Genesys Voice Platform Private Edition Guide

High availability and disaster recovery

5/19/2026

Find out how this service provides disaster recovery in the event the service goes down.

Related documentation:

-
-
-

RSS:

- For private edition

Name	High Availability	Disaster Recovery	Where can you host this service?
Voice Platform Configuration Server	N = 1 (singleton)	Active-spare	Primary or secondary unit
Voice Platform Media Control Platform	N = N (N+1)	Active-spare	Primary or secondary unit
Voice Platform Reporting Server	N = 1 (singleton)	Active-spare	Primary or secondary unit
Voice Platform Resource Manager	N = 2 (active-active)	Active-spare	Primary or secondary unit
Voice Platform Service Discovery	N = 1 (singleton)	Active-spare	Primary or secondary unit

See High Availability information for all services: High availability and disaster recovery

GVP Configuration Server

GVP Configuration Server is a singleton instance which connects to a highly available database.

Service Discovery

Service Discovery is a singleton service which will be restarted in case of crash or unavailabilty.

Reporting Server

A single-instance Reporting Server is be used and the POD is be re-started by Kubernetes Service in case of any error.

Resource Manager

High Availability for Resource Manager is achieved by combining two Resource Manager pods in an Active-Active HA-pair, where either one of the pods can process SIP requests. SIP Server acts as a load balancer and applies proprietary load-balancing rules (round-robin) when it forwards the SIP requests.

Service is Active-Active and replicates using in-memory data. Kubernetes stateful sets with replicas (2) are used to deploy the Active-Active RM pairs in the K8 cluster. SIP-Cluster in front of RM A-A pair takes care of load balancing.

Media Control Platform

For High Availability, MCP is deployed as a pool of instances (N+1) in a region and calls are routed to available MCPs from Resource Manager (RM). RM detects when an MCP instance goes down and marks that instance as unavailable. Future calls will not be routed to that instance.

SIP Server/RM has a recovery mechanism where existing recordings which started on a MCP, which has now become unavailable, are then re-routed to a different MCP.

It is recommended to deploy the MCP pool across multiple AZs (min 2 AZs) so that there is redundancy in case of specific AZ issues.

In case of DR, MCP pool in another region should be configured along with other GVP components.

Auto-scaling

MCP supports 2 types of auto-scaling: a time-based schedule scaling and a CPU-based scaling. A combination of both types of scaling can be used to provide the most efficient and agile autoscaling policy. For example, pre-scaling at the start of the work day and scaling down at the end of the day and the ability to react to bursts of traffic using CPU-based scaling.

- Cron schedule scaling

MCP can be pre-scaled based on a time schedule using KEDA cron scaler. The following parameters are available to customize:

```
useKeda: true # If this is set to true, use Keda for scaling, or use HPA directly
keda:
    enabled: true
    preScaleStart: "0 14 * * *"
    preScaleEnd: "0 2 * * *"
    preScaleDesiredReplicas: 4
    pollingInterval: 15
    cooldownPeriod: 300
```

- CPU-based scaling

MCP scaling is also triggered by CPU usage using the Horizontal Pod Autoscaler (HPA).

```
hpa:
  enabled: true
  # minReplicas => replicaCount is used instead
  maxReplicas: 4
  targetCPUAverageUtilization: 20
```

```
scaleupPeriod: 15
scaleupPods: 4
scaleupPercent: 50
scaleupStabilizationWindow: 0
scaleupPolicy: Max
scaledownPeriod: 300
scaledownPods: 2
scaledownPercent: 10
scaledownStabilizationWindow: 3600
scaledownPolicy: Min
```