Guidelines for Custom Reverse Proxy Deployment

This appendix provides you with guidelines for deploying an appropriate reverse proxy.

Note These guidelines are provided as best effort and Cisco does not claim support for any custom reverse proxy deployments.

Reverse proxy selection and configuration for digital channel interactions

Minimum and additional requirements

Minimum requirements

Contact Center administrators must select an appropriate reverse proxy. Any reverse proxy that meets the following minimum requirements can be used:

- Supports HTTP2/TLS 1.2.
- Has proper logging mechanism for easy debugging of issues and includes Tracking ID to easily track the task requests.
- Supports failover between the Cloud Connect nodes with health check.
- Supports X-Forwarded headers. The solution uses these headers to decide how to handle a request when front-ended with load balancer.

Additional Requirements

Some desirable requirements in a reverse-proxy are as follows:

- Consider deploying proxies that are built on non-blocking IO-based technology instead of the traditional thread-per-request architecture, to scale better.

- Apply rate limiting and configure allowed list of Webex Connect or Load balancer IPs.

Performance and hardware recommendation

For details, see Performance and Hardware Recommendations.

Configure custom reverse proxy

Install the host OS and reverse-proxy of your choice. Consider the following points while configuring the reverse-proxy:

- Configure SSL certificates as required.
- Configure the Mutual Transport Layer Security (mTLS) authentication between reverse proxy and Cloud Connect.
    - Add the list of trusted reverse proxy IP addresses and the corresponding hostnames on the publisher and subscriber nodes of Cloud Connect. For details, see Add Proxy IP.
    - Configure SSL certificate verification to establish communication between the reverse proxy host and the Digital Routing service. For details, see Configure reverse proxy host verification.
- Configure both nodes (publisher and subscriber) of Cloud Connect for task requests. Implement HTTP health check and failover to the subscriber node. The health check API that the Digital Routing service supports is */drapi/v1/ping*.
- The DataConn callback requests are routed through the reverse proxy. Configure the DataConn requests to the upstream Cloud Connect publisher node. The DataConn service runs only on the publisher node of CloudConnect.

Host header configuration

The following are the mandatory HTTP headers that reverse-proxy has to set along with the actual headers set by the client before forwarding the headers to the Finesse server.

Table 1: Host header and description

Header	Description
X-Client-IP X-Real-IP	The reverse-proxy must populate this custom header as the client's IP address before forwarding it to Cloud Connect.

Header	Description
Host	The Host request header specifies the host and port number of the server to which the request is being sent. If no port is included, the default port for the service requested (for example, 443 for an HTTPS URL and 80 for an HTTP URL) is used. An HTTP/1.1 proxy ensures that any request message it forwards contains an appropriate Host header field to identify the service being requested by the proxy. This value is used by Cloud Connect to find if the request is sent via the allowed list of proxies configured in Cloud Connect.
X-Forwarded-For	The `X-Forwarded-For` (XFF) header is used for identifying the originating IP address of a client connecting to a web server through an HTTP proxy or a load balancer. The IP of the reverse-proxy has to be appended or set. Cloud Connect uses this header to find if the request is from the allowed list of reverse-proxies. When the request is forwarded through multiple reverse-proxies or load balancer, the values of all reverse-proxies are appended to the rightmost value of this header.
X-Forwarded-Port	The reverse-proxy should set the listening port on this header. Cloud Connect server receives all the requests internally via 8445 port.
Connection	Any Connection value in the HTTP header that is set by the client must be cleared and forwarded to the Cloud Connect server so that the server decides the connection management and not the client. This prevents security outages.

Reverse proxy selection and configuration for VPN-less access to Finesse Desktop

Minimum and additional requirements

Minimum requirements

Contact Center administrators must select an appropriate reverse-proxy. Any reverse-proxy that meets the following minimum requirements can be used:

- Supports HTTP2/TLS 1.2 and secure Websockets.

- Has proper logging mechanism for easy debugging of issues
- Supports multiple Finesse, IdS, and CUIC servers from a single reverse-proxy.
- Supports periodic revalidation of cached content. This is required because any updates or installations on the internal hosts don't require a manual intervention to clear the cached content of the proxy.
- Supports custom authentications or provides alternative mechanisms such as an enterprise login to prevent unauthenticated access of solution components.

Note When you use Cisco-provided reverse-proxy configuration, the requests are authenticated at the proxy before they are forwarded to the upstream servers. When you are configuring a custom reverse-proxy, you must create this authentication layer if they have to be as secure as the Cisco provided configuration. You should consider this configuration step while planning to implement VPN-less access to Finesse using a custom reverse-proxy.

- Enables caching of static resources with support for cache-control header to reduce DoS/DDoS attack vectors and to scale the proxy. Any proxy that needs to support more than a few hundred users and does not provide response caching features should be deployed with a Content Delivery Network (CDN) with support for cache-control headers so that load and security guidelines are met.

Note CDN deployment is also recommended with caching proxies such as OpenResty® Nginx to eliminate the impact of DDoS attacks.

- Supports X-Forwarded headers. These headers are used by the solution to decide how to handle a request.

Additional Requirements

Some desirable requirements in a reverse-proxy are as follows:

- Consider deploying proxies that are built on non-blocking IO-based technology instead of the traditional thread-per-request architecture, to scale better.
- Consider proxies that provide response substitution capabilities which allow workarounds for custom gadgets as custom gadgets may not work with reverse-proxy directly.

Note Finesse Desktop Chat over reverse-proxy requires response substitution capability.

- Support for port-based forwarding can be used to reduce the cost of deployment by avoiding the need for multiple externally resolvable hostnames, public DNS records, and corresponding certificates for each internal server that has to be accessed.
- Support for custom plugin/modules, which can be used to enhance the authentication model and provide a more robust security posture.

Performance and hardware recommendation

For details, see Performance and Hardware Recommendations.

Configure Reverse-Proxy

Install the host OS and reverse-proxy of your choice. Consider the following points while configuring the reverse-proxy:

- Configure SSL certificates as required.
- Refer to the specific proxy documentation and configure the proxy rules for each service with the same host and port that is configured in the mapping file.
- IdS and IdP trust should be configured before proxy mapping configuration is done. Otherwise, proxy configuration changes will not be processed by IdS.
- For IdS hosts, if proxy configuration is changed, the administrator must re-establish trust on IdP for new IdS proxy hosts after downloading new metadata file from IdS admin.
- For Finesse hosts, if proxy configuration is changed, the administrator must manually add or update the allowed Finesse client redirect URIs from IdS administration interface.
- Whenever SAML certificate is regenerated or IdP metadata is uploaded, proxy configurations are generated afresh.

To secure the reverse-proxy, refer to the *Security Guidelines* section in the Security Guide for Cisco Unified ICM/Contact Center Enterprise .

Host Header Configuration

The following are the mandatory HTTP headers that reverse-proxy has to set along with the actual headers set by the client before forwarding the headers to the Finesse server.

Table 2:

Header	Description
X-Client-IP	The reverse-proxy should populate this custom header as the client's IP address before forwarding it to the Finesse server. This is used to log the client's IP in the Finesse server.

Header	Description
Host	The Host request header specifies the host and port number of the server to which the request is being sent. If no port is included, the default port for the service requested (for example, 443 for an HTTPS URL and 80 for an HTTP URL) is used. An HTTP/1.1 proxy ensures that any request message it forwards contains an appropriate Host header field to identify the service being requested by the proxy. This value is used by Finesse to find if the request is sent via the allowed list of proxies configured in Finesse. The hostname and port value of the reverse-proxy should be set. Otherwise, the Finesse validation fails and returns HTTP 400 Error.
X-Forwarded-For	The `X-Forwarded-For` (XFF) header is used for identifying the originating IP address of a client connecting to a web server through an HTTP proxy or a load balancer. The IP of the reverse-proxy has to be appended or set. Finesse uses this header to find if the request is from the allowed list of reverse-proxies. When the request is forwarded through multiple reverse-proxies, the values of all reverse-proxies are appended to the rightmost value of this header.
X-Forwarded-Port	The reverse-proxy should set the listening port on this header. Finesse server receives all the requests internally via 8445 port. This header value helps Finesse to set the valid configuration.

The following are the standard headers manipulated by the proxy:

Table 3:

Header	Description
Connection	Any Connection value in the HTTP header that is set by the client should be cleared and forwarded to the Finesse server. This has to be done so that the Finesse server decides the connection management and not the Finesse client. This prevents security outages.
Accept-Encoding	The reverse-proxy clears the Accept-Encoding header to have better control over compression aspects of the response.