



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.

Deployment Guide

Load Balancing

Load Balancing



Purpose: To describe how to implement Load Balancing for your web engagement servers.

Contents

- [1 Load Balancing](#)
 - [1.1 Introduction](#)
 - [1.2 Architecture](#)

Introduction

Genesys Web Engagement supports any third-party load balancer as long as the load balancing features include cookie support and URL encoding-based routing methods.

Load balancing deployment should occur at the latest stage of your deployment. Genesys recommends that you follow these steps:

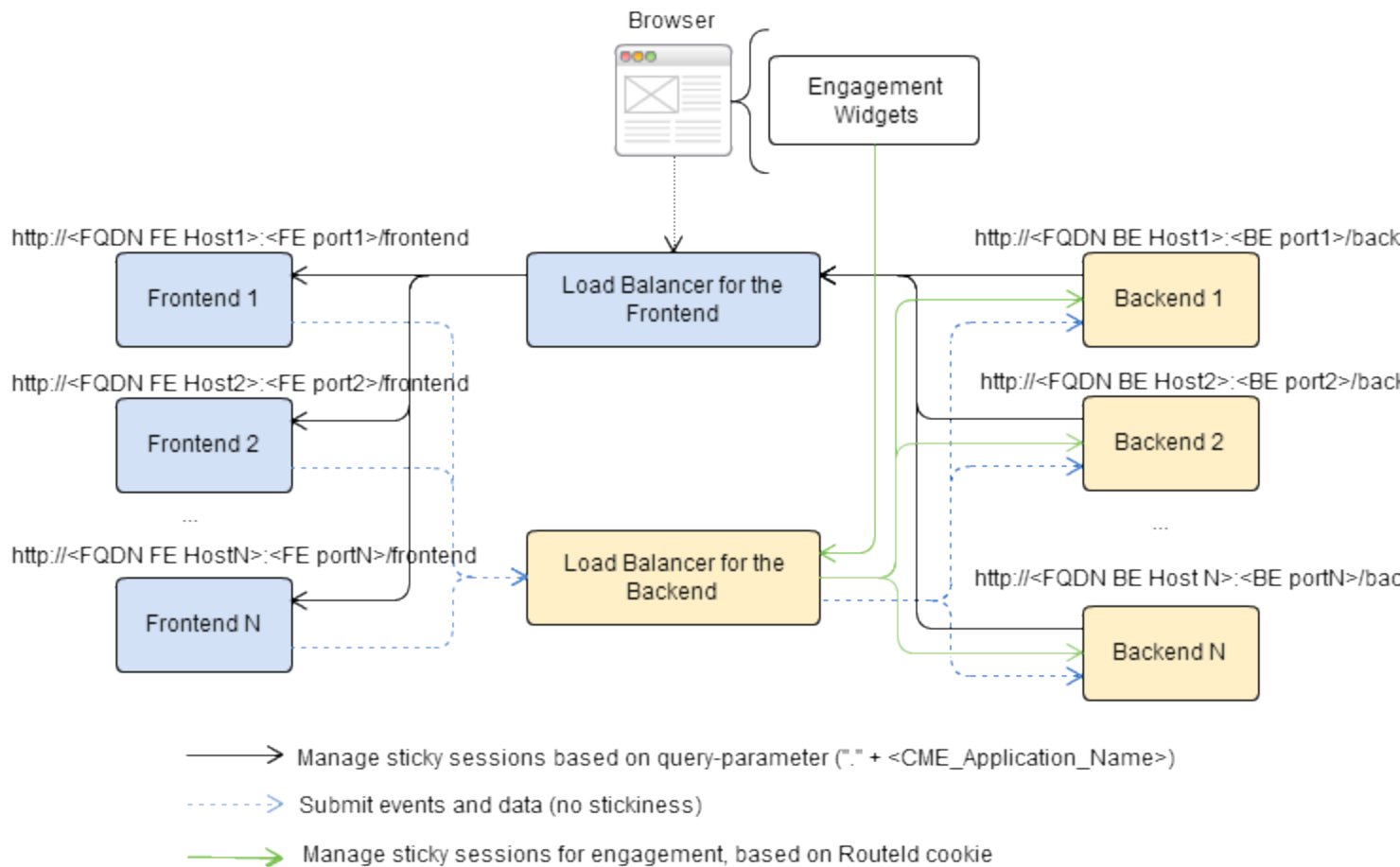
1. Install Genesys Web Engagement, as described in the [Installation chapter](#).
2. Develop and test a Web Engagement Application, as described in [Application Development Workflow](#).
3. Follow the [Load Balancing](#) guidelines to install your servers.
4. Deploy your application on the Web Engagement servers, as detailed in [Deploying to Production Environment](#)

Important

Frontend and Backend load balancers are mandatory elements for a production deployment. Interconnecting the Frontend and Backend servers directly is only appropriate for a development environment.

Architecture

The following diagram shows how you can implement a load balancing configuration for your Web Engagement servers.



Sample of Deployment for Load Balancing

In the above example:

- The load balancer for the Frontend Server implements sticky sessions for the deployed Web Engagement application;
- The load balancer for the Backend Server implements sticky sessions to route IP addresses when customers are engaged.

Sticky sessions

Genesys Web Engagement uses sticky sessions as follows:

- The load balancer for the Frontend Server implements sticky sessions based on URL encoding static parameters for the deployed Web Engagement application. Web Engagement creates the parameter as follows:
`"." + <CME_Application_Name>`.
Note: If you are using Apache, you do not need to add `"."` to your static parameter; this will be done by the Web Engagement Server.
- The load balancer for the Backend Server implements sticky sessions based on cookies to route IP addresses when the customers are engaged. The load balancer for the Backend Server must support

the following features:

- Cookie-based stickiness to enable engagement.
- Storage of sticky parameters into cookies.

Important

All the cookies are created by the Load Balancing system, not by the Genesys Web Engagement servers.

Configurations

Apache is used by default as a sample in the configuration instructions, which are available on the following page:

- [Implement Load Balancing](#)