This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

Web Real-Time Communications Deployment Guide

Scalability, Availability, Failover

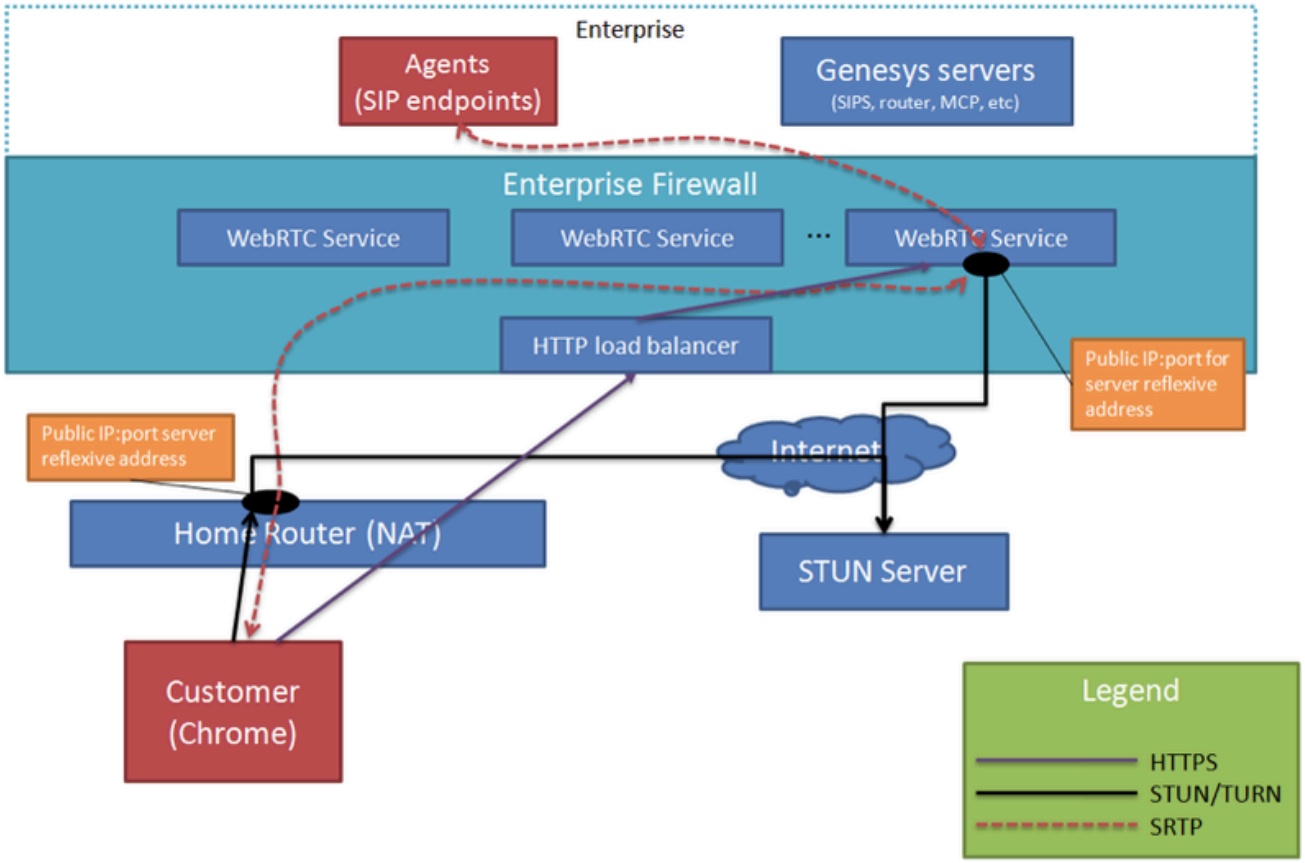
12/17/2025

Contents

Scalability, Availability, Failover

Scalability

The Genesys WebRTC Service is scalable to multiple instances. This allows it to support a large number of simultaneous media sessions with the use of an HTTP load balancer in front of the group of WebRTC Service instances. Since the WebRTC Service handles user sessions and media sessions, it is important for the HTTP load balancer to support session persistence with the use of session cookies. This will ensure that all communication with a user is always bound to a specific instance of the WebRTC Service.



Availability

In order for the WebRTC Service to achieve maximum availability—that is, in order to ensure that it can service one hundred percent of capacity at all times—you should deploy redundant instances.

In general, if the WebRTC Service is deployed as a cluster of `N` instances and you wish to prepare for the potential failure of `M` instances, you can maintain maximum availability of the service by deploying `N + M` instances. The HTTP load balancer should be configured to detect the failure of any instances and to direct incoming sessions to the remaining instances.

Failover

Currently, the Genesys WebRTC Service does not provide continuity of the media session when a failure occurs at a Genesys WebRTC Service instance. When a failure happens during a media session, the session is considered failed.

Outside of a media session, a user who is connected to the WebRTC Service to receive inbound calls will need to establish a new session with a different instance of the WebRTC Service. When the JavaScript library detects a failure, it will attempt to re-establish the session. At this time, the HTTP load balancer will also detect the failure and direct new requests to another available instance of WebRTC Service.