Genesys Voice Platform Private Edition Guide

6/4/2026

Table of Contents

Contents

Find links to all the topics in this guide.

Related documentation:

-
-

RSS:

- For private edition

Genesys Voice Platform is a service available with the Genesys Multicloud CX private edition offering.

Overview

Learn more about Genesys Voice Platform, its architecture, and how to support high availability and disaster recovery.

- About Genesys Voice Platform
- Architecture
- High availability and disaster recovery

Configure and deploy

Find out how to configure and deploy Genesys Voice Platform.

- Before you begin
- Configure Genesys Voice Platform
- Provision Genesys Voice Platform
- Deploy Genesys Voice Platform
- Upgrade, rollback, or uninstall Genesys Voice Platform

Upgrade, roll back, or uninstall

Find out how to upgrade, roll back, or uninstall the Genesys Voice Platform services.

- Upgrade, rollback, or uninstall Genesys Voice Platform

Observability

Learn how to monitor Genesys Voice Platform with metrics and logging.

- Logging

About Genesys Voice Platform

Contents

Learn about Genesys Voice Platform and how it works in Genesys Multicloud CX private edition.

Related documentation:

-
-
-

RSS:

- For private edition

Genesys Voice Platform

Genesys Voice Platform (GVP) is a software-only, standards-based voice portal that provides cost-effective customer interactions, 24x7, for businesses using voice, video, the web, and the cloud. Functioning beyond traditional IVR systems, GVP provides touch-tone access to applications and incorporates speech recognition technology and video for conversational exchanges, better to identify and resolve customer requests.

GVP employs a VoiceXML-based media server for network service providers and enterprise customers. GVP is a self-service system that comes with Genesys Media Server and can provide media services simultaneously with VoiceXML self-service applications.

Media services available with GVP are:

- Call parking
- Call qualification
- Call progress detection
- Third-party call recording support
- Call conferencing
- Audio/video streaming

So GVP can be used to provide augmented routing and agent services in addition to self-service applications, proactive contact solutions and outbound calling media.

Genesys Voice Platform (GVP) comprises the following services:

- GVP Configuration Server
- Service Discovery
- Reporting Server
- Resource Manager

- Media Control Platform

Supported Kubernetes platforms

GVP services are supported on the following cloud platforms:

- Google Kubernetes Engine (GKE)
- Azure Kubernetes Service (AKS)

See the Genesys Voice Platform Release Notes for information about when support was introduced.

GVP Configuration Server

GVP Configuration Server service is the internal application that connects to the database for the GVP service.

Service Discovery

Service Discovery:

- Allows MCP pods to be discovered via consul as they are scaled out and added to LRG in Config Server
- Checks the tenant configmap and if there are new tenant information changes that are not in GVP Configuration Server
- Creates/updates the tenant configuration, such as IVR profile in GVP Configuration Server

Reporting Server

Reporting Server (RS) receives the data and statistics submitted by the reporting clients (Resource Manager, Media Control Platform, and MRCP Proxy)

RS provides this service: storage in the SQLServer DB is used for billing and reporting purposes. RS uses persistent volume for storing the Active MQ journal files.

Resource Manager

Resource Manager (RM) is the first element to process requests for GVP services, and it interacts with the GVP Configuration Server to determine the tenant, the IVR profile, and the resource required to deliver the service. It then forwards the request to the resource that can deliver the service, such as

Media Control Platform (MCP) and others.

Resource Manager acts as a SIP proxy for SIP traffic between any two SIP components in the GVP architecture.

Resource Manager also acts as a SIP notifier, accepting SIP SUBSCRIBE requests from SIP Server and maintaining multiple independent subscriptions for the same or different SIP devices.

K8s headless service is created to expose both the RM addresses. The K8s RM headless service name is configured in the sip-server cluster MSML DN (VOIP DN). SIP Server is enabled to use SRV, and the RM headless service name is used as SRV record for contacting RM.

When RM pairs are upgraded, the K8s RM headless service name remains the same, so there is no need to update the SIP DNs.

Media Control Platform

Media Control Platform (MCP) provides media services such as:

- Call parking
- Call qualification
- Call Progress Detection
- Third-party call recording support
- Call conferencing
- Audio/video streaming

Architecture

Contents

Learn about Genesys Voice Platform architecture

Related documentation:

-
-
-

RSS:

- For private edition

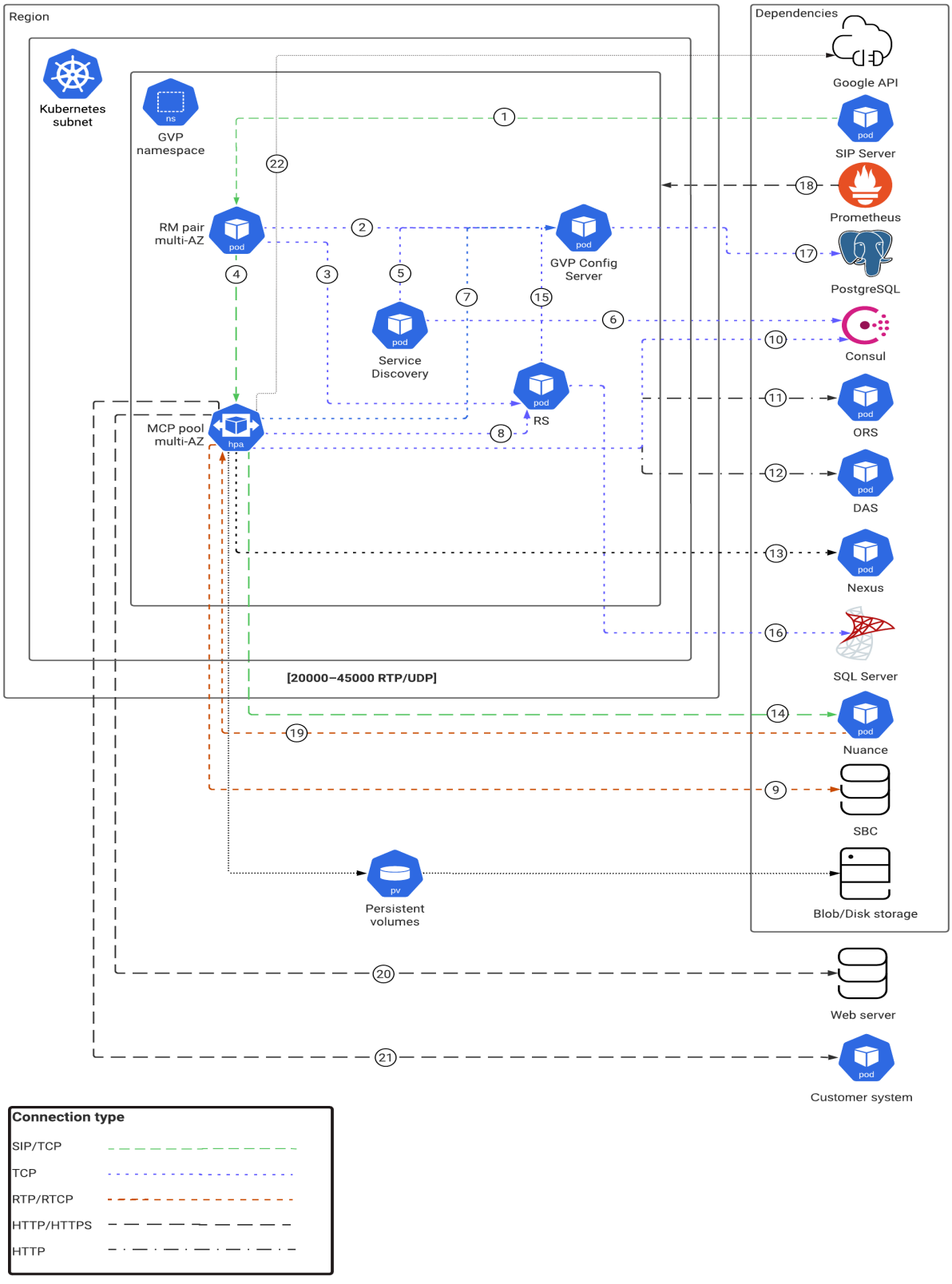
Introduction

For information about the overall architecture of Genesys Multicloud CX private edition, see the high-level Architecture page.

See also High availability and disaster recovery for information about high availability/disaster recovery architecture.

Architecture diagram — Connections

The numbers on the connection lines refer to the connection numbers in the table that follows the diagram. The direction of the arrows indicates where the connection is initiated (the source) and where an initiated connection connects to (the destination), from the point of view of Genesys Voice Platform as a service in the network.


Region
Kubernetes subnet
ns
GVP namespace
RM pair multi-AZ
pod
GVP Config Server
Service Discovery
RS
MCP pool multi-AZ
hpa
[20000–45000 RTP/UDP]
pv
Persistent volumes
Dependencies
Google API
SIP Server
Prometheus
PostgreSQL
Consul
ORS
DAS
Nexus
SQL Server
Nuance
SBC
Blob/Disk storage
Web server
Customer system
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
Connection type
SIP/TCP
TCP
RTP/RTCP
HTTP/HTTPS
HTTP

Connections table

The connection numbers refer to the numbers on the connection lines in the diagram. The **Source**, **Destination**, and **Connection Classification** columns in the table relate to the direction of the arrows in the Connections diagram above: The source is where the connection is initiated, and the destination is where an initiated connection connects to, from the point of view of Genesys Voice Platform as a service in the network. *Egress* means the Genesys Voice Platform service is the source, and *Ingress* means the Genesys Voice Platform service is the destination. *Intra-cluster* means the connection is between services in the cluster.

Connection	Source	Destination	Protocol	Port	Classification	Data that travels on this connection
1	RM pair multi-AZ	SIP Server	SIP/TCP	5090	Egress	SIP Protocol messages.
2	RM pair multi-AZ	GVP Config Server	TCP	8888	Egress	TCP messages. RM connects to GVP Configuration Server to get configuration data.
3	RM pair multi-AZ	RS	TCP	61616	Egress	ActiveMQ messages RM posts billing data to RS.
4	RM pair multi-AZ	MCP pool multi-AZ	SIP/TCP	5070	Egress	SIP Protocol messages.
5	Service Discovery	GVP Config Server	TCP	8888	Egress	TCP messages. SD connects to configuration server to Check/Add/Delete MCP applications.
6	Service Discovery	Consul	TCP	8500/8501	Egress	TCP messages. SD periodically syncs with Consul to get information of MCPs registered in Consul. SD writes tenant-related

Connection	Source	Destination	Protocol	Port	Classification	Data that travels on this connection
						information to Consul KV.
7	MCP pool multi-AZ	GVP Config Server	TCP	8888	Egress	TCP messages. MCP connects to configuration server to get recording certificate details.
8	MCP pool multi-AZ	RS	TCP	61616	Egress	ActiveMQ messages. MCP posts billing data to RS.
9	MCP pool multi-AZ	SBC	RTP/RTCP	20000-45000	Egress	RTP messages.
10	MCP pool multi-AZ	Consul	TCP	8500/8501	Egress	TCP messages. Service Handler container inside MCP Registers MCP to the Consul.
11	MCP pool multi-AZ	ORS	HTTP/HTTPS	11200	Egress	HTTP messages
12	MCP pool multi-AZ	DAS	HTTP/HTTPS	80	Egress	HTTP messages. MCP connects to DAS to fetch vxml applications.
13	MCP pool multi-AZ	Nexus		443	Egress	Websocket messages. MCP connects to Nexus for Voicebot and Agent Assist services. **Note**: The protocol is WebSocket.
14	MCP pool multi-AZ	Nuance	SIP/TCP	5060	Egress	SIP Messages

Connection	Source	Destination	Protocol	Port	Classification	Data that travels on this connection
						(port is 5060). Also, RTSP messages (port range is 14000-15999) and RTP (port range is 20000-45000). MCP connects to Nuance for ASR/TTS services.
15	RS	GVP Config Server	TCP	8888	Egress	TCP messages. RS connects to configuration server to fetch configuration data.
16	RS	SQL Server	TCP	1433	Egress	TCP Messages. RS connection to database.
17	GVP Config Server	PostgreSQL	TCP	5432	Egress	TCP messages. GVP Config Server connection to database.
18	Prometheus	GVP namespace	HTTP		Ingress	HTTP messages
19	Nuance	MCP pool multi-AZ	RTP/RTCP	20000-45000/ 14000-15999		Nuance provides ASR/TTS services. RTSP messages (port range is 14000-15999) and RTP (port range is

Connection	Source	Destination	Protocol	Port	Classification	Data that travels on this connection
						20000-45000.
20	MCP pool multi A-Z	Web server	HTTP/HTTPS	80	Egress	HTTP messages. MCP connects to web server to fetch vxml applications.
21	MCP pool multi A-Z	Customer system	HTTP/HTTPS	80	Egress	HTTP messages.
22	MCP pool multi A-Z	Google API		443	Egress	GRPC messages. MCP connects to Google APIs for TTS service.

Architecture - Configuration Server

Contents

Learn about Genesys Voice Platform- configuration server architecture

Related documentation:

-
-
-

RSS:

- For private edition

Introduction

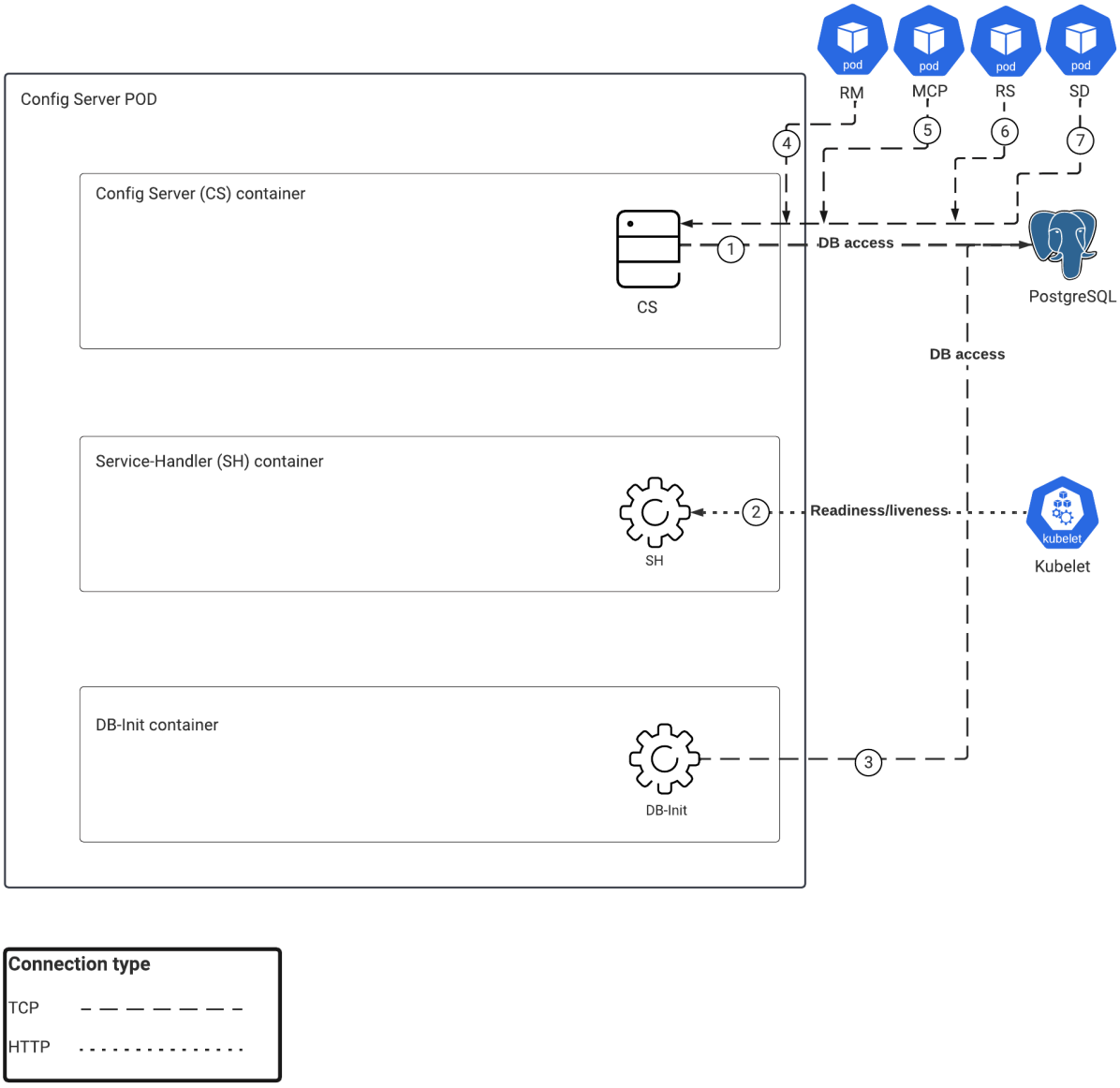
The following diagram displays the architecture of GVP Configuration Server.

For information about the overall architecture of Genesys Multicloud CX private edition, see the high-level Architecture page.

See also High availability and disaster recovery for information about high availability/disaster recovery architecture.

Architecture diagram — Connections

The numbers on the connection lines refer to the connection numbers in the table that follows the diagram. The direction of the arrows indicates where the connection is initiated (the source) and where an initiated connection connects to (the destination), from the point of view of Genesys Voice Platform as a service in the network.



Connections table

The connection numbers refer to the numbers on the connection lines in the diagram. The **Source**, **Destination**, and **Connection Classification** columns in the table relate to the direction of the arrows in the Connections diagram above: The source is where the connection is initiated, and the destination is where an initiated connection connects to, from the point of view of Genesys Voice Platform as a service in the network. *Egress* means the Genesys Voice Platform service is the source, and *Ingress* means the Genesys Voice Platform service is the destination. *Intra-cluster* means the connection is between services in the cluster.

Connection	Source	Destination	Protocol	Port	Classification	Data that travels on this connection
1	CS	PostgreSQL	TCP	5432	Egress	TCP messages. GVP Config Server connection to database.
2	Kubelet	SH	HTTP	8300	Ingress	HTTP GET Requests and for liveness and readiness checks.
3	DB-Init	PostgreSQL	TCP	5432	Egress	TCP messages.
4	RM	CS	TCP	8888	Egress	TCP messages. RM connects to GVP Configuration Server connects to get configuration data.
5	MCP	CS	TCP	8888	Egress	TCP messages. MCP connects to Configuration Server to get recording certificate details and Google keys.
6	RS	CS	TCP	8888	Egress	TCP messages. RS connects to configuration server to fetch configuration data.
7	SD	CS	TCP	8888	Egress	TCP messages. Service Discovery connects to Configuration

Connection	Source	Destination	Protocol	Port	Classification	Data that travels on this connection
						Server to Check/Add/ Delete MCP applications.

Architecture - Service Discovery

Contents

Learn about Genesys Voice Platform- service discovery architecture

Related documentation:

-
-
-

RSS:

- For private edition

Introduction

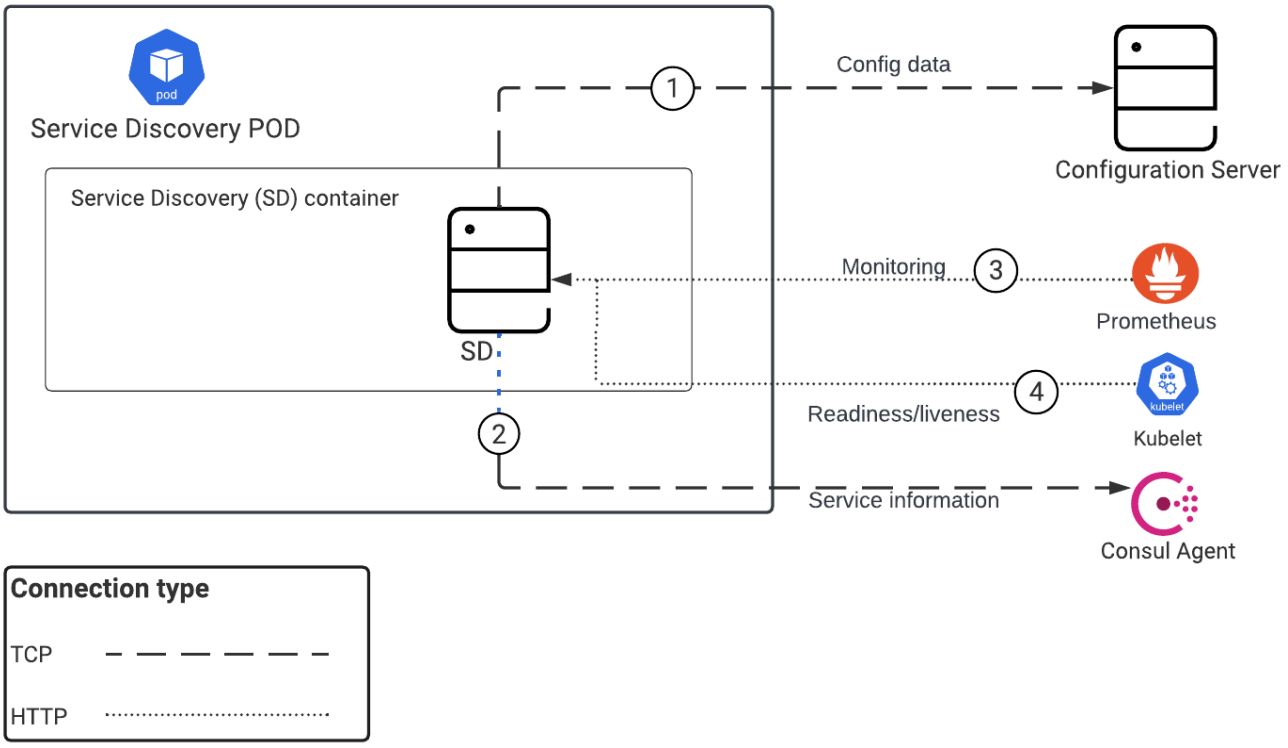
The following diagram displays the architecture of GVP Service Discovery.

For information about the overall architecture of Genesys Multicloud CX private edition, see the high-level Architecture page.

See also High availability and disaster recovery for information about high availability/disaster recovery architecture.

Architecture diagram — Connections

The numbers on the connection lines refer to the connection numbers in the table that follows the diagram. The direction of the arrows indicates where the connection is initiated (the source) and where an initiated connection connects to (the destination), from the point of view of Genesys Voice Platform as a service in the network.



Connections table

The connection numbers refer to the numbers on the connection lines in the diagram. The **Source**, **Destination**, and **Connection Classification** columns in the table relate to the direction of the arrows in the Connections diagram above: The source is where the connection is initiated, and the destination is where an initiated connection connects to, from the point of view of Genesys Voice Platform as a service in the network. *Egress* means the Genesys Voice Platform service is the source, and *Ingress* means the Genesys Voice Platform service is the destination. *Intra-cluster* means the connection is between services in the cluster.

Connection	Source	Destination	Protocol	Port	Classification	Data that travels on this connection
1	SD	Configuration Server	TCP	8888	Egress	TCP messages. SD connects to configuration server to Check/Add/ Delete MCP applications.
2	SD	Consul	TCP	8500/8501	Egress	TCP

Connection	Source	Destination	Protocol	Port	Classification	Data that travels on this connection
		Agent				messages. SD periodically syncs with Consul to get information of MCPs registered in Consul.
3	Prometheus	SD	HTTP	9090	Ingress	HTTP messages. SD metric upload to Prometheus.
4	Kubelet	SD	HTTP	8080	Ingress	HTTP GET Requests and for liveness and readiness checks.

Architecture - Reporting Server

Contents

Learn about Genesys Voice Platform- reporting server architecture

Related documentation:

-
-
-

RSS:

- For private edition

Introduction

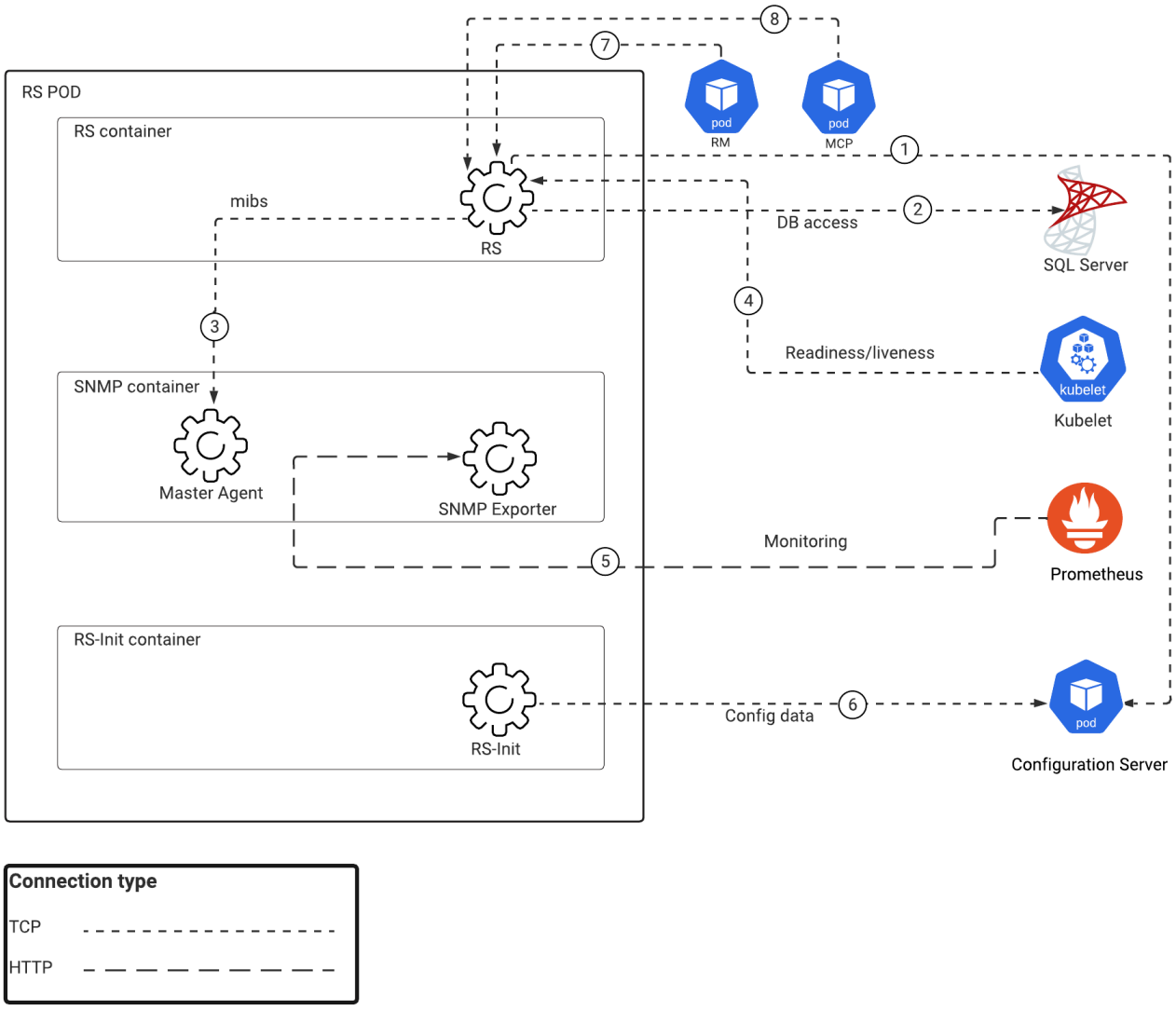
The following diagram displays the architecture for GVP Reporting Server.

For information about the overall architecture of Genesys Multicloud CX private edition, see the high-level Architecture page.

See also High availability and disaster recovery for information about high availability/disaster recovery architecture.

Architecture diagram — Connections

The numbers on the connection lines refer to the connection numbers in the table that follows the diagram. The direction of the arrows indicates where the connection is initiated (the source) and where an initiated connection connects to (the destination), from the point of view of Genesys Voice Platform as a service in the network.



Connections table

The connection numbers refer to the numbers on the connection lines in the diagram. The **Source**, **Destination**, and **Connection Classification** columns in the table relate to the direction of the arrows in the Connections diagram above: The source is where the connection is initiated, and the destination is where an initiated connection connects to, from the point of view of Genesys Voice Platform as a service in the network. *Egress* means the Genesys Voice Platform service is the source, and *Ingress* means the Genesys Voice Platform service is the destination. *Intra-cluster* means the connection is between services in the cluster.

Connection	Source	Destination	Protocol	Port	Classification	Data that travels on this connection
1	RS	Config Server	TCP	8888	Egress	TCP messages. RS connects to configuration server to fetch configuration data.
2	RS	SQL Server	TCP	1433	Egress	TCP messages. RS connection to database.
3	RS	Master Agent	TCP	1705	Egress	TCP messages. RS posts SNMP metric and traps to SNMP MA.
4	Kubelet	RS	TCP	61616 / 8080		61616 for liveness and 8080 for readiness. **Note**: The protocol is not just TCP, but TCP/ HTTP.
5	Prometheus	SNMP Exporter	HTTP	9116	Ingress	HTTP Messages. RS Custom SNMP metric upload to Prometheus.
6	RS-Init	Config Server	TCP	8888	Egress	TCP messages. RS-Init container connects to GVP CS to create RS application.
7	RM	RS	TCP	61616	Ingress	ActiveMQ messages. RM posts billing data to RS.
8	MCP	RS	TCP	61616	Ingress	ActiveMQ

Connection	Source	Destination	Protocol	Port	Classification	Data that travels on this connection
						messages. MCP posts billing data to RS.

Architecture - Resource Manager

Contents

Learn about Genesys Voice Platform- resource manager architecture

Related documentation:

-
-
-

RSS:

- For private edition

Introduction

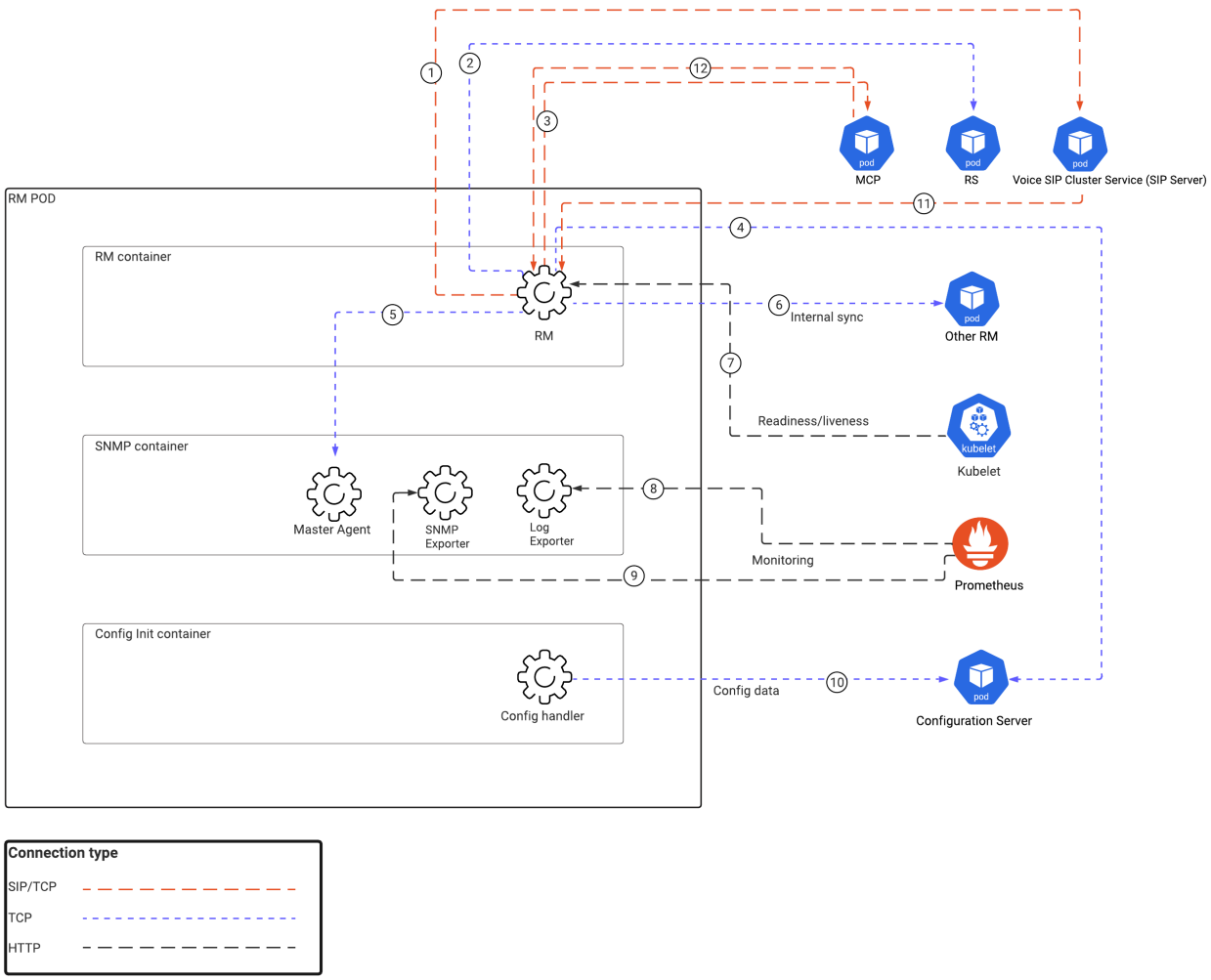
The following diagram displays the architecture for Resource Manager.

For information about the overall architecture of Genesys Multicloud CX private edition, see the high-level Architecture page.

See also High availability and disaster recovery for information about high availability/disaster recovery architecture.

Architecture diagram — Connections

The numbers on the connection lines refer to the connection numbers in the table that follows the diagram. The direction of the arrows indicates where the connection is initiated (the source) and where an initiated connection connects to (the destination), from the point of view of Genesys Voice Platform as a service in the network.



Connections table

The connection numbers refer to the numbers on the connection lines in the diagram. The **Source**, **Destination**, and **Connection Classification** columns in the table relate to the direction of the arrows in the Connections diagram above: The source is where the connection is initiated, and the destination is where an initiated connection connects to, from the point of view of Genesys Voice Platform as a service in the network. *Egress* means the Genesys Voice Platform service is the source, and *Ingress* means the Genesys Voice Platform service is the destination. *Intra-cluster* means the connection is between services in the cluster.

Connection	Source	Destination	Protocol	Port	Classification	Data that travels on this connection
1	RM	SIP Server	SIP/TCP	5090	Egress	SIP Protocol

Connection	Source	Destination	Protocol	Port	Classification	Data that travels on this connection
						messages.
2	RM	RS	TCP	61616	Egress	ActiveMQ messages. RM posts billing data to RS.
3	RM	MCP	SIP/TCP	5070	Egress	SIP Protocol messages.
4	RM	Config Server	TCP	8888	Egress	TCP messages. RM connects to GVP CS to get configuration data.
5	RM	SNMP Master Agent	TCP	1705	Egress	TCP Messages. RM posts SNMP metric and traps to SNMP MA.
6	RM	Other RM	TCP	9801	Egress	TCP messages. Internode communication between RMs.
7	Kubelet	RM	HTTP	8300		For liveness and readiness checks
8	Prometheus	Log Exporter	HTTP	8200	Ingress	HTTP messages. RM log metric upload to Prometheus.
9	Prometheus	SNMP Exporter	HTTP	9116	Ingress	HTTP Messages. RM custom SNMP metric upload to Prometheus.
10	Config handler	Config Server	TCP	8888	Egress	TCP messages. Config Handler container

Connection	Source	Destination	Protocol	Port	Classification	Data that travels on this connection
						connects to GVP CS to create RM application and LRG.
11	SIP Server	RM	SIP/TCP	5060	Ingress	SIP Protocol messages.
12	MCP	RM	SIP/TCP	5060	Ingress	SIP Protocol messages.

Architecture - Media Control Platform

Contents

Learn about Genesys Voice Platform- media control platform architecture

Related documentation:

-
-
-

RSS:

- For private edition

Introduction

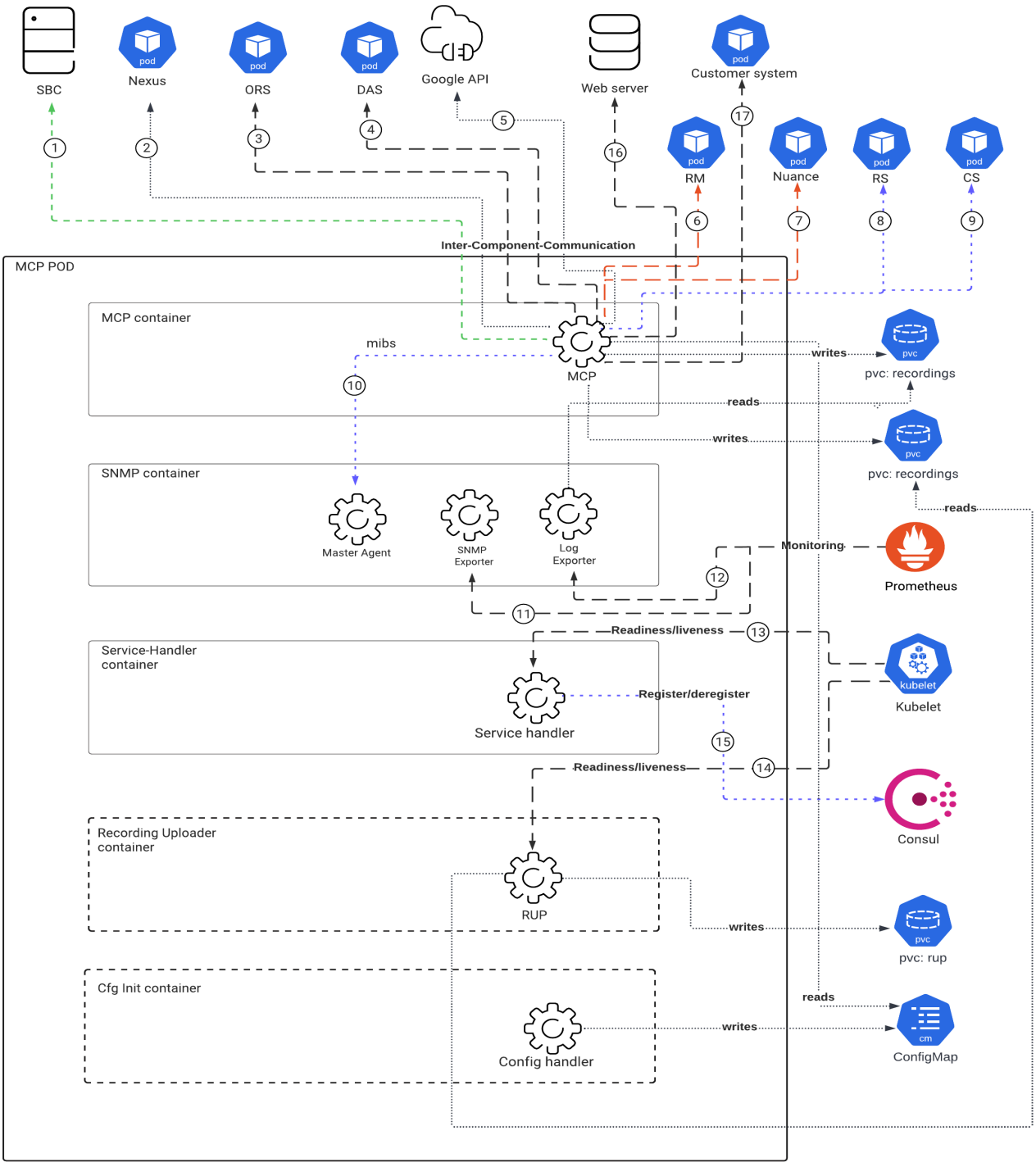
The following diagram displays the architecture for Media Control Platform.

For information about the overall architecture of Genesys Multicloud CX private edition, see the high-level Architecture page.

See also High availability and disaster recovery for information about high availability/disaster recovery architecture.

Architecture diagram — Connections

The numbers on the connection lines refer to the connection numbers in the table that follows the diagram. The direction of the arrows indicates where the connection is initiated (the source) and where an initiated connection connects to (the destination), from the point of view of Genesys Voice Platform as a service in the network.


SBC
Nexus
ORS
DAS
Google API
Web server
Customer system
RM
Nuance
RS
CS
Inter-Component-Communication
MCP POD
MCP container
MCP
mibs
writes
reads
pvc: recordings
SNMP container
Master Agent
SNMP Exporter
Log Exporter
Monitoring
Prometheus
Service-Handler container
Service handler
Register/deregister
Readiness/liveness
Kubelet
Consul
Recording Uploader container
RUP
pvc: rup
Cfg Init container
Config handler
ConfigMap
Connection type
RTP/RTCP
HTTP
SIP/TCP
TCP

Connections table

The connection numbers refer to the numbers on the connection lines in the diagram. The **Source**, **Destination**, and **Connection Classification** columns in the table relate to the direction of the arrows in the Connections diagram above: The source is where the connection is initiated, and the destination is where an initiated connection connects to, from the point of view of Genesys Voice Platform as a service in the network. *Egress* means the Genesys Voice Platform service is the source, and *Ingress* means the Genesys Voice Platform service is the destination. *Intra-cluster* means the connection is between services in the cluster.

Connection	Source	Destination	Protocol	Port	Classification	Data that travels on this connection
1	MCP	SBC	RTP/RTCP	20000-45000	Egress	RTP messages.
2	MCP	Nexus	HTTP	443	Egress	Websocket messages. MCP connects to Nexus for Voicebot and Agent Assist services.
3	MCP	ORS	HTTP	11200	Egress	HTTP messages.
4	MCP	DAS	HTTP	80	Egress	HTTP messages. MCP connects to DAS to fetch VXML applications.
5	MCP	Google API	HTTP	443	Egress	GRPC messages. MCP connects to Google APIs for TTS service.
6	MCP	RM	SIP/TCP	5060	Egress	SIP Protocol messages.
7	MCP	Nuance	SIP/TCP	5060	Egress	SIP messages. Also, RTSP messages (protocol is RTSP and port is 14000-15999) and RTP (protocol is

Connection	Source	Destination	Protocol	Port	Classification	Data that travels on this connection
						RTP and port is 20000-45000). MCP connects to Nuance for ASR/TTS services.
8	MCP	RS	TCP	61616	Egress	ActiveMQ messages. MCP posts billing data to RS.
9	MCP	CS	TCP	8888	Egress	TCP messages. MCP connects to configuration server to get recording certificate details.
10	MCP	Master Agent	TCP	1705	Egress	TCP messages. MCP posts SNMP metric and traps to SNMP MA.
11	Prometheus	SNMP Exporter	HTTP	9116	Ingress	HTTP messages. MCP Custom SNMP metric upload to Prometheus.
12	Prometheus	Log Exporter	HTTP	8200	Ingress	HTTP messages. MCP log metric upload to Prometheus.
13	Kubelet	Service Handler	HTTP	8300	Ingress	HTTP GET Requests and for liveness and readiness checks.
14	Kubelet	RUP	HTTP	8080	Ingress	HTTP GET Requests and for

Connection	Source	Destination	Protocol	Port	Classification	Data that travels on this connection
						liveness and readiness checks.
15	Service Handler	Consul	TCP	8500/8501	Egress	TCP messages. Service Handler container inside MCP Registers MCP to the Consul.
16	MCP	Web server	HTTP	80	Egress	HTTP messages. MCP connects to web server to fetch vxml applications.
17	MCP	Customer system	HTTP/HTTPS	80	Egress	HTTP messages.

High availability and disaster recovery

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
```

Before you begin

Contents

Find out what to do before deploying Genesys Voice Platform.

Related documentation:

-
-
-

RSS:

- For private edition

Limitations and assumptions

- Resource Manager does not use gateway LRG configurations. Instead, it uses the contact center ID coming from SIP Server as gvp-tenant-id in the INVITE message to identify the tenant and pick the IVR Profiles.
- Only single MCP LRG is supported per GVP deployment.
- Only the specific component configuration options documented in Helm values.yaml overrides can be modified. Other configuration options can't be changed.
- DID/DID groups are managed as part of Designer applications (Applications)
- SIP TLS / SRTP are currently not supported.

Download the Helm charts

You will have to download the GVP related Docker containers and Helm charts from the JFrog repository. For docker container and helm chart versions, refer to Helm charts and containers for Genesys Voice Platform.

For more information on JFrog, refer to the Downloading your Genesys Multicloud CX containers topic in the *Setting up Genesys Multicloud CX private edition* document.

Third-party prerequisites

Third-party services

Name	Version	Purpose	Notes
A container image		Used for downloading	

Name	Version	Purpose	Notes
registry and Helm chart repository		Genesys containers and Helm charts into the customer's repository to support a CI/CD pipeline. You can use any Docker OCI compliant registry.	
PostgreSQL	11.x	Relational database.	The GVP Configuration Server is separate from the Tenant Configuration Server and requires its own database.
Consul	1.8	Service discovery, service mesh, and key/value store.	The following features must be enabled in Consul: Service Discovery – to register MCP service for auto discovery of MCP pods by RM
MS SQL Server	2016 or later	Relational database. Required only for GVP.	

Storage requirements

Media Control Platform

Storage requirement for production (min)

Persistent Volume	Size	Type	IOPS	Functionality	Container	Critical	Backup needed
recordings-volume	100Gi	RWO	high	Storing recordings, dual AZ,	gvp-mcp, rup	Y	Y
rup-volume	40Gi	RWO	high	Storing recordings temporarily, dual AZ,	rup	Y	Y
log-pvc	50Gi	RWO	medium	storing log files	gvp-mcp	Y	Y

Storage requirements for Sandbox

Persistent Volume	Size	Type	IOPS	Functionality	Container	Critical	Backup needed
recordings-volume	50Gi	RWO	high	Storing recordings, dual AZ,	gvp-mcp, rup	Y	Y

rup-volume	20Gi	RWO	high	Storing recordings temporarily, dual AZ,	rup	Y	Y
log-pvc	25Gi	RWO	medium	storing log files	gvp-mcp	Y	Y

Resource Manager

Storage requirement for production (min)

Persistent Volume	Min Size	Type	IOPS	Functionality	Container	Critical	Backup needed
billingpvc	20Gi	RWO	high	billing	gvp-rm	Y	Y
log-pvc	50Gi	RWO	medium	storing log files	gvp-rm	Y	Y

Storage requirements for Sandbox

Persistent Volume	Min Size	Type	IOPS	Functionality	Container	Critical	Backup needed
billingpvc	20Gi	RWO	high	billing	gvp-rm	Y	Y
log-pvc	10Gi	RWO	medium	storing log files	gvp-rm	Y	Y

Service Discovery

Not applicable

Reporting Server

Storage requirement for production (min)

Persistent Volume	Min Size	Type	IOPS	Functionality	Container	Critical	Backup needed
billing-pvc	20Gi	RWO	High	Stores ActiveMQ data and config information	gvp-rs	Y	Y

Storage requirement for Sandbox

Persistent Volume	Min Size	Type	IOPS	Functionality	Container	Critical	Backup needed
billing-pvc	10Gi	RWO	High	Stores ActiveMQ data and	gvp-rs	Y	Y

Persistent Volume	Min Size	Type	IOPS	Functionality	Container	Critical	Backup needed
				config information			

GVP Configuration Server

Not applicable

Network requirements

Media Control Platform

Ingress

Not applicable

HA/DR

MCP is deployed with autoscaling in all regions. For more details, see the section Auto-scaling.

Calls are routed to active MCPs from GVP Resource Manager (RM) and in case of a MCP instance terminating, the calls are then routed to a different MCP instance.

Cross-region bandwidth

MCPs are not expected to be doing cross-region requests in normal mode of operation.

External connections

Not applicable

Pod Security Policy

All containers running as genesys user (500) and non-root user.

```
podSecurityContext:
  fsGroup: 500
  runAsUser: 500
  runAsGroup: 500
  runAsNonRoot: true
```

SMTP Settings

Not applicable

TLS/SSL Certificates configurations

Not applicable

Resource Manager

Ingress

Not applicable

HA/DR

Resource Manager is deployed as the Active and Active pair.

Cross-region bandwidth

Resource Manager is deployed per region. There is no cross region deployment.

External connections

Not applicable

Pod Security Policy

All containers running as genesys user (500) and non-root user.

```
podSecurityContext:
  fsGroup: 500
  runAsUser: 500
  runAsGroup: 500
  runAsNonRoot: true
```

SMTP Settings

Not applicable

TLS/SSL Certificates configurations

Not applicable

Service Discovery

Ingress

Not applicable

HA/DR

Service Discovery is a singleton service which will be restarted if it shuts down unexpectedly or becomes unavailable.

Cross-region bandwidth

Service Discovery is not expected to be doing cross-region requests in normal mode of operation.

External connections

Not applicable

Pod Security Policy

All containers running as genesys user (500) and non-root user.

```
podSecurityContext:
  fsGroup: 500
  runAsUser: 500
  runAsGroup: 500
  runAsNonRoot: true
```

SMTP Settings

Not applicable

TLS/SSL Certificates configurations

Not applicable

Reporting Server

Ingress

Not applicable

HA/DR

Reporting Server is deployed as a single pod service.

Cross-region bandwidth

Reporting Server is deployed per region. There is no cross region deployment.

External connections

Not applicable

Pod Security Policy

All containers running as genesys user (500) and non-root user.

```
podSecurityContext:
  fsGroup: 500
  runAsUser: 500
```

```
  runAsGroup: 500
  runAsNonRoot: true
```

SMTP Setting

Not applicable

TLS/SSL Certificates configurations

Not applicable

GVP Configuration Server

Ingress

Not applicable

HA/DR

GVP Configuration Server is deployed as a singleton. If the GVP Configuration Server crashes, a new pod will be created. The GVP services will continue to service calls if the GVP Configuration Server is unavailable and only new configuration changes, such as new MCP pods, will not be available.

Cross-region bandwidth

GVP Configuration Server is not expected to be doing cross-region requests in normal mode of operation.

External connections

External service	Functionality
PostGresSQL	database

Pod Security Policy

All containers running as genesys user (500) and non-root user.

```
podSecurityContext:
  fsGroup: 500
  runAsUser: 500
  runAsGroup: 500
  runAsNonRoot: true
```

SMTP Settings

Not applicable

TLS/SSL Certificates configurations

Not applicable

Browser requirements

N/A

Genesys dependencies

Media Control Platform

Service	Functionality
Consul	Consul service must be deployed before deploying MCP for proper service registration in GVP Configuration Server and RM.

Resource Manager

Service	Functionality
GVP Configuration Server	GVP Configuration Server must be deployed before deploying RM for proper working.

Service Discovery

Service	Functionality
Consul	Consul service must be deployed before deploying Service Discovery for proper service registration in GVP Configuration Server and Resource Manager.

Reporting Server

Service	Functionality
GVP Configuration Server	GVP Configuration Server must be deployed before deploying RS for proper working.

GVP Configuration Server

N/A

GDPR support

This section describes product-specific aspects of Genesys Voice Platform support for the European Union's General Data Protection Regulation (GDPR) in premise deployments. For general information about Genesys support for GDPR compliance, see General Data Protection Regulation.

Warning

Disclaimer: The information contained here is not considered final. This document will be updated with additional technical information.

Data Retention Policies

GVP has configurable retention policies that allow expiration of data. GVP allows aggregating data for items like peak and call volume reporting. The aggregated data is anonymous. Detailed call detail records include DNIS and ANI data. The Voice Application Reporter (VAR) data could potentially have personal data, and would have to be deleted when requested. The log data files would have sensitive information (possibly masked), but requires the data to be rotated/expired frequently to meet the needs of GDPR.

Configuration Settings

Media Server

Media Server is capable of storing data and sending alarms which can potentially contain sensitive information, but by default, the data will typically be automatically cleansed (by the log rollover process) within 40 days.

The location of these files can be configured in the GVP Media Control Platform Configuration [default paths are shown below]:

- vxmli:recordutterance-path = $InstallationRoot$/utterance/
- vxmli:recording-basepath = $InstallationRoot$/record/
- Netann:record-basepath = $InstallationRoot$/record
- msml:cpd-record-basepath = $InstallationRoot$/record/
- msml:record-basepath = $InstallationRoot$
- msml:record-irrecoverablerecordpostdir = $InstallationRoot$/cache/record/failed
- mpc:recordcachedir = $InstallationRoot$/cache/record
- calllog:directory = $InstallationRoot$/callrec/Log files and temporary files can be saved.

The location of these files can be configured in the GVP Media Control Platform Configuration [default paths are shown below]:

- vxmli:logdir = $InstallationRoot$/logs/
- vxmli:tmpdir = $InstallationRoot$/tmp/
- vxmli:directories-save_tempfiles = $InstallationRoot$/tmp/

Note: Changing default values is not really supported in the initial Private Edition release for any of the above MCP options.

Also, additional sinks are available where alarms and potentially sensitive information can be captured. See **Table 6** and **Appendix H** of the Genesys Voice Platform User's Guide for more information. The metrics can be configured in the GVP Media Control Platform configuration:

- ems.log_sinks = MFSINK I DATAC I TRAPSINK
- ems:metricsconfig-DATAC = *
- ems:dc-default-metricsfilter = 0-16,18,25,35,36,41,52-55,74,128,136-141
- ems.metricsconfig.MFSINK = 0-16,18-41,43,52-56,72-74,76-81,127-129,130,132-141,146-152

GVP Resource Manager

Resource Manager is capable of storing data and sending alarms and potentially sensitive information, but by default, the data will typically be automatically cleansed (by the log rollover process) within 40 days.

Customers are advised to understand the GVP logging (for all components) and understand the sinks (destinations) for information which the platform can potentially capture. See **Table 6** and **Appendix H** of the Genesys Voice Platform User's Guide for more information.

GVP Reporting Server

The Reporting Server is capable of storing/sending alarms and potentially sensitive information, but by default, these components process but do not store consumer PII. Customers are advised to understand the GVP logging (for all components) and understand the sinks (destinations) for information which the platform can potentially capture. See **Table 6** and **Appendix H** of the Genesys Voice Platform User's Guide for more information.

By default, Reporting Server is designed to collect statistics and other user information. Retention period of this information is configurable, with most data stored for less than 40 days. Customers should work with their application designers to understand what information is captured as part of the application, and, whether or not the data could be considered sensitive.

These settings could be changed by the customer as per their need by using a Helm chart override values.yaml.

Data Retention Specific Settings

- rs.db.retention.operations.daily.default: "40"
- rs.db.retention.operations.monthly.default: "40"
- rs.db.retention.operations.weekly.default: "40"
- rs.db.retention.var.daily.default: "40"
- rs.db.retention.var.monthly.default: "40"
- rs.db.retention.var.weekly.default: "40"
- rs.db.retention.cdr.default: "40"

Identifying Sensitive Information for Processing

The following example demonstrates how to find this information in the Reporting Server database – for the example where 'Session_ID' is considered sensitive:

- select * from dbo.CUSTOM_VARS where session_ID = '018401A9-100052D6';
- select * from dbo.VAR_CDRS where session_ID = '018401A9-100052D6';

- select * from dbo.EVENT_LOGS where session_ID = '018401A9-100052D6';
- select * from dbo.MCP_CDR where session_ID = '018401A9-100052D6';
- select * from dbo.MCP_CDR_EXT where session_ID = '018401A9-100052D6';

An example of a SQL query which might be used to understand if specific information is sensitive:

```
USE [ems-rs]
DECLARE @SearchStr nvarchar(100) = '018401A9-100052D6'
DECLARE @Results TABLE (ColumnName nvarchar(370), ColumnValue nvarchar(3630))

SET NOCOUNT ON

DECLARE @TableName nvarchar(256), @ColumnName nvarchar(128), @SearchStr2 nvarchar(110)
SET  @TableName = ''
SET @SearchStr2 = QUOTENAME('%' + @SearchStr + '%','''')

WHILE @TableName IS NOT NULL

BEGIN
    SET @ColumnName = ''
    SET @TableName =
    (
        SELECT MIN(QUOTENAME(TABLE_SCHEMA) + '.' + QUOTENAME(TABLE_NAME))
        FROM     INFORMATION_SCHEMA.TABLES
        WHERE         TABLE_TYPE = 'BASE TABLE'
            AND    QUOTENAME(TABLE_SCHEMA) + '.' + QUOTENAME(TABLE_NAME) > @TableName
            AND    OBJECTPROPERTY(
                    OBJECT_ID(
                        QUOTENAME(TABLE_SCHEMA) + '.' + QUOTENAME(TABLE_NAME)
                         ), 'IsMSShipped'
                           ) = 0
    )

    WHILE (@TableName IS NOT NULL) AND (@ColumnName IS NOT NULL)

    BEGIN
        SET @ColumnName =
        (
            SELECT MIN(QUOTENAME(COLUMN_NAME))
            FROM     INFORMATION_SCHEMA.COLUMNS
            WHERE         TABLE_SCHEMA    = PARSENAME(@TableName, 2)
                AND    TABLE_NAME    = PARSENAME(@TableName, 1)
                AND    DATA_TYPE IN ('char', 'varchar', 'nchar', 'nvarchar', 'int', 'decimal')
                AND    QUOTENAME(COLUMN_NAME) > @ColumnName
        )

        IF @ColumnName IS NOT NULL

        BEGIN
            INSERT INTO @Results
            EXEC
            (
                'SELECT ''' + @TableName + '.' + @ColumnName + ''', LEFT(' + @ColumnName + ',
3630)
                FROM ' + @TableName + ' (NOLOCK) ' +
                ' WHERE ' + @ColumnName + ' LIKE ' + @SearchStr2
            )
        END
    END
END
```

```
SELECT ColumnName, ColumnValue FROM @Results
```

Configure Genesys Voice Platform

Contents

Learn how to configure Genesys Voice Platform.

Related documentation:

-
-
-

RSS:

- For private edition

Override Helm chart values

The following sections provide information on the various settings that have to be configured in Genesys Voice Platform.

- Configuration Server
- Service Discovery
- Reporting Server
- Resource Manager
- Media Control Platform

This page also includes the section Configure Kubernetes.

GVP Configuration Server

The following tables list the configurable parameters of the GVP ConfigServer chart and their default values.

NOTE: There may be other parameters that could be changed, but changing them is not recommended, unless you know exactly what you are doing.

You must set service.port and serviceHandler.port to 8888 and 8300. This limitation will be addressed in a future release.

Parameter	Description	Default Value	Valid Values	Notes
`podLabels`	Custom labels to be added for each pod	(none)	Valid set of labels as "name: value"	

Parameter	Description	Default Value	Valid Values	Notes
`podAnnotations`	Custom annotations to be added for each pod	(none)	Valid set of annotations as "name: value"	
`replicaCount`	Number of replicas in deployment (should be 1 for CS)	`1`	A number > 0	
`partOf`	Namespace where the application is deployed	`gvp`	A string	
`image..registry`	Image registry URL for the container	`genesysengagedev.azurecr.io`	A valid registry URI	
`image..repository`	Repository path for the container in the above registry	`gvp/gvp_confserv`, `gvp_configserver_servicehandler`, `gvp/gvp_configserver_configserverinit`	A valid repository path/name	
`image..pullPolicy`	Pull policy for the container	`IfNotPresent`	`IfNotPresent` or `Always`	
`image..tag`	Container version tag	`{{.Chart.AppVersion }}`	A valid tag string	
`configserver.alerts.cpuUtilizationAlertLimit`	CPU utilization % limit above which an alert will be generated	`70`	A number > 0	The condition must last for 5 mins
`configserver.alerts.workingMemoryUtilizationAlertLimit`	Working memory utilization % limit above which an alert will be generated	`90`	A number > 0	The condition must last for 5 mins
`configserver.alerts.workingMemoryAlertLimit`	Working memory size limit in GB above which an alert will be generated	`7`	A number > 0	The condition must last for 5 mins
`configserver.alerts.maxRestarts`	A limit on number of container restarts in last 5 mins above which an alert will be generated	`2`	A number > 0	
`service.type`	Service type - ClusterIP/ NodePort/ LoadBalancer	`ClusterIP`	ClusterIP, NodePort, or LoadBalancer	
`service.host`	Host name of the service/config server	`gvp-configserver-0`	A valid Config Server name	

Parameter	Description	Default Value	Valid Values	Notes
service.port	Service port number	8888	A valid port number	
service.targetPort	Application port within the container	8888	A valid port number	
serviceHandler.port	Port of serviceHandler	8300	A valid port number	
secrets.imagePull	Secret names used for container image pulls	pureengage-docker-dev, pureengage-docker-staging	Valid secret names	
secrets.configServer.secretName	K8s secret name for Config Server	configserver-secret	A valid secret name	
secrets.configServer.secretUserKey	K8s secret user key for Config Server	username	A valid user key	
secrets.configServer.secretPwdKey	K8s secret password key for Config Server	password	A valid password key	
secrets.postgres.dbname	Postgres DB name	gvp	A valid DB name key	
secrets.postgres.dbport	Postgres DB port	5432	A valid DB port number	
secrets.postgres.secretName	K8s secret name for Postgres	postgres-secret	A valid secret name	
secrets.postgres.secretAdminUserKey	K8s secret user key for Postgres admin user	db-username	A valid DB user name key	
secrets.postgres.secretAdminPwdKey	K8s secret password key for Postgres admin user	db-password	A valid DB user password key	
secrets.postgres.secretHostnameKey	K8s secret hostname key for Postgres	db-hostname	A valid DB host name key	
secrets.postgres.secretDbNameKey	K8s secret DB name key for Postgres	db-name	A valid DB name key	
resources.requests.memory	Guaranteed memory allocation for CS container	128Mi	A valid k8s memory size value	
resources.requests.cpu	Guaranteed CPU allocation for CS container	100m	A valid k8s CPU unit value	
resources.limits.memory	Maximum amount of Memory K8s allocates for CS container	1Gi	A valid k8s memory size value	

Parameter	Description	Default Value	Valid Values	Notes
`resources.limits.cpu`	Maximum amount of CPU K8s allocates for CS container	`1`	A valid k8s CPU unit value	
`podSecurityContext.fsGroup`	Pod SecurityContext GID for volumes	`500`	A valid GID number	
`podSecurityContext.runAsUser`	Pod SecurityContext user ID for processes	`500`	A valid UID number	
`podSecurityContext.runAsGroup`	Pod SecurityContext primary group ID for processes	`500`	A valid GID number	
`podSecurityContext.runAsNonRoot`	Pod SecurityContext flag to require running as non-root	`true`	`true` or `false`	
`securityContext.*`	Same as podSecurityContext settings above but for containers			See above
`priorityClassName`	K8s priorityClassName for the pod (optional parameter, but recommended)	(none)	A valid priority class name	
`affinity`	Affinity for pod assignment (optional parameter)	(none)	Valid affinity settings	
`nodeSelector`	Node selector labels	(none)	Valid nodeSelector settings	
`tolerations`	Settings for tolerations	(none)	Valid name:value pairs	
`prometheus.createRule`	Whether to create the Prometheus rules or not	`true`	`true` or `false`	
`prometheus.logExporterPort`	Port for allowing Prometheus scrapes for log exporter. Do not change.	`8200`	A valid port number	
`prometheus.snmpExporterPort`	Port for allowing Prometheus scrapes for SNMP exporter. Do not	`9116`	A valid port number	

Parameter	Description	Default Value	Valid Values	Notes
	change.			
`grafana.enabled`	Whether to create the Grafana dashboard or not	`true`	`true` or `false`	
`grafana.k8sSidecarTargetDirectory`	Directory in Grafana where the gvp dashboard should be stored. Internal to Genesys	`/var/lib/grafana/dashboards/gvp`	`/var/lib/grafana/dashboards/gvp`	
`networkPolicies.enabled`	Whether to enable network policies or not	`false`	`true` or `false`	
`networkPolicies.dnsUdpTcp`	Port for allowing DNS lookup. Do not change.	53	A valid port number	
`dnsConfig.options.ndots`	Required number of dots in an FQDN to trigger a DNS query	3	A valid number of dots	

Service Discovery

The following tables list the configurable parameters of the GVP SD chart and their default values.

Note: There may be other parameters that could be changed, but changing them is not recommended, unless you know exactly what you are doing. SD runs periodically, every SYNC_PERIOD.

Parameter	Description	Default Value	Valid Values	Notes
`podLabels`	Custom labels to be added for each pod	(none)	Valid set of labels as "name: value"	
`podAnnotations`	Custom annotations to be added for each pod	(none)	Valid set of annotations as "name: value"	
`replicaCount`	Number of replicas in deployment (should be 1 for SD)	1	A number > 0	
`smtp`	Whether SMTP is supported or not (for email)	`allowed`	`allowed`, or `none`	

Parameter	Description	Default Value	Valid Values	Notes
`partOf`	Namespace where the application is deployed	`gvp`	A string value for name space	
`serviceName`	Service name	`gvp-sd`		Not to be changed
`image.registry`	Image registry URL for the SD container	`genesysengagedev.azurecr.io`	A valid registry URI	
`image.repository`	Repository path for the SD container in the above registry	`gvp/gvp_sd`		
`image.pullPolicy`	Pull policy for the container	`IfNotPresent`	`IfNotPresent` or `Always`	
`image.tag`	Container version tag	`{{.Chart.AppVersion }}`	A valid tag number	
`env.EXTERNAL_CONSUL_SERVER`	Complete URL of external Consul server, if used	none (empty)	URI such as https://consul.genesyscloud.com:8501	The port is included in this URL. If defined, env.CONSUL_PORT will not be used.
`env.CONSUL_PORT`	Consul Server port	`8501`	Valid port number	
`env.CONFIG_SERVER_HOST`	Name of the Config Server	`gvp-configserver`	Valid Config Server name	
`env.CONFIG_SERVER_PORT`	Port used by Config Server	`8888`	Valid port number	
`env.CONFIG_SERVER_USER`	Application name used for Config Server access	`default`	Valid application name	
`env.HTTP_SERVER_PORT`	Port used by the local HTTP Server (used for health checks)	`8080`	Valid port number	
`env.METRICS_EXPORTER_PORT`	Port used for metrics exporter	`9090`	Valid port number	
`env.DEF_MCP_FOLDER`	MCP apps CU (Configuration Unit) folder in Config Server (CS)	`MCP_Configuration_Unit\\MCP_LRG`	Valid CU folder in CS	
`env.TEST_MCP_FOLDER`	MCP apps CU (Configuration Unit) folder for testing	`MCP_Configuration_Unit_Test\\MCP_LRG`	Valid CU folder in CS	
`env.SYNC_INIT_DELAY`	Initial delay in ms before app is executed	`10000`	A Valid number of ms	
`env.SYNC_PERIOD`	Time in ms between successive executions	`60000`	A valid number of ms	

Parameter	Description	Default Value	Valid Values	Notes
`env.MCP_PURGE_PERIOD_MINS`	Time in minutes between attempts to purge extra MCPs in CME	`0`	A valid number of minutes	0 means, try every time
`env.EMAIL_METERING_FACTOR`	Error emails are sent once every this many SYNC_PERIODs	`10`	A valid number of seconds	
`` `env.RECORDINGS_CONTAINER` ``	Recordings container name to use in the Annex section of Transaction object in CME	`recordings`	A valid container name	
`env.TENANT_KV_FOLDER`	KV tenants folder name in Consul	`tenants`	A valid folder name	
`env.TENANT_CONFIGMAP_FOLDER`	Local folder to mount tenant config files	`/etc/config`	A valid folder path	Should not be changed
`env.SMTP_SERVER`	SMTP server URI	`smtp-relay.smtp.svc.cluster.local`	A valid URI	
`secrets.imagePull`	Secret names used for container image pulls	`pureengage-docker-dev`, `pureengage-docker-staging`	Valid secret names	
`secrets.configServer.k8s`	Whether k8s secret is used or csi/vault is used for CS	`true`	`true` or `false`	
`secrets.configServer.k8sSecretName`	K8s secret name for Config Server	`configserver-secret`	Valid k8s secret name	
`secrets.configServer.k8sUserKey`	K8s secret user key for Config Server	`username`	Valid user name	
`secrets.configServer.k8sPasswordKey`	K8s secret password key for Config Server	`password`	Valid password	
`secrets.configServer.vaultSecretName`	Vault secret name for Config Server	`/configserver-secret`	Valid valult secret name	
`secrets.configServer.vaultUserKey`	Vault secret user key for Config Server	`configserver-username`	Valid user name	
`secrets.configServer.vaultPasswordKey`	Vault secret password key for Config Server	`configserver-password`	Valid password	
`secrets.consul.k8s`	Whether k8s secret is used or csi/vault is used for Consul	`true`	`true` or `false`	

Parameter	Description	Default Value	Valid Values	Notes
`secrets.consul.k8sTokenName`	K8s token name for Consul	`shared-consul-consul-gvp-token`	Valid k8s token name	
`secrets.consul.k8sTokenKey`	K8s token key for Consul	`consul-consul-gvp-token`	Valid k8s token key	
`secrets.consul.vaultSecretName`	Vault secret name for Consul	`/consul-secret`	Valid k8s secret name	
`secrets.consul.vaultSecretKey`	Vault secret key for Consul	`consul-consul-gvp-token`	Valid k8s secret key	
`secrets.gtts.k8s`	Whether k8s secret is used or csi/vault is used for GTTS	`false`	`true` or `false`	
`secrets.gtts.k8sSecretName`	K8s secret name for GTTS	`gtts-secret`	Valid k8s secret name	
`secrets.gtts.EncryptedKey`	K8s encrypted key for GTTS	`encrypted-key`	Valid k8s encrypted key	
`secrets.gtts.PasswordKey`	K8s secret password key for GTTS	`password`	Valid password key	
`resources.requests.memory`	Guaranteed memory allocation for SD container	`128Mi`	A valid k8s memory size value	
`resources.requests.cpu`	Guaranteed CPU allocation for SD container	`100m`	A valid k8s CPU unit value	
`resources.limits.memory`	Maximum amount of Memory K8s allocates for SD container	`2Gi`	A valid k8s memory size value	
`resources.limits.cpu`	Maximum amount of CPU K8s allocates for SD container	`1000m`	A valid k8s CPU unit value	
`podSecurityContext.fsGroup`	Pod SecurityContext GID for volumes	`500`	A valid GID number	
`podSecurityContext.runAsUser`	Pod SecurityContext user ID for processes	`500`	A valid UID number	
`podSecurityContext.runAsGroup`	Pod SecurityContext primary group ID for processes	`500`	A valid GID number	
`podSecurityContext.runAsNonRoot`	Pod SecurityContext flag to require running as non-	`true`	`true` or `false`	

Parameter	Description	Default Value	Valid Values	Notes
	root			
`securityContext.*`	Same as podSecurityContext settings above but for containers	(See above)		
`priorityClassName`	K8s priorityClassName for the pod (optional parameter, but recommended)	(none)	Valid priority class name	
`affinity`	Affinity for pod assignment (optional parameter)	(none)	Valid affinity settings	
`nodeSelector`	Node selector labels	(none)	Valid nodeSelector settings	
`tolerations`	Settings for tolerations	(none)	Valid name:value pairs	
`prometheus.enabled`	Whether to enable Prometheus scraping or not	`true`	`true` or `false`	
`prometheus.podMonitor.enabled`	Whether to enable Prometheus podMonitor rules or not	`true`	`true` or `false`	
`networkPolicies.enabled`	Whether to enable network policies or not	`false`	`true` or `false`	
`dnsConfig.options.ndots`	Required number of dots in an FQDN to trigger a DNS query	`3`	A valid number of dots	

Reporting Server

The following tables list the configurable parameters of the GVP Reporting Server chart and their default values.

Parameter	Description	Default	Valid values	Notes
`replicaCount`	Number of replicas in deployment	`1`		
`image.gvprsrepository`	Image repository location for RS container	pureengage-docker-staging.jfrog.io/gvp/gvp_rs		

Parameter	Description	Default	Valid values	Notes
`image.snmprepository`	Image repository location for snmp container	pureengage-docker-staging.jfrog.io/gvp/multicloud/gvp_snmp		
`image.dbinitrepository`	Image repository location for RS init container	pureengage-docker-staging.jfrog.io/gvp/gvp_rs_init		
`imagePullSecrets.name`	Image pull secret	`pureengage-docker-staging`		
`namespace`	Namespace where the component is deployed	`gvp`	Per environment	
`prometheus.metric.port`	SNMP exporter port at which Prometheus would scrape data	9116	Same as default/ fixed	
`service.restapiport`	The RS rest API port	8080	Standard Port range : 1 to 65535. Changing this would require RS application configuration changes.	
`service.activemqport`	The RS ActiveMQ port	61616	Standard Port range : 1 to 65535. Changing this would require RS application configuration changes.	
`service.envinjectport`	Port used for reading secret details	443	Fixed	
`service.dnsport`	Port used for DNS resolution	53	Per environment	
`service.configserverport`	The port of the GVP Configuration Server	8888	Per environment	
`service.snmpport`	The SNMP container port to which RS sends mib data	1705	Same as default/ fixed	
`livenessValues.initialDelaySeconds`	Initial delay before the first liveness probe	30	Same as default/ fixed. It is not advisable to change the values unless instructed by Genesys.	

Parameter	Description	Default	Valid values	Notes
`livenessValues.periodSeconds`	The interval between each liveness probe request	120	Same as default/ fixed. It is not advisable to change the values unless instructed by Genesys.	
`livenessValues.timeoutSeconds`	The timeout for each liveness probe request	3	Same as default/ fixed. It is not advisable to change the values unless instructed by Genesys.	
`livenessValues.failureThreshold`	Threshold for liveness check failure and container restart	3	Same as default/ fixed. It is not advisable to change the values unless instructed by Genesys.	
`readinessValues.path`	Readiness probe URI path	/ems-rs/readiness	Same as default/ fixed. It is not advisable to change the values unless instructed by Genesys.	
`readinessValues.initialDelaySeconds`	Initial delay before the first readiness probe	10	Same as default/ fixed. It is not advisable to change the values unless instructed by Genesys.	
`readinessValues.periodSeconds`	The interval between each readiness probe request	60	Same as default/ fixed. It is not advisable to change the values unless instructed by Genesys.	
`readinessValues.timeoutSeconds`	The timeout for each readiness probe request	3	Same as default/ fixed. It is not advisable to change the values unless instructed by Genesys.	
`readinessValues.failureThreshold`	Threshold for readiness check failure	3	Same as default/ fixed. It is not advisable to change the values unless instructed by Genesys.	
`pvc.activemqAndLocalConfigPath`	ActiveMQ and local config cache path	/billing/gvp-rs	Same as default/ fixed. It is not advisable to change the values	

Parameter	Description	Default	Valid values	Notes
			unless instructed by Genesys.	
`resourceForRS.requests.memory`	Guaranteed Memory allocation for RS container	4Gi		
`resourceForRS.requests.cpu`	Guaranteed CPU allocation for RS container	600m		
`resourceForRS.limits.memory`	Maximum amount of Memory K8s allocates for RS container	13Gi		
`resourceForRS.limits.cpu`	Maximum amount of CPU K8s allocates for RS container	1000m		
`resoueceForSnmp.requests.memory`	Guaranteed Memory allocation for snmp container	500Mi		
`resoueceForSnmp.requests.cpu`	Guaranteed CPU allocation for snmp container	1000m		
`resoueceForSnmp.limits.memory`	Maximum amount of Memory K8s allocates for snmp container	8Gi		
`resoueceForSnmp.limits.cpu`	Maximum amount of CPU K8s allocates for snmp container	1500m		
`env.CFGAPP`	The GVP configuration Server application name	default	Per environment	
`env.GVP_RS_SERVICE_HOSTNAME`	Hostname of the RS	gvp-rs.gvp.svc.cluster.local	Per environment - eg : .gvp.svc.cluster.local	
`env.CFG_HOST`	The Hostname of the GVP Configuration Server	gvp-configserver.gvp.svc.cluster.local	Per environment	
`env.CFG_PORT`	The port of the GVP Configuration Server	8888	Per environment	
`env.CMDLINE`	Command line set in the Start info tab of RS application	./rs_startup.sh	Same as default / fixed	
`env.DBNAME`	The name of the	gvp-rs	Per environment	

Parameter	Description	Default	Valid values	Notes
	RS DB to be used			
env.DBUSER	RSDB user which has read and write access on the DB	mssql	Per environment	
env.rsDbSharedUsername	RSDB username which has read-only access on the DB	mssqlreader	Per environment	
env.DBPORT	RS SQl server DB port	1433	Per environment	
env.ENVTYPE	Environment on which RS is going to be deployed	""		
env.GenesysIURegion	The region on which RS is going to be deployed	""		
env.localconfigcachepath	The local GVP Configuration Server cache path	/billing/gvp-rs/data/cache	. Changing this would require RS application configuration changes.	
env.HOSTFOLDER	The RS host object folder in GVP Configuration Server	Hosts		
env.HOSTOS	The OS of the Host on which RS is going to be deployed	CFGRedHatLinux	fixed	
env.LCAPORT	The LCA port to be set on the RS Host object	4999		
env.MSSQLHOST	The Hostname of the MSSQL server which contains the RS DB	mssql00-gvp-dev-westus2.privatelink.database.windows.net	mssql00-gvp--.privatelink.database.windows.net	
env.RSAPP	The name of the RS application object	azure_rs		
env.RSJVM_INITIALHEAPSIZE	The initial and minimum JVM heap size	500m	Per environment	
env.RSJVM_MAXHEAPSIZE	The maximum JVM heap size	1536m	Per environment	
env.RSFOLDER	The RS application folder in GVP Configuration Server	Applications		

Parameter	Description	Default	Valid values	Notes
env.RS_VERSION	The RS version to be set in the RS application	9.0.032.22		
env.STDOUT	Determines whether init container writes console logs or not	true		
env.WRKDIR	The RS working directory to be set on the RS application	/usr/local/genesys/rs/	. Changing this would require RS application configuration changes.	
env.SNMPAPP	The RS SNMP application name	azure_rs_snmp		
env.SNMP_WORKDIR	The SNMP working directory to be set on the SNMP application	/usr/sbin		
env.SNMP_CMDLINE	The SNMP command line to be set in the snmp application	snmpd	Same as default/ fixed	
env.SNMPFOLDER	The snmp application folder in GVP Configuration Server	Applications		

Reporting Server configuration options can be modified using values.yaml in the section **RS config**. For example, as below:

```
RSCONFIG:
  log:
    verbose: "trace"
    trace: "stdout"
```

Genesys recommends using the below configuration option for RS in values.yaml:

```
RSCONFIG:
  persistence:
    rs.storage.metricsfilter: "0-15,25,36,52-55,74,136-141,148-151"
```

Resource Manager

The following tables list the configurable parameters of the GVP RM chart and their default values.

Note: There is a typo in names of a few parameters in the table. "resource" is spelt "resouece". Regardless of the typo, the parameters function correctly. The typo will be corrected in a future version.

Parameter	Description	Default	Valid values	Notes
`replicaCount`	Number of replicas in deployment	`2`		
`image.gvprmrepository`	Image repository location for RM container	pureengage-docker-staging.jfrog.io/gvp/gvp_rm		
`image.cfghandlerrepository`	Image repository location for cfghandler container	pureengage-docker-staging.jfrog.io/gvp/gvp_rm_cfghandler		
`image.snmprepository`	Image repository location for snmp container	pureengage-docker-staging.jfrog.io/gvp/multicloud/gvp_snmp		
`image.gvprmtestrepository`	Image repository location for gvprmtest container	pureengage-docker-staging.jfrog.io/gvp/gvp_rm_test		
`image.pullSecret`	Image pull secret	`pureengage-docker-staging`		
`namespace`	Namespace where the component is deployed	`gvp`	As per Environment	
`prometheus.metric.port`	SNMP exporter port at which Prometheus would scrape data	9116	same as default/ fixed	
`prometheus.log.port`	Log exporter port at which Prometheus would scrape data	8200		
`service.port`	RM service port	5060	Changing this would require RM application configuration changes.	
`service.rmHealthCheckAPIPort`	Node JS server port for liveness and readiness probes	8300		
`livenessValues.path`	Liveness probe URI path	/rm/liveness	Same as default/ fixed. The values are not advisable to be changed unless instructed by genesys.	
`livenessValues.initialDelaySeconds`	Initial delay before the first liveness	60	Same as default/ fixed. The values	

Parameter	Description	Default	Valid values	Notes
	probe		are not advisable to be changed unless instructed by genesys.	
`livenessValues.periodSeconds`	The interval between each liveness probe request	90	Same as default/ fixed. The values are not advisable to be changed unless instructed by genesys.	
`livenessValues.timeoutSeconds`	The timeout for each liveness probe request	20	Same as default/ fixed. The values are not advisable to be changed unless instructed by genesys.	
`livenessValues.failureThreshold`	Threshold for liveness check failure and container restart	3	Same as default/ fixed. The values are not advisable to be changed unless instructed by genesys.	
`readinessValues.path`	Readiness probe URI path	/rm/readiness	Same as default/ fixed. The values are not advisable to be changed unless instructed by genesys.	
`readinessValues.initialDelaySeconds`	Initial delay before the first readiness probe	10	Same as default/ fixed. The values are not advisable to be changed unless instructed by genesys.	
`readinessValues.periodSeconds`	The interval between each readiness probe request	60	Same as default/ fixed. The values are not advisable to be changed unless instructed by genesys.	
`readinessValues.timeoutSeconds`	The timeout for each readiness probe request	20	Same as default/ fixed. The values are not advisable to be changed unless instructed by genesys.	
`readinessValues.failureThreshold`	Threshold for readiness check failure	3	Same as default/ fixed. The values are not advisable to be changed unless instructed by genesys.	

Parameter	Description	Default	Valid values	Notes
env.cfghandler.CFGSERVER	Primary Configuration Server	gvp-configserver.gvp.svc.cluster.local	Per environment	
env.cfghandler.CFGSERVERBACKUP	Backup Configuration Server	gvp-configserver.gvp.svc.cluster.local	Per environment	
env.cfghandler.CFGPORT	Configuration Server port	8888	Per environment	
env.cfghandler.CFGAPP	Configuration Server name for login	default	Per environment	
env.cfghandler.RMAPP	Prefix to RM application name	azure_rm		
env.cfghandler.RMFOLDER	RM application folder in Configuration Server	Applications\ RM_MicroService\ RM_Apps		
env.cfghandler.HOSTFOLDER	RM host folder in Configuration Server	Hosts\ RM_MicroService		
env.cfghandler.MCPFOLDER	MCP CU and LRG folder to which RM points to	MCP_Configuration_Unit\ MCP_LRG		
env.cfghandler.SNMPFOLDER	SNMP application folder in Configuraion Server	Applications\ RM_MicroService\ SNMP_Apps		
env.cfghandler.Environment	Environment on which RM is going to be deployed	prod		
env.cfghandler.CONFSERVERAPP	Configuration Server application to be tied to RM	confserv	Per environment	
env.cfghandler.RSAPP	RS application to be tied to RM	azure_rs		
env.cfghandler.SNMPAPP	SNMP application name prefix to be tied to RM	azure_rm_snmp		
env.cfghandler.STDOUT	Determines whether init container writes console logs or not	true		
env.cfghandler.VOICEMAILSERVICEIDNUMBER	voicemail number configuration for tenant creation	5551111		
resourceForRM.requestsmemory	Guaranteed memory allocation for RM container	4Gi		

Parameter	Description	Default	Valid values	Notes
`resourceForRM.requests.cpu`	Guaranteed CPU allocation for RM container	2000m		
`resourceForRM.limits.memory`	Maximum amount of Memory K8s allocates for RM container	14Gi		
`resourceForRM.limits.cpu`	Maximum amount of CPU K8s allocates for RM container	4000m		
`resoueceForSnmp.requests.memory`	Guaranteed memory allocation for snmp container	500Mi		
`resoueceForSnmp.requests.cpu`	Guaranteed CPU allocation for snmp container	1000m		
`resoueceForSnmp.limits.memory`	Maximum amount of Memory K8s allocates for snmp container	8Gi		
`resoueceForSnmp.limits.cpu`	Maximum amount of CPU K8s allocates for snmp container	1500m		
`fluentBitSidecar.enabled`	Enable/disable FluentBit sidecar for console logging	`false`	`true` or `false`	Warning: The fluent-bit sidecar feature is being provided as-is w/o support and requires a third-party container image that Genesys does not provide or support.
`rmLogStorage.volumeType.persistentVolumeEnabled`	Enable/disable Persistent Volume (PV) for logging	`false`	`true` or `false`	
`rmLogStorage.volumeType.persistentVolumeClaim.storageClass`	Storage class for PV claim of log storage	`disk-premium`	A valid PVC storage class name	
`rmLogStorage.volumeType.persistentVolumeClaim.accessMode`	Log storage PVC access mode	`ReadWriteOnce`	A valid PVC storage access mode	
`rmLogStorage.volumeType.persistentVolumeClaim.claimSize`	Size of log storage of PVC type	`50Gi`	A valid storage size	
`rmLogStorage.volumeType.hostPath.enabled`	Enable/disable hostPath for logging	`true`	`true` or `false`	

Parameter	Description	Default	Valid values	Notes
`rmLogStorage.volumeType.hostPath.path`	Enable/disable logging	`true`	`true` or `false`	
`rmLogStorage.volumeType.hostPath.path`	Log storage host path for RM logs	`/mnt/log`	A valid storage path	Do not change, as other resources may depend on this.
`rmLogStorage.volumeType.emptyDir.enal`	Enable/disable logging	`false`	`true` or `false`	

Resource Manager configuration options can be modified using values.yaml in the section **RM config**. For example, as below:

```
RMCONFIG:
   log:
      verbose: "trace"
```

Media Control Platform

The following tables list the configurable parameters of the GVP MCP chart, their default values, etc.

Note:

- There may be other parameters that could be changed, but changing them is not recommended, unless you know exactly what you are doing.
- You must set mcp.sipPort to 5070. This limitation will be addressed in a future release.
- All the necessary overrides exist allowing for integration with Nuance V11 running on VMs. Using overrides in this manner requires network connectivity between the Kubernetes cluster and host(s) running Nuance. For more details on this integration option, refer to Nuance 11 GVP case study.

Parameter	Description	Default Value	Valid Values	Notes
`podLabels`	Custom labels to be added for each pod	(none)	Valid set of labels as "name: value"	
`podAnnotations`	Custom annotations to be added for each pod	(none)	Valid set of annotations as "name: value"	
`deploymentEnv`	Name of environment - should start with dev, stage, or prod	"UPDATE_ENV" (dummy)	A valid environment name	The prefix is used to determine the configuration map to use. It may also be used by the deployment

Parameter	Description	Default Value	Valid Values	Notes
				pipeline.
`replicaCount`	Number of minimum replicas required in the deployment 2	A number > 0	2 or higher is recommended for HPA or rolling upgrade with podDisruptionBudget	
`podManagementPolicy`	Pod Management Policy for Launch/ Termination	`Parallel`	`Parallel` or `OrderedReady`	
`terminationGracePeriod`	Time to wait for graceful shutdown	`3600 seconds`	A valid time value	
`dashboardReplicaStatefulSetFilterOverride`	This value is used to filter MCP replica sets or StatefulSets in dashboards. The default is the full Chart name.	`(mcp)`	A valid filter string	
`partOf`	Namespace where the application is deployed	`gvp`	A string value	
`image..registry`	Image registry URL for the container	`genesysengagedev.azurecr.io`	A valid registry URI	
`image..repository`	Repository path for the container in the above registry	`gvp/multicloud/gvp_mcp`, `gvp/multicloud/gvp_mcp_servicehandler`, `gvp/multicloud/gvp_mcp_confighandler`, `gvp/multicloud/gvp_snmp`, or `cce/recording-provider`	A valid repository path/name	
`image..pullPolicy`	Pull policy for the container	`IfNotPresent`	`IfNotPresent` or `Always`	
`image..tag`	Container version tag	`{{.Chart.AppVersion }}`, or a particular tag for some containers - see values.yaml	A valid tag string	
`mcp.sipPort`	SIP port used by MCP	`5070`	A valid port number	This has to match MCP configuration
`mcp.sipProtocol`	SIP transport Protocol used by MCP	`TCP`	A valid transport protocol	Has to match MCP configuration
`mcp.logicalResourceGroup`	Name of the LRG in the Config Server	`MCP_Configuration_Unit_Test`	A valid LRG name	
`mcp.alerts.*`	Threshold value	A number	A number > 0	This value may

Parameter	Description	Default Value	Valid Values	Notes
	for each MCP alert			require tuning
`mcpsnmp.logPrefixName`	Prefix of the MCP log files to query, should be MCP	`MCP`	A valid MCP log file prefix	
`rup.rupVolume.storageClass`	Storage class for PV Claims of RUP volume	`disk-premium`	A valid PVC storage class name	
`rup.rupVolume.accessModes`	PVC access mode of RUP volume	`ReadWriteOnce`	A valid PVC storage access mode	
`rup.rupVolume.volumeSize`	Size of RUP volume	`40Gi`	A valid storage size	
`rup.recordingsFolder`	RUP recordings folder	`/pvolume/recordings`	Recordings folder used MCP and RUP	Do not change
`rup.recordingsCache`	Recordings cache folder, where MCP creates recordings files	`/pvolume/recording_cache`	Recordings cache folder used MCP and RUP	Do not change
`rup.decommisionDestType`	Destination type for RUP recordings on decommision/ shutdown	`WebDAV`	A valid destination type	
`rup.decommisionDestWebdavUrl`	WebDAV destination URL to be used on decommision	`http://gvp-central-rup:8180`	A valid URL	
`rup.diskFullDestType`	Destination type for RUP recordings on disk-full condition	`WebDAV`	A valid destination type	
`rup.diskFullDestWebdavUrl`	WebDAV destination URL to be used on disk-full condition	`http://gvp-central-rup:8180`	A valid URL	
`rup.cpUrl`	RUP Conversation Provider URL	`http://cce-conversation-provider.cce.svc.cluster.local`	A valid URL	
`rup.unrecoverableLostAction`	Action for unrecoverable/lost recordings	`uploadtodefault`	A valid action type	
`rup.unrecoverableDestType`	Destination type for RUP unrecoverable/lost recordings	`Azure`	A valid destination type	
`rup.unrecoverableDestAzureAccountName`	Account name for RUP lost recordings for Azure dest type	`gvpwestus2dev`	Azure account name	

Parameter	Description	Default Value	Valid Values	Notes
`rup.unrecoverableDestAzureContainerNa`	Container name for RUP lost recordings for Azure dest type	`ccerp-unrecoverable`	A container name	
`rup.logJsonEnable`	Enable RUP logging in Json format	`true`	`true` or `false`	
`rup.logLevel`	Log level for RUP file logging	`INFO`	A valid Java log level string	
`rup.logConsoleLevel`	Log level for RUP console logging	`INFO`	A valid Java log level string	
`rup.resources.requests.memory`	Guaranteed memory allocation for RUP container	`120Mi`	A valid k8s memory size value	
`rup.resources.requests.cpu`	Guaranteed CPU allocation for RUP container	`100m`	A valid k8s CPU unit value	
`rup.resources.requests.ephemeral-storage`	Ephemeral storage size for RUP container	`10Gi`	A valid k8s storage size	
`rup.resources.limits.memory`	Maximum amount of Memory K8s allocates for RUP container	`2Gi`	A valid k8s memory size value	
`rup.resources.limits.cpu`	Maximum amount of CPU K8s allocates for RUP container	`1000m`	A valid k8s CPU unit value	
`recordingStorage.storageClass`	Storage class for PV claim of recording storage	`disk-preminum`	A valid PVC storage class name	
`recordingStorage.accessModes`	PVC access mode of recording storage	`ReadWriteOnce`	A valid PVC storage access mode	
`recordingStorage.volumeSize`	Size of recording volume of PVC type	`40Gi`	A valid storage size	
`fluentBitSidecar.enabled`	Enable/disable FluentBit sidecar for console logging	`false`	`true` or `false`	Warning: The fluent-bit sidecar feature is being provided as-is w/o support and requires a third-party container image that Genesys does not provide or support.
`mcpLogStorage.volumeType.persistentV`	Enable/disable Persistent Volume	`false.enabled`	`true` or `false`	

Parameter	Description	Default Value	Valid Values	Notes
	(PV) for logging			
`mcpLogStorage.volumeType.persistentVo`	Storage class for PVC type persistent storage	`disk-premium`	A valid PVC storage class name	
`mcpLogStorage.volumeType.persistentVo`	Log storage PVC access mode	`ReadWriteOnce`	A valid PVC storage access mode	
`mcpLogStorage.volumeType.persistentVo`	Size of log storage of PVC type	`50Gi`	A valid storage size	
`mcpLogStorage.volumeType.hostPath.ena`	Enable/disable hostPath for logging	`true`	`true` or `false`	
`mcpLogStorage.volumeType.hostPath.pa`	Log storage host path for MCP logs	`/mnt/log`	A valid storage path	Do not change, as other resources may depend on this
`mcpLogStorage.volumeType.emptyDir.en`	Enable/disable emptyDir for logging	`false`	`true` or `false`	
`serviceHandler.serviceHandlerPort`	Port of serviceHandler	`8300`	A valid port	Do not change, as other resources may depend on this
`serviceHandler.consulExternalHost`	External Consul host name, if used	``	A valid host FQDN or address	
`serviceHandler.consulPort`	Consul service port	`8501`	A valid port	Do not change, as other resources may depend on this
`serviceHandler.registrationInterval`	Consul registration interval in ms	`10000`	Time in milliseconds	
`serviceHandler.mcpHealthCheckInterva`	Consul Health Check interval	`30s`	A time value	
`serviceHandler.mcpHealthCheckTimeout`	Consul Health Check timeout	`10s`	A time value	
`configServer.host`	GVP Config Server host name	`gvp-configserver`	Name of Config Server	
`configServer.port`	GVP Config Server port number	`8888`	Service port of Config Server	
`configServer.app`	GVP Config Server application name	`default`	A valid Config Server application name	
`secrets.imagePull`	Secret names used for container image pulls	`pureengage-docker-dev`, `pureengage-docker-staging`	Valid secret names	

Parameter	Description	Default Value	Valid Values	Notes
secrets.configServer.k8s	Whether k8s secret is used or csi/vault is used for CS	true	true or false	
secrets.configServer.secretName	K8s secret name for Config Server	configserver-secret	A valid secret name	
secrets.configServer.userKey	K8s secret DB user key for Config Server	username	A valid user key	
secrets.configServer.passwordKey	K8s secret DB password key for Config Server	password	A valid password key	
secrets.configServer.csiSecretProviderClass	CSI secret provider class for Config Server	keyvault-gvp-gvp-acs-configserver-secret	A valid csiSecretProviderClass attribute string	
secrets.consul.k8s	Whether k8s secret is used or csi/vault is used for Consul	true	true or false	
secrets.consul.secretName	K8s secret name for Consul	shared-consul-consul-gvp-token	A valid secret name	
secrets.consul.secretKey	K8s secret key for Consul	consul-consul-gvp-token	A valid secret key	
secrets.consul.csiSecretProviderClass	CSI secret provider class for Consul	keyvault-consul-consul-gvp-token	A valid csiSecretProviderClass attribute string	
resourcesMcp.requests.memory	Guaranteed memory allocation for MCP container	500Mi	A valid k8s memory size value	
resourcesMcp.requests.cpu	Guaranteed CPU allocation for MCP container	250m	A valid k8s CPU unit value	
resourcesMcp.requests.ephemeral-storage	Ephemeral storage size for MCP container	10Gi	A valid storage size	
resourcesMcp.limits.memory	Maximum amount of Memory K8s allocates for MCP container	8Gi	A valid k8s memory size value	
resourcesMcp.limits.cpu	Maximum amount of CPU K8s allocates for MCP container	4000m	A valid k8s CPU unit value	
resourcesDefault.requests.memory	Guaranteed memory allocation for other/default container	120Mi	A valid k8s memory size value	

Parameter	Description	Default Value	Valid Values	Notes
`resourcesDefault.requests.cpu`	Guaranteed CPU allocation for default container	`100m`	A valid k8s CPU unit value	
`resourcesDefault.limits.memory`	Maximum amount of Memory K8s allocates for default container	`128Mi`	A valid k8s memory size value	
`resourcesDefault.limits.cpu`	Maximum amount of CPU K8s allocates for default container	`100m`	A valid k8s CPU unit value	
`podSecurityContext.fsGroup`	Pod SecurityContext GID for volumes	`500`	A valid GID number	
`podSecurityContext.runAsUser`	Pod SecurityContext User ID for processes	`500`	A valid UID number	
`podSecurityContext.runAsGroup`	Pod SecurityContext primary Group ID for processes	`500`	A valid GID number	
`podSecurityContext.runAsNonRoot`	Pod SecurityContext flag to require running as non-root	`true`	`true` or `false`	
`securityContext.*`	Same as podSecurityContext settings above but for containers	(See above)		
`priorityClassName`	K8s priorityClassName for the pod (optional parameter, but recommended)	(none)	A valid priority class name	
`affinity`	Affinity for pod assignment (optional parameter)	`(none)`	Valid affinity settings	
`nodeSelector`	Node selector labels	`genesysengage.com/nodepool: realtime`	Valid nodeSelector settings	
`tolerations`	Settings for tolerations	`key: "k8s.genesysengage.com/nodepool", operator: Exists, effect: NoSchedule`	Valid name:value pairs	

Parameter	Description	Default Value	Valid Values	Notes
`hpa.enabled`	A 'true' value enables HPA (HPA needs to be enabled when Keda is enabled)	`true`	`true` or `false`	
`hpa.maxReplicas`	Number of maximum replicas	`4`	A number >= replicaCount	
`hpa.targetCPUAverageUtilization`	CPU utilization target value in percentage	`20`	A number 1 to 100	
`hpa.scaleupPeriod`	Time period in seconds to use with "scaleupPods" as well as "scaleupPercent"	`15`	A number >= 1	
`hpa.scaleupPercent`	Maximum percentage of replicas allowed to be scaled up in "scaleupPeriod"	`50`	A number 1 to 100	
`hpa.scaleupPods`	Maximum number of replicas allowed to be scaled up in "scaleupPeriod"	`4`	A number >= 1	
`hpa.scaleupStabilizationWindow`	Stabilization window in seconds for scale-ups	`0`	A number >= 0	`0` is recommended
`hpa.scaleupPolicy`	Scale-up policy allowing minimum or maximum change in replica count	`Max`	`Min`, `Max`, or `Disabled`	
`hpa.scaledownPeriod`	Time period in seconds to use with "scaledownPods" as well as "scaledownPercent"	`300`	A number >= 1	
`hpa.scaledownPercent`	Maximum percentage of replicas allowed to be scaled down in "scaledownPeriod"	`10`	A number 1 to 100	
`hpa.scaledownPods`	Maximum number of replicas allowed to be scaled down in "scaledownPeriod"	`2`	A number >= 1	
`hpa.scaledownStabilizationWindow`	Stabilization window in seconds	`3600`	A number >= 0	

Parameter	Description	Default Value	Valid Values	Notes
	for scale-downs			
`hpa.scaledownPolicy`	Scale-down policy allowing minimum or maximum change in replica count	`Min`	`Min`, `Max`, or `Disabled`	
`hpa.keda.enabled`	Whether to enable Keda with HPA for auto scaling or not.	`true`	`true` or `false`	HPA must be enabled when Keda is enabled
`hpa.keda.preScaleStart`	Cron schedule for starting Keda scaling	`0 14 * * *`	A valid string with cron schedule	
`hpa.keda.preScaleEnd`	Cron schedule for ending Keda scaling	`0 2 * * *`	A valid string with cron schedule	
`hpa.keda.preScaleDesiredReplica`	Number of desired replicas for Keda scaling	`4`	A valid number > 1	
`hpa.keda.pollingInterval`	Interval for polling Kedа trigger source, in seconds	`15`	Number of seconds	
`hpa.keda.cooldownPeriod`	Seconds to wait after last trigger before scaling down	`300`	Number of seconds	
`prometheus.enabled`	Whether to enable Prometheus scraping or not	`true`	`true` or `false`	
`prometheus.podMonitor.enabled`	Whether to enable Prometheus podMonitor rules or not	`true`	`true` or `false`	
`grafana.enabled`	Whether to enable Grafana dashboard or not	`false`	`true` or `false`	
`podDisruptionBudget.enabled`	Enable or disable podDisruptionBudget	`true`	`true` or `false`	
`podDisruptionBudget.maxUnavailable`	Maximum number of pods that can be unavailable during a rolling update	`1`	A number >= 1	
`networkPolicies.enabled`	Whether to enable network policies or not	`false`	`true` or `false`	
`dnsConfig.options.ndots`	Required number of dots in an FQDN to trigger a DNS	`3`	A valid number of dots	

Parameter	Description	Default Value	Valid Values	Notes
	query			

The following chart parameters are MCP configurable options in the ConfigMap that can be overridden.

Parameter	Description	Default Value	Valid Values	Notes
`mcpConfig.mcp.ems.default.metrics`	Specifies the default filter for metrics to be delivered to the Reporting Server for Call Events reporting.	`0-15,25,36,52-55,74,136-141,148-151`	Comma separated list of metric values or ranges. A metric value must be between 0 and 151 inclusive. The values * and blank are also allowed.	
`mcpConfig.mcp.ems.dc.enableSQA`	This flag determines if the Data Collection Sink is configured to perform Service Quality analysis.	`false`	True or false	
`mcpConfig.mcp.mpc.dispatchthreads`	Number of media dispatching threads	`4`	A number >= 1	2 or more matching CPU cores is recommended
`mcpConfig.mcp.log.verbose`	MCP log verbose level	`interaction`	`all`, `debug`, `trace`, `interaction`, or `standard`	
`mcpConfig.mcp.mpc.codec`	List of codecs to support	`pcmu pcma telephone-event`	Any combination of: pcmu, pcma, g722, opus, g726, g729, gsm, amr, amr-wb, tfci, h263, h263-1998, h264, vp8 or telephone-event	
`mcpConfig.mcp.mpc.transcoders`	List of trancoders to enable	`PCM MP3`	Any combination of: G722, GSM, G726, G729, AMR, AMR-WB, MP3, OPUS, H263, H264 and VP8, or "none"	
`mcpConfig.mcp.mpc.playcache.enable`	Enable or disable media play cache	`1`	`0` or `1`	0 - disable, 1 - enable
`mcpConfig.mcp.fm.http_proxy`	Address of HTTP proxy to be used	(none)	A valid address, possibly with port	
`mcpConfig.mcp.fm.https_proxy`	Address of HTTPS proxy to be used	(none)	A valid address, possibly with port	
`mcpConfig.nexus_asr1_pool.size`	Size of Nexus ASR1 pool	0	`0` or a positive number	A `0` value disables this

Parameter	Description	Default Value	Valid Values	Notes
				resource pool
mcpConfig.nexus_asr1.provision.vrm.client.resource.uri	A list/array of URIs within square brackets delimited by commas	[\"ws://nexus-production.nexus.svc.cluster.local/nexus/v3/bot/connection\"]	A valid list of URIs	
mcpConfig.nexus_asr1.provision.vrm.client.resource.type	The type of resource	ASR	ASR	Do not change
mcpConfig.nexus_asr1.provision.vrm.client.resource.name	The name of the resource	ASRC	ASRC	Do not change
mcpConfig.nexus_asr1.provision.vrm.client.resource.engine	A list of engine names	nexus	nexus	This name is used by the application
mcpConfig.nexus_asr1.provision.vrm.client.resource.engine.nexus.audio.codec	Name of the audio codec used	mulaw	mulaw	Only "mulaw" supported currently
mcpConfig.nexus_asr1.provision.vrm.client.resource.engine.nexus.audio.samplerate	Audio sampling rate	8000	8000	Do not change
mcpConfig.nexus_asr1.provision.vrm.client.resource.engine.nexus.enableMaxSpeechTimeout	Whether to enable `maxspeechtimeout` or not	true	`true`, `false`	Do not change
mcpConfig.nexus_asr1.provision.vrm.client.resource.engine.nexus.serviceAccountKey	The string value of service account key	(none)	A valid key, or empty if not needed	
mcpConfig.nexus_asr1.provision.vrm.client.resource.privateKey	The private key to use with the resource, if needed	(none)	A valid key	
mcpConfig.nexus_asr1.provision.vrm.client.resource.proxy	The proxy URI to use	(none)	A valid URI	
mcpConfig.nexus_asr1.provision.vrm.client.resource.transportProtocol	Transport protocol used with the server	WEBSOCKET	WEBSOCKET	Do not change

Configuring MCPs to connect to Nexus (for Bot use cases)

- Set one or more URIs for Nexus service using parameter `mcpConfig.nexus_asr1.provision.vrm.client.resource.uri`, separated by commas.
- Set the value of `mcpConfig.nexus_asr1.pool.size` to the number of Nexus service URIs in the previous step.

Note: To test the connectivity between the MCP's network and the Nexus service, you could use a program like 'curl'. Though the service would return a 404 error for the HTTP request, this would tell you if a connection could be made between the two. Here is an example of a curl request to Nexus:

```
$ curl -v https://nex-2.genhtcc.com/api/v1beta/asrConnector/openConnection
...
* Connected to nex-2.genhtcc.com (xx.yyy.zzz.172) port 443 (#0)
...
> GET /api/v1beta/asrConnector/openConnection HTTP/2
> Host: nex-2.genhtcc.com
```

```
> User-Agent: curl/7.61.1
> Accept: */*
>
* Connection state changed (MAX_CONCURRENT_STREAMS == 128)!
```

Configure Kubernetes

Media Control Platform

Secrets

Secrets used by MCP:

Secret Name	Secret key	Type of secret	Notes
secrets.configServer.secretName	secrets.configServer.dbUserKey	Kubernetes	GVP Configuration Server username
	secrets.configServer.dbPasswordKey	Kubernetes	GVP Configuration Server password
secrets.consul.secretName	secrets.consul.secretKey	Kubernetes	Consul token

ConfigMaps

The following ConfigMaps are used:

ConfigMap	Used by	Notes
-mcp-config	confighandler, gvp-mcp	Template for MCP configuration
-log-alerts-dashboard	Grafana	Dashboard for MCP alerts based on parsing MCP logs
-utilization-dashboard	Grafana	Dashboard for MCP utilization

Resource Manager

ConfigMaps

ConfigMap	Used by	Notes
-configmap	cfghandler, gvp-rm	Template for RM configuration

Secrets

Secret Name	Secret key	Type of secret	Notes
secrets.configserverProviderClassName	cfgSecretFileNameForCfgUsername	Kubernetes	GVP Configuration Server username
	cfgSecretFileNameForCfgPassword	Kubernetes	GVP Configuration

Secret Name	Secret key	Type of secret	Notes
			Server password

Service Discovery

ConfigMaps

The following config maps are used:

ConfigMap	Used by	Notes
tenant-inventory	gvp-sd	Tenant information to be synced in GVP Configuration Server

Secrets

Secret Name	Secret key	Type of secret	Notes
secrets.configServer.secretName	secrets.configServer.dbUserKey	Kubernetes	GVP Configuration Server username
	secrets.configServer.dbPasswordKey	Kubernetes	GVP Configuration Server password
secrets.consul.k8sTokenName	secrets.consul.k8sTokenKey	Kubernetes	Consul token in k8s
secrets.consul.vaultSecretName	secrets.consul.vaultSecretKey	CSI	Consul token in CSI

Reporting Server

ConfigMaps

ConfigMap	Used by	Notes
	init, gvp-rs	Template for RS configuration

Secrets

Secret Name	Secret key	Type of secret	Notes
secrets.configserverProviderClassName	cfgSecretFileNameForCfgUsername	Kubernetes	GVP Configuration Server username
	cfgSecretFileNameForCfgPassword	Kubernetes	GVP Configuration Server password

Secret Name	Secret key	Type of secret	Notes
secret.rsSecretName	dbadminsecretFileName	Kubernetes	RS DB Password
	dbreadersecretFileName	Kubernetes	RS DB Password

SQLServer password is created as Kubernetes secrets and is used by Reporting Server for connecting with SQL DB. They will be mounted as a Volume in RS init / RS container for DB initialization and DB connection

1. SQL secrets should be created in the Cluster with admin and reader passwords.
    1. RS admin file name - RS admin password created during DB deployment which will be used reporting server.
    2. RS reader file name - Random password string which will be associated to 'read-only' user created by RS init container. They will be shared to BDS namespace for billing purposes

To use Kubernetes secret, set the following parameter to 'false'.

```
secret: keyVaultSecret: false
```

GVP Configuration Server

ConfigMaps

N/A

Secrets

Secrets used by GVP Configuration Server:

Secret Name	Secret key	Type of secret	Notes
secrets.postgres.secretName	secrets.postgres.secretHostnameKey	Kubernetes	Database hostname
	secrets.postgres.secretDbNameKey	Kubernetes	Database name
	secrets.postgres.secretAdminUserKey	Kubernetes	Database admin user
	secrets.postgres.secretAdminPwdKey	Kubernetes	Database password

Secrets to be created after GVP Configuration Server deployment:

Secret Name	Secret key	Type of secret	Notes
secrets.configServer.secretName	secrets.configServer.dbUserKey	Kubernetes	GVP Configuration Server username
	secrets.configServer.dbPasswordKey	Kubernetes	GVP Configuration Server password

Provision Genesys Voice Platform

Contents

- Administrator

Learn how to provision Genesys Voice Platform.

Related documentation:

-
-
-

RSS:

- For private edition

This page includes the following sections:

- Tenant provisioning
- Provisioning Nexus connection in GVP

Tenant provisioning

GVP Service Discovery (**SD**) container is used for provisioning tenant object in GVP Configuration Server and creating application objects for - Configuration Server, Media Control Platform app objects under Resource Manager logical resource group.

Overview

GVP Service Discovery (**SD**) container (running in **K8s**) does the following:

1. Runs a timer that gets invoked every **`1 min`** by default (this is configurable).
2. Checks **Consul** for the registered MCPs and then checks **GVP Configuration Server (CS)** for the MCPs present there and does the necessary addition/removal of MCPs from CS to sync with Consul data.
3. Checks the **`tenant-inventory`** configmap in the **`gvp`** namespace of the **K8s cluster** and, if configmap is updated from the last run, then based on the new data creates/updates tenant information.
4. Note that SD processes one tenant at a time.

The following **objects** are created in CS as part of GVP Tenant creation, and unless specified SD uses default values for those objects:

- The **Tenant** object itself with properties set in the **Annex** section
- The following **IVR Profiles**:
  - IVRAppDefault

    - conference
    - cpd
    - record
    - media
- The following **Transactions** object (used by **Recording Uploader**):
    - hybrid_integration

For Tenant creation, the following parameters **SHOULD** be specified:

- As part of Tenant provisioning, the tenant is registered to **GWS** and you get a **GWS-CCID**. This parameter is **mandatory** for GVP tenant creation.
- The **id** for the tenant is a **mandatory** parameter. This can be arbitrary string, but the preferable value is the **last 4-digits of the GWS-CCID**.
- The tenant **name** parameter is **preferred** to be populated.
- The **default-application** parameter should be set to **IVRAppDefault** always.
- The provisioned parameter should be set to 'true', unless the git pipeline workflow does that. Once this is set, then Service Discovery populates the tenant's contact center id (uuid) and the tenant's default IVR-Profile's dbid in Consul KV store.

Provision Consul to provide `gvp_config_dbid` and `ivr_app_dbid` parameters with Tenant ID and default IVR application ID values from the GVP Configuration Server database.

Where,

- **gvp_config_dbid** is the DB ID for a particular tenant.
- **ivr_app_dbid** is the DB ID for the default IVR application for that tenant.

Example request:

```
curl -H "Authorization: Bearer " https:///v1/kv/tenants//gvp
{
"gvp_config_dbid": "161",
"ivr_app_dbid"  : "217"
```

Use Case 1 - Deploy New Tenant

- As mentioned above, deploying a new tenant with SD and CS simply boils down to creating a configmap in your K8s cluster under gvp namespace.

Note: SD doesn't support creating multiple tenants in bulk, so you need to provide one JSON data file per tenant and repeat the process for multiple tenants.

- A bare minimum JSON should contain the minimal set of parameters mentioned above:

```
{
```

```
    "name": "CustomerX",
    "id": "2026",
    "gws-ccid": "285bd12f-5e4a-4c76-ad93-752ee1a82026",
    "default-application": "IVRAppDefault"
}
```

- Delete the existing **tenant-inventory** configmap if any: **`kubectl -n gvp delete configmap tenant-inventory --ignore-not-found`**
- Create the configmap with your JSON file: **`kubectl -n gvp create configmap tenant-inventory --from-file tenant-2026.json`**
- This configmap is mounted as a **volume** in SD - so the new JSON is updated in the **`/etc/config`** folder of the SD container.

```
[genesys@gvp-sd-bfcdd567f-8hrzm config]$ pwd
/etc/config
[genesys@gvp-sd-bfcdd567f-8hrzm config]$ ls -ls
total 0
0 lrwxrwxrwx 1 root root 27 May  5 12:59 tenant-2026.json > ..data/tenant 2026.json
```

- In the **next cycle**, SD will detect the new file and process it - thus creating/updating tenant and the associated objects.
- Once processed, SD **ignores** the file in subsequent cycles.

Use Case 2 - Update Existing Tenant

Note: Service Discovery cannot change/update configuration for Environment tenant. This limitation will be addressed in future release.

The mechanism for updating existing tenant is very similar to deploying new tenant. In this case, the existing JSON for the tenant needs to be updated with new parameters.

Note: You MUST always keep the minimal set of parameters present in the JSON even if in the case of update.

Example 1 - Recording destination provisioning for recording uploader

WEM provisioning is supported through the **hybrid_integration** transactions object. Only change is to specify the additional **WEM parameters** in the **Tenant JSON** file. An example JSON file with the WEM parameters may look like the following:

```
{
    "name": "CustomerX",
    "id": "2026",
    "gws-ccid": "285bd12f-5e4a-4c76-ad93-752ee1a82026",
    "default-application": "IVRAppDefault",
    "provisioned": "true",
    "transactions": {
```

```
        "name": "hybrid_integration",
        "recording-uploader.destFolder": {
            "destType": "Folder"
        },
        "recording-uploader.destFolder.mediaUpload": {
            "folder_path": "/rup/recordings"
        }
    }
}
```

The rest of the steps are the same as above. Delete existing configmap and create new with updated JSON.

Provisioning Nexus connection in GVP MCP

Integrating MCP to Nexus for Dialogflow voicebot

This section explains how to integrate MCP to Nexus for Dialogflow voicebot.

Pre-requisites

The following are the pre-requisites for integrating MCP to Nexus for Dialogflow voicebot:

1. The Nexus bot endpoint URL. For example:

   ```
   ws://nexus-production.nexus.svc.cluster.local/nexus/v3/bot/connection
   ```

2. You should register the Tenant to Nexus.
3. Verify that the Nexus API key is configured for the tenant by logging into the Tenant's Configuration Server > Transactions > DesginerEnv > Nexus.

Configuration

Update the following parameters in the section **mcpConfig** in the **values.yaml** file and deploy/redeploy MCP:

```
nexus_asr1.pool.size: 1

  nexus_asr1.provision.vrm.client.resource.uri: "[\"ws://nexus-
production.nexus.svc.cluster.local/nexus/v3/bot/connection\"]"

  nexus_asr1.provision.vrm.client.resource.type: "ASR"

  nexus_asr1.provision.vrm.client.resource.name: "ASRC"

  nexus_asr1.provision.vrm.client.resource.engines: "nexus"

  nexus_asr1.provision.vrm.client.resource.engine.nexus.audio.codec: "mulaw"

  nexus_asr1.provision.vrm.client.resource.engine.nexus.audio.samplerate: 8000
```

```
nexus_asr1.provision.vrm.client.resource.engine.nexus.enableMaxSpeechTimeout: true

nexus_asr1.provision.vrm.client.resource.certificate: ""

nexus_asr1.provision.vrm.client.resource.privatekey: ""

nexus_asr1.provision.vrm.client.resource.proxy: ""

nexus_asr1.provision.vrm.client.TransportProtocol: "WEBSOCKET"
```

The mandatory parameters are:

```
nexus_asr1.pool.size
nexus_asr1.provision.vrm.client.resource.uri.
```

You can leave the remaining parameters as default.

For more information on the above-mentioned parameters, refer to Configure Genesys Voice Platform.

Validation

Make test calls and check the following statements in MCP logs:

```
Int 50148 5A1A9632-10053FC2 140635978869056 asr_open 5A1A9632-10053FC2-asr-nexus-471699/
success
Int 50149 5A1A9632-10053FC2 140635978869056 asr_close 5A1A9632-10053FC2-asr-nexus-471699
Int 50159 5A1A9632-10053FC2-asr-nexus-471699 140635978869056 ws_stats 5c49970a-d41a-91f5-6ece-
af6d71bf2120 Tx: total 477/76320, sent 477/76320, failed 0/0, dropped 0/0 Rx: 0/0
```

Integrating MCP to Nexus for Agent Assist

This section explains how to integrate MCP to Nexus for Agent Assist.

Pre-requisites

The following are the pre-requisites for Integrating MCP to NEXUS for Agent Assist:

1. The Nexus Agent Assist endpoint URL. For example:

   ```
   ws://nexus-production.nexus.svc.cluster.local/athena/v1/agent-assist/voice/connection
   ```

2. You should register the Tenant to Nexus.
3. The GWS URL and client secrets. For example:

   ```
   https://gauth-int.nlb02-westus2.int.dev.genazure.com/auth/v3/oauth/token
   ```

Configuration

For configuration, complete these steps:

1. Create a new Kubernetes secret:

   ```
   apiVersion: v1
    kind: Secret
   ```

```
metadata:
  name: shared-gauth-gvp-client-secret
  namespace: gvp
type: Opaque
data:
 gauth-gvp-client-secret: $CLIENT_SECRET
```

Note: As regards Client secret, you must obtain the value from your GWS and replace “$CLIENT_SECRET " with the actual secret.

2. Update the following parameters in the section **secrets** in the **values.yaml** file:

```
gws:
 enabled: true
 clientName: "gvp_client"
 clientSecret:
   secretName: "shared-gauth-gvp-client-secret"
   secretKey: "gauth-gvp-client-secret"
```

3. Update the following parameters in the section **gws** in the **values.yaml** file and deploy/redeploy MCP:

```
gws:
  authEndpoint: $
```

Note: Replace “$" with the actual URL.

4. Add the following parameters to record the IVR profile of the Tenant:

```
"recordingclient.stream1.dest": "fixed,",
"recordingclient.stream1.dialogflow.engineId": "fixed,dfaa",
"recordingclient.stream1.encoding": "fixed,audio/mulaw",
"recordingclient.stream1.gauth": "fixed,true",
"recordingclient.stream1.xccid": "fixed,true",
"recordingclient.gvp.config.msml.record.enablesipfilerecording": "fixed,true"
```

Validation

Make test calls and check the following statements in the MCP logs:

```
Int 50156 F5099632-100028BD-0_268435867 140575662311744 streamer_open 0_268435867success
Int 50157 F5099632-100028BD-0_268435867 140575662311744 streamer_close 0_268435867
```

The MCP metrics section includes the following Nexus-related triggers:

MCP_WEBSOCKET_CLIENT_OPEN_ERROR	A websocket client error opening a connection to service, such as Nexus
MCP_WEBSOCKET_CLIENT_PROTOCOL_ERROR	A websocket client received a protocol error from the service, such as Nexus

Deploy Genesys Voice Platform

Contents

Learn how to deploy Genesys Voice Platform (GVP) into a private edition environment.

Related documentation:

-
-
-

RSS:

- For private edition

Assumptions

- The instructions on this page assume you are deploying the service in a service-specific namespace, named in accordance with the requirements on Creating namespaces. If you are using a single namespace for all private edition services, replace the namespace element in the commands on this page with the name of your single namespace or project.
- Similarly, the configuration and environment setup instructions assume you need to create namespace-specific (in other words, service-specific) secrets. If you are using a single namespace for all private edition services, you might not need to create separate secrets for each service, depending on your credentials management requirements. However, if you do create service-specific secrets in a single namespace, be sure to avoid naming conflicts.

Deploy

> **Important**
> Make sure to review Before you begin for the full list of prerequisites required to deploy Genesys Voice Platform.

Prerequisites

- Consul with Service Mesh and DNS
- Availability of shared Postgres for GVP Configuration Server
- Availability of SQL Server database for Reporting Server
  - Create DB in advance (for example, DB Name: **gvp_rs**).

- There is a requirement for one user to have admin or database owner (dbo) access and a second user with read only (ro) access.
- Use these credentials for the creation of Reporting Server secrets.

Environment setup

GKE

1. Log in to the gke cluster.

   ```
   gcloud container clusters get-credentials gke1
   ```

2. Create a JSON file to add the gvp namespace to the GKE cluster. For example, **create-gvp-namespace.json**:

   ```
   {
       "apiVersion":"v1",
       "kind":"Namespace",
       "metadata":{
          "name":"gvp",
          "labels":{
             "name":"gvp"
          }
       }
    }
   ```

3. Run the following command to create the namespace:

   ```
   kubectl apply -f apply create-gvp-namespace.json
   ```

4. Confirm the namespace was created:

   ```
   kubectl describe namespace gvp
   ```

AKS

1. Log in to the AKS cluster

   ```
   az aks get-credentials --resource-group $RESOURCE_GROUP --name $AKS_CLUSTER_NAME
   ```

2. Create gvp project in the AKS cluster using the following manifest file:

   ```
   {
   "apiVersion": "v1",
   "kind": "Namespace",
   "metadata": {
   "name": "gvp",
   "labels": {
   "name": "gvp"
   }
   }
   }
   kubectl apply -f apply create-gvp-namespace.json
   ```

3. Confirm the namespace creation.

```
kubectl describe namespace gvp
```

The order of installation matters with GVP. To deploy without errors, install in this order:

1. GVP Configuration Server
2. GVP ServiceDiscovery
3. GVP Reporting Server
4. GVP Resource Manager
5. GVP Media Control Platform

Helm chart release URLs

Download the GVP Helm charts from JFrog using your credentials:

gvp-configserver : https:///gvp-configserver-.tgz

gvp-sd : https:///gvp-sd-.tgz

gvp-rs : https:///gvp-rs-.tgz

gvp-rm : https:///gvp-rm-.tgz

gvp-mcp : https:///gvp-mcp-.tgz

For version numbers, refer to Helm charts and containers for Genesys Voice Platform.

1. GVP Configuration Server

Secrets creation

Create the following secrets that are required for the service deployment.

postgres-secret

db-hostname: Hostname of DB server

db-name: Database name

db-password: Password for DB user

db-username: Username for DB

server-name: Hostname of DB server

```
apiVersion: v1
```

```
kind: Secret
metadata:
  name: postgres-secret
  namespace: gvp
type: Opaque
data:
  db-username:
  db-password:
  db-hostname: cG9zdGdyZXMtcncuaW5mcmEuc3ZjLmNsdXN0ZXIubG9jYWw=
  db-name: Z3Zw
  server-name: cG9zdGdyZXMtcncuaW5mcmEuc3ZjLmNsdXN0ZXIubG9jYWw=
```

Run the following command:

```
kubectl apply -f postgres-secret.yaml
```

configserver-secret

password: Password to set for Config DB

username: Username to set for Config DB

```
apiVersion: v1
kind: Secret
metadata:
  name: configserver-secret
  namespace: gvp
type: Opaque
data:
  username:
  password:
```

Run the following command:

```
kubectl apply -f configserver-secret.yaml
```

Install Helm chart

Download the required Helm chart release from the JFrog repository and install. Refer to Helm Chart URLs.

```
helm install gvp-configserver ./ -f gvp-configserver-values.yaml
```

Set the following values in your values.yaml for Configuration Server:

priorityClassName >> Set to a priority class that exists on the cluster (or create it instead).

imagePullSecrets >> Set to your pull secret name.

gvp-configserver-values.yaml

```
# Default values for gvp-configserver.
# This is a YAML-formatted file.
# Declare variables to be passed into your templates.

## Global Parameters
## Add labels to all the deployed resources
##
podLabels: {}

## Add annotations to all the deployed resources
##
podAnnotations: {}

serviceAccount:
  # Specifies whether a service account should be created
  create: false
  # Annotations to add to the service account
  annotations: {}
  # The name of the service account to use.
  # If not set and create is true, a name is generated using the fullname template
  name:

## Deployment Configuration
## replicaCount should be 1 for Config Server
replicaCount: 1

## Base Labels. Please do not change these.
serviceName: gvp-configserver
component: shared
# Namespace
partOf: gvp

## Container image repo settings.
image:
  confserv:
    registry: pureengage-docker-staging.jfrog.io
    repository: gvp/gvp_confserv
    pullPolicy: IfNotPresent
    tag: "{{ .Chart.AppVersion }}"
  serviceHandler:
    registry: pureengage-docker-staging.jfrog.io
    repository: gvp/gvp_configserver_servicehandler
    pullPolicy: IfNotPresent
    tag: "{{ .Chart.AppVersion }}"
  dbInit:
    registry: pureengage-docker-staging.jfrog.io
    repository: gvp/gvp_configserver_configserverinit
    pullPolicy: IfNotPresent
    tag: "{{ .Chart.AppVersion }}"

## Config Server App Configuration
configserver:
  ## Settings for liveness and readiness probes
  ## !!! THESE VALUES SHOULD NOT BE CHANGED UNLESS INSTRUCTED BY GENESYS !!!
  livenessValues:
    path: /cs/liveness
    initialDelaySeconds: 30
    periodSeconds: 60
    timeoutSeconds: 20
    failureThreshold: 3
    healthCheckAPIPort: 8300

  readinessValues:
```

```
    path: /cs/readiness
    initialDelaySeconds: 30
    periodSeconds: 30
    timeoutSeconds: 20
    failureThreshold: 3
    healthCheckAPIPort: 8300

  alerts:
    cpuUtilizationAlertLimit: 70
    memUtilizationAlertLimit: 90
    workingMemAlertLimit: 7
    maxRestarts: 2

## PVCs defined
#  none

## Define service(s) for application
service:
  type: ClusterIP
  host: gvp-configserver-0
  port: 8888
  targetPort: 8888

## Service Handler configuration.
serviceHandler:
  port: 8300

## Secrets storage related settings - k8s secrets only
secrets:
  # Used for pulling images/containers from the respositories.
  imagePull:
    - name: pureengage-docker-dev
    - name: pureengage-docker-staging

  # Config Server secrets. If k8s is false, csi will be used, else k8s will be used.
  # Currently, only k8s is supported!
  configServer:
    secretName: configserver-secret
    secretUserKey: username
    secretPwdKey: password
    #csiSecretProviderClass: keyvault-gvp-gvp-configserver-secret

  # Config Server Postgres DB secrets and settings.
  postgres:
    dbName: gvp
    dbPort: 5432
    secretName: postgres-secret
    secretAdminUserKey: db-username
    secretAdminPwdKey: db-password
    secretHostnameKey: db-hostname
    secretDbNameKey: db-name
    #secretServerNameKey: server-name

## Ingress configuration
ingress:
  enabled: false
  annotations: {}
    # kubernetes.io/ingress.class: nginx
    # kubernetes.io/tls-acme: "true"
  hosts:
    - host: chart-example.local
      paths: []
  tls: []
```

```
  #  - secretName: chart-example-tls
  #    hosts:
  #      - chart-example.local

## App resource requests and limits
## ref: http://kubernetes.io/docs/user-guide/compute-resources/
##
resources:
  requests:
    memory: "512Mi"
    cpu: "500m"
  limits:
    memory: "1Gi"
    cpu: "1"

## App containers' Security Context
## ref: https://kubernetes.io/docs/tasks/configure-pod-container/security-context/#set-the-
security-context-for-a-container
##
## Containers should run as genesys user and cannot use elevated permissions
##
securityContext:
  runAsUser: null
  runAsGroup: 0
  # capabilities:
  #   drop:
  #   - ALL
  # readOnlyRootFilesystem: true
  # runAsNonRoot: true
  # runAsUser: null

podSecurityContext: {}
  # fsGroup: 0

## Priority Class
## ref: https://kubernetes.io/docs/concepts/configuration/pod-priority-preemption/
## NOTE: this is an optional parameter
##
priorityClassName: system-cluster-critical

## Affinity for assignment.
## Ref: https://kubernetes.io/docs/concepts/configuration/assign-pod-node/#affinity-and-anti-
affinity
##
affinity: {}

## Node labels for assignment.
## ref: https://kubernetes.io/docs/user-guide/node-selection/
##
nodeSelector: {}

## Tolerations for assignment.
## ref: https://kubernetes.io/docs/concepts/configuration/taint-and-toleration/
##
tolerations: []

## Service/Pod Monitoring Settings
## Whether to create Prometheus alert rules or not.
prometheusRule:
  create: true

## Grafana dashboard Settings
## Whether to create Grafana dashboard or not.
```

```
grafana:
  enabled: true

## Enable network policies or not
networkPolicies:
  enabled: false

## DNS configuration options
dnsConfig:
  options:
    - name: ndots
      value: "3"
```

Verify the deployed resources

Verify the deployed resources from the CLI.

2. GVP Service Discovery

NOTE: After GVP-SD (Service Discovery) pod gets deployed, you will notice a few errors. Please ignore them and move on to the next deployment. This will start working once Resource Manager (RM) and Media Control Platform (MCP) are deployed.

Secrets creation

Create the following secrets that are required for the service deployment.

shared-consul-consul-gvp-token

shared-consul-consul-gvp-token-secret.yaml

In regards to consul secret, you must obtain the token value from your consul deployment and replace "$CONSUL_TOKEN" with the actual token.

```
apiVersion: v1
kind: Secret
metadata:
  name: shared-consul-consul-gvp-token
  namespace: gvp
type: Opaque
data:
  consul-consul-gvp-token: $CONSUL_TOKEN
```

Run the following command:

```
kubectl create -f shared-consul-consul-gvp-token-secret.yaml
```

ConfigMap creation

Create the following ConfigMap that is required for the service deployment.

Caveat

If the tenant has not been deployed yet, then you will not have the information needed to populate the config map. An empty config-map can be created using:

```
kubectl create configmap tenant-inventory -n gvp
```

Create Config based on Tenant provisioning via Service Discovery Container.

t100.json

```
{
    "name": "t100",
    "id": "80dd",
    "gws-ccid": "9350e2fc-a1dd-4c65-8d40-1f75a2e080dd",
    "default-application": "IVRAppDefault"
}
```

Run the following command:

Add Config Map

```
kubectl create configmap tenant-inventory --from-file t100.json -n gvp
```

Install Helm chart

Download the required Helm chart release from the JFrog repository and install. Refer to Helm Chart URLs.

```
helm install gvp-sd ./ -f gvp-sd-values.yaml
```

gvp-sd-values.yaml

```
# Default values for gvp-sd.
# This is a YAML-formatted file.
# Declare variables to be passed into your templates.

## Global Parameters
## Add labels to all the deployed resources
##
podLabels: {}

## Add annotations to all the deployed resources
##
podAnnotations: {}

serviceAccount:
  # Specifies whether a service account should be created
  create: false
  # Annotations to add to the service account
```

```
  annotations: {}
  # The name of the service account to use.
  # If not set and create is true, a name is generated using the fullname template
  name:

## Deployment Configuration
replicaCount: 1
smtp: allowed

## Name overrides
nameOverride: ""
fullnameOverride: ""

## Base Labels. Please do not change these.
component: shared
partOf: gvp

image:
  registry: pureengage-docker-staging.jfrog.io
  repository: gvp/gvp_sd
  tag: "{{ .Chart.AppVersion }}"
  pullPolicy: IfNotPresent

## PVCs defined
# none

## Define service for application.
service:
  name: gvp-sd
  type: ClusterIP
  port: 8080

## Application configuration parameters.
env:
  MCP_SVC_NAME: "gvp-mcp"
  EXTERNAL_CONSUL_SERVER: ""
  CONSUL_PORT: "8501"
  CONFIG_SERVER_HOST: "gvp-configserver"
  CONFIG_SERVER_PORT: "8888"
  CONFIG_SERVER_APP: "default"
  HTTP_SERVER_PORT: "8080"
  METRICS_EXPORTER_PORT: "9090"
  DEF_MCP_FOLDER: "MCP_Configuration_Unit\\MCP_LRG"
  TEST_MCP_FOLDER: "MCP_Configuration_Unit_Test\\MCP_LRG"
  SYNC_INIT_DELAY: "10000"
  SYNC_PERIOD: "60000"
  MCP_PURGE_PERIOD_MINS: "0"
  EMAIL_METERING_FACTOR: "10"
  RECORDINGS_CONTAINER: "ccerp-recordings"
  TENANT_KV_FOLDER: "tenants"
  TENANT_CONFIGMAP_FOLDER: "/etc/config"
  SMTP_SERVER: "smtp-relay.smtp.svc.cluster.local"

## Secrets storage related settings
secrets:
  # Used for pulling images/containers from the respositories.
  imagePull:
    - name: pureengage-docker-dev
    - name: pureengage-docker-staging

  # If k8s is true, k8s will be used, else vault secret will be used.
  configServer:
    k8s: true
```

```
    k8sSecretName: configserver-secret
    k8sUserKey: username
    k8sPasswordKey: password
    vaultSecretName: "/configserver-secret"
    vaultUserKey: "configserver-username"
    vaultPasswordKey: "configserver-password"

  # If k8s is true, k8s will be used, else vault secret will be used.
  consul:
    k8s: true
    k8sTokenName: "shared-consul-consul-gvp-token"
    k8sTokenKey: "consul-consul-gvp-token"
    vaultSecretName: "/consul-secret"
    vaultSecretKey: "consul-consul-gvp-token"

  # GTTS key, password via k8s secret, if k8s is true.  If false, this data comes from tenant
profile.
  gtts:
    k8s: false
    k8sSecretName: gtts-secret
    EncryptedKey: encrypted-key
    PasswordKey: password

ingress:
  enabled: false
  annotations: {}
    # kubernetes.io/ingress.class: nginx
    # kubernetes.io/tls-acme: "true"
  hosts:
    - host: chart-example.local
      paths: []
  tls: []
  #  - secretName: chart-example-tls
  #    hosts:
  #      - chart-example.local

resources:
  requests:
    memory: "2Gi"
    cpu: "1000m"
  limits:
    memory: "2Gi"
    cpu: "1000m"

## App containers' Security Context
## ref: https://kubernetes.io/docs/tasks/configure-pod-container/security-context/#set-the-
security-context-for-a-container
##
## Containers should run as genesys user and cannot use elevated permissions
## Pod level security context
podSecurityContext:
  fsGroup: 0
  runAsUser: null
  runAsGroup: 0
  runAsNonRoot: true

## Container security context
securityContext:
  runAsUser: null
  runAsGroup: 0
  runAsNonRoot: true

## Priority Class
```

```
## ref: https://kubernetes.io/docs/concepts/configuration/pod-priority-preemption/
## NOTE: this is an optional parameter
##
priorityClassName: system-cluster-critical

## Affinity for assignment.
## Ref: https://kubernetes.io/docs/concepts/configuration/assign-pod-node/#affinity-and-anti-
affinity
##
affinity: {}

## Node labels for assignment.
## ref: https://kubernetes.io/docs/user-guide/node-selection/
##
nodeSelector: {}

## Tolerations for assignment.
## ref: https://kubernetes.io/docs/concepts/configuration/taint-and-toleration/
##
tolerations: []

## Service/Pod Monitoring Settings
prometheus:
  # Enable for Prometheus operator
  podMonitor:
    enabled: true

## Enable network policies or not
networkPolicies:
  enabled: false

## DNS configuration options
dnsConfig:
  options:
    - name: ndots
      value: "3"
```

Verify the deployed resources

Verify the deployed resources from the CLI.

3. GVP Reporting Server

Secrets creation

Create the following secrets that are required for the service deployment.

rs-dbreader-password

db_hostname: Hostname of DB server

db_name: Database name

db_password: Password for DB user

db_username: Username for DB

rs-dbreader-password-secret.yaml

```
apiVersion: v1
kind: Secret
metadata:
  name: rs-dbreader-password
  namespace: gvp
type: Opaque
data:
  db_username:
  db_password:
  db_hostname: bXNzcWxzZXJ2ZXJvcGVuc2hpZnQuZGF0YWJhc2Uud2luZG93cy5uZXQ=
  db_name: cnNfZ3Zw
```

Run the following command:

```
kubectl create -f rs-dbreader-password-secret.yaml
```

shared-gvp-rs-sqlserer-secret

db-admin-password: Password for DB admin

db-reader-password: Password for reader

shared-gvp-rs-sqlserer-secret.yaml

```
apiVersion: v1
kind: Secret
metadata:
  name: shared-gvp-rs-sqlserver-secret
  namespace: gvp
type: Opaque
data:
  db-admin-password:
  db-reader-password:
```

Run the following command:

```
kubectl create -f shared-gvp-rs-sqlserer-secret.yaml
```

Persistent Volumes creation

Create the following Persistent Volumes (PVs) that are required for the service deployment.

gvp-rs-0

gvp-rs-pv.yaml

```
apiVersion: v1
```

```
kind: PersistentVolume
metadata:
name: gvp-rs-0
namespace: gvp
spec:
capacity:
storage: 30Gi
accessModes:
- ReadWriteOnce
persistentVolumeReclaimPolicy: Retain
storageClassName: gvp
nfs:
path:  /export/vol1/PAT/gvp/rs-01
server: 192.168.30.51
```

Run the following command:

```
kubectl create -f gvp-rs-pv.yaml
```

Install Helm chart

Download the required Helm chart release from the JFrog repository and install. Refer to Helm Chart URLs.

```
helm install gvp-rs ./ -f gvp-rs-values.yaml
```

Set the following values in your values.yaml:

- priorityClassName >> Set to a priority class that exists on the cluster (or create it instead).
- imagePullSecrets >> Set to your pull secret name.
- keyVaultSecret: false >> Make sure this is false to force use of k8s secrets.
- storageClass: genesys-gvp >> Set to your storage class.

gvp-rs-values.yaml

```
## Global Parameters
## Add labels to all the deployed resources
##
labels:
  enabled: true
  serviceGroup: "gvp"
  componentType: "shared"

serviceAccount:
  # Specifies whether a service account should be created
  create: false
  # Annotations to add to the service account
  annotations: {}
  # The name of the service account to use.
  # If not set and create is true, a name is generated using the fullname template
  name:

## Primary App Configuration
```

```
##
# primaryApp:
# type: ReplicaSet
# Should include the defaults for replicas
deployment:
  replicaCount: 1
  strategy: Recreate
  namespace: gvp
nameOverride: ""
fullnameOverride: ""

image:
  registry: pureengage-docker-staging.jfrog.io
  gvprsrepository: gvp/gvp_rs
  snmprepository: gvp/gvp_snmp
  rsinitrepository: gvp/gvp_rs_init
  rstag:
  rsinittag:
  snmptag: v9.0.040.07
  pullPolicy: Always
  imagePullSecrets:
    - name: "pureengage-docker-staging"

## liveness and readiness probes
## !!! THESE OPTIONS SHOULD NOT BE CHANGED UNLESS INSTRUCTED BY GENESYS !!!
livenessValues:
  path: /ems-rs/components
  initialDelaySeconds: 30
  periodSeconds: 120
  timeoutSeconds: 3
  failureThreshold: 3

readinessValues:
  path: /ems-rs/components
  initialDelaySeconds: 10
  periodSeconds: 60
  timeoutSeconds: 3
  failureThreshold: 3

## PVCs defined
volumes:
  pvc:
    storageClass: managed-premium
    claimSize: 20Gi
    activemqAndLocalConfigPath: "/billing/gvp-rs"

## Define service(s) for application.  Fields many need to be modified based on `type`
service:
  type: ClusterIP
  restapiport: 8080
  activemqport: 61616
  envinjectport: 443
  dnsport: 53
  configserverport: 8888
  snmpport: 1705

## ConfigMaps with Configuration
## Use Config Map for creating environment variables
context:
  env:
    CFGAPP: default
    GVP_RS_SERVICE_HOSTNAME: gvp-rs.gvp.svc.cluster.local
    #CFGPASSWORD: password
```

```
    #CFGUSER: default
    CFG_HOST: gvp-configserver.gvp.svc.cluster.local
    CFG_PORT: '8888'
    CMDLINE: ./rs_startup.sh
    DBNAME: gvp_rs
    #DBPASS: 'jbIKfoS6LpfgaU$E'
    DBUSER: DBadmin
    rsDbSharedUsername: DBadmin
    DBPORT: 1433
    ENVTYPE: staging
    GenesysIURegion: westus2
    localconfigcachepath: /billing/gvp-rs/data/cache
    HOSTFOLDER: Hosts
    HOSTOS: CFGRedHatLinux
    LCAPORT: '4999'
    MSSQLHOST: mssqlserver.database.windows.net
    RSAPP: azure_rs
    RSJVM_INITIALHEAPSIZE: 500m
    RSJVM_MAXHEAPSIZE: 1536m
    RSFOLDER: Applications
    RS_VERSION: 9.0.032.22
    STDOUT: 'true'
    WRKDIR: /usr/local/genesys/rs/
    SNMPAPP: azure_rs_snmp
    SNMP_WORKDIR: /usr/sbin
    SNMP_CMDLINE: snmpd
    SNMPFOLDER: Applications

  RSCONFIG:
    messaging:
      activemq.memoryUsageLimit: "256 mb"
      activemq.dataDirectory: "/billing/gvp-rs/data/activemq"
    log:
      verbose: "trace"
      trace: "stdout"
    dbmp:
      rs.db.retention.operations.daily.default: "40"
      rs.db.retention.operations.monthly.default: "40"
      rs.db.retention.operations.weekly.default: "40"
      rs.db.retention.var.daily.default: "40"
      rs.db.retention.var.monthly.default: "40"
      rs.db.retention.var.weekly.default: "40"
      rs.db.retention.cdr.default: "40"

# Default secrets storage to k8s secrets with csi able to be optional
secret:
  # keyVaultSecret will be a flag to between secret types(k8's or CSI). If keyVaultSecret was
set to false k8's secret will be used
  keyVaultSecret: false
  #RS SQL server secret
  rsSecretName: shared-gvp-rs-sqlserver-secret
  # secretProviderClassName will not be used used when keyVaultSecret set to false
  secretProviderClassName: keyvault-gvp-rs-sqlserver-secret-00
  dbreadersecretFileName: db-reader-password
  dbadminsecretFileName: db-admin-password
  #Configserver secret
  #If keyVaultSecret set to false the below parameters will not be used.
  configserverProviderClassName: gvp-configserver-secret
  cfgSecretFileNameForCfgUsername: configserver-username
  cfgSecretFileNameForCfgPassword: configserver-password
  #If keyVaultSecret set to true the below parameters will not be used.
  cfgServerSecretName: configserver-secret
  cfgSecretKeyNameForCfgUsername: username
```

```
  cfgSecretKeyNameForCfgPassword: password

## Ingress configuration
ingress:
  enabled: false
  annotations: {}
    # kubernetes.io/ingress.class: nginx
  # kubernetes.io/tls-acme: "true"
  hosts:
    - host: chart-example.local
      paths: []
  tls: []
  #  - secretName: chart-example-tls
  #    hosts:
  #      - chart-example.local

networkPolicies:
  enabled: false

## primaryAppresource requests and limits
## ref: http://kubernetes.io/docs/user-guide/compute-resources/
##
resourceForRS:
  # We usually recommend not to specify default resources and to leave this as a conscious
  # choice for the user. This also increases chances charts run on environments with little
  # resources, such as Minikube. If you do want to specify resources, uncomment the following
  # lines, adjust them as necessary, and remove the curly braces after 'resources:'.
  requests:
    memory: "500Mi"
    cpu: "200m"
  limits:
    memory: "1Gi"
    cpu: "300m"

resoueceForSnmp:
  requests:
    memory: "500Mi"
    cpu: "100m"
  limits:
    memory: "1Gi"
    cpu: "150m"

## primaryApp containers' Security Context
## ref: https://kubernetes.io/docs/tasks/configure-pod-container/security-context/#set-the-
security-context-for-a-container
##
## Containers should run as genesys user and cannot use elevated permissions
securityContext:
  runAsNonRoot: true
  runAsUser: null
  runAsGroup: 0

podSecurityContext:
  fsGroup: 0

## Priority Class
## ref: https://kubernetes.io/docs/concepts/configuration/pod-priority-preemption/
##
priorityClassName: ""

## Affinity for assignment.
## Ref: https://kubernetes.io/docs/concepts/configuration/assign-pod-node/#affinity-and-anti-
affinity
```

```
##
affinity: {}

## Node labels for assignment.
## ref: https://kubernetes.io/docs/user-guide/node-selection/
##
nodeSelector: {}

## Tolerations for assignment.
## ref: https://kubernetes.io/docs/concepts/configuration/taint-and-toleration/
##
tolerations: []

## Extra labels
## ref: https://kubernetes.io/docs/concepts/overview/working-with-objects/labels/
##
#  labels: {}

## Extra Annotations
## ref: https://kubernetes.io/docs/concepts/overview/working-with-objects/annotations/
##
#  annotations: {}

## Service/Pod Monitoring Settings
monitoring:
  podMonitorEnabled: true
  prometheusRulesEnabled: true
  grafanaEnabled: true

monitor:
  prometheusPort: 9116
  monitorName: gvp-monitoring
  module: [if_mib]
  target: [127.0.0.1:1161]

##DNS Settings
dnsConfig:
  options:
    - name: ndots
      value: "3"
```

Verify the deployed resources

Verify the deployed resources from the CLI.

4. GVP Resource Manager

Note: Resource Manager and forward will not pass readiness checks until an MCP has registered properly. This is because this service is not available without MCPs.

Persistent Volumes creation

Create the following PVs that are required for the service deployment.

Note: If your deployment is capable of self-provisioning of Persistent Volumes, you can skip this step. The provisioner will create the volumes.

gvp-rm-01

gvp-rm-01-pv.yaml

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: gvp-rm-01
spec:
  capacity:
    storage: 30Gi
  accessModes:
    - ReadWriteOnce
  persistentVolumeReclaimPolicy: Retain
  storageClassName: gvp
  nfs:
    path:  /export/vol1/PAT/gvp/rm-01
    server: 192.168.30.51
```

Run the following command:

```
kubectl create -f gvp-rm-01-pv.yaml
```

gvp-rm-02

gvp-rm-02-pv.yaml

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: gvp-rm-02
spec:
  capacity:
    storage: 30Gi
  accessModes:
    - ReadWriteOnce
  persistentVolumeReclaimPolicy: Retain
  storageClassName: gvp
  nfs:
    path:  /export/vol1/PAT/gvp/rm-02
    server: 192.168.30.51
```

Run the following command:

```
kubectl create -f gvp-rm-02-pv.yaml
```

gvp-rm-logs-01

gvp-rm-logs-01-pv.yaml

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: gvp-rm-logs-01
spec:
  capacity:
    storage: 10Gi
  accessModes:
    - ReadWriteOnce
  persistentVolumeReclaimPolicy: Recycle
```

```
  storageClassName: gvp
  nfs:
    path:  /export/vol1/PAT/gvp/rm-logs-01
    server: 192.168.30.51
```

Run the following command:

```
kubectl create -f gvp-rm-logs-01-pv.yaml
```

gvp-rm-logs-02

gvp-rm-logs-02-pv.yaml

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: gvp-rm-logs-02
spec:
  capacity:
    storage: 10Gi
  accessModes:
    - ReadWriteOnce
  persistentVolumeReclaimPolicy: Recycle
  storageClassName: gvp
  nfs:
    path:  /export/vol1/PAT/gvp/rm-logs-02
    server: 192.168.30.51
```

Run the following command:

```
kubectl create -f gvp-rm-logs-02-pv.yaml
```

Install Helm chart

Download the required Helm chart release from the JFrog repository and install. Refer to Helm Chart URLs.

```
helm install gvp-rm ./ -f gvp-rm-values.yaml
```

Set the following values in your values.yaml for Configuration Server:

- priorityClassName >> Set to a priority class that exists on the cluster (or create it instead).
- imagePullSecrets >> Set to your pull secret name.
- Set cfgServerSecretName if you changed it from default.

gvp-rm-values.yaml

```
## Global Parameters
## Add labels to all the deployed resources
##
labels:
  enabled: true
  serviceGroup: "gvp"
  componentType: "shared"

## Primary App Configuration
##
```

```
# primaryApp:
# type: ReplicaSet
# Should include the defaults for replicas
deployment:
  replicaCount: 2
  deploymentEnv: "UPDATE_ENV"
  namespace: gvp
  clusterDomain: "svc.cluster.local"
nameOverride: ""
fullnameOverride: ""

image:
  registry: pureengage-docker-staging.jfrog.io
  gvprmrepository: gvp/gvp_rm
  cfghandlerrepository: gvp/gvp_rm_cfghandler
  snmprepository: gvp/gvp_snmp
  gvprmtestrepository: gvp/gvp_rm_test
  cfghandlertag:
  rmtesttag:
  rmtag:
  snmptag: v9.0.040.07
  pullPolicy: Always
  imagePullSecrets:
    - name: "pureengage-docker-staging"

dnsConfig:
  options:
    - name: ndots
      value: "3"

# Pod termination grace period 15 mins.
gracePeriodSeconds: 900

## liveness and readiness probes
## !!! THESE OPTIONS SHOULD NOT BE CHANGED UNLESS INSTRUCTED BY GENESYS !!!
livenessValues:
  path: /rm/liveness
  initialDelaySeconds: 60
  periodSeconds: 90
  timeoutSeconds: 20
  failureThreshold: 3

readinessValues:
  path: /rm/readiness
  initialDelaySeconds: 10
  periodSeconds: 60
  timeoutSeconds: 20
  failureThreshold: 3

## PVCs defined
volumes:
  billingpvc:
    storageClass: managed-premium
    claimSize: 20Gi
    mountPath: "/rm"
  logpvc:
    EnablePVForLogStorage: true
    storageClass: managed-premium
    claimSize: 5Gi
    accessMode: ReadWriteOnce
    mountPath: "/mnt/log"
    # If PV is not used for log storage by disabling the flag EnablePVForLogStorage: false,
the given host path will be used for log storage.
```

```
    LogStorageHostPath: /mnt/log

## Define service(s) for application.  Fields many need to be modified based on `type`
service:
  type: ClusterIP
  port: 5060
  rmHealthCheckAPIPort: 8300

## ConfigMaps with Configuration
## Use Config Map for creating environment variables
context:
  env:
    cfghandler:
      CFGSERVER: gvp-configserver.gvp.svc.cluster.local
      CFGSERVERBACKUP: gvp-configserver.gvp.svc.cluster.local
      CFGPORT: "8888"
      CFGAPP: "default"
      RMAPP: "azure_rm"
      RMFOLDER: "Applications\\RM_MicroService\\RM_Apps"
      HOSTFOLDER: "Hosts\\RM_MicroService"
      MCPFOLDER: "MCP_Configuration_Unit\\MCP_LRG"
      SNMPFOLDER: "Applications\\RM_MicroService\\SNMP_Apps"
      EnvironmentType: "prod"
      CONFSERVERAPP: "confserv"
      RSAPP: "azure_rs"
      SNMPAPP: "azure_rm_snmp"
      STDOUT: "true"
      VOICEMAILSERVICEDIDNUMBER: "55551111"

  RMCONFIG:
    rm:
      sip-header-for-dnis: "Request-Uri"
      ignore-gw-lrg-configuration: "true"
      ignore-ruri-tenant-dbid: "true"
    log:
      verbose: "trace"
    subscription:
      sip.transport.dnsharouting: "true"
      sip.headerutf8verification: "false"
      sip.transport.setuptimer.tcp: "5000"
      sip.threadpoolsize: "1"
    registrar:
      sip.transport.dnsharouting: "true"
      sip.headerutf8verification: "false"
      sip.transport.setuptimer.tcp: "5000"
      sip.threadpoolsize: "1"
    proxy:
      sip.transport.dnsharouting: "true"
      sip.headerutf8verification: "false"
      sip.transport.setuptimer.tcp: "5000"
      sip.threadpoolsize: "16"
      sip.maxtcpconnections: "1000"
    monitor:
      sip.transport.dnsharouting: "true"
      sip.maxtcpconnections: "1000"
      sip.headerutf8verification: "false"
      sip.transport.setuptimer.tcp: "5000"
      sip.threadpoolsize: "1"
    ems:
      rc.cdr.local_queue_path: "/rm/ems/data/cdrQueue_rm.db"
      rc.ors.local_queue_path: "/rm/ems/data/orsQueue_rm.db"

# Default secrets storage to k8s secrets with csi able to be optional
```

```
secret:
  # keyVaultSecret will be a flag to between secret types(k8's or CSI). If keyVaultSecret was
set to false k8's secret will be used
  keyVaultSecret: false
  #If keyVaultSecret set to false the below parameters will not be used.
  configserverProviderClassName: gvp-configserver-secret
  cfgSecretFileNameForCfgUsername: configserver-username
  cfgSecretFileNameForCfgPassword: configserver-password
  #If keyVaultSecret set to true the below parameters will not be used.
  cfgServerSecretName: configserver-secret
  cfgSecretKeyNameForCfgUsername: username
  cfgSecretKeyNameForCfgPassword: password

## Ingress configuration
ingress:
  enabled: false
  annotations: {}
    # kubernetes.io/ingress.class: nginx
  # kubernetes.io/tls-acme: "true"
  paths: []
  hosts:
    - chart-example.local
  tls: []
  #  - secretName: chart-example-tls
  #    hosts:
  #      - chart-example.local
networkPolicies:
  enabled: false
sip:
  serviceName: sipnode

## primaryAppresource requests and limits
## ref: http://kubernetes.io/docs/user-guide/compute-resources/
##
resourceForRM:
  # We usually recommend not to specify default resources and to leave this as a conscious
  # choice for the user. This also increases chances charts run on environments with little
  # resources, such as Minikube. If you do want to specify resources, uncomment the following
  # lines, adjust them as necessary, and remove the curly braces after 'resources:'.
  requests:
    memory: "1Gi"
    cpu: "200m"
    ephemeral-storage: "10Gi"
  limits:
    memory: "2Gi"
    cpu: "250m"

resoueceForSnmp:
  requests:
    memory: "500Mi"
    cpu: "100m"
  limits:
    memory: "1Gi"
    cpu: "150m"

## primaryApp containers' Security Context
## ref: https://kubernetes.io/docs/tasks/configure-pod-container/security-context/#set-the-
security-context-for-a-container
##
## Containers should run as genesys user and cannot use elevated permissions
securityContext:
  fsGroup: 500
  runAsNonRoot: true
```

```
  runAsUserRM: 500
  runAsGroupRM: 500
  runAsUserCfghandler: 500
  runAsGroupCfghandler: 500

## Priority Class
## ref: https://kubernetes.io/docs/concepts/configuration/pod-priority-preemption/
##
priorityClassName: ""

## Affinity for assignment.
## Ref: https://kubernetes.io/docs/concepts/configuration/assign-pod-node/#affinity-and-anti-
affinity
##
affinity: {}

## Node labels for assignment.
## ref: https://kubernetes.io/docs/user-guide/node-selection/
##
nodeSelector:


## Tolerations for assignment.
## ref: https://kubernetes.io/docs/concepts/configuration/taint-and-toleration/
##
tolerations: []

## Service/Pod Monitoring Settings
monitoring:
  podMonitorEnabled: true
  prometheusRulesEnabled: true
  grafanaEnabled: true

monitor:
  monitorName: gvp-monitoring
  prometheusPort: 9116
  prometheusPortlogs: 8200
  logFilePrefixName: RM
  module: [if_mib]
  target: [127.0.0.1:1161]
```

Verify the deployed resources

Verify the deployed resources from the CLI.

5. GVP Media Control Platform

Persistent Volumes creation

Create the following PVs that are required for the service deployment.

gvp-mcp-logs-01

gvp-mcp-logs-01-pv.yaml

```
apiVersion: v1
```

```
kind: PersistentVolume
metadata:
  name: gvp-mcp-logs-01
spec:
  capacity:
    storage: 10Gi
  accessModes:
    - ReadWriteOnce
  persistentVolumeReclaimPolicy: Recycle
  storageClassName: gvp
  nfs:
    path:  /export/vol1/PAT/gvp/mcp-logs-01
    server: 192.168.30.51
```

Run the following command:

```
kubectl create -f gvp-mcp-logs-01-pv.yaml
```

gvp-mcp-logs-02

gvp-mcp-logs-02-pv.yaml

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: gvp-mcp-logs-02
spec:
  capacity:
    storage: 10Gi
  accessModes:
    - ReadWriteOnce
  persistentVolumeReclaimPolicy: Recycle
  storageClassName: gvp
  nfs:
    path:  /export/vol1/PAT/gvp/mcp-logs-02
    server: 192.168.30.51
```

Run the following command:

```
kubectl create -f gvp-mcp-logs-02-pv.yaml
```

gvp-mcp-rup-volume-01

gvp-mcp-rup-volume-01-pv.yaml

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: gvp-mcp-rup-volume-01
spec:
  capacity:
    storage: 40Gi
  accessModes:
    - ReadWriteOnce
  persistentVolumeReclaimPolicy: Recycle
  storageClassName: disk-premium
  nfs:
    path:  /export/vol1/PAT/gvp/mcp-logs-01
    server: 192.168.30.51
```

Run the following command:

```
kubectl create -f gvp-mcp-rup-volume-01-pv.yaml
```

gvp-mcp-rup-volume-02

gvp-mcp-rup-volume-02-pv.yaml

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: gvp-mcp-rup-volume-02
spec:
  capacity:
    storage: 40Gi
  accessModes:
    - ReadWriteOnce
  persistentVolumeReclaimPolicy: Recycle
  storageClassName: disk-premium
  nfs:
    path:  /export/vol1/PAT/gvp/mcp-logs-02
    server: 192.168.30.51
```

Run the following command:

```
kubectl create -f gvp-mcp-rup-volume-02-pv.yaml
```

gvp-mcp-recording-volume-01

gvp-mcp-recordings-volume-01-pv.yaml

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: gvp-mcp-recording-volume-01
spec:
  capacity:
    storage: 40Gi
  accessModes:
    - ReadWriteOnce
  persistentVolumeReclaimPolicy: Recycle
  storageClassName: gvp
  nfs:
    path:  /export/vol1/PAT/gvp/mcp-logs-01
    server: 192.168.30.51
```

Run the following command:

```
kubectl create -f gvp-mcp-recordings-volume-01-pv.yaml
```

gvp-mcp-recording-volume-02

gvp-mcp-recordings-volume-02-pv.yaml

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: gvp-mcp-recording-volume-02
spec:
  capacity:
    storage: 40Gi
  accessModes:
```

```
    - ReadWriteOnce
  persistentVolumeReclaimPolicy: Recycle
  storageClassName: gvp
  nfs:
    path:  /export/vol1/PAT/gvp/mcp-logs-02
    server: 192.168.30.51
```

Run the following command:

```
kubecl create -f gvp-mcp-recordings-volume-02-pv.yaml
```

Install Helm chart

Download the required Helm chart release from the JFrog repository and install. Refer to Helm Chart URLs.

```
helm install gvp-mcp ./ -f gvp-mcp-values.yaml
```

Set the following values in your values.yaml:

- Set **logicalResourceGroup: "MCP_Configuration_Unit"** to add MCPs to the Real Configuration Unit (rather than test).

gvp-mcp-values.yaml

```
## Default values for gvp-mcp.
## This is a YAML-formatted file.
## Declare variables to be passed into your templates.

## Global Parameters
## Add labels to all the deployed resources
##
podLabels: {}

## Add annotations to all the deployed resources
##
podAnnotations: {}

serviceAccount:
  # Specifies whether a service account should be created
  create: false
  # Annotations to add to the service account
  annotations: {}
  # The name of the service account to use.
  # If not set and create is true, a name is generated using the fullname template
  name:

## Deployment Configuration
deploymentEnv: "UPDATE_ENV"
replicaCount: 2
terminationGracePeriod: 3600

## Name and dashboard overrides
nameOverride: ""
fullnameOverride: ""
dashboardReplicaStatefulsetFilterOverride: ""

## Base Labels. Please do not change these.
serviceName: gvp-mcp
component: shared
```

```
partOf: gvp

## Command-line arguments to the MCP process
args:
  - "gvp-configserver"
  - "8888"
  - "default"
  - "/etc/mcpconfig/config.ini"

## Container image repo settings.
image:
  mcp:
    registry: pureengage-docker-staging.jfrog.io
    repository: gvp/multicloud/gvp_mcp
    tag: "{{ .Chart.AppVersion }}"
    pullPolicy: IfNotPresent
  serviceHandler:
    registry: pureengage-docker-staging.jfrog.io
    repository: gvp/multicloud/gvp_mcp_servicehandler
    tag: "{{ .Chart.AppVersion }}"
    pullPolicy: IfNotPresent
  configHandler:
    registry: pureengage-docker-staging.jfrog.io
    repository: gvp/multicloud/gvp_mcp_confighandler
    tag: "{{ .Chart.AppVersion }}"
    pullPolicy: IfNotPresent
  snmp:
    registry: pureengage-docker-staging.jfrog.io
    repository: gvp/multicloud/gvp_snmp
    tag: v9.0.040.21
    pullPolicy: IfNotPresent
  rup:
    registry: pureengage-docker-staging.jfrog.io
    repository: cce/recording-provider
    tag: 9.0.000.00.b.1432.r.ef30441
    pullPolicy: IfNotPresent

## MCP specific settings
mcp:
  ## Settings for liveness and readiness probes of MCP
  ## !!! THESE VALUES SHOULD NOT BE CHANGED UNLESS INSTRUCTED BY GENESYS !!!
  livenessValues:
    path: /mcp/liveness
    initialDelaySeconds: 30
    periodSeconds: 60
    timeoutSeconds: 20
    failureThreshold: 3
    healthCheckAPIPort: 8300

  # Used instead of startupProbe.  This runs all initial self-tests, and could take some time.
  # Timeout is = 2
# maxUnavailable = 1
hpa:
  enabled: false
  minReplicas: 2
  maxUnavailable: 1
  maxReplicas: 4
  podManagementPolicy: Parallel
  targetCPUAverageUtilization: 20
  scaleupPeriod: 15
  scaleupPods: 4
  scaleupPercent: 50
  scaleupStabilizationWindow: 0
```

```
  scaleupPolicy: Max
  scaledownPeriod: 300
  scaledownPods: 2
  scaledownPercent: 10
  scaledownStabilizationWindow: 3600
  scaledownPolicy: Min

## Service/Pod Monitoring Settings
prometheus:
  mcp:
    name: gvp-mcp-snmp
    port: 9116

  rup:
    name: gvp-mcp-rup
    port: 8080

  podMonitor:
    enabled: true

grafana:
  enabled: false

  #log:
  #  name: gvp-mcp-log
  #  port: 8200

## Pod Disruption Budget Settings
podDisruptionBudget:
  enabled: true

## Enable network policies or not
networkPolicies:
  enabled: false

## DNS configuration options
dnsConfig:
  options:
    - name: ndots
      value: "3"

## Configuration overrides
mcpConfig:
  # MCP config overrides
  mcp.mpc.numdispatchthreads: 4
  mcp.log.verbose: "interaction"
  mcp.mpc.codec: "pcmu pcma telephone-event"
  mcp.mpc.transcoders: "PCM MP3"
  mcp.mpc.playcache.enable: 1
  mcp.fm.http_proxy: ""
  mcp.fm.https_proxy: ""

  #MRCP v2 ASR config overrides
  mrcpv2_asr.provision.vrm.client.connectpersetup: true
  mrcpv2_asr.provision.vrm.client.disablehotword: false
  mrcpv2_asr.provision.vrm.client.hotkeybasepath: "/usr/local/genesys/mcp/grammar/nuance/
hotkey"
  mrcpv2_asr.provision.vrm.client.noduplicatedgramuri: true
  mrcpv2_asr.provision.vrm.client.sendswmsparams: false
  mrcpv2_asr.provision.vrm.client.transportprotocol: "MRCPv2"
  mrcpv2_asr.provision.vrm.client.sendloggingtag: true
  mrcpv2_asr.provision.vrm.client.resource.name: "NuanceASRv2"
  mrcpv2_asr.provision.vrm.client.resource.uri: "sip:mresources@speech-server-clusterip:5060"
```

```
  mrcpv2_asr.provision.vrm.client.tlscertificatekey: "/usr/local/genesys/mcp/config/
x509_certificate.pem"
  mrcpv2_asr.provision.vrm.client.tlsprivatekey: "/usr/local/genesys/mcp/config/
x509_certificate.pem"
  mrcpv2_asr.provision.vrm.client.tlspassword: ""
  mrcpv2_asr.provision.vrm.client.tlsprotocoltype: "TLSv1"
  mrcpv2_asr.provision.vrm.client.confidencescale: 1
  mrcpv2_asr.provision.vrm.client.sendsessionxml: true
  mrcpv2_asr.provision.vrm.client.supportfornuance11: true
  mrcpv2_asr.provision.vrm.client.uniquegramid: true

  #MRCP v2 TTS config overrides
  mrcpv2_tts.provision.vrm.client.connectpersetup: true
  mrcpv2_tts.provision.vrm.client.speechmarkerencoding: "UTF-8"
  mrcpv2_tts.provision.vrm.client.transportprotocol: "MRCPv2"
  mrcpv2_tts.provision.vrm.client.sendloggingtag: true
  mrcpv2_tts.provision.vrm.client.resource.name: "NuanceTTSv2"
  mrcpv2_tts.provision.vrm.client.resource.uri: "sip:mresources@speech-server-clusterip:5060"
  mrcpv2_tts.provision.vrm.client.tlscertificatekey: "/usr/local/genesys/mcp/config/
x509_certificate.pem"
  mrcpv2_tts.provision.vrm.client.tlsprivatekey: "/usr/local/genesys/mcp/config/
x509_certificate.pem"
  mrcpv2_tts.provision.vrm.client.tlspassword: ""
  mrcpv2_tts.provision.vrm.client.tlsprotocoltype: "TLSv1"
  mrcpv2_tts.provision.vrm.client.nospeechlanguageheader: true
  mrcpv2_tts.provision.vrm.client.sendsessionxml: true
  mrcpv2_tts.provision.vrm.client.supportfornuance11: true
```

Verify the deployed resources

Verify the deployed resources from Google console/CLI.

Upgrade, rollback, or uninstall Genesys Voice Platform

Contents

Learn how to upgrade, rollback or uninstall Genesys Voice Platform.

Related documentation:

-
-
-

RSS:

- For private edition

Upgrade Genesys Voice Platform

Media Control Platform

MCP supports a canary upgrade, where a single instance of the statefulset (green) containing the new version is deployed along with the existing statefulset (blue).

Upgrade

Canary Upgrade

Add 1 instance of the new version in the green statefulset. Download the required Helm chart release from the JFrog repository and install. Refer to Helm Chart URLs.

Adding 1 instance of green with new version

```
helm install gvp-mcp-green ./ -f gvp-mcp-values.yaml --set hpa.maxReplicas=1 --set
replicaCount=1
```

Validating the upgrade

The readiness and liveness probes run some basic validation tests. To validate the deployment was successful, verify that all containers are running. and in Ready state.

Validation

```
kubectl get pods| grep gvp-mcp-green

NAME                                    READY   STATUS    RESTARTS   AGE
gvp-mcp-pipe-green-0                    4/4     Running   0          70m
```

Check for errors in the pod events.

Validation

```
Kubectl describe pod gvp-mcp-green-0
```

Complete the upgrade

Scale-up the green statefulset, assuming the overridevalues.yaml has the desired HPA min and max replicas.

Scale-up green with new version

```
helm upgrade gvp-mcp-green ./ -f gvp-mcp-values.yaml
```

Uninstall old blue statefulset to complete the upgrade.

```
helm uninstall gvp-mcp-blue
```

Resource Manager

Upgrade with Helm

1. Download RM helm.

2. Navigate to gvp-rm-microservice-master\helmcharts\gvp-rm

3. Check and update RM version in chart.yaml [Optional]

4. Check and update values.yaml [Optional]. Refer to the Resource Manager Override Helm chart values.

5. Navigate to gvp-rm-microservice-master\helmcharts

6. Issue the following command: gvp-rm >

Command: helm upgrade gvp-rm gvp-rm

```
PS C:\RM_microservice\gvp-rm-microservice-01042021\gvp-rm-microservice-master\helmcharts> helm upgrade gvp-rm-test gvp-rm
Release "gvp-rm-test" has been upgraded. Happy Helming!
NAME: gvp-rm-test
LAST DEPLOYED: Tue Apr 13 16:27:15 2021
NAMESPACE: gvp
STATUS: deployed
REVISION: 3
NOTES:
1. Get the application URL by running these commands:
  export POD_NAME=$(kubectl get pods --namespace gvp -l "app.kubernetes.io/name=gvp-rm,app.kubernetes.io/instance=gvp-rm-test" -o jsonpa
ame}")
  echo "Visit http://127.0.0.1:8080 to use your application"
  kubectl port-forward $POD_NAME 8080:80
```

Service Discovery

Service Discovery is upgraded with a rolling restart of the single instance.

Upgrade

Download the required Helm chart release from the JFrog repository and install. Refer to Helm Chart URLs.

```
helm upgrade gvp-sd ./ -f gvp-sd-values.yaml
```

Validating the upgrade

The readiness and liveness probes run some basic validation tests. To validate the deployment was successful, verify that all containers are running and in Ready state.

Validation

```
kubectl get pods| grep gvp-sd
```

```
NAME                                          READY   STATUS    RESTARTS   AGE
gvp-sd-5d8c7bf4cf-znrqt                        2/2     Running   0          70m
```

Check for errors in the pod events.

Validation

```
kubectl describe pod gvp-sd-5d8c7bf4cf-znrqt
```

Reporting Server

You need an SQL server deployed and secrets created in Kubernetes cluster.

DB is pre-initialised and has data [Optional]

1.Download RS helm.

2.Navigate to gvp-rs-microservice-master\helmcharts\gvp-rs

3.Check and update RS version in chart.yaml [Optional].

4.Check and update values.yaml. Configure DB details in values.yaml. Refer to the Reporting Server Override Helm chart values.

1. DB server name
2. DB name
3. DB username
4. secretProviderClassName
5. dbreadersecretFileName
6. dbadminsecretFileName

5.Navigate to gvp-rs-microservice-master\helmcharts

6.Issue the following command gvp-rs

Command: helm upgrade gvp-rs gvp-rs

GVP Configuration Server

GVP Configuration Server is upgraded with a rolling restart of the single instance.

Upgrade

Download the required Helm chart release from the JFrog repository and install. Refer to Helm Chart URLs.

```
helm upgrade gvp-configserver ./ -f gvp-configserver-values.yaml
```

Validating the upgrade

The readiness and liveness probes run some basic validation tests. To validate the deployment was successful, verify that all containers are running and in Ready state.

```
kubectl get pods| grep gvp-configserver
```

```
NAME                                        READY   STATUS    RESTARTS   AGE
gvp-configserver-0                          2/2     Running   0          70m
```

Check for errors in the pod events.

```
kubectl describe pod gvp-configserver-0
```

Rollback Genesys Voice Platform

Media Control Platform

The rollback procedure is to uninstall the new statefulset that was added during the canary upgrade.

Uninstall green statefulset during rollback

```
helm uninstall gvp-mcp-green
```

Scaling up/down the current stack

In the event that the current stack needs to be updated, for example making changes to the scaling policy but keeping the same version of the software, make the changes to the overridevalues.yaml and upgrade the current statefulset (blue in the example below).

Scale-up blue

```
helm upgrade gvp-mcp-blue ./ -f gvp-mcp-values.yaml
```

Resource Manager

Rollback can be done using the below command:

```
helm rollback gvp-rm
```

Service Discovery

The rollback procedure is to uninstall the new version and re-install the old version.

```
helm rollback gvp-sd
```

Reporting Server

Rollback can be done using the below command:

```
helm rollback
```

GVP Configuration Server

The rollback procedure is to uninstall the new version and re-install the old version.

```
helm rollback gvp-configserver
```

Uninstall Genesys Voice Platform

Media Control Platform

If there is only one MCP stack/statefulset, the uninstallation will cause service disruption.

Uninstall any statefulsets which are currently installed.

```
helm uninstall gvp-mcp-
```

Resource Manager

Uninstall Resource Manager using the following command:

```
helm uninstall gvp-rm
```

Service Discovery

Warning

Since there is only one Service Discovery instance, the uninstallation will cause service disruption.

Uninstall any statefulsets which are currently installed.

```
helm uninstall gvp-sd
```

Reporting Server

Uninstall Reporting Server by using the following command:

```
helm uninstall gvp-rs
```

GVP Configuration Server

Warning

Since there is only one GVP Configuration Server instance, the uninstallation will cause service disruption.

Uninstall any stateful sets which are currently installed.

```
helm uninstall gvp-configserver
```

Observability in Genesys Voice Platform

Contents

Learn about the logs, metrics, and alerts you should monitor for Genesys Voice Platform.

Related documentation:

-
-
-

RSS:

- For private edition

Monitoring

Private edition services expose metrics that can be scraped by Prometheus, to support monitoring operations and alerting.

- As described on Monitoring overview and approach, you can use a tool like Grafana to create dashboards that query the Prometheus metrics to visualize operational status.
- As described on Customizing Alertmanager configuration, you can configure Alertmanager to send notifications to notification providers such as PagerDuty, to notify you when an alert is triggered because a metric has exceeded a defined threshold.

The services expose a number of Genesys-defined and third-party metrics. The metrics that are defined in third-party software used by private edition services are available for you to use as long as the third-party provider still supports them. For descriptions of available Genesys Voice Platform metrics, see:

- Voice Platform Configuration Server metrics
- Voice Platform Media Control Platform metrics
- Voice Platform Service Discovery metrics
- Voice Platform Reporting Server metrics
- Voice Platform Resource Manager metrics

See also System metrics.

Enable monitoring

Service	CRD or annotations?	Port	Endpoint/ Selector	Metrics update interval
Voice Platform Configuration	Service/Pod Monitoring	Not applicable	See selector details on the	

Service	CRD or annotations?	Port	Endpoint/ Selector	Metrics update interval
Server	Settings		Voice Platform Configuration Server metrics and alerts page	
Voice Platform Media Control Platform	Service/Pod Monitoring Settings	9116, 8080, 8200	See selector details on the Voice Platform Media Control Platform metrics and alerts page	
Voice Platform Service Discovery	Automatic	9090	See selector details on the Voice Platform Service Discovery metrics and alerts page	
Voice Platform Reporting Server	ServiceMonitor / PodMonitor	9116	See selector details on the Voice Platform Reporting Server metrics and alerts page	
Voice Platform Resource Manager	ServiceMonitor / PodMonitor	9116, 8200	See selector details on the Voice Platform Resource Manager metrics and alerts page	

Configure metrics

The metrics that are exposed by Genesys Voice Platform services are available by default. No further configuration is required in order to define or expose these metrics. You cannot define your own custom metrics.

Alerting

Private edition services define a number of alerts based on Prometheus metrics thresholds.

> **Important**
>
> You can use general third-party functionality to create rules to trigger alerts based on metrics values you specify. Genesys does not provide support for custom alerts that you create in your environment.

For descriptions of available Genesys Voice Platform alerts, see:

- Voice Platform Configuration Server alerts
- Voice Platform Media Control Platform alerts
- Voice Platform Service Discovery alerts
- Voice Platform Reporting Server alerts
- Voice Platform Resource Manager alerts

Configure alerts

Private edition services define a number of alerts by default (for Genesys Voice Platform, see the pages linked to above). No further configuration is required.

The alerts are defined as **PrometheusRule** objects in a **prometheus-rule.yaml** file in the Helm charts. As described above, Genesys Voice Platform does not support customizing the alerts or defining additional **PrometheusRule** objects to create alerts based on the service-provided metrics.

Logging

Content coming soon

Voice Platform Configuration Server metrics and alerts

Contents

Find the metrics Voice Platform Configuration Server exposes and the alerts defined for Voice Platform Configuration Server.

Service	CRD or annotations?	Port	Endpoint/Selector	Metrics update interval

Voice Platform Configuration Server	Service/Pod Monitoring Settings	Not applicable	Service/Pod Monitoring Settings:
			`values.yaml`
			`------------`
			`prometheus:`
			`  createRule: true`
			Grafana Dashboard Settings:
			`values.yaml`
			`------------`
			`grafana:`
			` enabled: true`	

See details about:

- Voice Platform Configuration Server metrics
- Voice Platform Configuration Server alerts

Metrics

No metrics are defined for Voice Platform Configuration Server.

Metric and description	Metric details	Indicator of

Alerts

The following alerts are defined for Voice Platform Configuration Server.

Alert	Severity	Description	Based on	Threshold
ContainerCPUreachedHighPercentForConfigserver	P0Critical	The trigger will flag an alarm when the Configserver container CPU utilization goes	container_cpu_usage_seconds_total, container_spec_cpu_quota, container_spec_cpu_period	15mins

Alert	Severity	Description	Based on	Threshold
		beyond 70% for 15 mins		
ContainerMemoryUsedOver90PercentForConfigserver	HIGH	The trigger will flag an alarm when the Configserver container working memory use is over 90% of the limit for 15 mins	container_memory_working_set_bytes, kube_pod_container_resource_limits_memory_bytes	15mins
ContainerMemoryUsedOver1GBForConfigserver	HIGH	The trigger will flag an alarm when the Configserver container working memory has exceeded 1GB for 15 mins	container_memory_working_set_bytes	15mins
ContainerRestartsOverHourForConfigserver	HIGH	This alert is triggered when the Configserver container restarts in 15 mins exceeded 4	kube_pod_container_status_restarts_total	15mins
ContainerNotRunningForConfigserver	HIGH	This alert is triggered when the Configserver container has not been running for 15 minutes	kube_pod_container_status_running	15mins
ContainerRestartsOverHourForServiceHandler	MEDIUM	This alert is triggered when the service-handler container restarts exceeded 4 for 15 mins	kube_pod_container_status_running	15mins
ContainerNotRunningForServiceHandler	MEDIUM	This alert is triggered when the service-handler container has not been running for 15 minutes	kube_pod_container_status_running	15mins

Voice Platform Service Discovery metrics and alerts

Contents

Find the metrics Voice Platform Service Discovery exposes and the alerts defined for Voice Platform Service Discovery.

Service	CRD or annotations?	Port	Endpoint/Selector	Metrics update interval
Voice Platform Service Discovery	Automatic	9090	`env: METRICS_EXPORTER_PORT: "9090"`	

Metrics

Metric and description	Metric details	Indicator of
sdServiceCounter Consul and Configserver Sync Check Counter	**Unit:** Unsigned32 **Type:** Gauge **Label:** **Sample value:** 131	Useful for checking if SD is stuck or there is any deadlock
sdServiceLastRun Last Time When Consul and Configserver Sync Check has Run	**Unit:** Unix Time **Type:** Gauge **Label:** **Sample value:** 1634071196.054	Useful for checking if SD is stuck or there is any deadlock

Alerts

No alerts are defined for Genesys Voice Platform.

Voice Platform Reporting Server metrics and alerts

Contents

Find the metrics Voice Platform Reporting Server exposes and the alerts defined for Voice Platform Reporting Server.

Service	CRD or annotations?	Port	Endpoint/Selector	Metrics update interval

Voice Platform Reporting Server	ServiceMonitor / PodMonitor	9116	Metrics endpoint:
			`curl -v "http://:9116/snmp?target=127.0.0.1%3A1161&module=if_mib"`
			Enabling metrics:
			Service/Pod Monitoring Settings
			`prometheus:`
			`  enabled: true`
			`  metric:`
			`   port: 9116`
			Enable for Prometheus operator
			`podMonitor:`
			`  enabled: true`
			`  metric:`
			`   path: /snmp`
			`   module: [ if_mib ]`
			`   target: [ 127.0.0.1:1161 ]`
			monitoring:
			• prometheusRulesEnabled: true
			• grafanaEnabled: true
			monitor:
			• monitorName: gvp-monitoring	

See details about:

- Voice Platform Reporting Server metrics
- Voice Platform Reporting Server alerts

Metrics

Metric and description	Metric details	Indicator of
rsQueueName The name of the message queue (rsQueueName{gvpConfigDBID="172",rsQueueIndex="1",rsQueueName="rs.queue.remote_cdr.rm"}	**Unit:** DisplayString **Type:** Gauge **Label:** **Sample value:** 1	Information
rsQueueSize Used to get the size of the message queue. (rsQueueSize{gvpConfigDBID="172",rsQueueIndex="1"})	**Unit:** Unsigned32 **Type:** Gauge **Label:** **Sample value:** 1	Traffic
rsDequeueCount Used to get dequeue value of the message queue. (rsDequeueCount{gvpConfigDBID="172",rsQueueIndex="1"})	**Unit:** Counter64 **Type:** Counter **Label:** **Sample value:** 0	Traffic
rsEnqueueCount Used to get enqueue value of the message queue. (rsEnqueueCount{gvpConfigDBID="172",rsQueueIndex="1"})	**Unit:** Counter64 **Type:** Counter **Label:** **Sample value:** 4	Traffic
rsUptime The time (in hundredths of a second) since the RS was started.	**Unit:** Unsigned32 **Type:** Gauge **Label:** **Sample value:** 30619972	Information

Alerts

The following alerts are defined for Voice Platform Reporting Server.

Alert	Severity	Description	Based on	Threshold
PodStatusNotReady	CRITICAL	The trigger will flag an alarm when RS pod status is Not ready for 30 mins and this will be controlled through override-value.yaml file.	kube_pod_status_ready	30mins
ContainerRestartedRepeatedly	CRITICAL	The trigger will flag an alarm when the RS or RS SNMP	kube_pod_container_status_restarts_total	15mins

Alert	Severity	Description	Based on	Threshold
		container gets restarted 5 or more times within 15 mins		
InitContainerFailingRepeatedly	CRITICAL	The trigger will flag an alarm when the RS init container gets failed 5 or more times within 15 mins	kube_pod_init_container_status_restarts_total	15mins
ContainerCPUreachedHigh	HIGH	The trigger will flag an alarm when the RS container CPU utilization goes beyond 80% for 15 mins	container_cpu_usage_seconds_total, container_spec_cpu_quota, container_spec_cpu_period	15mins
ContainerMemoryUsageHigh	HIGH	The trigger will flag an alarm when the RS container Memory utilization goes beyond 80% for 15 mins	container_memory_usage_bytes, kube_pod_container_resource_limits_memory_bytes	15mins
RSQueueSizeCritical	HIGH	The trigger will flag an alarm when RS JMS message queue size goes beyond 15000 (3GB approx. backlog) for 15 mins	rsQueueSize	15mins
PVC50PercentFilled	HIGH	This trigger will flag an alarm when the RS PVC size is 50% filled	kubelet_volume_stats_used_bytes, kubelet_volume_stats_capacity_bytes	15mins
PVC80PercentFilled	CRITICAL	This trigger will flag an alarm when the RS PVC size is 80% filled	kubelet_volume_stats_used_bytes, kubelet_volume_stats_capacity_bytes	5mins

Voice Platform Resource Manager metrics and alerts

Contents

Find the metrics Voice Platform Resource Manager exposes and the alerts defined for Voice Platform Resource Manager.

Service	CRD or annotations?	Port	Endpoint/Selector	Metrics update interval

Voice Platform Resource Manager	ServiceMonitor / PodMonitor	9116,
		8200
			Metrics endpoints:
			`curl -v "http://:9116/snmp?target=127.0.0.1%3A1161&module=if_mib"`
			`curl -v "http://:8200/log`
			Enable metrics:
			Service/Pod Monitoring Settings
			`prometheus:`
			`enabled: true`
			`metric:`
			`port: 9116`
			`log:`
			`port: 8200`
			Enable for Prometheus operator
			`podMonitor:`
			`enabled: true`
			`metric:`
			`path: /snmp`
			`module: [ if_mib ]`
			`target: [ 127.0.0.1:1161 ]`
			`log:`
			`path: /log`
			monitoring:
			• prometheusRulesEnabled: true
			• grafanaEnabled: true
			monitor:
			• monitorName: gvp-monitoring
			• logFilePrefixName: RM	

See details about:

- Voice Platform Resource Manager metrics
- Voice Platform Resource Manager alerts

Metrics

Metric and description	Metric details	Indicator of
rmTotal5xxInviteSent Number of 5xx that were received for INVITE sent by RM	**Unit:** Unsigned32 **Type:** Gauge **Label:** **Sample value:** 514	Error
rmTotal4xxInviteSent Number of 4xx that were received for INVITE sent by RM	**Unit:** Unsigned32 **Type:** Gauge **Label:** **Sample value:** 10	Error
rmPRStatus GVP_RM_PhysicalResourceTable.Status: Current state of the resource (rmPRStatus{gvpConfigDBID="174",rmPRName="mcp-10.206.5.89",rmPRStatus="AVAILABLE"})	**Unit:** DisplayString **Type:** Gauge **Label:** **Sample value:** 1	Information
rmPRActiveCalls GVP_RM_PhysicalResourceTable.ActiveCalls: Number of calls that currently are handled by the resource. (rmPRActiveCalls{gvpConfigDBID="174",rmPRName="mcp-10.206.5.89"})	**Unit:** Unsigned32 **Type:** Gauge **Label:** **Sample value:**	Traffic
rmPRTotalCalls GVP_RM_PhysicalResourceTable.TotalCalls: Total number of calls that have been handled by this resource since it was connected to RM. (rmPRTotalCalls{gvpConfigDBID="174",rmPRName="mcp-10.206.5.89"})	**Unit:** Unsigned32 **Type:** Gauge **Label:** **Sample value:** 150	Traffic
rmTenantCurrentInboundCalls Number of active inbound calls that use this tenant.	**Unit:** Unsigned32 **Type:** Gauge **Label:** **Sample value:** 2	Traffic
rmTenantPeakCalls Maximum number of concurrent calls to this Tenant since it became active	**Unit:** **Type:** Gauge **Label:** **Sample value:** 10	Traffic

Alerts

The following alerts are defined for Voice Platform Resource Manager.

Alert	Severity	Description	Based on	Threshold
RM Service Down	CRITICAL	RM pods are not in ready state and RM service is not available	kube_pod_container_status_running	0
InitContainerFailingRepeatedly	CRITICAL	The trigger will flag an alarm when the RM init container gets failed 5 or more times within 15 mins.	kube_pod_init_container_status_restarts_total	15 mins
ContainerRestartedRepeatedly	CRITICAL	The trigger will flag an alarm when the RM or RM SNMP container gets restarted 5 or more times within 15 mins	kube_pod_container_status_restarts_total	15 mins
PodStatusNotReady	CRITICAL	The trigger will flag an alarm when RM pod status is Not ready for 30 mins and this will be controlled by override-value.yaml.	kube_pod_status_ready	30mins
RMTotal5XXErrorForINVITE	HIGH	The RM mib counter stats will be collected for every 30 seconds and if the mib counter total5xxInviteSent increments from its previous value by 5 within 5 minutes the trigger will flag an alarm.	rmTotal5xxInviteSent	5 mins
RMTotal4XXErrorForINVITE	MEDIUM	The RM mib counter stats will be collected for every 60 seconds and if the mib counter total4xxInviteSent increments from its previous value	rmTotal4xxInviteSent	1min

Alert	Severity	Description	Based on	Threshold
		by 10 within 60 seconds the trigger will flag an alarm.		
RMInterNodeConnectivityBroken	HIGH	Inter-node connectivity between RM nodes is lost for 5mins.	gvp_rm_log_parser_warn_total	5 mins
RMConfigServerConnectionLost	HIGH	RM lost connection to GVP Configuration Server for 5mins.	gvp_rm_log_parser_warn_total	5 mins
RMSocketInterNodeError	HIGH	RM Inter node Socket Error for 5mins.	gvp_rm_log_parser_eror_total	5mins
ContainerCPUreachedBOpercentForRM0	HIGH	The trigger will flag an alarm when the RM container CPU utilization goes beyond 80% for 15 mins	container_cpu_usage_seconds_total, container_spec_cpu_quota, container_spec_cpu_period	15mins
ContainerCPUreachedBOpercentForRM1	HIGH	The trigger will flag an alarm when the RM container CPU utilization goes beyond 80% for 15 mins	container_cpu_usage_seconds_total, container_spec_cpu_quota, container_spec_cpu_period	15mins
ContainerMemoryUsageBOpercentForRM0	HIGH	The trigger will flag an alarm when the RM container Memory utilization goes beyond 80% for 15 mins	container_memory_rss, kube_pod_container_resource_limits_memory_bytes	15mins
ContainerMemoryUsageBOpercentForRM1	HIGH	The trigger will flag an alarm when the RM container Memory utilization goes beyond 80% for 15 mins	container_memory_rss, kube_pod_container_resource_limits_memory_bytes	15mins
MCPPortsExceeded	HIGH	All the MCP ports in MCP LRG are exceeded	gvp_rm_log_parser_eror_total	1min
RMServiceDegradedThresholdPercentage	HIGH	One of the RM container is not in running state for 5mins	kube_pod_container_status_running	5mins
RMMatchingIVRTenantNotFound	MEDIUM	Matching IVR profile tenant could not be found for 2mins	gvp_rm_log_parser_eror_total	2mins

Alert	Severity	Description	Based on	Threshold
RMResourceAllocationFailed	MEDIUM	RM Resource allocation failed for 1mins	gvp_rm_log_parser_error_total	1

Voice Platform Media Control Platform metrics and alerts

Contents

Find the metrics Voice Platform Media Control Platform exposes and the alerts defined for Voice Platform Media Control Platform.

Service	CRD or annotations?	Port	Endpoint/Selector	Metrics update interval

Voice Platform Media Control Platform	Service/Pod Monitoring Settings	9116,
		8080,
		8200
			Metrics endpoints:
			`curl -v "http://:9116/snmp?target=127.0.0.1%3A1161&module=if_mib"`
			`curl -v "http://:8080/metrics`
			`curl -v "http://:8200/log`
			Service/Pod Monitoring Settings:
			`values.yaml`
			`------------`
			`prometheus:`
			`enabled: true`
			`mcp:`
			`port: 9116`
			`rup:`
			`port: 8080`
			`log:`
			`port: 8200`
			`podMonitor:`
			`enabled: true`
			Grafana dashboard settings:
			`values.yaml`
			`------------`
			`grafana:`
			`enabled: true`	

See details about:

- Voice Platform Media Control Platform metrics
- Voice Platform Media Control Platform alerts

Metrics

Metric and description	Metric details	Indicator of
mcpSipCurrentInboundSessions	**Unit:** Unsigned32	Traffic

Metric and description	Metric details	Indicator of
MCP Current Inbound Sessions	**Type:** Gauge **Label:** **Sample value:** 3	
mcpSipPeakInboundSessions MCP Peak Inbound Sessions	**Unit:** Unsigned32 **Type:** Gauge **Label:** **Sample value:** 100	Traffic
mcpSipTotalInboundSessions MCP Total Inbound Sessions	**Unit:** Unsigned32 **Type:** Gauge **Label:** **Sample value:** 9000	Traffic
mcpStatus MCP is RUNNING or not	**Unit:** 0/1 **Type:** Gauge **Label:** **Sample value:** 1	Information
gvp_mcp_log_parser_eror_total Used for alerting for various errors based on MCP log parsing. Errors are indexed by LogID's.	**Unit:** Unsigned32 **Type:** Gauge **Label:** **Sample value:** 35929	Errors

Alerts

The following alerts are defined for Voice Platform Media Control Platform.

Alert	Severity	Description	Based on	Threshold
ContainerCPUreached70PercentForMCP	Critical	The trigger will flag an alarm when the MCP container CPU utilization goes beyond 70% for 5 mins	container_cpu_usage_seconds_total, container_spec_cpu_quota, container_spec_cpu_period	15mins
ContainerMemoryUsedOver90PercentForMCP	Critical	The trigger will flag an alarm when the MCP container working memory use is over 90% of the limit for 5 mins	container_memory_working_set_bytes, kube_pod_container_resource_limits_memory_bytes	15mins
ContainerMemoryUsedOver7GBForMCP	Critical	The trigger will flag an alarm when the MCP container working memory has exceeded 7GB for 5 mins	container_memory_working_set_bytes	15mins

Alert	Severity	Description	Based on	Threshold
ContainerRestartsOverLimit	HIGH	The trigger will flag an alarm when the MCP container restarts exceeded 2 for 15 mins	kube_pod_container_status_restarts_total	15mins
MCP_MEDIA_ERROR_CRITICAL	CRITICAL	Number of LMSIP media errors exceeded critical limit	gvp_mcp_log_parser_eror_total {LogID="33008",endpoint="mcplog"...}	30mins
NGI_LOG_FETCH_RESOURCE_TIMEOUT	HIGH	Number of VXMLi fetch timeouts exceeded limit	gvp_mcp_log_parser_eror_total {LogID="40026",endpoint="mcplog"...}	1min
NGI_LOG_FETCH_RESOURCE_ERROR	HIGH	Number of VXMLi fetch errors exceeded limit	gvp_mcp_log_parser_eror_total {LogID="40026",endpoint="mcplog"...}	1min
NGI_LOG_PARSE_ERROR	WARNING	Number of VXMLi parse errors exceeded limit	gvp_mcp_log_parser_eror_total {LogID="40028",endpoint="mcplog"...}	1min
NGI_LOG_FETCH_RESOURCE_ERROR_4XX	WARNING	Number of VXMLi 4xx fetch errors exceeded limit	gvp_mcp_log_parser_eror_total {LogID="40032",endpoint="mcplog"...}	1min
MCP_WEBSOCKET_TOKEN_CREATE_ERROR	HIGH	There are errors creating a JWT token with a websocket client	gvp_mcp_log_parser_eror_total {LogID="40026",endpoint="mcplog"...}	N/A
MCP_WEBSOCKET_TOKEN_CONFIG_ERROR	HIGH	There are errors getting information for Auth token with a websocket client	gvp_mcp_log_parser_eror_total {LogID="40026",endpoint="mcplog"...}	N/A
MCP_WEBSOCKET_TOKEN_FETCH_ERROR	HIGH	There are errors fetching Auth token with a websocket client	gvp_mcp_log_parser_eror_total {LogID="40026",endpoint="mcplog"...}	N/A
MCP_WEBSOCKET_CLIENT_OPEN_ERROR	HIGH	There are errors opening a session with a websocket client	gvp_mcp_log_parser_eror_total {LogID="40026",endpoint="mcplog"...}	N/A
MCP_WEBSOCKET_CLIENT_PROTOCOL_ERROR	HIGH	There are protocol errors with a websocket client	gvp_mcp_log_parser_eror_total {LogID="40026",endpoint="mcplog"...}	N/A
MCP_SDP_PARSE_ERROR	WARNING	Number of SDP parse errors exceeded limit	gvp_mcp_log_parser_eror_total {LogID="33006",endpoint="mcplog"...}	N/A

Logging

Contents

Learn how to store logs for Genesys Voice Platform.

Related documentation:

-
-
-

RSS:

- For private edition

Media Control Platform

Container	logging destination	log rotation
gvp-mcp	mcp-log-pvc	100 MB
servicehandler	stdout	N/A
confighandler	stdout	N/A
mcp-snmp	stdout	N/A
rup	stdout	N/A

Log rotation and retention policy could be decided as per the requirements. Refer to MCP storage requirements for on log volume size and IOPS.

You may use the following settings to determine the retention period. For example:

```
mcpConfig:
  mcp.log.verbose: "interaction"
  mcp.log.segment:"100 MB"
  mcp.log.expire: 3
```

Refer to MCP storage requirements for the storage requirements for mcp-log-pvc pvc.

The referenced stdout through sidecar is the preferred method. If MCP logs need to be redirected to stdout, enable the following configuration in the Helm override values.yaml file:

```
fluentBitSidecar:

enabled: true
```

When fluent-bit is enabled, the log storage configuration could be any of type – **persistantVolume**, **hostPath**, or **emptyDir**.

For more information on how to configure fluent-bit, refer to Sidecar processed logging.

Note: The fluent-bit sidecar feature is being provided *as-is* without support and requires a third-party container image that Genesys does not provide or support.

Resource Manager

Log rotation and retention policy could be decided as per the requirements. Refer to Resource Manager storage requirements for on log volume size and IOPS.

You may use the log expire to determine the retention period. For example:

```
RMCONFIG:
   log:
     expire: 40 day
```

The referenced stdout through sidecar is the preferred method. If RM logs need to be redirected to stdout, enable the following configuration in the Helm override values.yaml file:

```
fluentBitSidecar:

enabled: true
```

When fluent-bit is enabled, the log storage configuration could be any of type – **persistantVolume**, **hostPath**, or **emptyDir**.

For more information on how to configure fluent-bit refer to, refer to Sidecar processed logging.

Note: The fluent-bit sidecar feature is being provided *as-is* without support and requires a third-party container image that Genesys does not provide or support.

Service Discovery

The following table describes information for logging for the different containers:

Container	logging destination	log rotation
gvp-sd	stdout	N/A

Log rotation and retention policy could be decided as per the requirements. Service Discovery uses stdout for logging. Service Discovery logs redirected from the stdout can be terminated from the Service Discovery log location based on the the retention period that you decide.

Reporting Server

Log rotation and retention policy could be decided as per the requirements. RS uses stdout for logging. RS logs redirected from the stdout can be terminated from the RS log location based on the the retention period that you decide.

```
RSCONFIG:
   log:
      verbose: "trace"
      trace: "stdout"
```

GVP Configuration Server

The following table describes information for logging for the different containers:

Container	logging destination	log rotation
servicehandler	stdout	N/A
gvp-configserver	stdout	N/A

Log rotation and retention policy could be decided as per the requirements. GVP Configuration Server uses stdout for logging. GVP Configuration Server logs redirected from the stdout can be terminated from the GVP Configuration Server log location based on the the retention period that you decide.