Orchestration Server Deployment Guide

High Availability

4/4/2025

Contents

High Availability

Starting with version 8.1.3, Orchestration server can operate in a high-availability (HA) configuration, providing you with redundant systems.

The Framework Management Layer currently supports two types of redundant configurations: warm standby and hot standby. ORS offers the warm standby redundancy type.

Warm Standby

Genesys uses the expression "warm standby" to describe a high-availability configuration in which a Backup-server remains initialized and ready to take over the operations of a Primary server.

HA architecture implies the existence of redundant applications. In the case of ORS, HA is achieved with Primary and Backup ORSs. These applications are configured with redundancy so if one fails, the other takes over its operations without significant loss of data or impact to business operations.

ORS Node

A pair of Primary and Backup Orchestration Servers is called an ORS "Node". An ORS Node has a Node ID, which is a DBID of an ORS Application configured as a Primary in Configuration Server.

An ORS application running in Backup mode does not:

- Create a new session
- Serve an existing session
- Perform the pulling of multimedia interactions
- Service the HTTP port

Note: Starting with Release 8.1.3, ORS does not support Super Nodes and Master Super Nodes in the ORS Cluster architecture.

HA Deployment and Session Creation

The ORS Node (Primary/Backup pair) that is responsible for a given interaction processing is the Assigned Node. Within the same cluster, an alternative Node (Primary/Backup pair), called a Reserved Node, can be another Node in single Data Center or another Node in another Data Center.

- When receiving an appropriate event on a DN loaded with a routing strategy, the Primary server/application of the assigned Node starts call processing by trying to create a session.

- A Backup application and a Reserved Node are watching the Primary application during the timeout specified in the `orchestration/call-watching-timeout` option. This continues until the session is created by Primary application (Assigned Node) or until the timeout expired.

If session for a call was successfully created, then the Primary continues the processing of the interaction.

Session Creation Sequence

If the Primary application failed to create a session during the specified timeout or the Backup application starts running in Primary mode, the new Primary will make an attempt to create a session for a given call. If the Assigned Node (Primary/Backup pair) fail to create a session, the Reserved Node will make an attempt to create a session.

Recovery of Failed Sessions

The ORS 8.1.3 and later HA architecture provides the possibility to recover a failed sessions for a given call. When the Management Layer detects failure of a Primary ORS, it starts running the Backup Orchestration server in Primary mode. The new Primary attempts to recover the persisted sessions.

Note: Not all persisted sessions can be recovered and some session events may be missed.

During an active session, when the Backup ORS application becomes a new Primary, the session can be explicitly restored when a proactive recovery mechanism is enabled in an SCXML application. Enabling proactive recovery for an SCXML application is controlled with the <scxml> tag attribute _sendSessionRecovered, which has values of `true` and `false`.

When set to `true`, ORS publishes a `session.recovered` event to a session. The SCXML application should specify its own transition to handle session recovery. See the *Orchestration Developer's Guide* for details.

Note: Composer 8.1.3 allows the _sendSessionRecovered attribute to be added to the root <scxml> element via the `Extensions` attribute of an interaction process diagram. For information on _sendSessionRecovered, see SCXML Elements in the SCXML Language Reference. Also see Interaction Process Diagrams.

Limitations for ORS HA Deployment

- In the case of a multimedia interaction, only the Node that pulls an interaction is allowed to work with it. There are no assigned or reserved Nodes in this case.

- A session that was created on one Node cannot be transferred to another Node.