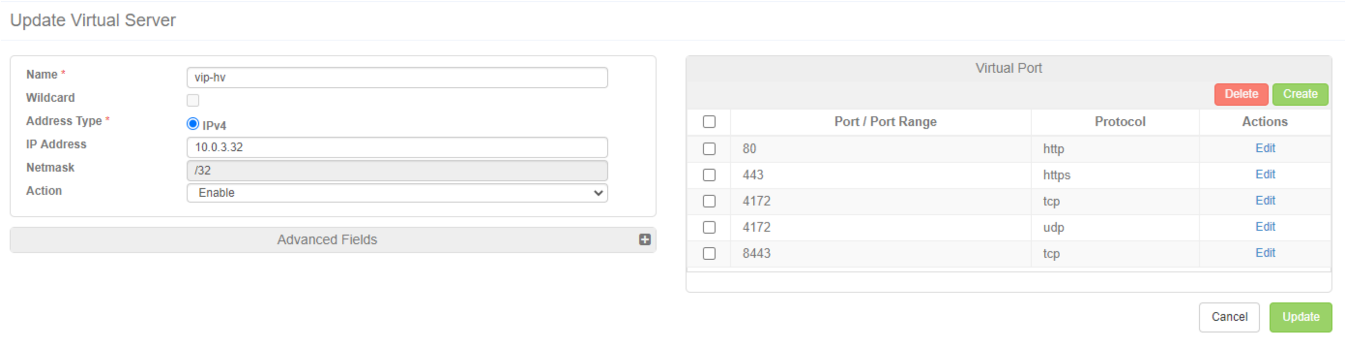
```
ADC(config)#slb virtual-server vip-hv 10.0.3.32 /32
ADC(config-slb vserver)#port 80 http
ADC(config-slb vserver-vport)#name vmware_view_vport_80_redirect
ADC(config-slb vserver-vport)#template http hv-http-redirect
ADC(config-slb vserver-vport)#port 443 https
ADC(config-slb vserver-vport)#name vmware_view_vport_443
ADC(config-slb vserver-vport)#source-nat pool hv-nat-pool
ADC(config-slb vserver-vport)#service-group sg-hv-443_https
ADC(config-slb vserver-vport)#template connection-reuse hv-conn-reuse
ADC(config-slb vserver-vport)#template persist source-ip src_ip_persist
ADC(config-slb vserver-vport)#template http hv-compression
ADC(config-slb vserver-vport)#template server-ssl hv-ssl-server
ADC(config-slb vserver-vport)#template client-ssl hv-ssl-client
ADC(config-slb vserver-vport)#port 3389 tcp
ADC(config-slb vserver-vport)#source-nat pool hv-nat-pool
ADC(config-slb vserver-vport)#service-group sg-hv-3389_rdp
ADC(config-slb vserver-vport)#template persist source-ip src_ip_persist
ADC(config-slb vserver-vport)#port 4172 tcp
ADC(config-slb vserver-vport)#name vmware_view_vport_4172
ADC(config-slb vserver-vport)#source-nat pool hv-nat-pool
ADC(config-slb vserver-vport)#service-group sg-hv-4172_pcoip
ADC(config-slb vserver-vport)#template persist source-ip src_ip_persist
ADC(config-slb vserver-vport)#port 4172 udp
ADC(config-slb vserver-vport)#name vmware_view_vport_4172_udp
ADC(config-slb vserver-vport)#source-nat pool hv-nat-pool
ADC(config-slb vserver-vport)#service-group sg-hv-4172_pcoip-udp
ADC(config-slb vserver-vport)#template persist source-ip src_ip_persist
ADC(config-slb vserver-vport)#port 8443 tcp
ADC(config-slb vserver-vport)#name vmware_view_vport_8443
ADC(config-slb vserver-vport)#source-nat pool hv-nat-pool
ADC(config-slb vserver-vport)#service-group sg-hv-8443_blast
ADC(config-slb vserver-vport)#template persist source-ip src_ip_persist
```



Create Virtual Port

Name	443
Protocol *	HTTPS
Port or Port Range *	443
Connection Limit	64000000
Action	Enable
Support HTTP2	□
Source NAT Pool	hv-nat-pool
Source NAT Auto	□
Source NAT Use CGNv6	□
Service Group	sg-hv-443
Template Client SSL	hv-ssl-client
Template Server SSL	hv-ssl-server
Template Cache	hv-cache
Template HTTP	hv-compression
Persist Type	○ Destination IP ● Source IP ○ Cookie
Template Persist Source IP	src_ip_persist

Figure 9: HTTPS (port 443) virtual port configuration



Update Virtual Server

Name *	vip-hv
Wildcard	□
Address Type *	● IPv4
IP Address	10.0.3.32
Netmask	/32
Action	Enable

Virtual Port

□	Port / Port Range	Protocol	Actions
□	80	http	Edit
□	443	https	Edit
□	4172	tcp	Edit
□	4172	udp	Edit
□	8443	tcp	Edit

Figure 10: Virtual Server configuration

(Optional) Enable Integrated DDoS Protection

Thunder ADC provides integrated DDoS protection features and can be configured to defend against common DDoS attacks. To configure integrated DDoS protection, follow the steps below.

Via Web GUI

1. Navigate **Security > DDoS**.
2. Enter DDoS Protection configuration.
   a. IP Anomaly Drop: Drop All
   b. Bad Content: 24
   c. Out of Sequence: 24
   d. Zero Window: 24
   e. ICMP Rate Limiting Rate: 2000

Via CLI

```
ADC(config)#ip anomaly-drop drop-all
ADC(config)#ip anomaly-drop bad-content 24
ADC(config)#ip anomaly-drop out-of-sequence 24
ADC(config)#ip anomaly-drop zero-window 24
ADC(config)#icmp-rate-limit 2000
```

***Note**: For more detailed information on how to configure DDoS protection and other security features such as Web Application Firewall (WAF) and SSO/MFA as part of Application Access Management (AAM), refer to the ACOS Configuration Guide.*

Deployment Verification

To verify that the VMware Horizon View deployment is working fine, check the status of virtual server (VIP) and service ports.

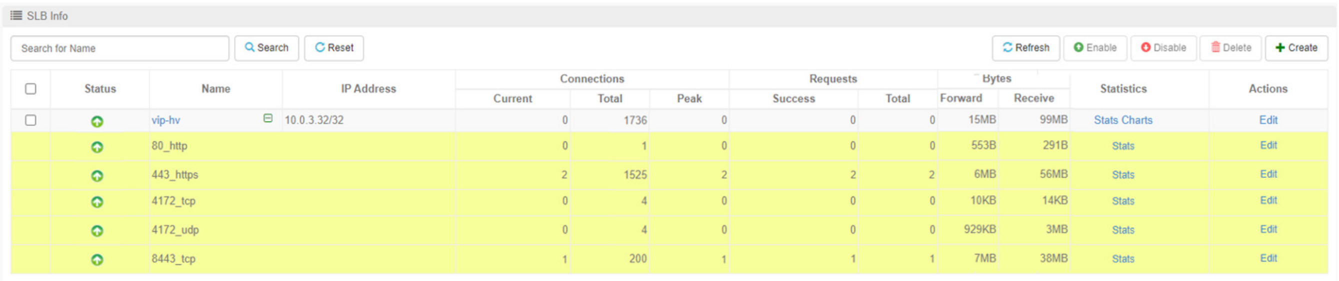
Verify the Status of VIP and ADC Services

Via Web GUI

1. Navigate to **Dashboard > ADC > SLB Info**, and expand the virtual service **vip-hv** for Horizon View.
2. Make sure that service status for all services ports are UP. ⬆

Via CLI

```
ADC#show slb virtual-server
ADC#show slb virtual-server bind
ADC#show slb virtual-server vip-hv
ADC#show slb server hv-conn-svr1
ADC#show slb server hv-conn-svr2
```

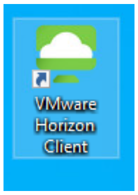


□	Status	Name	IP Address	Connections			Requests		Bytes		Statistics	Actions
				Current	Total	Peak	Success	Total	Forward	Receive		
□	⬆	vip-hv ⊟	10.0.3.32/32	0	1736	0	0	0	15MB	99MB	Stats Charts	Edit
	⬆	80_http		0	1	0	0	0	553B	291B	Stats	Edit
	⬆	443_https		2	1525	2	2	2	6MB	56MB	Stats	Edit
	⬆	4172_tcp		0	4	0	0	0	10KB	14KB	Stats	Edit
	⬆	4172_udp		0	4	0	0	0	929KB	3MB	Stats	Edit
	⬆	8443_tcp		1	200	1	1	1	7MB	38MB	Stats	Edit

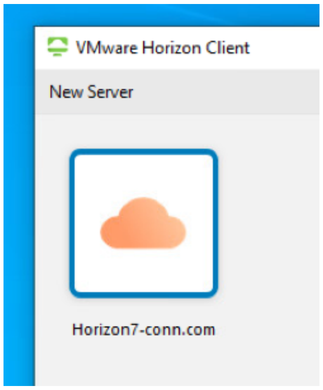
Figure 11: SLB VIP status and statistics dashboard

Validate Access to the VMware Horizon View and VDI

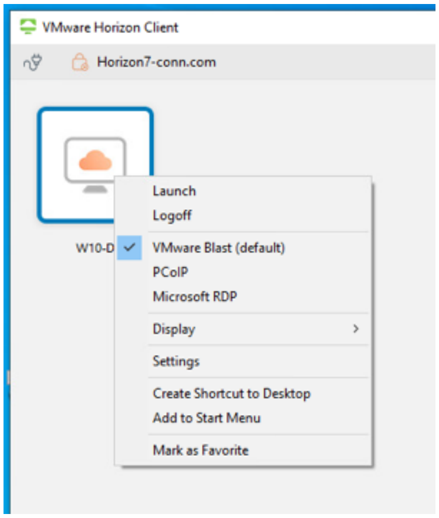
1. Launch VMware Horizon View Client, and login to authenticate.



2. Once successful, the desktop pool(s) will be available to launch depending on access authorization.



3. Right click to select a connection protocol (default: Blast). Once launched, you will be connected to one of the virtual desktops in the pool.



Summary

This document describes how to configure Thunder ADC as a load balancer to support VMware Horizon View 7.x deployment. The A10 Thunder ADC, powered by Advanced Core Operating System (ACOS®), supports easy and flexible deployment options, helps IT operators enable reliable virtual desktop services, strengthens high availability, and maximizes elasticity and performance for Horizon View deployment. Deploying Horizon View 7.x with Thunder ADC provides the following benefits:

- Load balancing and high availability of VMware Connection Servers
- Layer 4 and Layer 7 full proxy service with optimization features
- Support for various deployments such as secure tunnel connections with Unified Access Gateway (UAG)
- Relaxing CPU-intensive tasks from SSL/TLS transactions using lighter cipher on the VMware Connection Servers while ensuring highly secure communication using stronger cipher on the Thunder ADC

For more information about Thunder ADC, please refer to:

- https://www.a10networks.com/products/thunder-adc/
- https://documentation.a10networks.com/ACOS/510x/ACOS_5_1_0/index.html

For more information about VMware Horizon View, please refer to: VMware Horizon 7.x Reference Architecture Guides and VMware Horizon View Protocols and Ports.

Appendix A – Thunder ADC Configuration

Enter CLI configs here using the format below.

```
!
hostname ADC
!
ip anomaly-drop bad-content 24
ip anomaly-drop drop-all
ip anomaly-drop out-of-sequence 24
ip anomaly-drop zero-window 24
!
icmp-rate-limit 2000
!
interface management
  ip address 172.21.50.12 255.255.255.0
  ip control-apps-use-mgmt-port
  ip default-gateway 172.21.50.1
!
interface ethernet 1
  name uplink_dmz
  enable
  ip address 10.0.3.30 255.255.255.0
!
interface ethernet 2
  name server_firm
  enable
  ip address 10.0.2.1 255.255.255.0
!
ip nat pool hv-nat-pool 10.0.2.12 10.0.2.12
netmask /32
!
ip route 0.0.0.0 /0 10.0.3.1
!
slb common
  enable-l7-req-acct
!
health monitor hm-https
  method https port 443 expect response-code 200
url GET /
!
slb template cipher hv-client-ciphers
  TLS1_DHE_RSA_AES_256_GCM_SHA384
  TLS1_DHE_RSA_AES_128_GCM_SHA256
  TLS1_ECDHE_RSA_AES_128_SHA priority 10
  TLS1_ECDHE_RSA_AES_256_SHA priority 10
  TLS1_ECDHE_RSA_AES_128_SHA256 priority 10
  TLS1_ECDHE_RSA_AES_128_GCM_SHA256 priority 10
  TLS1_ECDHE_ECDSA_AES_256_SHA priority 10
  TLS1_ECDHE_ECDSA_AES_128_SHA256 priority 10
  TLS1_ECDHE_ECDSA_AES_128_GCM_SHA256 priority 10
  TLS1_ECDHE_ECDSA_AES_128_SHA priority 10
!
slb template server-ssl hv-ssl-server
!
slb server hv-conn-svr1 10.0.2.164
  health-check ping
  port 443 tcp
    health-check hm-https
  port 4172 tcp
  port 4172 udp
  port 8443 tcp
!
slb server hv-conn-svr2 10.0.2.165
  health-check ping
  port 443 tcp
    health-check hm-https
  port 4172 tcp
  port 4172 udp
  port 8443 tcp
!
slb service-group sg-hv-443_https tcp
  method least-connection
  health-check hm-https
  member hv-conn-svr1 443
  member hv-conn-svr2 443
!
slb service-group sg-hv-4172_pcoip tcp
  method least-connection
  member hv-conn-svr1 4172
  member hv-conn-svr2 4172
!
slb service-group sg-hv-8443_blast tcp
  method least-connection
  member hv-conn-svr1 8443
  member hv-conn-svr2 8443
!
slb service-group sg-hv-4172_pcoip-udp udp
  method least-connection
  member hv-conn-svr1 4172
  member hv-conn-svr2 4172
!
slb template client-ssl hv-ssl-client
  cert <<your certificate>>
  enable-tls-alert-logging fatal
  key <<your key>>
  template cipher hv-client-ciphers
  disable-sslv3
  version 33 31
!
slb template connection-reuse hv-conn-reuse
!
slb template persist source-ip src_ip_persist
  match-type server
!
slb template http hv-http-redirect
  redirect secure
!
slb template http hv-compression
  compression enable
!
slb template cache hv-cache
!
slb virtual-server vip-hv 10.0.3.32 /32
  port 80 http
    name vmware_view_vport_80_redirect
    service-group sg-hv-443
    template http hv-http-redirect
    redirect-to-https
  port 443 https
    name vmware_view_vport_443
    source-nat pool hv-nat-pool
    service-group sg-hv-443_https
    template connection-reuse hv-conn-reuse
    template persist source-ip src_ip_persist
    template http hv-compression
    template server-ssl hv-ssl-server
    template client-ssl hv-ssl-client
  port 4172 tcp
    name vmware_view_vport_4172
    source-nat pool hv-nat-pool
    service-group sg-hv-4172_pcoip
    template persist source-ip src_ip_persist
  port 4172 udp
    name vmware_view_vport_4172_udp
    source-nat pool hv-nat-pool
    service-group sg-hv-4172_pcoip-udp
    template persist source-ip src_ip_persist
  port 8443 tcp
    name vmware_view_vport_8443
    source-nat pool hv-nat-pool
    service-group sg-hv-8443_blast
    template persist source-ip src_ip_persist
!
sflow setting local-collection
!
sflow collector ip 127.0.0.1 6343
!
!
end
```

Appendix B – aFleX Policy for JSESSIONID-Based Persistence

If you have Layer 7 service port/VIP (e.g., port 443 HTTPS) and want to use JSESSIONID cookies for client persistence, create an aFleX policy and apply it to the virtual server.

aFleX policy Name: hv-persist

Via Web GUI	Via CLI

1. Navigate to ADC > aFleX and click Create.
2. Name: hv-persist
3. Definition: Copy below.
4. Apply the aFleX “hv-persist” to the port 443 HTTP of the VIP.
	`Thunder (config)#aflex create hv-persit`
	`<< Copy below aFleX policy>>`
	Note*: With CLI configuration, type “.” on a line by itself when done.*
	`ADC(config)#slb virtual-server vip-hv 10.0.3.32`
	`ADC(config-slb vserver)# port 443  https`
	`ADC(config-slb vserver-vport)# aflex hv-persist`

aFleX Policy: hv-persist

```
when HTTP_REQUEST {
 # Check if JSESSIONID exists
 if { [HTTP::cookie exists “JSESSIONID”] } {
 # JSESSIONID found in the request
 # we capture the first 32 characters
 set jsess_id [string range [HTTP::cookie “JSESSIONID”] 0 31]
 persist uie $jsess_id
 # Check if JSESSIONID exists in the uie persist table
 set p [persist lookup uie $jsess_id all]
 if { $p ne “” } {
 # JSESSIONID found in the persist table
 #log “JSESSIONID = \”$jsess_id\” found in persistency-table ([lindex $p 0] [lindex $p 1])”
 } else {
 # unknown JSESSIONID
 # (could be a fake JSESSIONID inserted by a bad end-user
 # or a user inactive for 30 minutes)
 #log “JSESSIONID = \”$jsess_id\” not found in persistency-table”
 }
 } else {
 # JSESSIONID not found in the request
 # (could be a new client)
 #log “No JSESSIONID cookie”
 }
}
when HTTP_RESPONSE {
 if { [HTTP::cookie exists “JSESSIONID”] } {
 set jsess_cookie [HTTP::cookie “JSESSIONID”]
 persist add uie $jsess_cookie 1800
 #log “Add persist entry for JSESSIONID \”$jsess_cookie\””
 }
}
```

About A10 Networks

A10 Networks (NYSE: ATEN) provides secure application services for on-premises, multi-cloud and edge-cloud environments at hyperscale. Our mission is to enable service providers and enterprises to deliver business-critical applications that are secure, available and efficient for multi-cloud transformation and 5G readiness. We deliver better business outcomes that support investment protection, new business models and help future-proof infrastructures, empowering our customers to provide the most secure and available digital experience. Founded in 2004, A10 Networks is based in San Jose, Calif. and serves customers globally.

For more information, visit: a10networks.com or tweet @a10Networks

Learn More

About A10 Networks

Contact Us

a10networks.com/contact

Part Number: A10-DG-16175-EN-01 JULY 2020