Chapter 4
Deployment Mode

Training Team

HCSA-ADC Official Training

Integrative Cybersecurity

Visionary. AI-powered. Accessible.

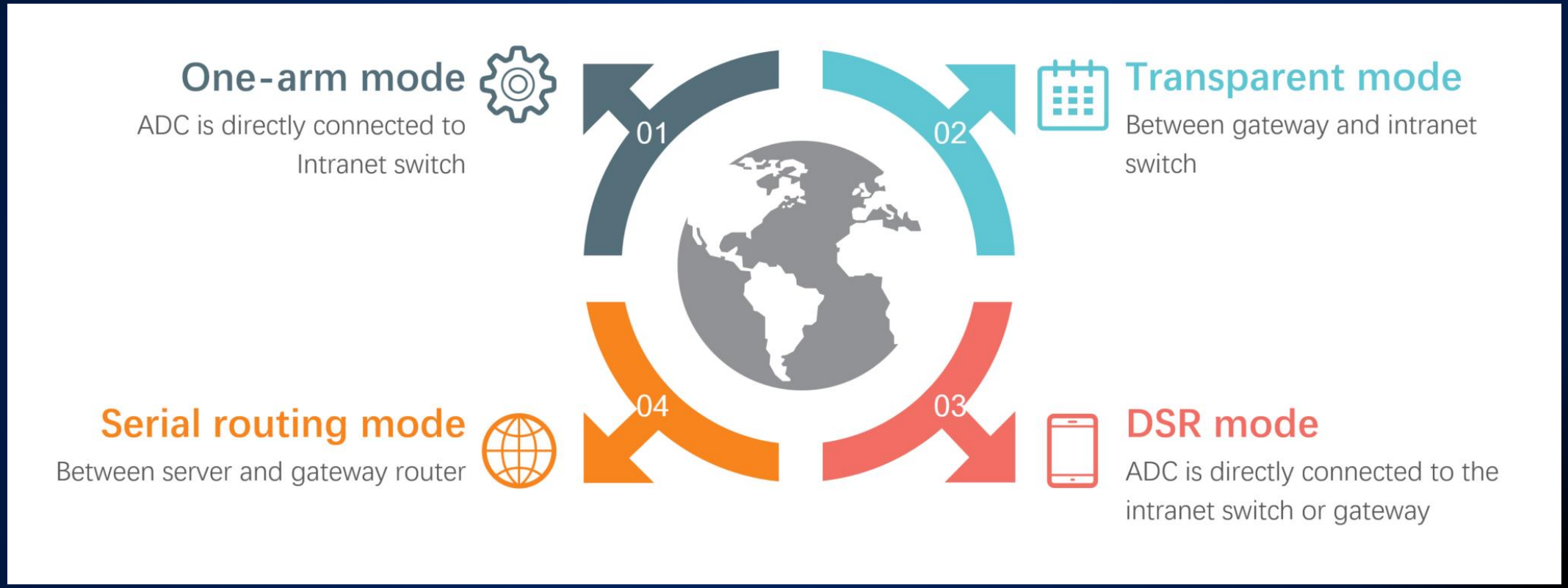
Agenda

- Deployment Mode
- Health Check

1 Deployment Mode

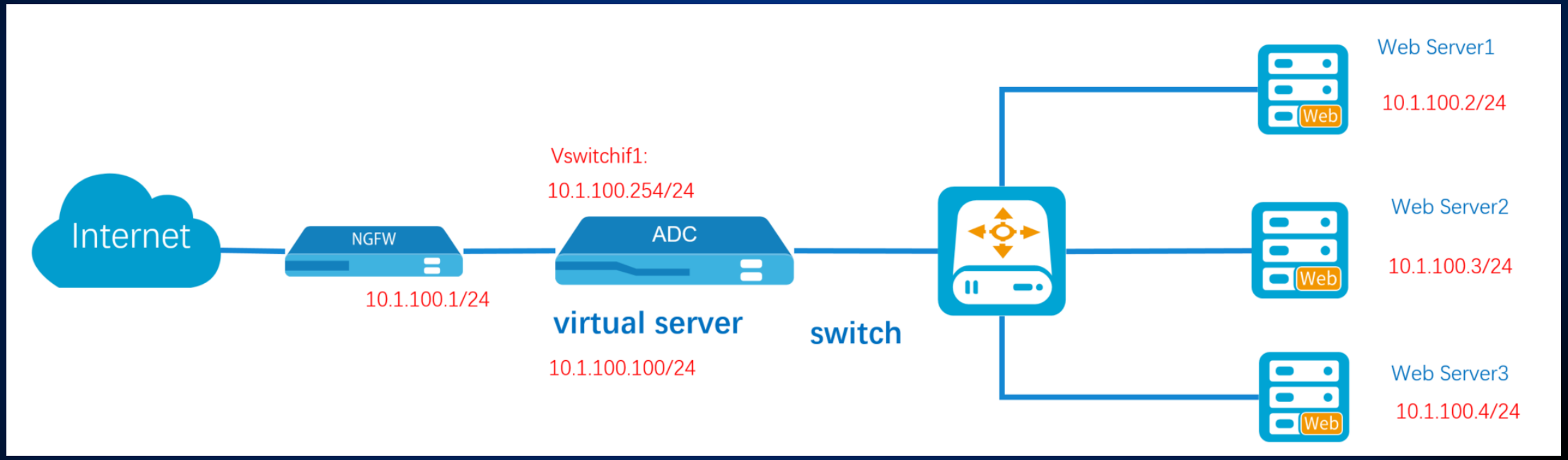
ADC Working Mode

- ADC provides four basic deployment modes, which are as follows



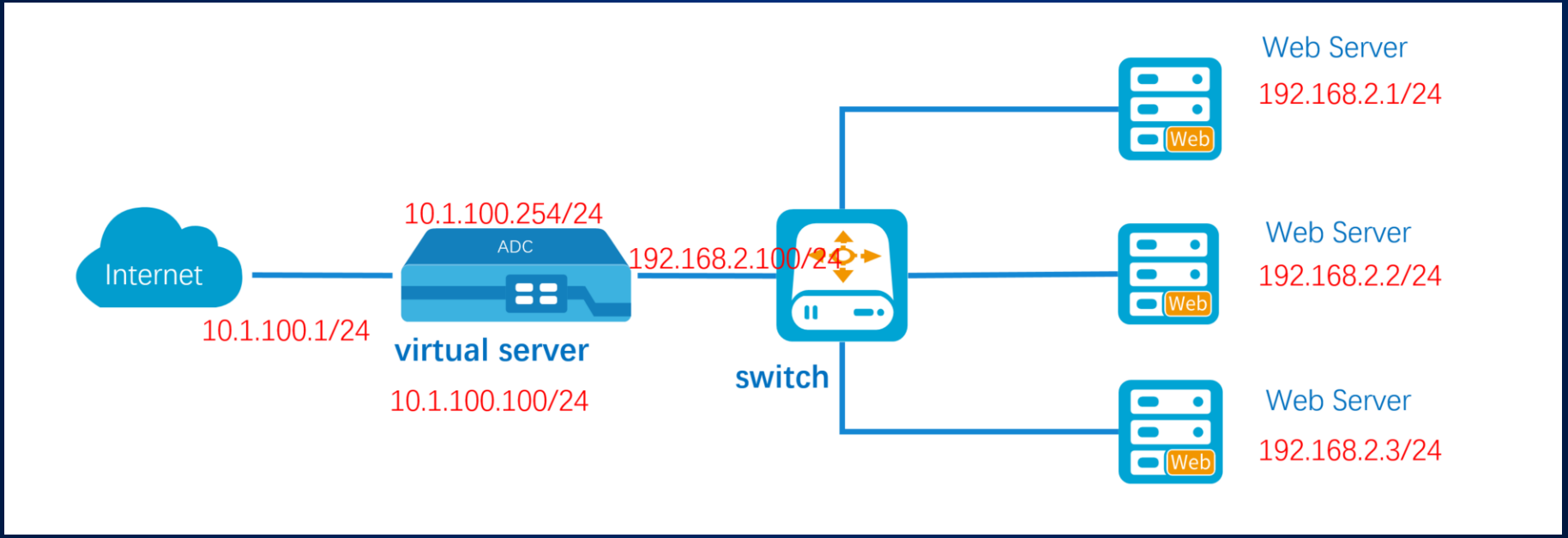
Transparent Mode

- Transparent mode is also known as bridge mode or transparent bridging mode. It is used when the IT administrator does not wish to change the existing network layout, which has already been set up with routers and switches. The deployment is simple and easy to use, and is applicable to most network environments.



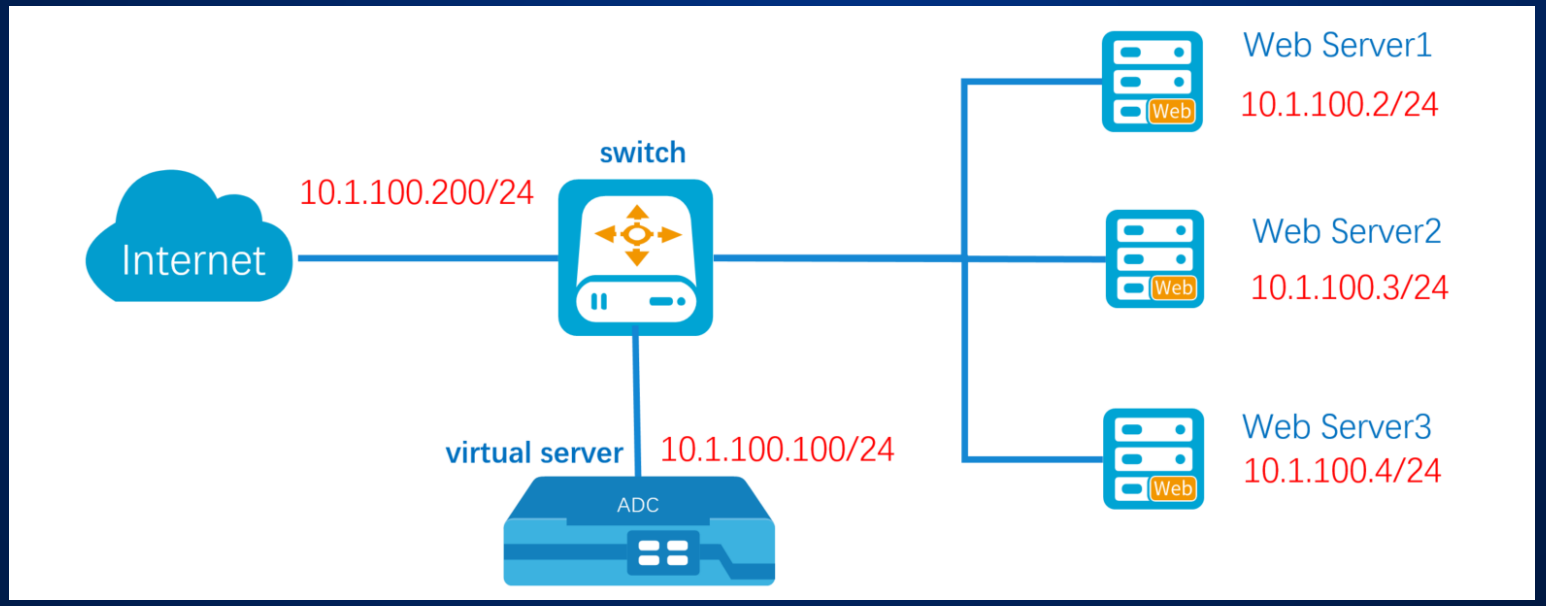
Serial Routing Mode

- Serial routing mode is characterized by deploying the ADC device between a server and a gateway, which can maximize the effects of load balance. The two sides of the ADC device (i.e., the server and the gateway) are on different network segments, so that the server can be isolated and the server security can be ensured.



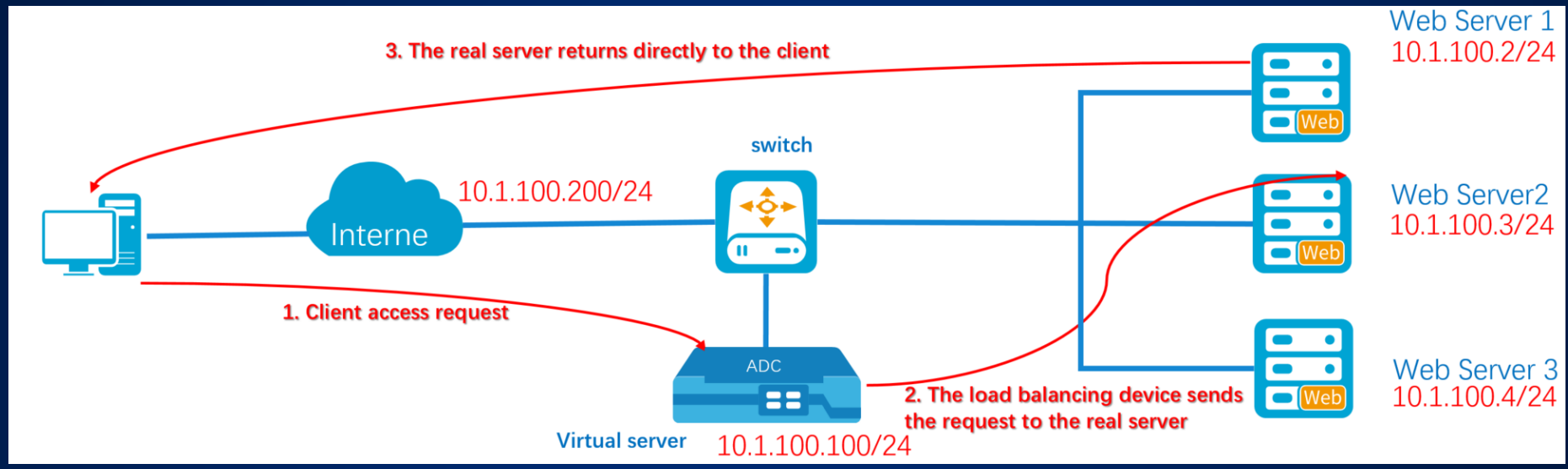
One-Arm Mode

- One-arm mode is often used when the IT administrator wants to deploy the ADC device in a bypass manner to the existing network environment without affecting the entire network's performance. When the server load balance is implemented in the one-arm mode, the real IP of the client is invisible to the server side.
- **In one-arm mode, the device is usually directly connected to the internal network switch.**



DSR Mode (Direct Server Return)

- Direct Server Return (DSR) mode is also called the triangle mode, which is characterized by that a server responds directly to a client. In DSR mode, a request sent by a client to a server will be processed by the ADC device first, and then forwarded to the server. However, a response returned by the server will be sent directly to the client instead of passing through the device.



- Features:

✓ The asymmetric deployment mode can avoid performance bottlenecks caused by load balancers, and reduce network Latency.

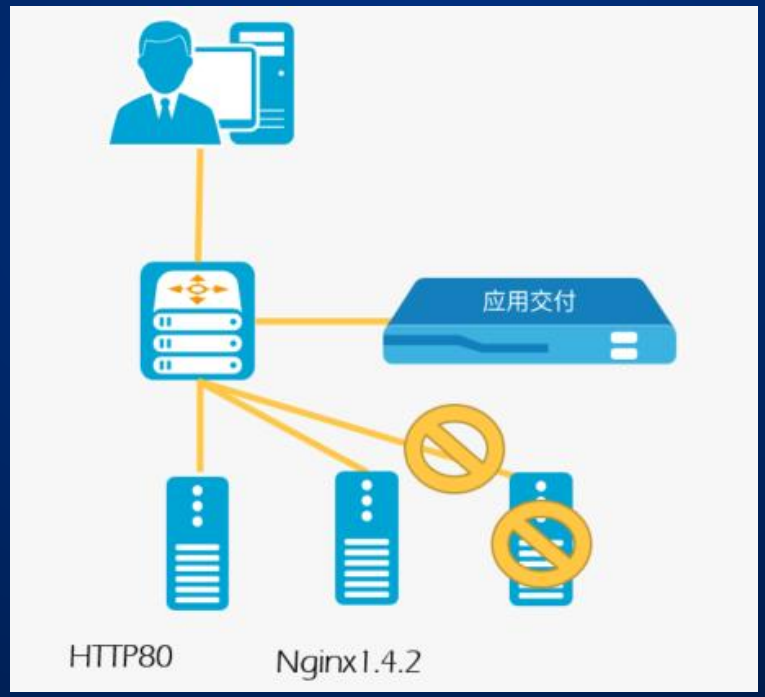
✓ This mode is only supported on Layer 4, but not on Layer 7.

✓ Commonly used scenarios: It is often used in network scenarios that require low latency, such as voice and video applications

2 Health Check

Health Check

- Health Check: By sending probe messages, the server's status, network, performance, and service conditions are probed, which will help to exclude servers that are unreachable, services that are not enabled, servers that have reached performance bottlenecks, or servers that have changed their service content from the list of resources available for allocation, in order to ensure that client requests are distributed to real servers that can work normally.

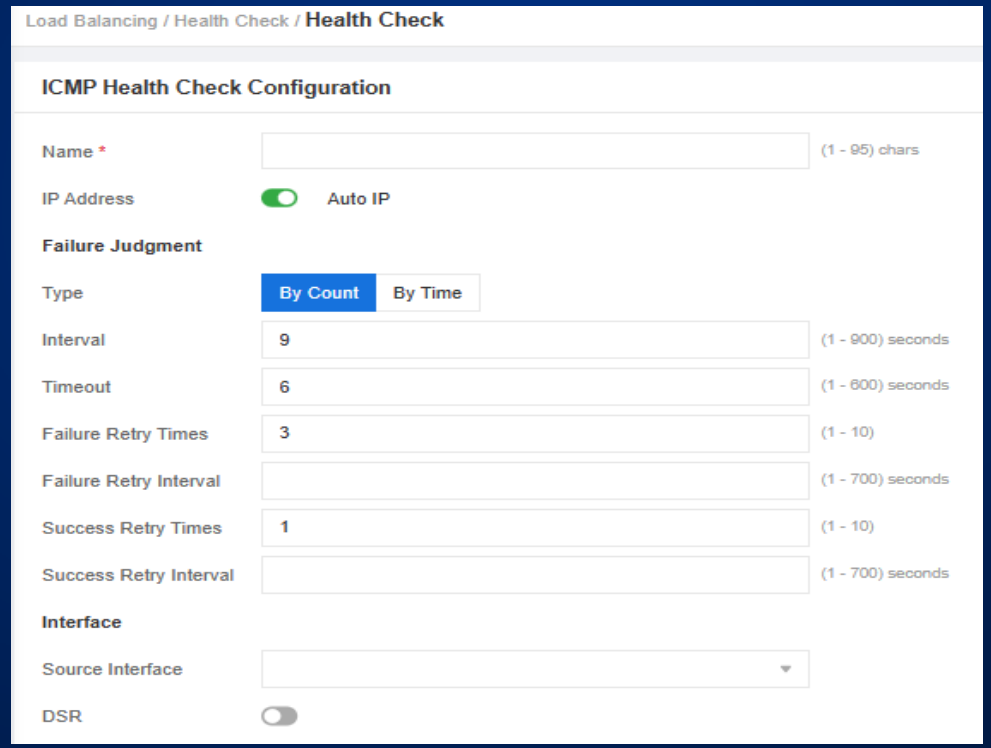


Health Check Methods

- AX provides multiple health check methods.
  - Based on network and transport layer techniques such as use method like ICMP, TCP-ECHO, etc. to check the connectivity.
  - Based on application layer techniques such as TCP, UDP, TCP-HALF-OPEN, HTTP, HTTPS, DNS, SMTP, POP3, DNS, FTP, RADIUS, WEBSOCKET, SNMP, SIP, and HTTP/TCP passive detection.
  - Content customization and third-party script is also available for health check implementation.
  - Health check combination: AND, OR, Threshold.

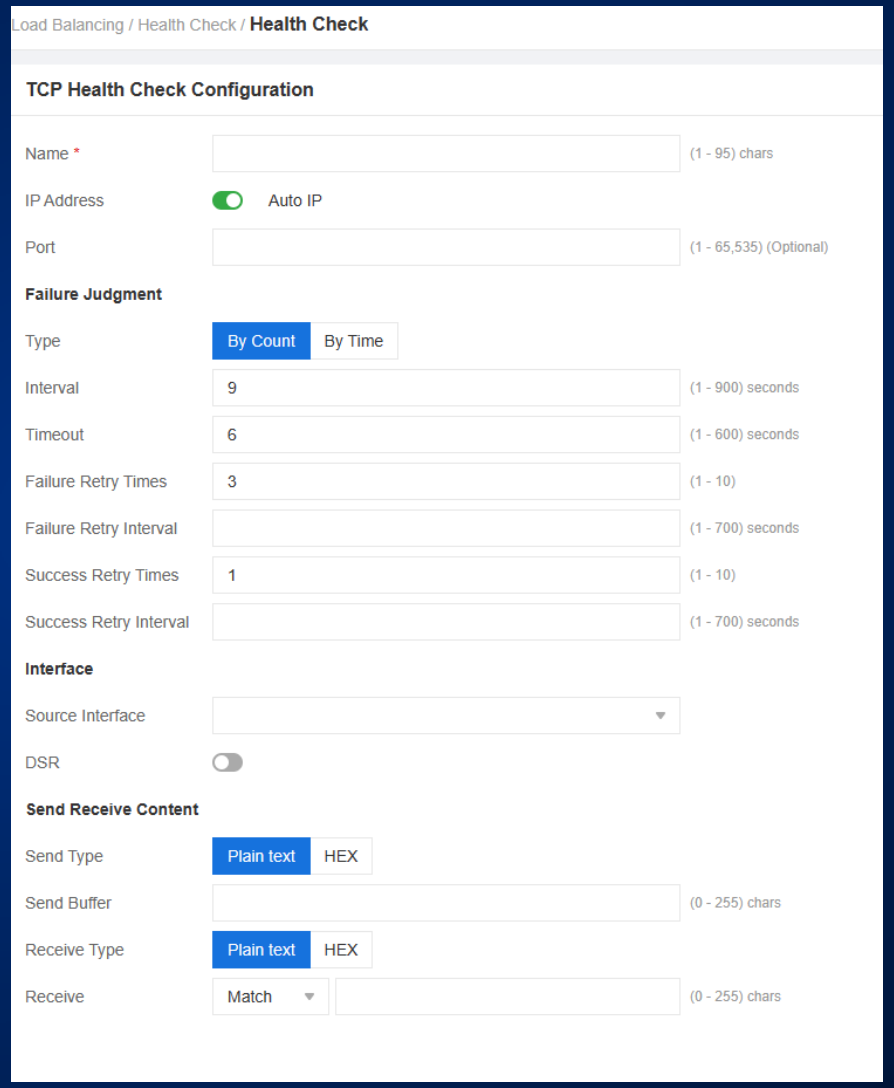
ICMP Health check

- Used for simple checks on real server's network. During the check, the device sends ICMP ECHO packets to the real service. If a response message is received from the real service within the timeout period, the ICMP health check is considered successful. Otherwise, a health check will be reattempted. If the number of consecutive retries exceeds the "retry count" set by the user, the ICMP health check is considered to be failed.



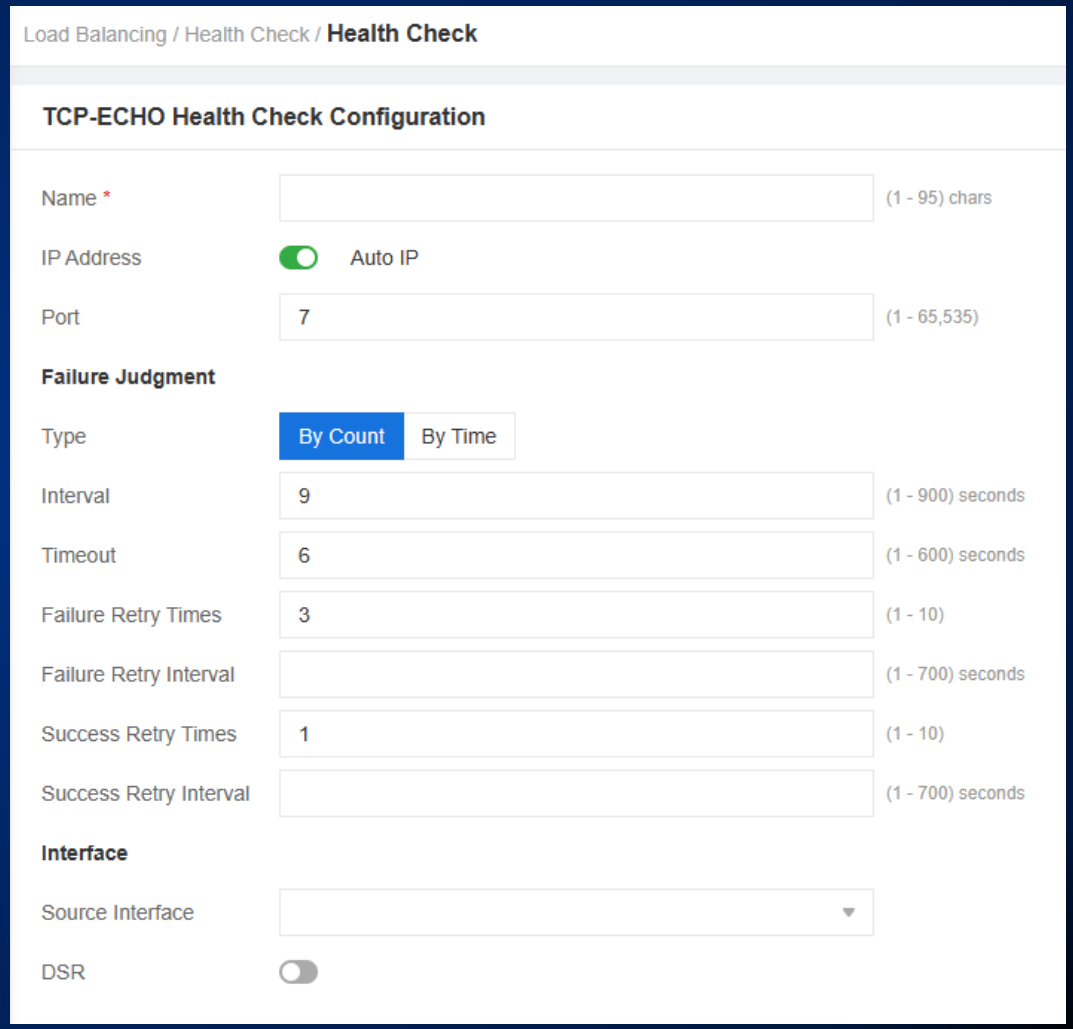
TCP Health Check

- It is used to check the TCP connection service between device and real server. During the check, the device establishes a TCP connection with the real server and sends specific content. If the content returned by the real server within the timeout period contains the expected content configured, the TCP health check is considered successful. Otherwise, a health check will be performed again. If the number of consecutive retries exceeds the "retry times" set by the user, the TCP health check is considered to be failed.



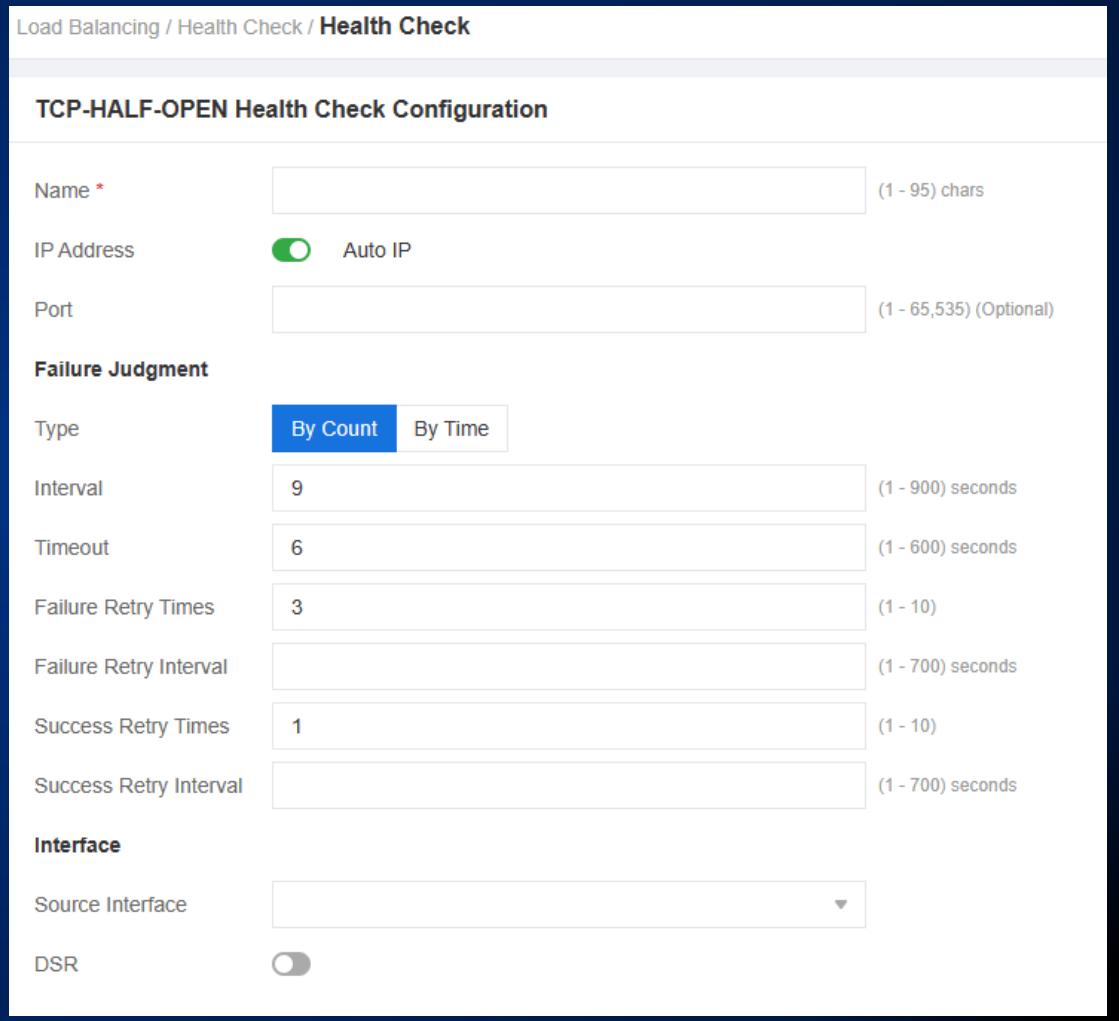
TCP-ECHO Health Check

- Used to check the TCP connection status between a device and a real service. During the check, the device establishes a TCP connection with the real server through the ECHO port (default port is 7) and sends a specific string. If the real server returns the same string as the one sent, the TCP-ECHO health check is considered to be successful. Otherwise, the health check will be performed again. If the number of consecutive retries exceeds the "retry count" set by the user, the TCP-ECHO health check is considered failed.



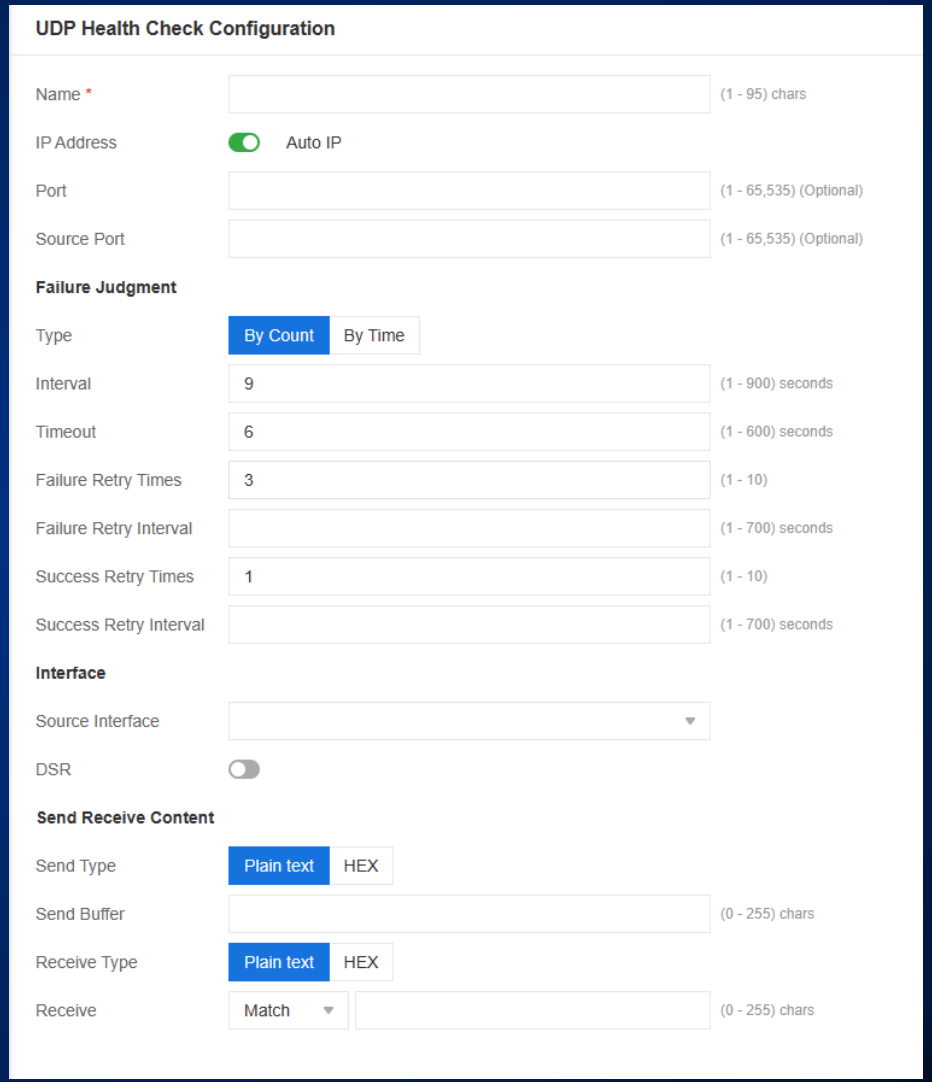
TCP-HALF-OPEN Health Check

- Used to check the TCP connection between a device and a real service. During the check, the device establishes a TCP half-connection with the real server for the check. The device sends a SYN packet to the real server, and if the real server returns a SYN ACK packet, the TCP half-connection health check is considered successful. Otherwise, the health check will be performed again. If the number of consecutive retries exceeds the "retry count" set by the user, the TCP half-connection health check is considered to be failed.



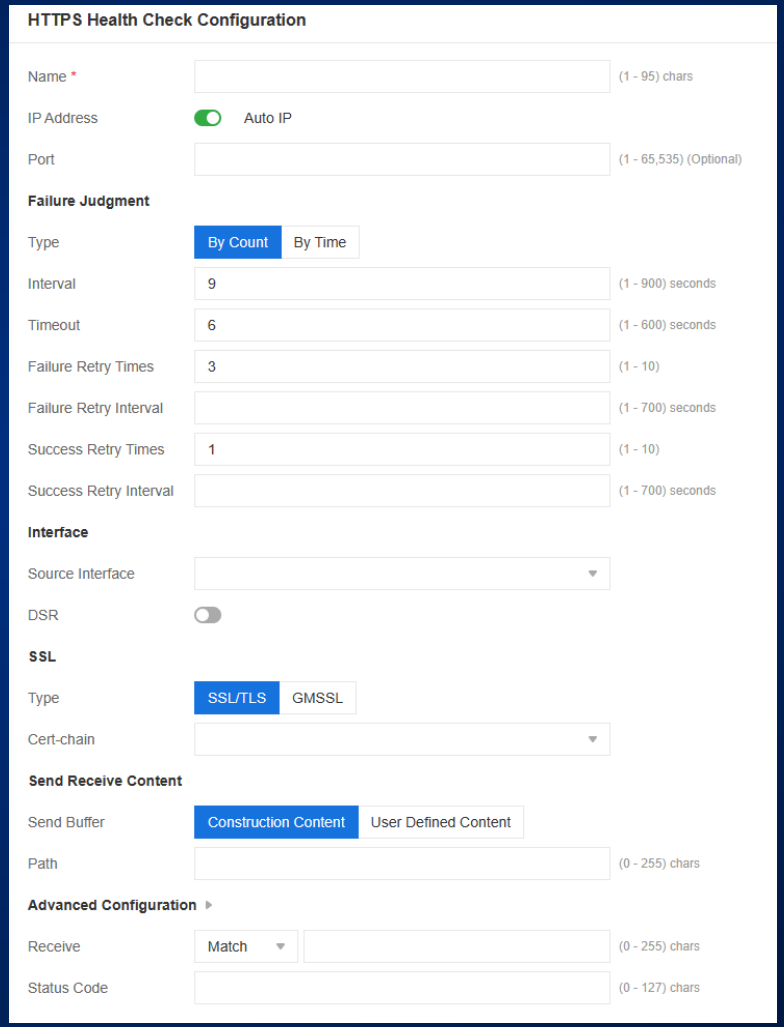
UDP Health Check

- Used to check the UDP service between device and real server. During the check, the device sends specific content to the real service through UDP. Within the timeout period, if the content returned by the real service contains the expected content configured, the UDP health check is considered to be successful. If the number of consecutive failures exceeds the retry count, the UDP health check is considered to be failed.



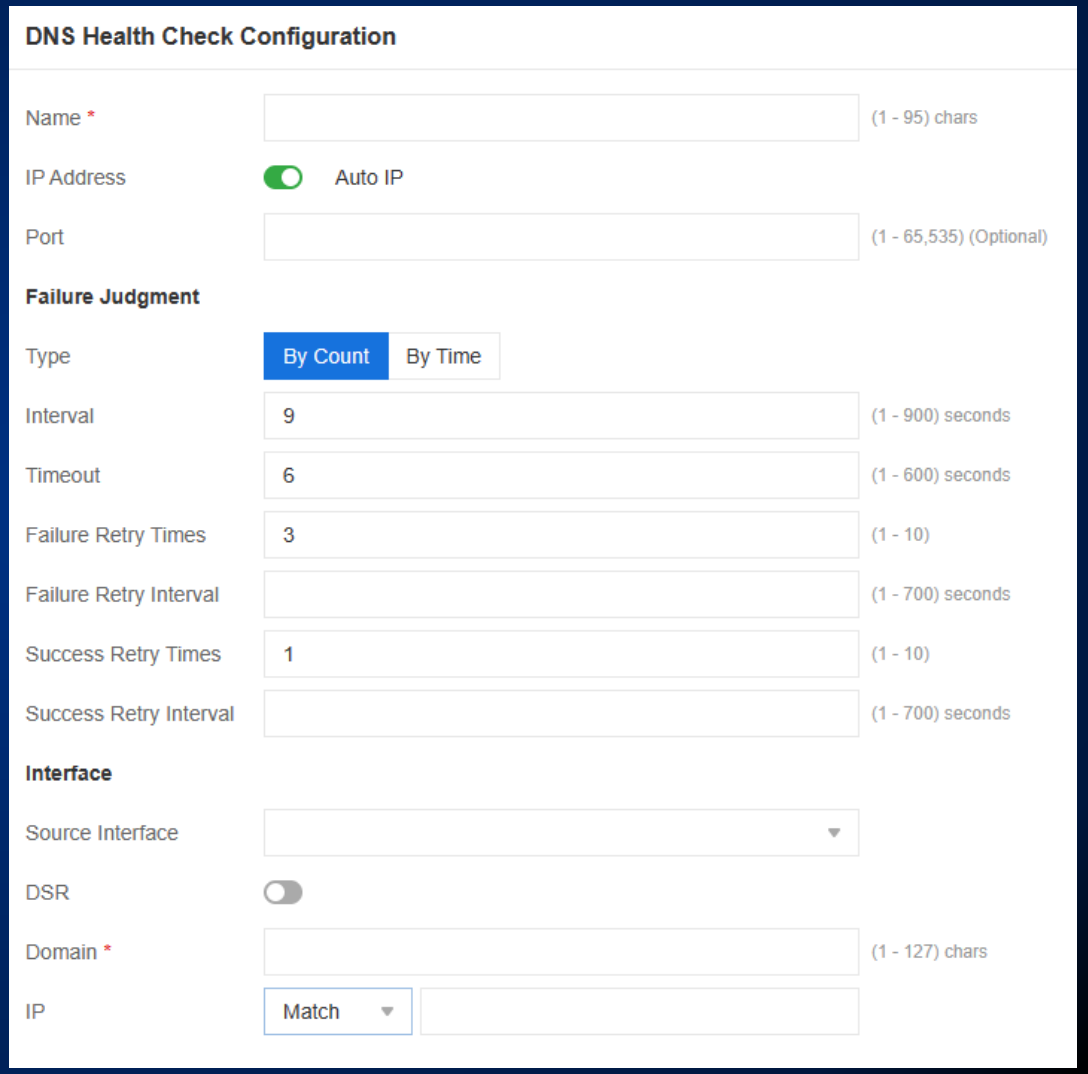
HTTP/HTTPS Health Check

- Used to check the HTTP/HTTPS service of a real server. During the check, the device sends a request to the real service through HTTP/HTTPS. Within the timeout period, if the content returned by the real server contains the expected content configured, the HTTP/HTTPS health check is considered to be successful. Otherwise, the health check will be performed again. If the number of consecutive retries exceeds the "retry count" set by the user, the HTTP/HTTPS health check is considered to be failed.



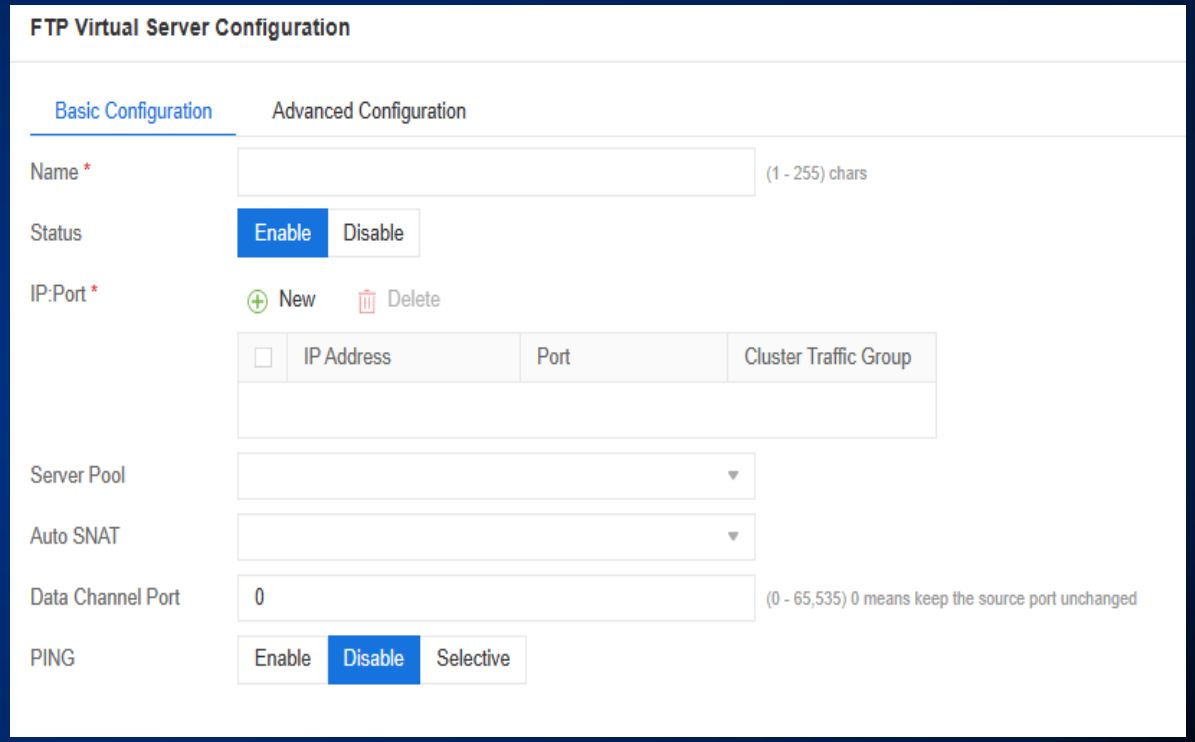
DNS Health Check

- Used for performing health checks on the DNS service status of real server. During the check, the device sends a specific domain name to the real server, and if the data returned by the real server within the timeout period contains the IP address corresponding to that domain name, the DNS health check is considered to be successful. Otherwise, a new health check will be performed. If the number of consecutive retries exceeds the "retry times" set by the user, the DNS health check is considered to be failed.



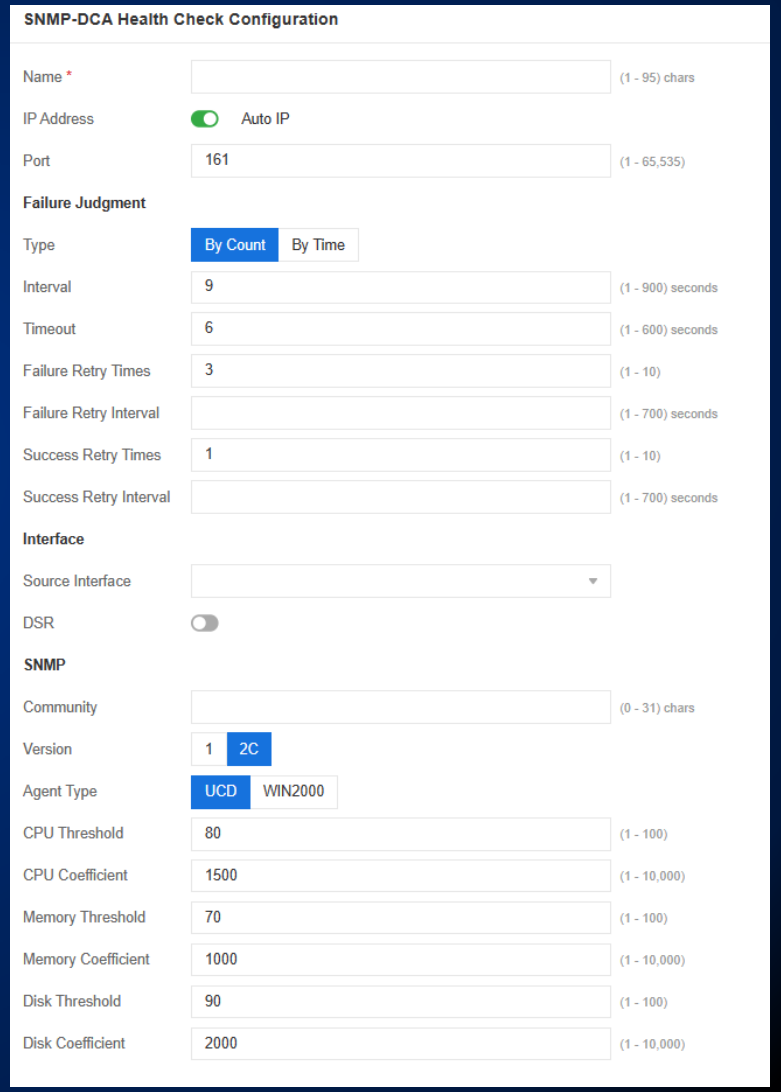
FTP Health Check

- Used for performing health checks on the FTP service status of real server. During the check, if the device can successfully connect to the real service by using the specified username and password within the timeout period, and download the specified file, the FTP health check is considered to be successful. Otherwise, a new health check will be performed. If the number of consecutive retries exceeds the "retry times" set by the user, the FTP health check is considered to be a failure.



SNMP Health Check

Config Item	Description
CPU Threshold	The maximum utilization rate of the CPU for the server is being checked. If this value is exceeded, the system will generate log information at the "warning" level.
CPU Coefficient	The CPU coefficient value for calculating the weight of the computing ratio for health checks, which is used to provide feedback on the selection policy of the real server.
Memory Threshold	The maximum utilization rate of the Memory for the server is being checked. If this value is exceeded, the system will generate log information at the "warning" level.
Memory Coefficient	The memory coefficient value for calculating the weight of the computing ratio for health checks
Disk Threshold	The maximum utilization rate of the Disk for the server is being checked. If this value is exceeded, the system will generate log information at the "warning" level.
Disk Coefficient	The disk coefficient value for calculating the weight of the computing ratio for health checks



SNMP Health Check

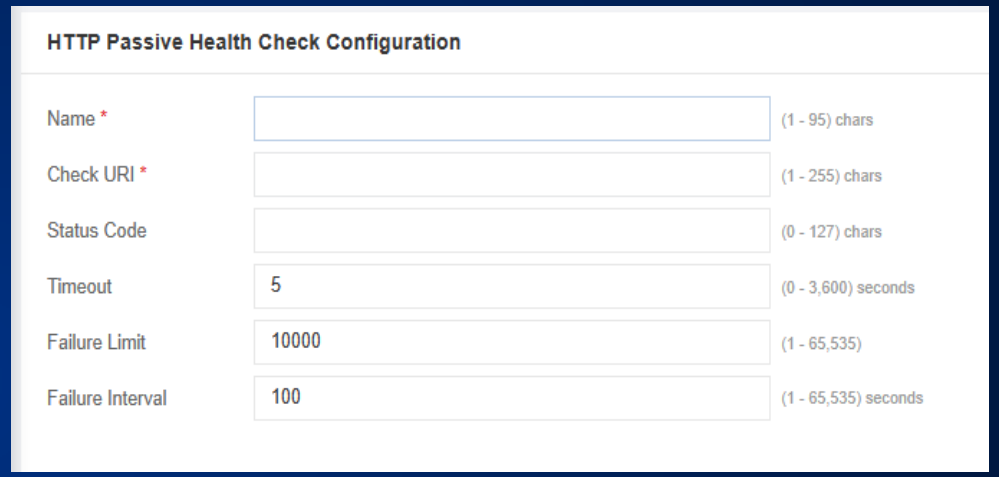
- By using SNMP to detect the usage of CPU, memory, and disk on the backend server, the status of the server can be determined like whether it is work normally. If SNMP is not working, the server is considered to be down.
  - weight = 10^(($cc*(ct-cur)$)/ct)+10^(($mc*(mt-mur)$)/mt)+10^(($dc*(dt-dur)$)/dt)
  - cur: CPU utilization threshold，//Obtain CPU usage rate
  - ct: CPU threshold
  - cc: CPU coefficient
  - mur: MEM utilization threshold，//Obtain Memory usage rate
  - mt: MEM threshold
  - mc: MEM coefficient
  - dur: DISK utilization threshold，//Obtain Disk usage rate
  - dt: DISK threshold
  - dc: DISK coefficient

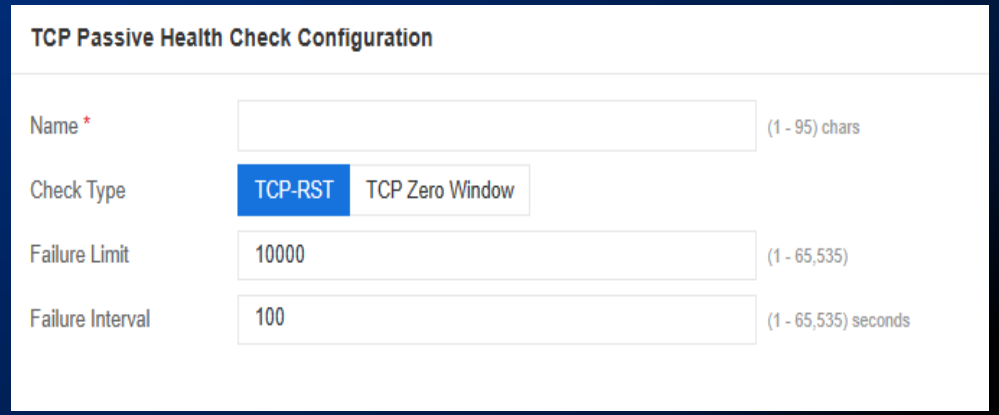
Passive Health Check

- Passive health checks: Real-time monitoring of the access data between the client and server. Once application access failure or other similar information is detected in the access data, the user request can be promptly redirected to another server, and the server can be defined as not working.
  - Passive TCP health check: By monitoring whether the server replies with TCP RST and zero window, and setting a threshold for the number of TCP RST and zero window replies, the server status can be set to be "down" when the number of replies reaches the threshold.
  - Passive HTTP health check: By monitoring the status code returned by URL and setting a threshold for the number of returned status codes, the server status can be set to be "down" when the number of replies reaches the threshold.
- The advantages of passive health checks are:
  - they can complement the defects of active health checks that are unable to perceive the server status in certain situations (such as intermittent client requests, the server will be determined to be down only when all the active detections are failed, and the count will be reset if a successful attempt is detected during multiple checks).
  - Passive health checks can also detect if the server is busy even if the port is reachable for TCP checks and the service is responding for HTTP checks, which cannot be achieved by active health checks alone.

Passive Health Check

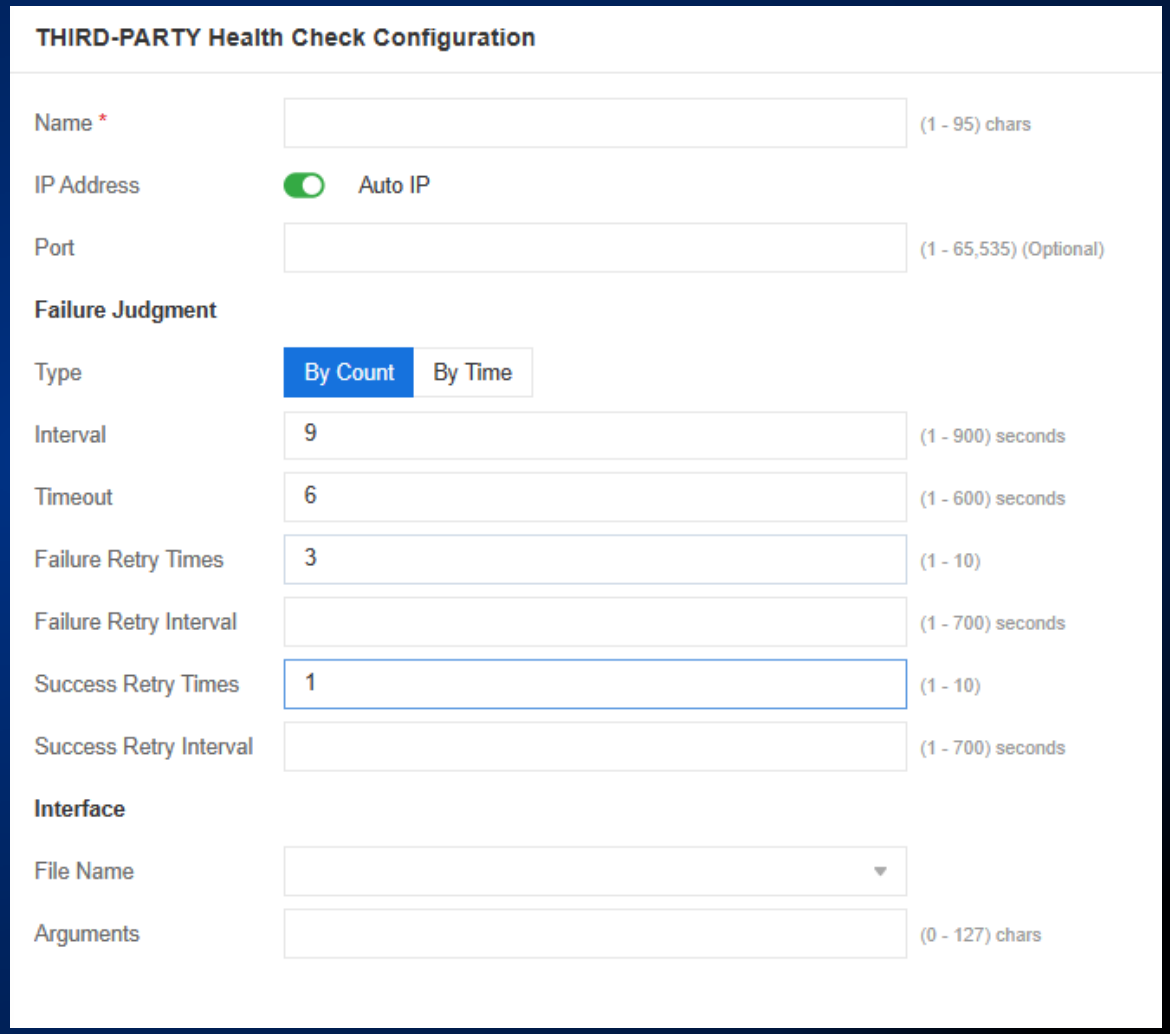
- Passive health check:
  - Check URL: The resource path that should be checked in the passive health check.
  - Status code: By determining the upper limit of error status codes returned by the server, the server status can be determined.
  - Timeout: After the device sends an HTTP request to the client, the timer starts, and if the connection is abnormal after the timeout, the number of failures will increase by 1.
  - Failure limit and Failure interval: If the number of exceptions reaches the set threshold during the exception statistics period, the server will be judged as down.
  - TCP-RST: If the device receives an RST packet, the number of failures will increase by 1.
  - TCP zero window: If the device receives a TCP window size 0 packet, the number of failures will increase by 1.





Third-Party Health Check

- Using third-party script files for health checks. Currently, uploading Python script files (UNIX-format .py files) is supported. In addition, the device comes with a third-party health check script based on Exchange, which is used to perform health checks on Exchange servers. During the check, the device executes the relevant health checks based on the script content. If the script executes successfully within the timeout period, the health check is considered to be successful.



Hillstone NETWORKS

Integrative Cybersecurity

Visionary. AI-powered. Accessible.

+1 408 508 6750
inquiry@hillstonenet.com
5201 Great America Pkwy, #420
Santa Clara, CA 95054
www.hillstonenet.com