



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.

# User's Guide

## Load Balancing for a Single UCS Database

12/14/2025

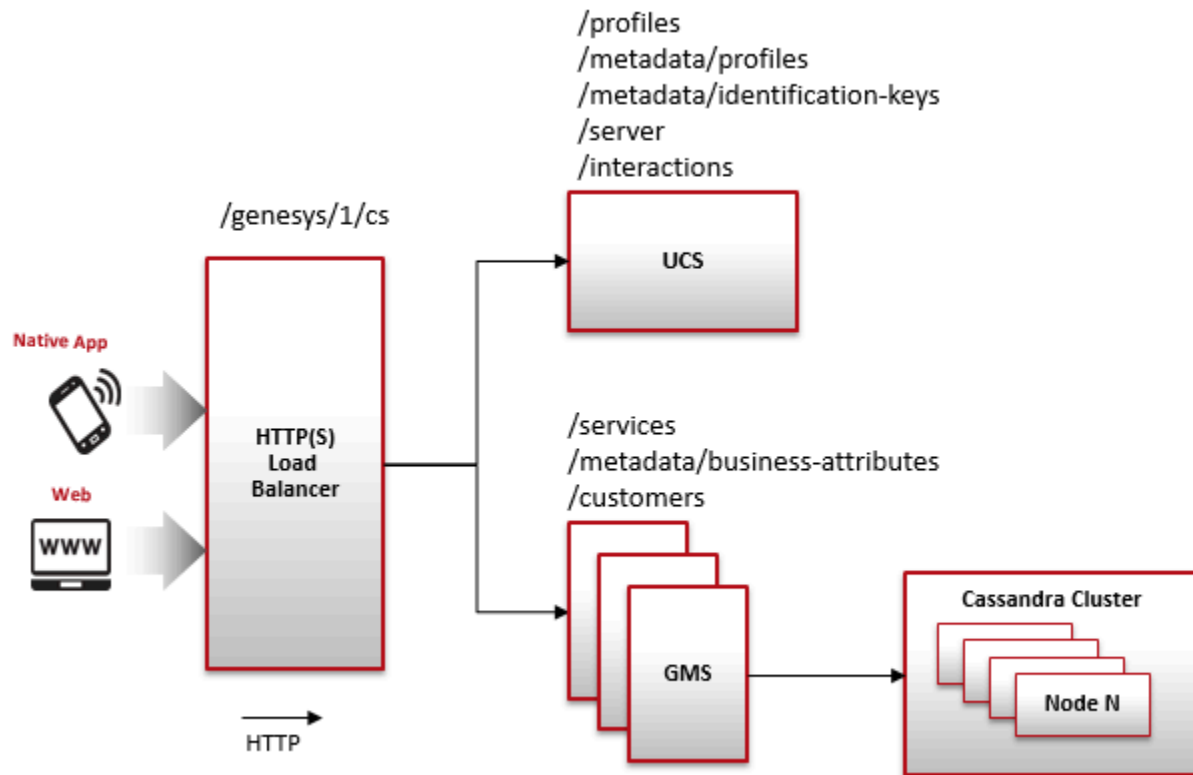
# Load Balancing for a Single UCS Database

**Purpose:** To configure Apache HTTP load balancing for the UCS database.

## Contents

- [1 Load Balancing for a Single UCS Database](#)
  - [1.1 UCS Configuration](#)
  - [1.2 Limitation](#)

This solution provides a deployment example of multiple Context Services instances, running with a single UCS database. The requests to Context Services are balanced between the servers using the Apache HTTP load balancing server. This type of configuration is beneficial for those environments experiencing high traffic to the Context Services server by providing high availability or redirecting requests to another site based on bandwidth consumption.



### Important

For examples of proxy settings, refer to the additional configuration instructions to set for UCS [here](#).

## UCS Configuration

### UCS/Context Services Configuration

The UCS application must be configured to run in Context Services mode. All other services must be disabled in the [Configuration Server](#).

### Apache HTTP Server Configuration

The Apache HTTP server uses the `mod_proxy` module for load balancing configuration. This `mod_proxy` module is directly maintained by Apache and allows more features for better performance compared to other modules (for example, `mod_jk`). The Apache server must **load the modules**. Requests to Context Services are forwarded to a cluster member, depending on the load factor. If the cluster member fails, requests are sent to the hot standby members. Apache see these members as hot stand-by; however, the Genesys configuration has them configured as Primary.

The `mod_proxy` module uses the `lbmethod` load balancing scheduler. It has three algorithms:

- `byrequests`—performs weighted request counting.
- `bytraffic`—performs weighted traffic byte count balancing.
- `bybusyness` (Apache HTTP Server 2.2.10 and later)—performs pending request balancing.

The default configuration uses the `byrequests` algorithm. For more information, see the Apache documentation for `mod_proxy`.

### Limitation

If the schema changes after creating the extension on one Context Services server instance, you must refresh other instances internal caches by calling the `/metadata/cache` URI.