Voice Microservices Private Edition Guide

6/7/2026

Table of Contents

Contents

Find links to all the topics in this guide.

Related documentation:

-
-

RSS:

- For private edition

Voice Microservices is a service available with the Genesys Multicloud CX private edition offering. Voice Microservices includes the Tenant Service, however there is a separate Private Edition Guide for the Tenant Service. For information about the Tenant Service, including provisioning, configuration, and deployment information, see the *Tenant Service Private Edition Guide*.

Overview

Learn more about Voice Microservices and how to get started.

- About Voice Microservices
- Architecture
- High availability and disaster recovery

Configure and deploy

Find out how to configure and deploy Voice Microservices.

- Before you begin
- Consul requirements for Voice services
- Redis requirements for Voice services
- Configure Voice Microservices
- Provision Voice Microservices
- Deploy Voice Microservices
- Upgrade, rollback, or uninstall Voice Microservices

Observability

Learn how to monitor Voice Microservices with metrics and logging.

- Observability in Voice Microservices
- Agent State Service metrics and alerts
- Call State Service metrics and alerts
- Config Service metrics and alerts
- Dial Plan Service metrics and alerts
- FrontEnd Service metrics and alerts
- ORS metrics and alerts
- Voice Registrar Service metrics and alerts
- Voice RQ Service metrics and alerts
- Voice SIP Cluster Service metrics and alerts
- Voice SIP Proxy Service metrics and alerts
- Voicemail metrics and alerts

Functionality

Find information about the differences between Voice Microservices and legacy applications.

- Feature support and known limitations

About Voice Microservices

Contents

Learn about Voice Microservices and how it works in Private Edition.

Related documentation:

-
-
-

RSS:

- For private edition

Supported Kubernetes platforms

Voice Microservices are supported on the following Kubernetes platforms:

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

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

Voice Microservices

Voice Microservices is an application cluster that provides the following functionality:

- Handle incoming voice (SIP) interactions
- Route voice and digital (IXN) interactions
- Support outbound interactions
- Provide events stream for reporting
- Support agents across regions

Voice Microservices comprises the following microservices:

- Voice SIP Cluster Service
- Voice SIP Proxy Service
- Voice Tenant Service
- Voice Orchestration Service

- Voice Agent State Service
- Voice Call State Service
- Voice Dial Plan Service
- Voice Config Service
- Voice Registrar Service
- Voice Front End Service
- Voice Redis (RQ) Service
- Voice Voicemail Service

Voice SIP Cluster Service

The Voice SIP Cluster Service provides the following functionality:

- Handles SIP signaling by running multiple nodes: each node is tenant-independent and uses a Voice Dial Plan Service to resolve tenant-specific information.
- N+1 scalable: Each node starts from a predefined configuration file, which is the same for every node in the cloud.
- Includes a **js** controller providing traditional services to SIP Server (LCA, HA link), as well as:
  - Publishing TLib events and user data requests for Voice Call State, Voice Orchestration, and Voice Tenant Services.
  - Providing the Rest API to handle TLib requests from a Voice Front End Service.

Voice SIP Proxy Service

The Voice SIP Proxy Service is an intermediate interface among services and the Voice SIP Cluster Service. The Voice SIP Proxy Service provides the following functionality:

- Balances load of SIP signaling across Voice SIP Cluster Service instances.
- Processes SIP REGISTER requests and relays them to Voice Registrar Service.

SIP Proxy adds the following URL into the SIP messaging sent to the SBC:

```
voice-sipproxy.{{k8s-namespace }}.svc.cluster.local
```

This is an SRV record created in the K8s DNS when the SIP Proxy Service is deployed. This FQDN depends on the name of a namespace where SIP Proxy Service is deployed.

The DNS used by an SBC is integrated with the K8s DNS service to forward .svc.cluster.local FQDNs K8s DNS.

Voice Tenant Service

The Voice Tenant Service is a core service of the Genesys Multicloud CX platform that serves as an application layer between front-end Genesys Multicloud CX solutions and shared back-end core services in a region.

The Voice Tenant Service instances are dedicated to a tenant of Genesys Multicloud CX platform and provide these main functions: provisioning of tenant resources, such as agents and DNs; routing of interactions within a tenant; execution of outbound campaigns for a tenant; providing call control functionality; participation in authentication workflow for tenant's agents.

Voice Orchestration Service

The Voice Orchestration Service provides the following functionality:

- Interacts with each Voice Tenant Service.
- Provides routing instructions to a Voice Front End Service.
- Provides local routing session states through a storage system.
- Retrieves Route Points (RP) configuration with URLs and parameters of associated Designer SCXML Application from the Voice Config Service.
- Dynamically retrieves Applications from Designer Application Server.
- Compiles Designer Application into a javascript code to be executed with each session.
- Monitors Redis streams for new interactions from SIP Cluster Service, IXN Service or GWS. Orchestration Services retrieve triggering events in a round-robin fashion, thus new interactions are evenly distributed between Orchestration Nodes.
- Starts and executes Voice and digital Sessions when triggered by routing events.
- Reads from Voice RQ Service streams TLib events and user data requests published by Voice SIP Cluster Service.
- Reads from Voice RQ Service streams Interaction (IXN) events and user data requests published by IXN Service.
- Delivers call control and user data update requests to a proper Voice SIP Cluster Service node via the Restful API.
- Delivers new call control requests to a Voice Front End Service via the Restful API.
- Sends requests to URS via a corresponding Tenant Redis stream as a session requires.
- Reads from Voice RQ Service streams URS responses and events.
- Serializes context of sessions into Redis for HA.
- Recovers sessions from Redis in case of ORS failover and continues session execution from the last state it was serialized.
- Processes HTTP requests from MCP and sends events back.
- Provides monitoring and health metrics using the Prometheus API.

Voice Agent State Service

The Voice Agent State Service provides the following functionality:

- Maintains agent states in a storage system. Recovers agent states from failure and in case of auto-scaling events.
- Reads agent state requests (RequestAgentLogin, RequestAgentReady, ...) from a Voice Front End Service.
- Updates agent login sessions (through a Voice Config Service) based on those requests.
- Generates agent state events according to the TLib model and provides them to a Voice Tenant Service and reporting clients.
- Reads agent-related interaction events (EventRinging, ...) from a Voice Call State Service and updates agent session accordingly. Provides those events to reporting clients.
- Reads device notifications (in service/out of service) from a Voice Registrar Service and updates agent states accordingly.
- Reads agent reservation requests (RequestReserveAgent) from a Voice Front End Service and grants agent reservation to clients.

Voice Call State Service

The Voice Call State Service provides the following functionality:

- Reads interaction events from a Voice SIP Cluster Service.
- Reads user data requests from a Voice Front End Service and updates call user data states accordingly.
- Maintains call-thread states in a storage system.
- Recovers call-thread states from failure and in case of auto-scaling events.
- Produces agent-related call events to a Voice Agent State Service.

Voice Dial Plan Service

The Voice Dial Plan Service provides the following functionality:

- Provides the HTTP interface to the Voice SIP Cluster Service for device type resolution (internal, external) and dial plan execution, including the number translation.
- Supports Voicemail scenarios.
- Provides the following information to the SIP Cluster Service:
  - Device contact
  - Agent logged in on the device
  - Options configured on the DN or at Person CME object.

Voice Config Service

The Voice Config Service provides the following functionality:

- Provides access to tenant configuration data through the Rest API.
- Provides the Rest API for services to store and access device registration and agent login information.
- The following services access the configuration:
  - Voice Orchestration Service (for obtaining SCXML application details of a Route Point).
  - Voice SIP Cluster Service (for obtaining details about a tenant trunk and softswitch).
  - Voice Dial Plan Service (for obtaining details about tenants and Dial Plan provisioning).
  - Voice SIP Proxy Service (for obtaining details about tenants).
  - Voice Registrar Service (for saving details about device registration).
  - Voice Agent State Service (for saving details about agent logins).

Voice Registrar Service

The Voice Registrar Service provides the following functionality:

- Maintains device states by processing SIP REGISTER messages.
- Stores device registrations through a Voice Config Service.
- Distributes device notifications (EventDNBackInService, EventDNOutOfService) to a Voice Tenant Service. Device notifications can also be used by a Voice Agent State Service for agent state updates.

Voice Front End Service

The Voice Front End Service provides the following functionality:

- Delivers call control, user data updates, and distribute event requests to a proper Voice SIP Cluster Service node that handles the call.
- Writes agent state, agent reservation, DND status requests to a storage system (Kafka topic), consumed by a Voice Agent Service.

Voice Redis Queue Service

The Voice Redis Queue (RQ) Service provides the following functionality:

- Distributes TLib events for each voice call or digital interaction to a Voice Orchestration Service from other services, such as a Voice SIP Cluster Service and Interaction Service.

- The Voice RQ Service works as a cluster of nodes, where each node in the cluster accepts client connections and plays primary and backup roles.
- To interact with the Voice RQ Service, the rq-client library is used by other services that take care of computing the RQ node, to which TLib events are sent.

Voice Voicemail Service

The Voice Voicemail Service is part of the multi-tenant microservice architecture. It provides the following functionality:

- Provides deposit of voicemail messages to agent and agent group mailboxes.
- Provides access to voice mailboxes by dialing to a voicemail access number.
- Uses the Voice Config Service to retrieve agent configuration and states.
- Stores voicemail recordings and metadata in a storage system.
- Provisioning is done through Agent Setup.

Architecture

Contents

Learn about architecture

Related documentation:

-
-
-

RSS:

- For private edition

Introduction

The following diagram shows an example of the high-level architecture for Voice Microservices.

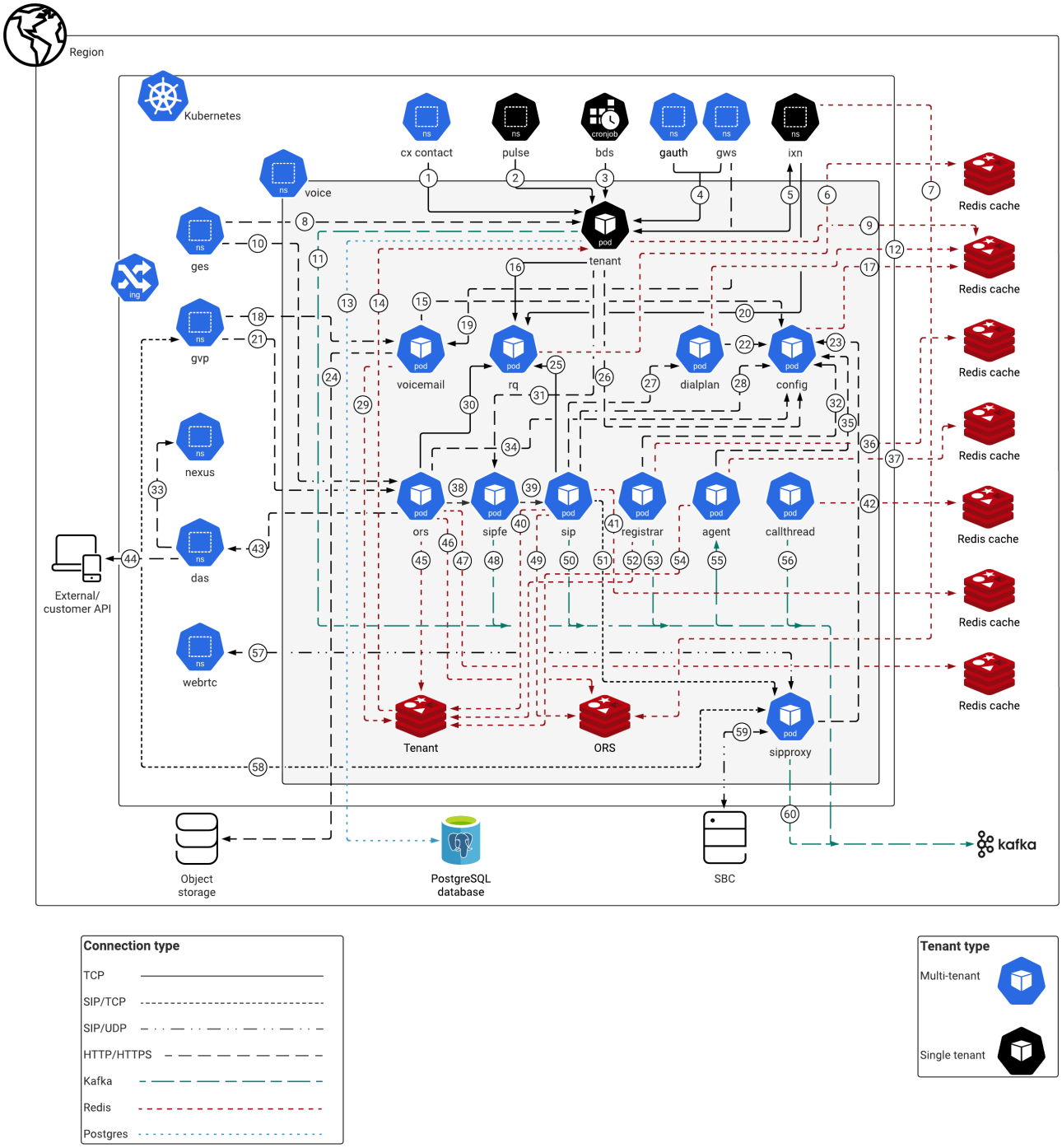
For information about voice connectivity network details, see Voice connectivity.

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 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 as a service in the network. *Egress* means the service is the source, and *Ingress* means the 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	CX Contact	Tenant Service	TCP	5050, 8888	Intra-cluster	Outbound campaigns provisioning and control performed by CX contact.
2	Genesys Pulse	Tenant Service	TCP	8888, 8000	Intra-cluster	Pulse obtains provisioning and real-time reporting data.
3	Billing Data Service	Tenant Service	TCP	8888	Intra-cluster	BDS reads the Tenant resource inventory.
4	Genesys Web Services and Applications	Tenant Service	TCP	8888, 8000, 2060	Intra-cluster	Provisioning and voice control/ observability requests to Tenant resources through GWS.
5	Interaction Server	Tenant Service	TCP	8888, 2060	Intra-cluster	Multimedia provisioning access and interaction status requests.
6	Voice RQ Service	Redis	Redis	6379*	Egress	Call-related Voice Microservices events for in-memory cache.
7	Interaction Server	Redis	Redis	6379	Egress	Interaction events.
8	Genesys Engagement	Tenant Service	HTTP	5080	Intra-cluster	Routing requests/

Connection	Source	Destination	Protocol	Port	Classification	Data that travels on this connection
	Service					events.
9	Tenant Service	Redis	Redis	6379*	Egress	Tenant configuration for in-memory cache.
10	Genesys Engagement Service	ORS	HTTP	9098	Intra-cluster	GES starts a session in ORS when it is time to put the callback in the queue for an agent. To initiate the ORS session, GES stores an entry in the Voice Microservice's Redis (using port 6379), rather than communicating directly with ORS. Once the ORS session is started, GES regularly queries the ORS session (using port 9098) for diagnostics information about the callback. In addition, GES might send events to control the ORS session; for example, when the callback is cancelled through the API or UI.

Connection	Source	Destination	Protocol	Port	Classification	Data that travels on this connection
11	Tenant Service	Kafka	Kafka	9092/9093*	Egress	Outbound data for reporting.
12	Dial Plan Service	Redis	Redis	6379*	Egress	Configuration and registration details for in-memory cache.
13	Tenant Service	PostgreSQL	Postgres	5432	Egress	Configuration data for persistent storage.
14	Redis 6.x	Tenant Service	Redis	6379*	Egress	Call and routing Voice Microservices events for the message bus.
15	Voicemail	Config Service	HTTP	9100	Intra-cluster	Fetches configuration data.
16	Tenant Service	Voice RQ Service	TCP	12100	Intra-cluster	Exchange of routing data for voice.
17	Config Service	Redis	Redis	6379*	Egress	In-memory cache for the Config Service.
18	Genesys Voice Platform	Voicemail	HTTP	8081	Intra-cluster	Communication to provide voicemail IVR pages.
19	Genesys Web Services and Applications	Voicemail	HTTP	8081	Intra-cluster	Voicemail uses GWS for user authentication. Agent Setup uses Voicemail service for Admin API functionalities. WWE uses Voicemail service for User API functionalities.

Connection	Source	Destination	Protocol	Port	Classification	Data that travels on this connection
20	Interaction Server	Voice RQ Service	TCP	12100	Intra-cluster	Exchange of routing data for digital.
21	Genesys Voice Platform	ORS	HTTP	11200	Intra-cluster	Provides call data to GVP.
22	Dial Plan Service	Config Service	HTTP	9100	Intra-cluster	Exchange of tenant information.
23	Voice SIP Proxy Service	Config Service	HTTP	9100	Intra-cluster	Exchange of tenant information.
24	Voicemail	Object storage	HTTP	N/A	Intra-cluster	Mailbox configurations and all the voicemail messages are saved in Azure Blob Storage/AWS S3 bucket. HTTP connection protocol is used without any specific port, whereas secret keys are used for establishing the connections.
25	Voice SIP Cluster Service	Voice RQ Service	TCP	12100	Intra-cluster	Exchange of voice call details.
26	Tenant Service	Config Service	HTTP	9100	Intra-cluster	Exchange of tenant information.
27	Voice SIP Cluster Service	Dial Plan Service	HTTP	8800	Intra-cluster	Provides routing instructions.
28	Voice SIP Cluster Service	Config Service	HTTP	9100	Intra-cluster	Exchange of tenant information.
29	Voicemail	Redis	Redis	6379*	Egress	In-memory cache for voicemail

Connection	Source	Destination	Protocol	Port	Classification	Data that travels on this connection
						data.
30	ORS	Voice RQ Service	TCP	12100	Intra-cluster	Fetches call and routing data.
31	Tenant Service	FrontEnd Service	HTTP	9101	Intra-cluster	Tenant Service provides Voice Microservices requests to the FrontEnd Service.
32	Voice Registrar Service	Config Service	HTTP	9100	Intra-cluster	Exchanges tenant information and stores agent registration.
33	Designer Application Server	Digital Channels	HTTP	80	Intra-cluster	Designer Applications use the ORS method to communicate with Nexus.
34	ORS	Config Service	HTTP	9100	Intra-cluster	ORS reads Route Points (RP) and Enhanced Routing Script (ERS) objects. ORS also implements keep-alive messages to detect if the connection to Config Service is alive.
35	Agent State Service	Config Service	HTTP	9100	Intra-cluster	Fetching of tenant, DN, person (agent) information, storing agent login/ logout.
36	Voice	Redis	Redis	6379*	Egress	In-memory

Connection	Source	Destination	Protocol	Port	Classification	Data that travels on this connection
	Registrar Service					cache for SIP registration.
37	Agent State Service	Redis	Redis	6379*	Egress	In-memory cache for agent activities.
38	ORS	FrontEnd Service	HTTP	9101	Intra-cluster	ORS requests are sent to the FrontEnd Service.
39	FrontEnd Service	Voice SIP Cluster Service	HTTP	11300	Intra-cluster	FrontEnd Service sends requests to SIP Server (SIP Cluster Service).
40	Voice SIP Cluster Service	Redis	Redis	6379*	Egress	Streaming of Voice Microservices events.
41	Voice SIP Cluster Service	Redis	Redis	6379*	Egress	In-memory cache for SIP Cluster Service.
42	Call State Service	Redis	Redis	6379*	Egress	In-memory cache for the Call State Service.
43	ORS	Designer Application Server	HTTP	80	Intra-cluster	ORS fetches Designer Applications (routing strategy).
44	Designer Application Server	External/ customer	HTTPS	443	Egress	External/ customer API requests. Designer Applications use the ORS method for external requests.
45	ORS	Redis	Redis	6379*	Egress	Streams routing events.

Connection	Source	Destination	Protocol	Port	Classification	Data that travels on this connection
46	ORS	Redis	Redis	6379*	Egress	Reads new calls and interactions in the system.
47	ORS	Redis	Redis	6379*	Egress	In-memory cache.
48	FrontEnd Service	Kafka	Kafka	9092/9093*	Egress	Provides data to reporting.
49	Voice SIP Cluster Service	Redis	Redis	6379*	Egress	Streams new call events.
50	Voice SIP Cluster Service	Kafka	Kafka	9092/9093*	Egress	Provides data to reporting.
51	Voice SIP Cluster Service	Voice SIP Proxy Service	SIP/TCP	5080	Intra-cluster	Exchange of SIP signals.
52	Voice Registrar Service	Redis	Redis	6379*	Egress	Streams DN registration details.
53	Voice Registrar Service	Kafka	Kafka	9092/9093*	Egress	Provides data to reporting.
54	Agent State Service	Redis	Redis	6379*	Egress	Streams agent-related events.
55	Kafka 2.x	Agent State Service	Kafka	9092/9093*	Egress	Provides data to reporting.
56	Call State Service	Kafka	Kafka	9092/9093*	Egress	Provides data to reporting.
57	Voice SIP Proxy Service	WebRTC Media Service	SIP/UDP	5070	Intra-cluster	Exchange of SIP signals.
58	Genesys Voice Platform	Voice SIP Proxy Service	SIP/TCP	5080	Intra-cluster	Exchange of SIP signals.
59	Voice SIP Proxy Service	Session Border Controller	SIP/TCP	Not known (configured as trunk DN level in customer	Intra-cluster	Exchange of SIP signals.

Connection	Source	Destination	Protocol	Port	Classification	Data that travels on this connection
				CME)		
60	Voice SIP Proxy Service	Kafka	Kafka	9092/9093*	Egress	Provides data to reporting.

* Configurable ports

Architecture - Cross-region

Contents

Learn about Voice Microservices- cross-region architecture

Related documentation:

-
-
-

RSS:

- For private edition

Introduction

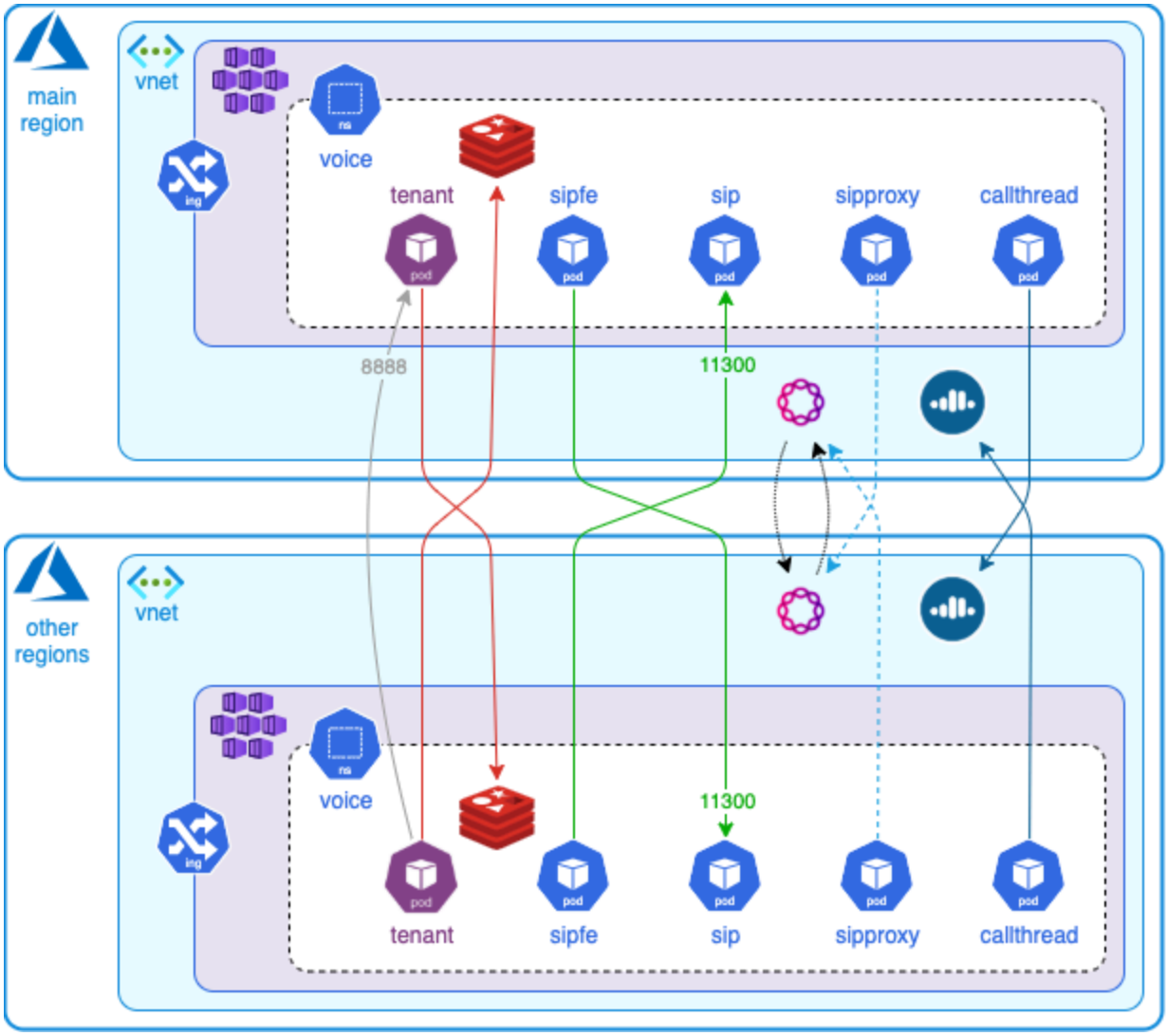
The following diagram shows an example of cross-region architecture for Voice Microservices.

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 Voice Microservices 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 Voice Microservices as a service in the network. *Egress* means the Voice Microservices service is the source, and *Ingress* means the Voice Microservices service is the destination. *Intra-cluster* means the connection is between services in the cluster.

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

Service	High Availability	Disaster Recovery	Where can you host this service?
Voice Microservices	N = N (N+1)	Active-spare	Primary or secondary unit

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

Before you begin

Contents

Find out what to do before deploying Voice Microservices.

Related documentation:

-
-
-

RSS:

- For private edition

Limitations and assumptions

Not applicable

Download the Helm charts

For information about how to download the Helm charts, see Downloading your Genesys Multicloud CX containers.

The following services are included with Voice Microservices:

- Voice Agent State Service
- Voice Config Service
- Voice Dial Plan Service
- Voice Front End Service
- Voice Orchestration Service
- Voice Registrar Service
- Voice Call State Service
- Voice RQ Service
- Voice SIP Cluster Service
- Voice SIP Proxy Service
- Voice Voicemail Service
- Voice Tenant Service

See Helm charts and containers for Voice Microservices for the Helm chart version you must

download for your release.

For information about the Voicemail Service, see Before you begin in the *Configure and deploy Voicemail* section of this guide.

For information about the Tenant service, also included with Voice Microservices, see the *Tenant Service Private Edition Guide*.

Third-party prerequisites

For information about setting up your Genesys Multicloud CX private edition platform, see Software Requirements.

The following table lists the third-party prerequisites for Voice Microservices.

Third-party services

Name	Version	Purpose	Notes
Redis	6.x	Used for caching. Only distributions of Redis that support Redis cluster mode are supported, however, some services may not support cluster mode.	
Consul	1.13.x	Service discovery, service mesh, and key/value store.	For additional information, see Voice services configuration in Consul.
Kafka	2.x	Message bus.	
An SMTP relay		Facilitates email communications in an environment where GCXI reports or voicemails are sent as emails to contact center personnel. Genesys recommends PostFix, but you can use any SMTP relay that supports standard mail libraries.	Required for Voice Voicemail Service if you integrate voicemails with email.
A container image registry and Helm chart repository		Used for downloading Genesys containers and Helm charts into the customer's repository to support a CI/CD pipeline. You can use any Docker OCI compliant registry.	

Storage requirements

Voice Tenant Service

Persistent Volume	Size	Type	IOPS	Functionality	Container	Critical	Backup needed
log-pvc	50Gi	RWO	medium	storing log files	tenant	Y	Y

SIP Cluster Service

Persistent Volume	Size	Type	IOPS	Functionality	Container	Critical	Backup needed
log-pvc	50Gi	RWO	medium	storing log files	voice-sip	Y	Y

VoiceMail Service

Persistent Volume	Type	IOPS	Functionality	Container	Critical	Backup needed
Azure blob storage v2	RWM	medium	storing voicemailbox settings and voicemail messages	tenant	Y	Y
AWS S3 Bucket	RWM	medium	storing voicemailbox settings and voicemail messages	tenant	Y	Y
File System	RWM	medium	storing voicemailbox settings and voicemail messages	tenant	Y	Y

For more information, see Storage requirements in the *Configure and deploy Voicemail* section of this guide.

Network requirements

For general network requirements, review the information on the suite-level Network settings page.

	Voice Voicemail Service	Voice Tenant Service
Cross-region bandwidth	Connect to other region Voicemail service to push MWI notification.	Need to connect to Tenant Service in other regions.

	Voice Voicemail Service	Voice Tenant Service
		Bandwidth for Redis cross-region connection.
External connections	Redis, Storage Account	Redis and Kafka: Supports secured (TLS) connection. Postgres: Supports secured (TLS, simple) connection between Tenant and Postgres server.
Pod Security Policy	All containers running as Genesys user (500) and non-root user	All containers running as Genesys user (500) and non-root user
SMTP Settings	SMTP enabled	Not applicable
TLS/SSL Certificates configurations	Not applicable	Not applicable
Ingress	Not applicable	Not applicable
Subnet sizing		Network bandwidth must be sufficient to handle the volume of data to be transferred into and out of Kafka and Redis. Subnet sizing to accommodate N+1 Tenant pods.
CNI for Direct Pod Routing	Not applicable	Not applicable

Browser requirements

Not applicable

Genesys dependencies

For detailed information about the correct order of services deployment, see Order of services deployment.

GDPR support

Multi-Tenant Inbound Voice: Voicemail Service

Customer data that is likely to identify an individual, or a combination of other held data to identify an individual is considered as Personally Identifiable Information (PII). Customer name, phone number, email address, bank details, and IP address are some examples of PII.

According to EU GDPR:

- When a customer requests to access personal data that is available with the contact center, the PII associated with the client is exported from the database in client-understandable format. You use the **Export Me** request to do this.
- When a customer requests to delete personal data, the PII associated with that client is deleted from the database within 30 days. However, the Voicemail service is designed in a way that the Customer PII data is deleted in one day using the **Forget Me** request.

Both **Export Me** and **Forget Me** requests depend only on Caller ID/ANI input from the customer. The following PII data is deleted or exported during the **Forget Me** or **Export Me** request process, respectively:

- Voicemail Message
- Caller ID/ANI

GDPR feature is supported only when **StorageInterface' *is configured as*** *BlobStorage****, and*** *Voicemail service* is configured with Azure storage account data store.

Adding caller_id tag during voicemail deposit

Index tag **caller_id** is included in voicemail messages and metadata blob files during voicemail deposit. Using the index tags, you can easily filter the **Forget Me** or **Export Me** instead of searching every mailbox.

GDPR multi-region support

In voicemail service, all voicemail metadata files are stored in master region and voicemail messages are deposited/stored in the respective region. Therefore, It is required to connect all the regions of a tenant to perform Forget Me, Undo Forget Me, or Export Me processes for GDPR inputs.

To provide multi-region support for GDPR, follow these steps while performing GDPR operation:

1. Get the list of regions of a tenant.
2. Ensure all regions storage accounts are up. If any one of storage accounts is down, you cannot perform the GDPR operation.
3. GDPR operates in the master region files, first.
4. Then, GDPR operates in all the non-master region files.

Consul requirements for Voice services

Contents

Find details about Voice services settings that you must configure in Consul before you proceed to configure the Voice Microservices. Some of the configuration in Consul must be performed when you deploy Consul.

Related documentation:

-
-
-

RSS:

- For private edition

Before you deploy the Voice Services, you must deploy the infrastructure services. See Third-party prerequisites for the list of required infrastructure services.

It is your responsibility to deploy and manage all required third-party services, however – in addition to any other Consul configuration you require – there are specific Consul features that you must enable for Voice services.

Complete the work on this page before you make any changes described in Configure Voice Microservices.

Configure Consul features for Voice services

You can find system-level information about Consul on the Software requirements, Network settings, and Order of services deployment pages in *Setting up Genesys Multicloud CX Private Edition*.

When you deploy Consul, you must enable the following features for the Voice services:

- connectinject – To deploy sidecar containers in Voice pods.
- controller – To provide service intention functionality.
- syncCatalog – To sync Kubernetes services to Consul. Set **toK8S: false** and **addK8SNamespaceSuffix: false** for syncing services from Kubernetes to Consul.
- AccessControlList – To enable ACL, set **manageSystemACLs: true**.
- storageclass – To set the storage class to a predefined storage class.
- TLS – To enable TLS, set **enabled: true**. Additional information is required to set up TLS; the following sample includes that information.

The following sample shows the features configuration in Consul:

```
# config.yaml
```

```
global:
  name: consul
  tls:
    enabled: true
    caCert:
      secretName: consul-ca-cert
      # The key of the Kubernetes secret.
      secretKey: tls.crt
    caKey:
      # The name of the Kubernetes secret.
      secretName: consul-ca-key
      # The key of the Kubernetes secret.
      secretKey: tls.key
  acls:
    manageSystemACLs: true
connectInject:
  enabled: true
controller:
  enabled: true
syncCatalog:
  enabled: true
  toConsul: true
  toK8S: false
  addK8SNamespaceSuffix: false
```

Create a Consul bootstrap token

When you enable an Access Control List (ACL) in Consul, you must ensure that Voice services have access to read and write to Consul. To provide access, you create a token with permissions for Voice services in the Consul UI.

1. You can create the ACL bootstrap token when you deploy Consul, although it is possible to do this configuration later as part of the Voice Services deployment. You use the bootstrap token to log into the Consul UI to create a new ACL. Use the following command to get the bootstrap token:

   ```
   kubectl get secret consul-bootstrap-acl-token -n  -o go-template='{{.data.token |
   base64decode}}
   ```

2. Create a new token to which you'll assign the permissions required for Voice services. For example, we'll create a token with a value of a7529f8a-1146-e398-8bd7-367894c4b37b. You create a Kubernetes secret with this token. For example:

   ```
   kubectl create secret generic consul-voice-token -n voice --from-literal='consul-
   consul-voice-token=a7529f8a-1146-e398-8bd7-367894c4b37b'
   ```

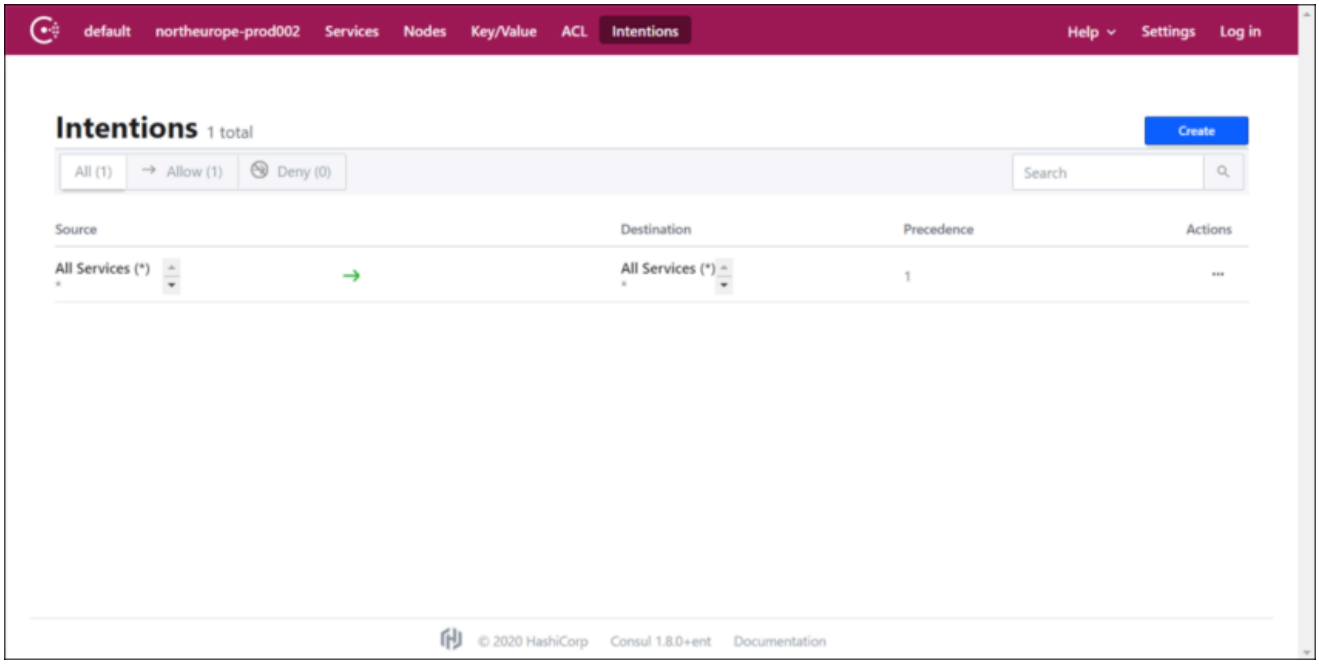
3. Create a policy (voice-policy) with the following list of permissions and assign it to the new token:

   ```
   service_prefix "" {
     policy = "read"
     intentions = "read"
   }
   service_prefix "" {
     policy = "write"
     intentions = "write"
   }
   node_prefix "" {
     policy = "read"
   }
   ```

```
node_prefix "" {
  policy = "write"
}
agent_prefix "" {
  policy = "read"
}
agent_prefix "" {
  policy = "write"
}
session_prefix "" {
  policy = "write"
}
session_prefix "" {
  policy = "read"
}
namespace_prefix "" {
 key_prefix "" {
  policy = "write"
 }
 session_prefix "" {
  policy = "write"
 }
}
key_prefix "" {
  policy = "read"
 }
key_prefix "" {
  policy = "write"
 }
```

Create Intentions in the Consul UI

Voice services use the Consul service mesh to connect between services. Consul has provision to either allow or deny the connection between services. This is done using *intentions*. Log into the **Intentions** tab using the bootstrap token and create a new intention to allow all source services to all destination services as shown in the following screenshot.


default
northeurope-prod002
Services
Nodes
Key/Value
ACL
Intentions
Help
Settings
Log in
Intentions 1 total
Create
All (1)
Allow (1)
Deny (0)
Search
Source
Destination
Precedence
Actions
All Services (*)
All Services (*)
1
© 2020 HashiCorp
Consul 1.8.0+ent
Documentation

Redis requirements for Voice services

Contents

Register services and endpoints that connect to Redis.

Related documentation:

-
-
-

RSS:

- For private edition

Before you deploy the Voice services, you must deploy the infrastructure services. See Third-party prerequisites for the list of required infrastructure services. It is your responsibility to deploy and manage all required third-party services.

This page describes how to register Redis services in Consul to enable connections from the Voice services. Complete the work on this page before you make any changes described in Configure Voice Microservices.

Register the Redis service in Consul

After you create the Redis cluster, register the Redis IP address with Consul. You must create cluster information for the Kubernetes services and endpoints that connect to Redis. Once the Kubernetes services are created, Consul will automatically sync those services and register them in Consul.

Create Kubernetes services and endpoints

Perform Redis registration for all of the following Redis service names. The Voice services use these service names to connect to the Redis cluster.

```
redis-agent-state
redis-call-state
redis-config-state
redis-ors-state
redis-ors-stream
redis-registrar-state
redis-rq-state
redis-sip-state
redis-tenant-stream
```

Manifest file

For all the preceding Redis service names, create a separate service and endpoint using the following example:

```
apiVersion: v1
kind: Service
metadata:
  name:  (ex, redis-agent-state)
  namespace:  (ex, voice)
  annotations:
    "consul.hashicorp.com/service-sync": "true"
spec:
  clusterIP: None
---
apiVersion: v1
kind: Endpoints
metadata:
  name:  (ex, redis-agent-state)
  namespace:  (ex, voice)
subsets:
  - addresses:
      - ip:  (ex, 51.143.122.147)
    ports:
      - port:  (ex, 6379)
        name: redisport
        protocol:  (ex, TCP)
```

Use the following command to get the cluster IP for the Redis service:

```
kubectl get service infra-redis-redis-cluster -n infra -o jsonpath='{.spec.clusterIP}'
```

Configure Voice Microservices

Contents

Learn how to configure Voice Microservices.

Related documentation:

-
-
-

RSS:

- For private edition

Override Helm chart values

For general information about overriding Helm chart values, see Overriding Helm Chart values in the *Genesys Multicloud CX Private Edition Guide*.

When deploying Voice services, certain parameters must be enabled or modified based on customer requirements and environment. For each of the Voice services, an override **values.yaml** file must be created that overrides certain sections of the default configuration for the service. In this document, we use the following format for creating an override **values.yaml** file: **_override_values.yaml**.

The **override_values.yaml** file contains the following sections:

- Deployment
- Image
- Config
- Secrets
- HPA
- Resources
- Log volume

Deployment section

This section can be used to specify minimum and max instances that will be started for each service. By default, the minimum replica count is 1, and the maximum replica count is 10. You can modify it per your load requirements. For RQ service alone it is recommended to set replica count to 2 or more based on load for high availability.

```
deployment:
namespace: voice      # Namespace of voice service
replicaCount: 1       # Min replica count when service is deployed
```

```
maxReplicas: 10       # Max replica count to which the service will scale.
```

Image section

This section has information about the registry from which the voice services will be deployed.

```
image:
  registry: pureengage-docker-staging.jfrog.io  # registry from where image needs to be
deployed
  pullPolicy: Always                            # whether to pull image always
  imagePullSecrets: "mycred"                    # Secrets needed for pulling image from
registry
```

Config section

The config section contains configuration parameters that need to be overridden for all voice services.

Additional information needs to be passed for SIP Cluster Service: dnsServer. Get the DNS Server value from the above section (Configure DNS server for voice-sip).

```
# Set the redis port to be used.
context:
  envs:
    redis:
      port: 6379             # Redis port
    dnsServer: "10.202.0.10" # DNS server address. Needed only for SIP Service.
```

Secrets section

This section captures all the secrets needed by voice services for connecting to infraservices (Consul, Kafka, Redis). The default values for Redis and Kafka secrets are the same as what is created above.

```
# set the secrets
secrets:
  redisCache:
    general:
      enabled: true
  consulACL:
    volumes:
    - name: consul-shared-secret
      secret:
        secretName: consul-voice-token
```

HPA section

The HPA section captures whether HPA is enabled for a service or not and what is the CPU and memory percentage used for scale up and scale down. Common HPA for the following voice services: Agent Service, Config Service, Call State Service, Registrar Service, SIP Front End service, Dial Plan Service.

```
hpa:
  targetCPUPercent: 60         # Average CPU percentage which determine scale up and down
  targetMemoryPercent: 60      # Average Memory percentage which determine scale up and down
  enabled: true                # Horizontal Pod scalar enabled
```

For SIP Proxy and RQ Services, HPA is set to false:

```
hpa:
  enabled: false                # Horizontal Pod scalar enabled
```

For SIP Cluster and Orchestration Services, HPA is set as follows:

```
hpa:
  targetCPUPercent: 50         # Average CPU percentage which determine scale up and down
  targetMemoryPercent: 50      # Average Memory percentage which determine scale up and down
  enabled: true                # Horizontal Pod scalar enabled
```

Resources section

This section captures the resource request and limits for each voice service. The default resource given below is set for each service. You can modify this request and limit based on your load requirement.

```
resources:
  requests:
    cpu: "250m"
    memory: "256Mi"
  limits:
    cpu: "500m"
    memory: "512Mi"
```

For Orchestration and SIP Cluster Services, the CPU and memory requirement is high. Genesys recommends the following setting:

```
resources:
  requests:
    cpu: "500m"
    memory: "1Gi"
  limits:
    cpu: "1500m"
    memory: "4Gi"
```

Log volume

This section captures parameters pertaining to log volumes needed by SIP Cluster Service. These parameters are needed for storing logging of SIP Server binary, which runs inside the SIP Cluster Service. Configure the values for **storageClass** and **volumeName** based on the recommendation given in the Persistent Volume section.

```
# pvc will be created for logs
volumes:
  pvcLog:
    create: true
    claim: sip-log-pvc
    storageClass:
    volumeName:

  pvcJsonLog:
    create: true
    claim: sip-json-log-pvc
    storageClass:
    volumeName:
```

```
  log:
    mountPath:

  jsonLog:
    mountPath:
```

Configure Kubernetes

For information, see the following resources:

- Override Helm chart values
- Configure security
- Secrets for Voice services
- Deploy Voice Microservices

Configure security

Before you deploy the Voice Microservices, be sure to read Security Settings in the *Setting up Genesys Multicloud CX Private Edition* guide.

Security context configuration

The security context settings define the privilege and access control settings for pods and containers. For more information, see the Kubernetes documentation.

By default, the user and group IDs are set in the **values.yaml** file as `500:500:500`, meaning the **genesys** user.

```
containerSecurityContext:
  readOnlyRootFilesystem: false
  runAsNonRoot: true
  runAsUser: 500
  runAsGroup: 500

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

Secrets for Voice services

Create the following Kubernetes secrets for other infrastructure services:

1. Kafka
2. docker-registry

3. Redis

Kafka secrets

Kafka secrets must be created when Kafka is deployed. The secret is referenced in the Voice Microservices **values.yaml** file.

When Kafka is deployed without authentication, create the secret for Kafka as follows:

```
kubectl create secret generic -n voice kafka-secrets-token --from-literal=kafka-
secrets={\"bootstrap\":}
for ex, kubectl create secret generic -n voice kafka-secrets-token --from-literal=kafka-
secrets={\"bootstrap\":\"infra-kafka-cp-kafka.infra.svc.cluster.local:9092\"}
```

When Kafka is deployed with authentication, create the secret for Kafka using this method:

```
kubectl create secret generic -n voice kafka-secrets-token --from-literal=kafka-
secrets={\"bootstrap\":, \"username\": , \"password\": }
for ex, kubectl create secret generic -n voice kafka-secrets-token --from-literal=kafka-
secrets={\"bootstrap\":\"infra-kafka-cp-
kafka.infra.svc.cluster.local:9092\",\"username\":\"kafka-user\",\"password\":\"kafka-
password\"}
```

Redis secrets

Ensure Redis is installed before you deploy the Voice Services.

Use the following commands to create Redis secrets:

```
export REDIS_PASSWORD=$(kubectl get secret infra-redis-redis-cluster -n infra -o
jsonpath="{.data.redis-password}" | base64 --decode)
kubectl create secret generic -n voice redis-agent-token --from-literal=redis-agent-
state={\"password\":\"$REDIS_PASSWORD\"}
kubectl create secret generic -n voice redis-callthread-token --from-literal=redis-call-
state={\"password\":\"$REDIS_PASSWORD\"}
kubectl create secret generic -n voice redis-config-token --from-literal=redis-config-
state={\"password\":\"$REDIS_PASSWORD\"}
kubectl create secret generic -n voice redis-tenant-token --from-literal=redis-tenant-
stream={\"password\":\"$REDIS_PASSWORD\"}
kubectl create secret generic -n voice redis-registrar-token --from-literal=redis-registrar-
state={\"password\":\"$REDIS_PASSWORD\"}
kubectl create secret generic -n voice redis-sip-token --from-literal=redis-sip-
state={\"password\":\"$REDIS_PASSWORD\"}
kubectl create secret generic -n voice redis-ors-stream-token --from-literal=redis-ors-
stream={\"password\":\"$REDIS_PASSWORD\"}
kubectl create secret generic -n voice redis-ors-token --from-literal=redis-ors-
state={\"password\":\"$REDIS_PASSWORD\"}
kubectl create secret generic -n voice redis-rq-token --from-literal=redis-rq-
state={\"password\":\"$REDIS_PASSWORD\"}
```

JFrog secrets

Use the following commands to create JFrog secrets:

```
kubectl create secret docker-registry   --docker-server= --docker-username="$JFROG_USER" --
docker-password="$JFROG_PASSWORD" -n voice
```

Provision Voice Microservices

Contents

- Administrator

Learn how to provision Voice Microservices.

Related documentation:

-
-
-

RSS:

- For private edition

Other than the provisioning of Tenant and Voicemail Services, no specific provisioning is required for the Voice services.

Tenant provisioning

For information about Tenant provisioning, see Provision the Tenant Service.

Voicemail provisioning

For information about voicemail provisioning, see Provision the Voicemail Service.

Deploy Voice Microservices

Contents

Learn how to deploy Voice Microservices 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.

> **Important**
>
> Make sure to review Before you begin for the full list of prerequisites required to deploy Voice Microservices.
>
> To deploy the Tenant service, see the *Tenant Service Private Edition Guide*.
>
> For information about deploying Voicemail Service, see Deploy Voicemail.

General deployment prerequisites

Before you deploy the Voice Services, you must deploy the infrastructure services. See Third-party prerequisites for the list of required infrastructure services.

In addition, see Consul requirements for Voice services and Redis requirements for Voice services for information about specific configuration that must be completed in Consul before you configure or deploy Voice Microservices.

To override values for both the infrastructure services and voice services, see Override Helm chart values.

Deployment order for Voice Microservices

Genesys recommends the following order of deployment for the Voice Microservices:

- Voice Services
- Tenant Service
- Voicemail Service

Create the Voice namespace

Before deploying Voice Services and their dependencies, create a namespace using the following command:

```
kubectl create ns voice
```

In all Voice Services and the configuration files of their dependencies, the namespace is **voice**. If you want a specific, custom namespace, create the namespace (using the preceding command) and remember to change the namespace in files, as required.

Deploy Voice services

Voice Services require a Persistent Volume Claim (PVC); the Voice SIP Cluster Service uses a persistent volume to store traditional SIP Server logs. Before deploying Voice Services, create the PVC.

Storage class and Claim name

The created persistent volume must be configured in the **sip_node_override_values.yaml** file as shown below:

```
# pvc will be created for logs
volumes:
  pvcLog:
    create: true
    claim: sip-log-pvc
    storageClass: voice
    volumeName:  (ex sip-log-pv)

  pvcJsonLog:
    create: true
    claim: sip-json-log-pvc
    storageClass: voice
    volumeName:  (ex sip-log-pv)
```

Configure the DNS Server for voice-sip

The Voice SIP Cluster Service requires the DNS server to be configured in its **sip_node_override_values.yaml** file. Follow the steps in the Kubernetes documentation to install a **dnsutils** pod. Using the **dnsutils** pod, get the **dnsserver** that's used in the environment.

The default value in the SIP Helm chart is `10.0.0.10`. If the **dnsserver** address is different, update it in the **sip_node_override_values.yaml** file as shown below:

```
# update dns server ipaddress
  context:
    envs:
      dnsServer: "10.202.0.10"
```

Voice Service Helm chart deployment

Deploy the Voice Services using the provided Helm charts.

```
helm upgrade --install --force --wait --timeout 300s -n voice -f ./voice_helm_values/
agent_override_values.yaml voice-agent /voice-agent-.tgz --set version= --username
"$JFROG_USER" --password "$JFROG_PASSWORD"

helm upgrade --install --force --wait --timeout 300s -n voice -f ./voice_helm_values/
callthread_override_values.yaml voice-callthread /voice-callthread-.tgz --set version= --
username "$JFROG_USER" --password "$JFROG_PASSWORD"

helm upgrade --install --force --wait --timeout 200s -n voice -f ./voice_helm_values/
config_override_values.yaml voice-config /voice-config-.tgz --set version= --username
"$JFROG_USER" --password "$JFROG_PASSWORD"

helm upgrade --install --force --wait --timeout 300s -n voice -f ./voice_helm_values/
dialplan_override_values.yaml voice-dialplan /voice-dialplan-.tgz --set version= --username
"$JFROG_USER" --password "$JFROG_PASSWORD"

helm upgrade --install --force --wait --timeout 200s -n voice -f ./voice_helm_values/
ors_node_override_values.yaml voice-ors /voice-ors-.tgz --set version= --username
"$JFROG_USER" --password "$JFROG_PASSWORD"

helm upgrade --install --force --wait --timeout 300s -n voice -f ./voice_helm_values/
registrar_override_values.yaml voice-registrar /voice-registrar-.tgz --set version= --
username "$JFROG_USER" --password "$JFROG_PASSWORD"

helm upgrade --install --force --wait --timeout 200s -n voice -f ./voice_helm_values/
rq_node_override_values.yaml voice-rq /voice-rq-.tgz --set version= --username "$JFROG_USER"
--password "$JFROG_PASSWORD"

helm upgrade --install --force --wait --timeout 200s -n voice -f ./voice_helm_values/
sip_node_override_values.yaml voice-sip /voice-sip-.tgz --set version= --username
"$JFROG_USER" --password "$JFROG_PASSWORD"

helm upgrade --install --force --wait --timeout 300s -n voice -f ./voice_helm_values/
sipfe_override_values.yaml voice-sipfe /voice-sipfe-.tgz --set version= --username
"$JFROG_USER" --password "$JFROG_PASSWORD"

helm upgrade --install --force --wait --timeout 300s -n voice -f ./voice_helm_values/
sipproxy_override_values.yaml voice-sipproxy /voice-sipproxy-.tgz --set version= --username
"$JFROG_USER" --password "$JFROG_PASSWORD"
```

The following table contains a list of the minimum recommended Helm chart versions that should be used:

Service name	Helm chart version
voice-config	voice-config-9.0.11.tgz
voice-dialplan	voice-dialplan-9.0.08.tgz
voice-registrar	voice-registrar-9.0.14.tgz
voice-agent	voice-agent-9.0.10.tgz
voice-callthread	voice-callthread-9.0.12.tgz
voice-sip	voice-sip-9.0.22.tgz
voice-sipfe	voice-sipfe-9.0.06.tgz
voice-sipproxy	voice-sipproxy-9.0.09.tgz
voice-rq	voice-rq-9.0.08.tgz
voice-ors	voice-ors-9.0.08.tgz

Deploy the Tenant service

The Tenant Service is included with the Voice Microservices, but has its own deployment procedure. To deploy the Tenant Service, see Deploy the Tenant Service.

Validate the deployment

Follow the steps below to validate the successful deployment of voice microservices.

1. Verify the helm deployments using the following command.

   ```
   helm list -n voice
   ```

   Sample output:

   ```
   NAME                                        NAMESPACE       REVISION
    UPDATED                                  STATUS          CHART

    APP VERSION
    voice-agent-latest                                  voice           4
    2022-08-18 13:22:12.355810905 +0000 UTC deployed        voice-
    agent-100.0.1000006           1.0
    voice-callthread-latest                             voice           70
    2022-08-18 09:44:07.078583581 +0000 UTC deployed        voice-
    callthread-100.0.1000006      1.0
    voice-config-latest                                 voice           61
    2022-08-19 01:33:02.039668264 +0000 UTC deployed        voice-
    config-100.0.1000006          1.0
    voice-dialplan-latest                               voice           5
    2022-08-18 12:33:31.223393121 +0000 UTC deployed        voice-
    dialplan-100.0.1000009        1.0
    voice-ors-latest                                    voice           1
   ```

```
2022-08-15 21:40:32.013855856 +0000 UTC deployed        voice-
ors-100.0.1000018             1.0
voice-registrar-latest                          voice           108
2022-08-18 13:41:26.37007884 +0000 UTC  deployed        voice-
registrar-100.0.1000007       latest-aa9f28a
voice-rq-latest                                 voice           14
2022-08-18 13:44:07.187279228 +0000 UTC deployed        voice-
rq-100.0.1000004              1.0
voice-sip-latest                                voice           193
2022-08-10 23:06:05.057511521 +0000 UTC deployed        voice-
sip-100.0.1000018             1.0
voice-sipfe-latest                              voice           73
2022-08-10 23:49:45.166013304 +0000 UTC deployed        voice-
sipfe-100.0.1000006           1.0
voice-sipproxy-latest                           voice           5
2022-08-11 17:13:30.894221491 +0000 UTC deployed        voice-
sipproxy-100.0.1000007        1.0
voice-voicemail-latest                          voice           67
2022-08-18 15:18:47.347509225 +0000 UTC deployed        voice-
voicemail-100.0.1000015       1.0
```

2. Verify readiness state of Kubernetes objects using the kubectl commands.
    1. Run the following command to check the deployments:

        ```
        kubectl get deployments -n voice
        ```

        Sample output:

        ```
        NAME                  READY   UP-TO-DATE   AVAILABLE   AGE
        voice-agent           2/2     2            2           40d
        voice-callthread      3/3     3            3           704d
        voice-config          1/1     1            1           704d
        voice-dialplan        1/1     1            1           41d
        voice-registrar       1/1     1            1           703d
        voice-sip-debug-kpan  2/2     2            2           68d
        voice-sipfe           3/3     3            3           727d
        voice-voicemail       1/1     1            1           87d
        ```

    2. Run the following command to check the Statefulsets:

        ```
        kubectl get statefulset -n voice
        ```

        Sample output:

        ```
        NAME             READY   AGE
        voice-ors        50/50   40d
        voice-rq         20/20   40d
        voice-sip        30/30   703d
        voice-sipproxy   5/5     40d
        ```

3. Check if all the pods are running and in Ready state.
    1. Run the following command to check the readiness of the pods.

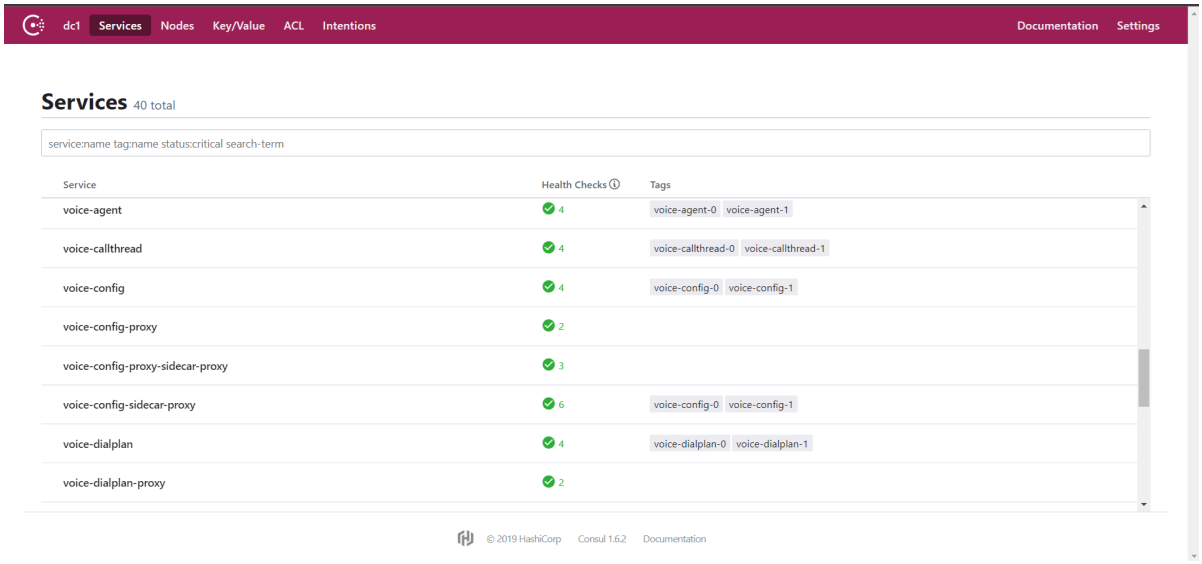
        ```
        kubectl get pods -n voice
        ```

        Sample output:

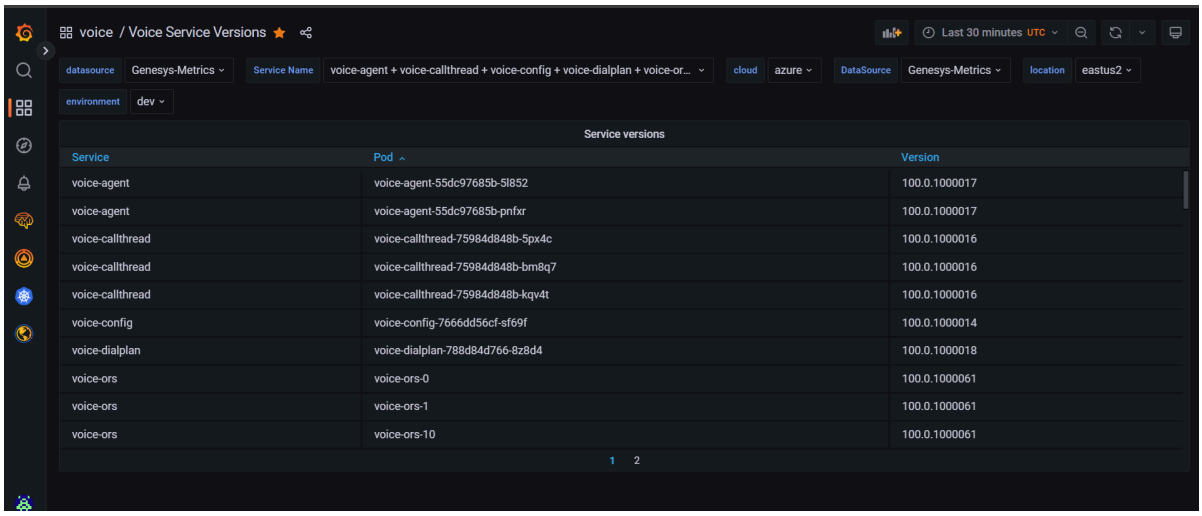
        ```
        NAME                                READY   STATUS        RESTARTS        AGE
        t2100-0                             3/3     Running       0
        4d23h
        voice-agent-55dc97685b-pnfxr        2/2     Running       0
        ```

```
170m
voice-callthread-75984d848b-bm8q7       2/2     Running       0
170m
voice-callthread-75984d848b-kqv4t       2/2     Running       0
170m
voice-config-7666dd56cf-sf69f           2/2     Running       0              39h
voice-dialplan-788d84d766-8z8d4         2/2     Running       0              37h
voice-ors-0                             2/2     Running       0              18h
voice-ors-1                             2/2     Running       0
6d5h
voice-registrar-6c54c6bc9-tkvk2         2/2     Running       0              39h
voice-rq-0                              2/2     Running       0              38h
voice-rq-1                              2/2     Running       0
4d17h
voice-sip-0                             3/3     Running       0              39h
voice-sip-1                             3/3     Running       0              11d
voice-sipfe-56c7bc77dd-7fpkh            2/2     Running       0
170m
voice-sipproxy-0                        2/2     Running       0              11d
voice-voicemail-66f745448b-wqmfc        2/2     Running       0
4d20h
```

4. Verify the health status of the pods in Consul dashboard.
   If the services are running and in Ready state, the health check will be marked as Green in Consul dashboard.



5. Check the versions of microservices in Grafana dashboard.
   Only if voice-dashboards are deployed in the voice namespace, you can perform this check in the dashboard.



6. Check for any crash, KafkaJS or Redis connection errors in Prometheus, Grafana dashboards and/or logs of the respective microservices.

From a functional point of view, you can validate the voice microservices deployment by performing the following steps.

1. Before you can validate the voice microservices, you must create few objects in the Tenant configdb to start the verification.
    1. Port forward the Tenant instance at 8888 port and access the tenant objects through Configuration Manager application.

        ```
        kubectl port forward t2100-0 8888:8888 -n voice
        ```

    2. Create a few Directory Numbers (DNs) under the Sip_Cluster switch with the following options:

        ```
        [TServer]
                contact=*
                dual-dialog-enabled=false
                infra-class=2
                make-call-rfc3725-flow=1
                refer-enabled=false
                sip-cti-control=talk,hold
                sip-ring-tone-mode=1
                use-contact-as-dn=true
                use-register-for-service-state=false
        ```

    3. Create a Place object and map the DNs created.
    4. Create new Agents with username and password, under the "Persons" section.
    5. Map the Place to the agent.
2. Once the objects are created successfully, follow the steps below to validate the voice microservices deployment..
    1. Register the DNs from Endpoints.
    2. Login/Logout the Agents from Workspace Web Edition or a similar application and change the states - Ready, Not Ready and Logout.
    3. Make few test calls between the agents.

4. Perform other call functionalities like - hold/retrieve, conference, transfer, after call work, and so on.
5. If Designer is available, load different strategies onto route points (external facing SBC Numbers) and validate if the inbound call made from PSTN is being routed to the agent/skill group configured.

3. Additionally, you can also check the below after the deployment of voice microservices.
    1. Verify whether the Grafana dashboards of the voice microservices are updated with relevant data and they reflect the status of the services correctly.
    2. Check if the alerts and alarms are configured for the voice microservices and are active.

Upgrade, rollback, or uninstall Voice Microservices

Contents

Learn how to upgrade, rollback or uninstall Voice Microservices.

Related documentation:

-
-
-

RSS:

- For private edition

Upgrade Voice Microservices

Because Voice Services are real-time services, you use canary-based deployment to upgrade. Canary deployment is a technique of deploying the new software version to one or more *canary* instances. You verify that the new version works as expected and that is also works with the previous version. Deploying only one or two canary instances is sufficient to discover a faulty version and to minimize the risk of adding a new version into production. After you have verified that the new software version works correctly, you can proceed with the upgrade.

The upgrade procedure consists of these major steps:

1. Canary deployment
2. Upgrade
3. Delete canary

Canary deployment

For any new Voice service version, you first deploy a *canary* instance of it. After you confirm that the new canary version is performing correctly, you roll out the version to all instances of the Voice service using the procedure described in the Service upgrade section. For the Voice RQ service, see Upgrade the RQ node service.

IMPORTANT

When upgrading from early Helm versions to versions equal to or later than the following, delete the configmap (`service name>-configmap`) for each service before deploying the canary instance.

Service	Version
Agent State	100.0.100.0003
Call State	100.0.100.0003

Service	Version
Config	100.0.100.0003
Dial Plan	100.0.100.0006
ORS	100.0.100.0007
Registrar	100.0.100.0003
SIP Cluster	100.0.100.0009
Front End	100.0.100.0003
SIP Proxy	100.0.100.0003

For the canary deployment, some parameters in the **canary_override_values.yaml** file must be overridden. The following sample shows the overrides. The **canary_override_values.yaml** file is passed to the Helm chart during the deployment of the canary instance. When upgrading SIP Cluster Service to version 100.0.100.0009 or later, there are some changes to the following sample for some sections. To review the changes, see the **canary_override_values.yaml** file for SIP Cluster Service sample.

```
# serviceaccount is created during initial deployment
serviceAccount:
create: false

deployment:
postfix: canary

# configmap is already created during initial deployment
context:
create: false

# this is needed for SIP canary only
loggingSidecar:
context:
create: false

# this is also needed for SIP canary only
volumes:
pvcLog:
create: false
pvcJsonLog:
create: false

# podmonitor is not needed for canary, but metric server enabling is needed
prometheus:
podMonitor:
enabled: false
metricServer:
enabled: true

# canary does not need HPA
hpa:
enabled: false
```

Sample: **canary_override_values.yaml** file for SIP Cluster Service version 100.0.100.0009 and later

Starting with version 100.0.100.0009, the **context**, **volumes**, and **logging sidecar** sections in the Voice SIP Cluster Service **canary_override_values.yaml** file differ from the preceding sample. The

following sample shows the changes.

```
context:
  create: true
  envs:
    sbcAddress: ""
    enableSharedTrunk: true # Enable/Disable shared trunks configured in SIPNode
    enableSharedSoftswitch: true # Enable/Disable shared softswitches configured in SIPNode

volumes:
  pvcLog:
    create: false

  pvcJsonLog:
    create: false

  log:
    mountPath: /opt/genesys/logs/volume
    volumePath: /mnt/log

  jsonLog:
    emptyDir: true
    mountPath: "/opt/genesys/logs/sip_node/JSON"

loggingSidecar:
  image:
    registry: genesysengageprod001.azurecr.io
    repository: sre/fluent-bit
    pullPolicy: Always
    tag: 1.6.1
```

The following commands deploy a canary instance:

```
helm upgrade --install --force --wait --timeout 300s -n voice -f ./voice_helm_values/
canary_override_values.yaml voice-agent-canary /voice-agent-.tgz --set version= --username
"$JFROG_USER" --password "$JFROG_PASSWORD"

helm upgrade --install --force --wait --timeout 300s -n voice -f ./voice_helm_values/
canary_override_values.yaml voice-callthread-canary /voice-callthread-.tgz --set version= --
username "$JFROG_USER" --password "$JFROG_PASSWORD"

helm upgrade --install --force --wait --timeout 200s -n voice -f ./voice_helm_values/
canary_override_values.yaml voice-config-canary /voice-config-.tgz --set version= --username
"$JFROG_USER" --password "$JFROG_PASSWORD"

helm upgrade --install --force --wait --timeout 300s -n voice -f ./voice_helm_values/
canary_override_values.yaml voice-dialplan-canary /voice-dialplan-.tgz --set version= --
username "$JFROG_USER" --password "$JFROG_PASSWORD"

helm upgrade --install --force --wait --timeout 200s -n voice -f ./voice_helm_values/
canary_override_values.yaml voice-ors-canary /voice-ors-.tgz --set version= --username
"$JFROG_USER" --password "$JFROG_PASSWORD"

helm upgrade --install --force --wait --timeout 300s -n voice -f ./voice_helm_values/
canary_override_values.yaml voice-registrar-canary /voice-registrar-.tgz --set version= --
username "$JFROG_USER" --password "$JFROG_PASSWORD"

helm upgrade --install --force --wait --timeout 200s -n voice -f ./voice_helm_values/
canary_override_values.yaml voice-rq-canary /voice-rq-.tgz --set version= --username
"$JFROG_USER" --password "$JFROG_PASSWORD"

helm upgrade --install --force --wait --timeout 200s -n voice -f ./voice_helm_values/
canary_override_values.yaml voice-sip-canary /voice-sip-.tgz --set version= --username
```

```
"$JFROG_USER" --password "$JFROG_PASSWORD"

helm upgrade --install --force --wait --timeout 300s -n voice -f ./voice_helm_values/
canary_override_values.yaml voice-sipfe-canary /voice-sipfe-.tgz --set version= --username
"$JFROG_USER" --password "$JFROG_PASSWORD"

helm upgrade --install --force --wait --timeout 300s -n voice -f ./voice_helm_values/
canary_override_values.yaml voice-sipproxy-canary /voice-sipproxy-.tgz --set version= --
username "$JFROG_USER" --password "$JFROG_PASSWORD"
```

Service upgrade

After you validate the canary deployment of a Voice service, use the following commands to upgrade the current version of a Voice service to the new version:

```
helm upgrade --install --force --wait --timeout 300s -n voice -f ./voice_helm_values/
agent_override_values.yaml voice-agent /voice-agent-.tgz --set version= --username
"$JFROG_USER" --password "$JFROG_PASSWORD"

helm upgrade --install --force --wait --timeout 300s -n voice -f ./voice_helm_values/
callthread_override_values.yaml voice-callthread /voice-callthread-.tgz --set version= --
username "$JFROG_USER" --password "$JFROG_PASSWORD"

helm upgrade --install --force --wait --timeout 200s -n voice -f ./voice_helm_values/
config_override_values.yaml voice-config /voice-config-.tgz --set version= --username
"$JFROG_USER" --password "$JFROG_PASSWORD"

helm upgrade --install --force --wait --timeout 300s -n voice -f ./voice_helm_values/
dialplan_override_values.yaml voice-dialplan /voice-dialplan-.tgz --set version= --username
"$JFROG_USER" --password "$JFROG_PASSWORD"

helm upgrade --install --force --wait --timeout 200s -n voice -f ./voice_helm_values/
ors_node_override_values.yaml voice-ors /voice-ors-.tgz --set version= --username
"$JFROG_USER" --password "$JFROG_PASSWORD"

helm upgrade --install --force --wait --timeout 300s -n voice -f ./voice_helm_values/
registrar_override_values.yaml voice-registrar /voice-registrar-.tgz --set version= --
username "$JFROG_USER" --password "$JFROG_PASSWORD"

helm upgrade --install --force --wait --timeout 200s -n voice -f ./voice_helm_values/
rq_node_override_values.yaml voice-rq /voice-rq-.tgz --set version= --username "$JFROG_USER"
--password "$JFROG_PASSWORD"

helm upgrade --install --force --wait --timeout 200s -n voice -f ./voice_helm_values/
sip_node_override_values.yaml voice-sip /voice-sip-.tgz --set version= --username
"$JFROG_USER" --password "$JFROG_PASSWORD"

helm upgrade --install --force --wait --timeout 300s -n voice -f ./voice_helm_values/
sipfe_override_values.yaml voice-sipfe /voice-sipfe-.tgz --set version= --username
"$JFROG_USER" --password "$JFROG_PASSWORD"

helm upgrade --install --force --wait --timeout 300s -n voice -f ./voice_helm_values/
sipproxy_override_values.yaml voice-sipproxy /voice-sipproxy-.tgz --set version= --username
"$JFROG_USER" --password "$JFROG_PASSWORD"
```

Delete the canary instance

When the upgrade of a Voice service is successful, use the following commands to delete the canary instance of the service:

```
helm delete voice-agent-canary -n voice
helm delete voice-callthread-canary -n voice
helm delete voice-config-canary -n voice
helm delete voice-dialplan-canary -n voice
helm delete voice-ors-canary -n voice
helm delete voice-registrar-canary -n voice
helm delete voice-sip-canary -n voice
helm delete voice-sipfe-canary -n voice
helm delete voice-sipproxy-canary -n voice
```

Upgrade the RQ node service

The upgrade procedure for the RQ node service differs from other Voice services. Use the following steps to upgrade the Voice RQ service.

1. Set the strategy to **OnDelete** in the **rq_node_override_values.yaml** file (the strategy is set, by default, to **RollingUpdate** when a fresh RQ node service is deployed).
   Example:

   ```
   deployment:
     deploymentType: statefulset
     strategy: OnDelete
   ```

2. Use the following command to upgrade the voice-rq **values.yaml** file to the new version:

   ```
   helm upgrade --install --force --wait --timeout 200s -n voice -f ./voice_helm_values/
   rq_node_override_values.yaml voice-rq https:///voice-rq/voice-rq-9.0.07.tgz --set
   version=9.0.6 --username "$JFROG_USER" --password "$JFROG_PASSWORD"
   ```

3. Delete the voice-rq-0 pod. This causes the voice-rq-0 pod to be automatically upgraded to the new version. The upgraded Helm version is applied to new pods only when a pod is deleted. You can then validate the upgrade using this canary pod (voice-rq-0) to ensure it works with other RQ nodes.
   If you delete other RQ node pods, they are also upgraded automatically to the new version. Genesys recommends that you avoid this type of random upgrade of RQ nodes. Before deleting and upgrading any other RQ pods, use the new version on the canary pod (voice-rq-0) to test and validate the upgrade.

4. If the canary pod (voice-rq-0) works correctly with other pods and in the environment, then you can upgrade the voice-rq Helm **values.yaml** file to the new version (see step 2). When that upgrade is complete, delete the remaining RQ pods. The new RQ node pods have the new version.

Rollback Voice Microservices

For Voice Microservices, you can perform a service rollback at the following times:

1. After performing the canary deployment.
2. After upgrading the service.

After canary deployment

If you deploy the canary instance as a new version and that version is not working as expected, then

you can delete the canary deployment using the following command:

```
helm delete voice-agent-canary -n voice
```

After service upgrade

After you upgrade a service to a new version and that version is found to have issues, then you can roll back to the previous version using the following command:

```
helm upgrade --install --force --wait --timeout 300s -n voice -f ./voice_helm_values/
agent_override_values.yaml voice-agent https://pureengage-helm-staging-local.jfrog.io/voice-
agent-.tgz --set version= --username "$JFROG_USER" --password "$JFROG_PASSWORD"
```

Uninstall Voice Microservices

To uninstall a Voice service, use the following command:

```
helm uninstall  -n voice
```

For more information, see the Helm documentation.

Before you begin

Contents

Find out what to do before deploying the Voicemail Service.

Related documentation:

-
-
-

RSS:

- For private edition

Limitations and assumptions

Voice Voicemail Service is integrated with Workspace Web Edition and Web Services and Applications. The integration brings an appearance of actionable voice mailbox information in the Workspace Web Edition UI, presenting users with Message Waiting Indicator(s) for each voice mailbox assigned to them either directly as a personal mailbox or as a group mailbox via membership in a group(s) having a mailbox provisioned. Users still have access to voicemail from Workspace Web Edition by dialing directly to a voicemail access number, which is 5555.

Download the Helm charts

For information about how to download the Helm charts, see Downloading your Genesys Multicloud CX containers.

The following table identifies the Helm chart version associated with the Voicemail service.

Service name	Helm chart version
voice-voicemail	voice-voicemail-100.0.xx.tgz

Third-party prerequisites

See the Third-party prerequisites for the Voice Services.

Storage requirements

Choosing Voicemail storage

To store mailbox metadata and messages, consider the following supported options for storage in the Private Edition deployment:

1. Persistent Volumes & Persistent Volume Claims
2. Azure Blob Storage
3. AWS S3 Bucket

See the following sections to learn how to use these storage options and to find information about their limitations.

Persistent Volume & Claim

- Persistent Volume (PV) is a piece of storage that can be mounted to a Voicemail Service deployment inside the Kubernetes cluster.
- Voicemail Service requires a separate storage class and PV to be created for a Voicemail storage.
- If the customer wants to extend the deployment to more than one Kubernetes cluster, Voicemail Service requires to mount the same PV for all the Kubernetes cluster for that customer.
- Create the Persistent Volume Claim (PVC) from the Voicemail PV.
- The access mode for the PVC must be **ReadWriteMany**, since the Voicemail Service will edit the existing data while updating the mailbox settings or the message state.
- Use the sizing doc, which you can find on the Genesys SIP Feature Server landing page, to calculate the required storage space.

Here is the sample Kubernetes YAML file for creating PVCs for a Voicemail Service. The PVC creation is controlled by the Voicemail Service Helm chart by overriding the **values.yaml**.

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: voice-voicemail-pvc
  namespace: voice
  labels:
    servicename: voice-voicemail
spec:
  storageClassName: voice-voicemail
  volumeMode: Filesystem
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 20Gi
```

Limitations

1. Replication strategies are not available for the data.
2. Retention limit: Admins can't configure the auto-expiration for a Voicemail message.

3. When a customer has more than one Kubernetes cluster deployed, the PV for all the Kubernetes clusters must be created from a single storage drive, so that the data from one Kubernetes cluster is shared among other Kubernetes clusters.

Azure blob

- Unlike PV, the Azure Blob Storage provides options to replicate and configure Time to live for the files and can be accessed from any Kubernetes cluster by using the storage access keys.
- Create the Azure Storage with the blob storage.
- The access keys for the blob storage must be securely mounted to the Voicemail pod. You can do one of the following:
  - Store access keys in Azure Key Vault and mount it via a Container Storage Interface (CSI) driver.
  - Create access keys as a Kubernetes secret and volume mount the Kubernetes secret. (This option is considered less secured than the CSI driver approach.)
- The **values.yaml** file can be overridden for configuring either a Kubernetes secret or CSI driver, which is explained in Override Helm chart values.

AWS S3 Bucket

- Like Azure Blob Storage, S3 bucket provides options to replicate and configure Time to live for the files and can be accessed from any Kubernetes cluster by using the access and secret keys.
- Create a new S3 bucket or a folder inside the existing bucket.
- The access and secret keys for blob storage needs to be securely mounted to the Voicemail pod as below.
  - Create access/secret keys as Kubernetes secret and volume mount the Kubernetes secret.
- The values.yaml file can be overridden for configuring Kubernetes secret, which is explained in Override Helm chart values.

Network requirements

For more information, see Network requirements in the *Configure and deploy* section of this guide.

Browser requirements

Not applicable.

Genesys dependencies

For information about dependencies for Voicemail Service, see additional prerequisites on the Deploy Voicemail page. For detailed information about the correct order of services deployment, see Order of

services deployment.

GDPR support

Customer data that is likely to identify an individual, or a combination of other held data to identify an individual is considered as Personally Identifiable Information (PII). Customer name, phone number, email address, bank details, and IP address are some examples of PII.

Multi-Tenant Inbound Voice: Voicemail Service

According to EU GDPR:

- When a customer requests to access personal data that is available with the contact center, the PII associated with the client is exported from the database in client-understandable format. You use the **Export Me** request to do this.
- When a customer requests to delete personal data, the PII associated with that client is deleted from the database within 30 days. However, the Voicemail service is designed in a way that the Customer PII data is deleted in one day using the **Forget Me** request.

Both **Export Me** and **Forget Me** requests depend only on Caller ID/ANI input from the customer. The following PII data is deleted or exported during the **Forget Me** or **Export Me** request process, respectively:

- Voicemail Message
- Caller ID/ANI

GDPR feature is supported only when **StorageInterface** is configured as **BlobStorage**, and **Voicemail service** is configured with Azure storage account data store.

Adding caller_id tag during voicemail deposit

Index tag **caller_id** is included in voicemail messages and metadata blob files during voicemail deposit. Using the index tags, you can easily filter the **Forget Me** messages instead of searching every mailbox.

GDPR multi-region support

In voicemail service, all voicemail metadata files are stored in master region and voicemail messages are deposited/stored in the respective region. Therefore, it is required to connect all the regions of a tenant to perform Forget Me, Undo Forget Me, or Export Me processes for GDPR inputs.

To provide multi-region support for GDPR, follow these steps while performing GDPR operation:

1. Get the list of regions of a tenant.
2. Ensure all regions storage accounts are up. If any one of storage accounts is down, you cannot perform the GDPR operation.
3. GDPR operates in the master region files, first.

4. Then, GDPR operates in all the non-master region files.

Standalone Scripts

You can invoke the **Forget Me** and **Export Me** APIs from a standalone Node.js script. This script can be executed by a user or an automated scheduler. When a user executes the script:

- The script authenticates with the user auth token.
- The user must have the bearer or the basic token.

In the case of an automated scheduler, the script uses the client credential (also known as system account) and processes the request. In this scenario, the user has to configure the GWS URL as an environment variable. The script would generate the auth token for the client and access the GDPR APIs. The script can be integrated into the GitHub Actions pipeline and invoked from the GitHub pipeline.

Script Inputs

Parameter	Value	Is it mandatory	Description
-i or --input	file-path	Yes	Input.json is the same as the JSON input passed to the REST API. The client credentials ccid (contact center id) must be included as key-value pair in the input.json file because ccid cannot be fetched from auth token of the client credentials. Sample value "ccid" : "2c5ea4c0-4067-11e9-8bad-9b1deb4d3b7d"

-o or --output	output-location	Yes	output.json
			
			Forget Me Operation:
			
			The output.json is the same as the response from the Forget Me API.
			
			Export Me Operation:
			
			{
			
			"caseid":"123456789",
			"consumers":
			{
			
			//The message media is exported in the output location and the filename is the same as the message IDs.
			"555551212":
			["filename of

			message1", "filename of message2"], "555556161": [], "555556162": ["filename of message1"] } } Undo Operation: The output.json is the same as the response from Undo API. execution.log Execution logs are available in the execution.log file
-u or --user	User token	Either user token or client credentials	User token is fetched from GWS
-c or --client	Client credentials	Either user token or client credentials	Client credential is required when scheduling the script. Client credentials can be obtained by requesting the GWS team.
-p or --operation	forgetme exportme undoforgot	Yes	Type of operation to be done when the script is executed.

User token example:

```
node gdpr.js -i "t2026" -o "t2026" -u "dDIwMjZcXGRlZmF1bHQ6cGFzc3dvvnW=" -p "forgetme" --basic

node gdpr.js -i "t2026" -o "t2026" -u "dDIwMjZcXGRlZmF1bHQ6cGFzc3dvvnW=" -p "undoforget" --
basic

node gdpr.js -i "t2026" -o "t2026" -u "dDIwMjZcXGRlZmF1bHQ6cGFzc3dvvnW=" -p "exportme" --basic
```

Client credentials example:

```
node gdpr.js -i "t2026" -o "t2026" -c "iPdZIMR5qHAohE4wsCOLaOeAopUyJDZalmwN6FPH9rjUcztZ" -p
"forgetme"

node gdpr.js -i "t2026" -o "t2026" -c "iPdZIMR5qHAohE4wsCOLaOeAopUyJDZalmwN6FPH9rjUcztZ" -p
"undoforget"

node gdpr.js -i "t2026" -o "t2026" -c "iPdZIMR5qHAohE4wsCOLaOeAopUyJDZalmwN6FPH9rjUcztZ" -p
"exportme"
```

Limitations

MWI count is not updated automatically on deleting the files during **Forgetme** operation. It is updated during the next voicemail message deposit or voicemail message delete of a mailbox.

- If the **Forgetme** rule is first executed at 10:00 UTC in the day, then the file **X** marked for **Forgetme** at 10.01 UTC same day, the **Forgetme** rule does not delete the file 'X' on the second day at 10:00 UTC since it does not meet the **file has not been modified in one day** condition. However, it gets deleted in next day.
- If the message is deposited and not read by any agent, the **Forget Me** API is executed and marked for deletion. Before deleting the file, if the agent reads the message/forward, the message metadata and the last modified time are updated. In such cases, the file may not be deleted in one day because the last modified date condition is not met.

Configure the Voicemail Service

Contents

Learn how to configure the Voicemail Service.

Related documentation:

-
-
-

RSS:

- For private edition

Override Helm chart values

For general information about overriding Helm chart values, see Overriding Helm Chart values in the *Genesys Multicloud CX Private Edition Guide*.

To assist you, Genesys provides override values for Voice Microservices in the Values.yaml file. If you need to change the default configuration, you must modify the Values file. The following section describes some common changes to these values.

Overriding values.yaml for the Voicemail Service

Common changes:

```
#Configure the GWS Base URL
context:
...
  envs:
    ...
    gwsBaseUrl:  # sample URL https:///auth/v3
    ...
```

You can override the Voicemail Service values for the following parameters based on what kind of storage you're using.

Persistent Volumes & Persistent Volume Claims

For PV and PVCs, use the following example:

```
#Blob Storage to be disabled and no value for mounts and volumes
  blobStorage:
    general:
      mode: 'k8s | csi'
      enabled: false
    mounts:
    volumes:
```

```
context:
...
  envs:
    ...
    storageInterface: "FileSystem"
    voicemailHome: "/storage/data"
    ...
```

Azure blob storage

For Azure Blob storage, use the following examples.

Azure Blob storage with the CSI driver:

1. Store the Azure Blob storage account access keys into Key Vault with a key-value pair:
   - key=
   - value=
2. Create a CSI driver from the Key Vault into the Kubernetes cluster with the name "keyvault-voice-voicemail-storage-csi-secrets".
3. Do the following changes in values.yaml:

```
#Blob Storage is enabled
   blobStorage:
     general:
       mode: 'k8s | csi' # Secrets needs to mounted via K8s or CSI driver
       enabled: true
     mounts:
       - name: voicemail-secrets
         readOnly: true
         mountPath: "/opt/genesys/katana/voicemail/secret"
     volumes:
       - name: voicemail-secrets
         csi:
           driver: secrets-store.csi.k8s.io
           readOnly: true
           volumeAttributes:
             secretProviderClass: keyvault-voice-voicemail-storage-csi-secrets


context:
...
  envs:
    ...
    storageInterface: "AzureBlob"
    voicemailHome: ""
    ...
```

Azure Blob storage with a Kubernetes secret:

1. Create a K8s secret cluster with the name "voicemail-storage-secrets" having a key-value pair:
   - key=
   - value=
2. Do the following changes in **values.yaml**

```
#Blob Storage is enabled
   blobStorage:
     general:
       mode: 'k8s | csi' # Secrets needs to mounted via K8s or CSI driver
       enabled: true
     mounts:
       - name: voicemail-secrets
         readOnly: true
         mountPath: "/opt/genesys/katana/voicemail/secret"
     volumes:
       - name: voicemail-secrets

secret:
          secretName: voicemail-storage-secrets


context:
...
  envs:
    ...
    storageInterface: "AzureBlob"
    voicemailHome: ""
    ...
```

AWS S3 bucket

For AWS S3 bucket with a Kubernetes secret, use the following examples.

1. Create a Kubernetes secret cluster with the name "voice voice-voicemail-s3-secrets" having a key-value pair:
   - accessKey=
   - secretkey=
   - bucketName=
   - bucketPath=

```
{
  "accessKey": "",
  "secretkey": "",
  "bucketName": "",
  "bucketPath": ""
}
```

2. Do the following changes in **values.yaml**:

```
#AWS S3 is enabled
   s3Storage:
     general:
       mode: 'k8s' # Secrets needs to mounted via K8s
       enabled: true
     mounts:
       - name: voicemail-s3-secrets
         readOnly: true
         mountPath: "/opt/genesys/katana/voicemail/secret"
     volumes:
       - name: voicemail-s3-secrets
         secret:
           secretName: voice-voicemail-s3-secrets
```

```
context:
...
  envs:
    ...
    storageInterface: "AWS_S3"
    voicemailHome: ""
    ...
```

You can now run the Helm install to deploy a Voicemail Service.

Create the IVR profile

The IVR profile is required for GVP to connect with the Voicemail Service.

To create the IVR profile for the Voicemail service:

1. Configure the following parameters:
   - Name = "voicemailservice"
   - Display name = "voicemailservice"
   - Annex options:
     - [gvp.general]\service-type=voicexml
     - [gvp.policy] section with default values
     - [gvp.service-parameters]\voicexml.gvp.appmodule="fixed,VXML-NG
     - [gvp.service-prerequisite]\initial-page-url and alternate-voice-xml = "http://voice-voicemail-service.voice.svc.cluster.local:8081/fs"
     - [gvp.service-prerequisite]\REQUESTURI_TIMEOUT = 5
2. In Tenant > Annex options:
   - [gvp.dn-groups]\voicemailservice = "55551111"
   - [gvp.dn-group-assignments]\voicemailservice = DB id of the IVR profile ("voicemailservice")
3. In the GVP namespace, add port 8081 to allow outbound communication. The Voicemail service runs in the Voice namespace with port 8081.

Provision the Voicemail Service

Contents

Learn how to provision the Voicemail Service.

Related documentation:

-
-
-

RSS:

- For private edition

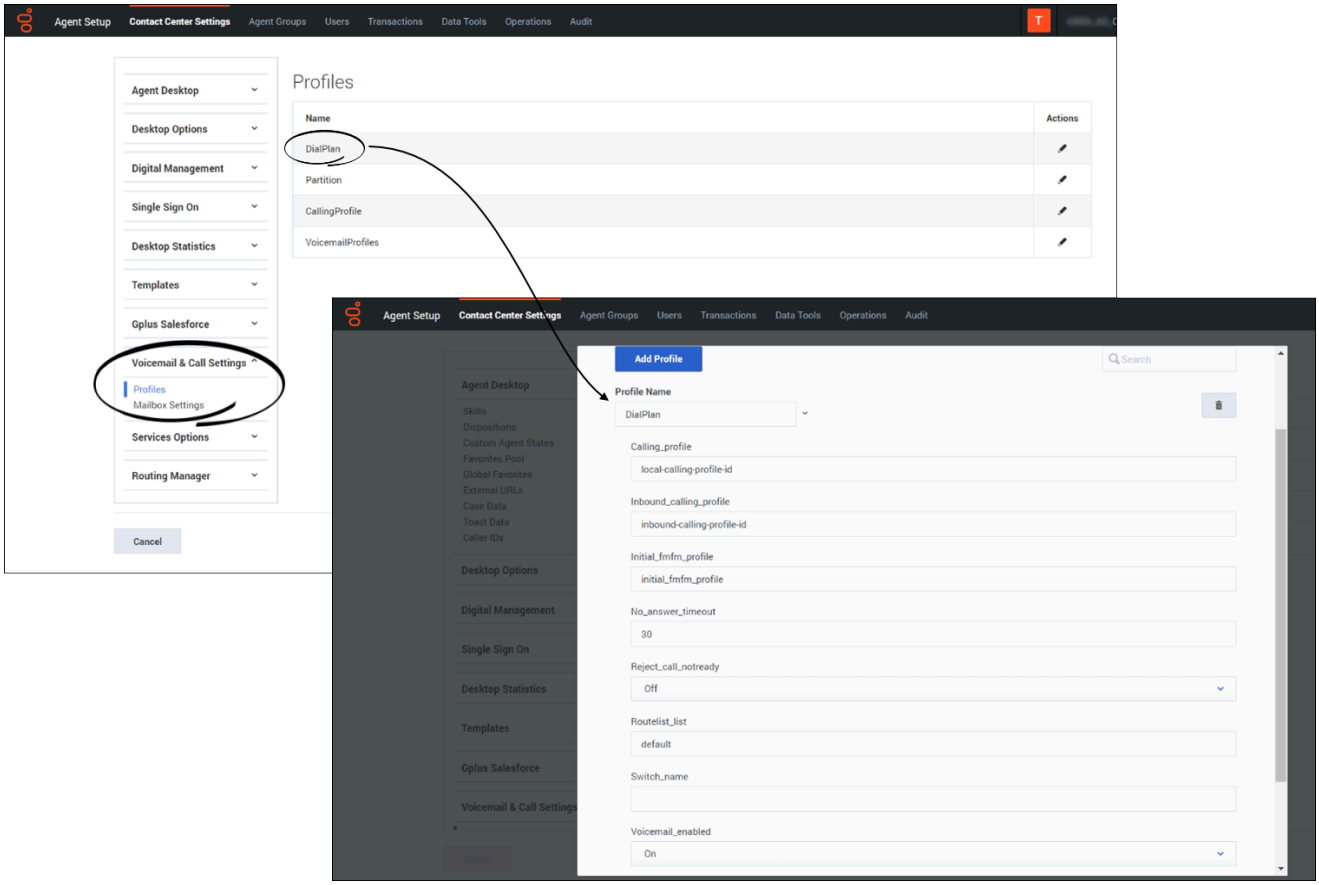
You provision the voicemail service using Agent Setup. Provisioning consists of the following tasks:

1. Enable voicemail.
2. Create voicemail profiles.
3. Configure mailbox settings, as required.
4. Configure greetings.
5. Bulk-provision mailboxes.

Enabling Voicemail

To provision the voicemail service in Agent Setup, you must first enable voicemail:

1. Log in to Agent Setup.
2. Navigate to **Contact Center Settings** > **Voicemail & Call Settings** > **Profiles**.
3. Select the **DialPlan** profile.
4. Select 0n in the **Voicemail_enabled** field.



Managing voicemail profiles

You use voicemail profiles to assign voicemail settings. To create and manage voicemail profiles:

1. Log in to Agent Setup.
2. Navigate to **Contact Center Settings** > **Voicemail & Call Settings** > **Profiles** > **VoicemailProfiles**.
3. To create a new profile, click **Add Profile**. To edit an existing profile, select it in the list.
4. The following table describes the options you can configure for a voicemail profile.

Option	Values (bold indicates the default value)	Description
Email Body	"Mailbox has a new message from ", or any text	The body of the notification email. It can contain any of the following parameter tokens: • `CallerID` is the phone number of the caller.

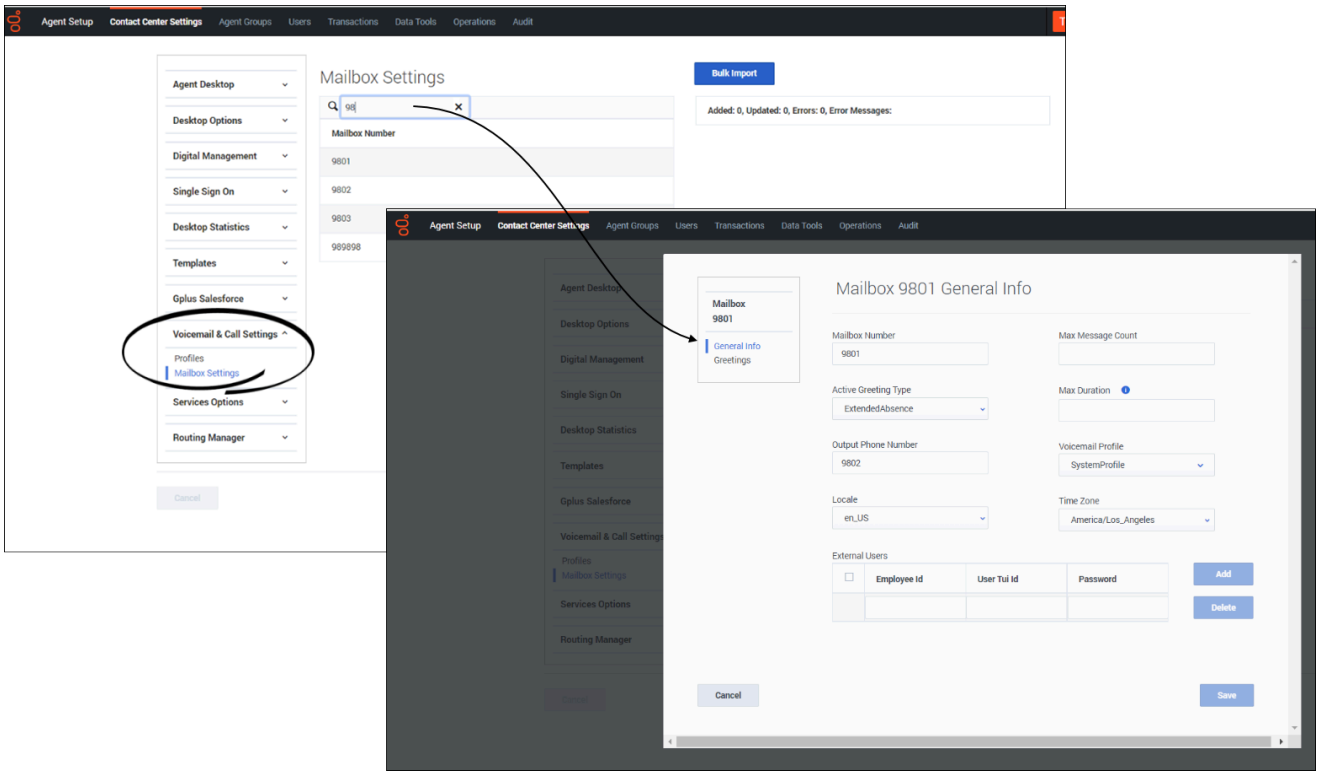
Option	Values (bold indicates the default value)	Description
		• `MailboxID` is the mailbox that contains the message. • `MsgPriority` is the message priority set by the caller, if enabled. • `MsgReceivedDate` is the date on which the caller left the message. • `UserEmail` is the email address of the recipient. • `UserPhone` is the phone number of the recipient. • `VoicemailAccessURL` is the URL that the recipient can click to retrieve their message online. • `VoicemailAccessNumber` is the phone number that the user can dial to listen to their message. To insert a parameter, type . The message also includes any static text you type.
Email From Address	user@domain	The email address from which you want to send notifications.
Email Notification	true, **false**	Enable or disable email notifications.
Email Subject	"Genesys Voicemail Notification: New Message from ", or any subject line	The subject line of the notification email. It can contain any of the parameter tokens available in the Email Body.
Max Duration	**30**, or any positive integer	Specifies, in seconds, the maximum message length.
Max Message Count	**100**, or any positive integer	Type a value to set a new maximum number of messages.
Retention Limit	10, 20, 30, 40, 50, 60	Deletes the voicemail from storage after the configured number of days.
Voicemail Forwarding	true, **false**	When the **Voicemail Forwarding** option is set to `true`, you can use your telephone to forward voicemail messages left in your mailbox to any mailbox.

Configuring mailbox settings

To configure and manage mailbox settings:

1. Log into Agent Setup.
2. Navigate to **Contact Center Settings** > **Voicemail & Call Settings** > **Mailbox Settings**.
3. Search and select the mailbox for which the settings need to be modified. The selected mailbox opens in a new window.
4. On the **General Info** tab, use the following options to provision the selected mailbox.

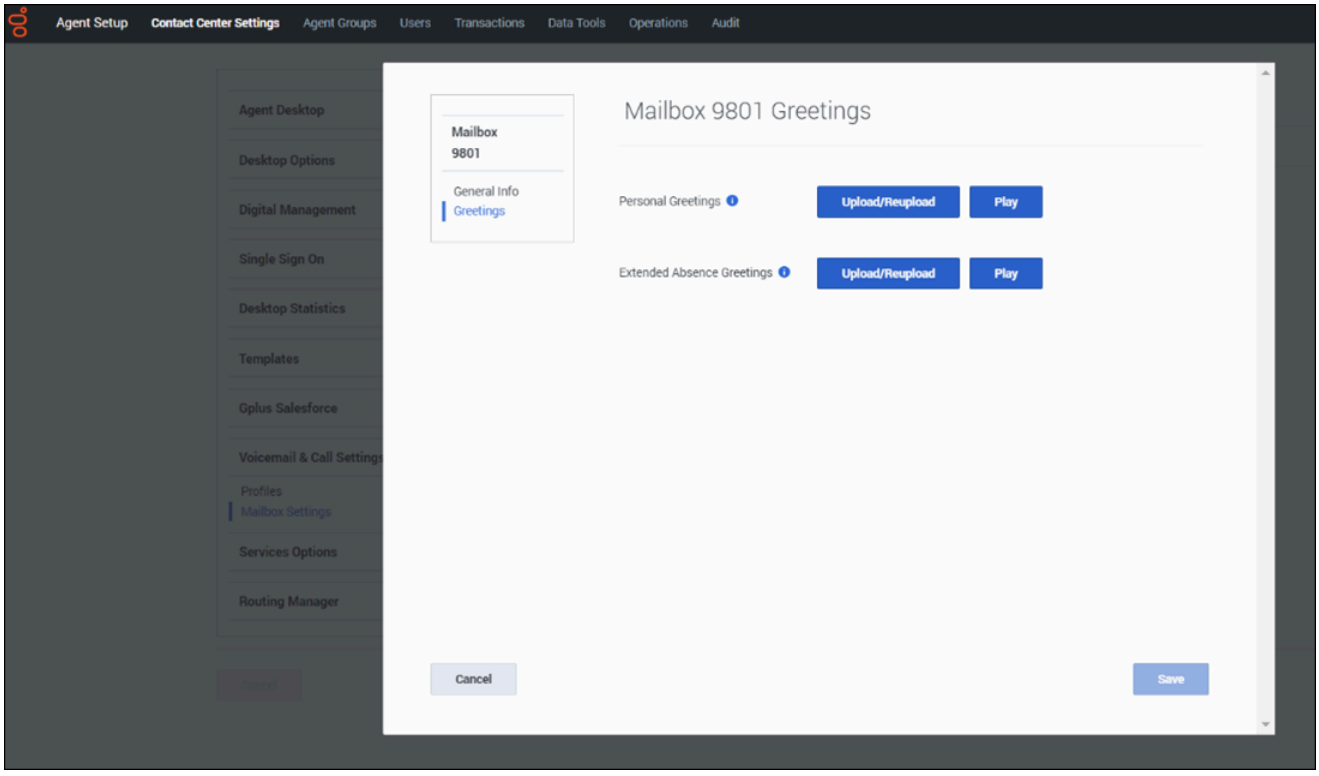
Setting	Values (bold indicates the default value)	Description
Max Message Count	**100**, or any positive integer	Type a value to set a new maximum number of messages.
Active Greeting Type	**Standard**, or a greeting type from the menu	Specifies the type of mailbox greeting.
Max Duration	**30**, or any positive integer	Specifies, in seconds, the maximum message length.
Output Phone Number	**System (Not Set)**, or any phone number or routing point	When set, enables a caller to transfer out of voicemail to the specified destination at any time during a call. Select the radio button and type a value to set a new optout phone number. Select **System** to restore the value to the number in parentheses, which is the value set at the application or switch level for the configuration option `voicemail-optout-destination`.
Voicemail Profile	**SystemProfile**, or a profile from the menu	Specifies the profile that the mailbox uses.
Locale	**en-US**, or other locale strings	Specifies the default locale for the Telephone User Interface (TUI).
Time Zone	**America/Los_Angeles**, or a time zone from the menu	Select a time zone from the menu to set a new time zone for all mailboxes that use the system (default) time zone. Select **System** to restore the system value.



Managing your greetings

To manage your voicemail greetings:

1. Log into Agent Setup.
2. Navigate to **Contact Center Settings** > **Voicemail & Call Settings** > **Mailbox Settings**.
3. Search and select the mailbox for which the settings need to be modified. The selected mailbox opens in a new window.
4. On the **Greetings** tab:
   - Click **Upload/Reupload** to upload a Personal or Extended Absence greeting.
   - Click **Play** to listen to the existing Personal or Extended Absence greeting.



Bulk-provisioning mailboxes

To provision many mailboxes simultaneously:

1. Log in to Agent Setup.
2. Navigate to **Contact Center Settings** > **Voicemail & Call Settings** > **Mailbox Settings**.
3. To add or modify mailbox settings simultaneously, click **Bulk Import**.
4. Create a CSV file with the following fields:
    - Mailbox Number
    - Active Greeting Type
    - Output Phone Number
    - Locale
    - Time Zone
    - Max Message Count
    - Max Duration
    - Voicemail Profile
5. Select the created CSV file. The mailbox settings will be updated for the mailboxes mentioned in CSV

file after successful Bulk Import.

Deploy Voicemail

Contents

Learn how to deploy Voicemail 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.

In addition to the general deployment prerequisites, the Voicemail Service has the following prerequisites; you must complete the following before deploying the Voicemail Service:

- Deploy Voice Services.
- Choose a storage option for Voicemail. For more information, see Storage requirements.

The following services are required before proceeding with Voicemail deployment testing:

- Tenant Service
- Genesys Voice Platform (GVP) deployment
- Web Services and Applications (GWS), including Agent Setup (for accessing the UI and configuring the Voicemail solution)

Voicemail deployment notes

1. Similar to Voice Services, a Voicemail Service also requires security context constraints for Genesys users. Since the security context already created for Voice Services, we reuse the same for the Voicemail Service. Use the following command to add the security context for a Voicemail Service:

```
oc adm policy add-scc-to-user genesys-restricted -z voice-voicemail -n voice
```

2. Edit the **voicemail_override_values.yaml** file in the **voice_helm_values** directory as per required storage. See Configure the Voicemail Service.
3. Install the Voicemail Helm chart with **overridden_values.yaml** (with the **-f** flag), below is the sample command.
   - The pass "--set version="

```
helm upgrade --install --force --wait --timeout 300s -n voice -f
 ./overridden_values.yaml voice-voicemail  https:///voice-voicemail/voice-
voicemail-9.0.07.tgz --set version=9.0.10 --username "$JFROG_USER" --password
"$JFROG_PASSWORD"
```

Validate the deployment

The procedure to validate the voicemail deployment is similar to voice microservices. For more information, see Validate the deployment in *Configure and deploy* section of this guide.

Upgrade, rollback, or uninstall the Voicemail Service

Contents

Learn how to upgrade, rollback or uninstall the Voicemail Service.

Related documentation:

-
-
-

RSS:

- For private edition

Upgrade Voicemail

The upgrade procedure consists of these major steps:

1. Canary deployment
2. Upgrade
3. Delete canary

Canary deployment

For the canary deployment, some parameters in the **canary_override_values.yaml** file must be overridden. This file is passed to the Helm chart during the deployment of the canary instance.

```
### Canary Override Values
serviceAccount:
  create: false   # Service account will be already created while initial deployment

service:
  canaryName: canary # Postfix that will be added for canary deployment

prometheus:
  podMonitor:
    enabled: false  # Podmonitor deployed during initial deployment will get metrics from all
pod instance including canary.

hpa:
  enabled: false  # HPA is not needed for canary
```

Deploy a canary instance:

```
helm upgrade --install --force --wait --timeout 500s -n voice -f
./voicemail_override_values.yaml -f ./voice_helm_values/canary_override_values.yaml voice-
voicemail-canary  https:///voice-agent/voice-voicemail-9.0.07.tgz --set version=9.0.10 --
username "$JFROG_USER" --password "$JFROG_PASSWORD"
```

Service upgrade

When the canary deployment of a Voicemail Service is ready for an upgrade, use the following command to upgrade the current version of a Voicemail Service to the desired version:

```
helm upgrade --install --force --wait --timeout 500s -n voice -f
./voicemail_override_values.yaml -f voice-voicemail-canary  https:///voice-agent/voice-
voicemail-9.0.07.tgz --set version=9.0.10 --username "$JFROG_USER" --password
"$JFROG_PASSWORD"
```

Delete the canary instance

If the upgrade of a Voicemail Service is successful, delete the canary instance of the service by using the following command:

```
helm uninstall voice-voicemailcanary -n voice
```

Rollback Voicemail

The procedure to rollback voicemail is similar to voice microservices rollback. For more information, see Rollback.

Uninstall Voicemail

The procedure to uninstall voicemail is similar to voice microservices uninstallation. For more information, see Uninstall.

Observability in Voice Microservices

Contents

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

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 Voice Microservices metrics, see:

- Agent State Service metrics
- Call State Service metrics
- Config Service metrics
- Dial Plan Service metrics
- FrontEnd Service metrics
- ORS metrics
- Voice Registrar Service metrics
- Voice RQ Service metrics
- Voice SIP Cluster Service metrics
- Voice SIP Proxy Service metrics
- Voicemail metrics

See also System metrics.

Deploy dashboard and alert dashboards

Deploy dashboards and alert rules using these Helm charts:

- **voice-dashboards** – This installs the dashboards that are created to monitor various Voice Services.
- **voice-alertrules** – This installs the alert rules that specify what type of alarm must be triggered based on the metrics.

```
helm repo add helm-staging https:// --username "$JFROG_USER" --password "$JFROG_PASSWORD"
helm repo update

helm install voice-alertrules -n voice https:///voice-monitoring/voice-alertrules-1.0.5.tgz --
username "$JFROG_USER" --password "$JFROG_PASSWORD"

helm install voice-dashboards -n voice https:///voice-monitoring/voice-dashboards-1.0.8.tgz --
username "$JFROG_USER" --password "$JFROG_PASSWORD"
```

Enable monitoring

You can expose metrics on a service-by-service basis. To do so, edit the **Values.yaml** file associated with each service, and enable metrics using either the **Prometheus** operator, or **Prometheus Annotation**.

```
prometheus:
  # Enable for Prometheus Annotation
  metricServer:
    enabled: false
    path: /metrics
```

OR

```
  # Enable for Prometheus operator
  podMonitor:
    enabled: false
    path: /metrics
    interval: 30s
```

Service	CRD or annotations?	Port	Endpoint/ Selector	Metrics update interval
Agent State Service	PodMonitor	11000	http://:11000/ metrics	30 seconds
Call State Service	Supports both CRD and annotations	11900	http://:11900/ metrics	30 seconds
Config Service	Supports both CRD and annotations	9100	http://:9100/ metrics	30 seconds
Dial Plan Service	Supports both CRD and annotations	8800	http://:8800/ metrics	30 seconds
FrontEnd Service	Supports both CRD and annotations	9101	http://:9101/ metrics	30 seconds
ORS	Supports both CRD	11200	http://:11200/	30 seconds

Service	CRD or annotations?	Port	Endpoint/ Selector	Metrics update interval
	and annotations		metrics	
Voice Registrar Service	Supports both CRD and annotations	11500	http://:11500/ metrics	30 seconds
Voice RQ Service	Supports both CRD and annotations	12000	http://:12000/ metrics	30 seconds
Voice SIP Cluster Service	Supports both CRD and annotations	11300	http://:11300/ metrics	30 seconds
Voice SIP Proxy Service	Supports both CRD and annotations	11400	http://:11400/ metrics	30 seconds
Voicemail	Supports both CRD and annotations	8081	http://:8081/ metrics	30 seconds

Configure metrics

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

The Metrics pages linked to above show some of the metrics the Voice Microservices expose. You can also query Prometheus directly or via a dashboard to see all the metrics available from the Voice Microservices.

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 Voice Microservices alerts, see:

- Agent State Service alerts
- Call State Service alerts
- Config Service alerts
- Dial Plan Service alerts
- FrontEnd Service alerts
- ORS alerts

- Voice Registrar Service alerts
- Voice RQ Service alerts
- Voice SIP Cluster Service alerts
- Voice SIP Proxy Service alerts
- Voicemail alerts

Configure alerts

Private edition services define a number of alerts by default (for Voice Microservices, 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, Voice Microservices does not support customizing the alerts or defining additional **PrometheusRule** objects to create alerts based on the service-provided metrics.

Logging

Voice Microservices can write logs generated by internal components to the following locations:

- Persistent Volume/Persistent Volume Claim with RWX storage. For more information, see Log volume, Deploy the Voice Services, and Persistent volumes.
- Ephemeral volume (emptyDir) with a Fluent Bit logging sidecar that tails log files and sends them to standard output (stdout). For more information, see Forwarding logs to stdout.

Forwarding logs to stdout

You can optionally forward logs from internal components to stdout using a logging sidecar (Genesys currently supports Fluent Bit) and an ephemeral volume (emptyDir). The Fluent Bit sidecar tails the logs and sends them to stdout. For more information, see Sidecar processed logging in the *Genesys Multicloud CX Private Edition Operations guide*.

The SIPNode logs within the SIP Cluster service can be forwarded to stdout. By default, forwarding logs to stdout is disabled. To enable the log forwarding option, set the following parameters in the voice-sip Helm Chart **values.yaml** file:

```
volumes:
   # Mount an Ephemeral Volume for storing the legacy SIPS logs.
  sipsLog:
   mountPath: "/opt/genesys/logs/sip_node/SIPS"

sipsLoggingSidecar:
  enabled: true                           # sips-logging-sidecar container will be created
  image:
    registry: genesysengagedev.azurecr.io   # Registry from where the images will be fetched
    repository: sre/fluent-bit              # repository and folder where the particular
```

```
service image is located
    pullPolicy: Always                      # Policy to pull always or only when the image is
not there
    tag: 1.8.x                              # fluent-bit version that will be used by the
logging sidecar
  context:
    create: true                            #Create configmap for sips-logging-sidecar.Set to
true in Values.yaml and set to false in canary_values.yaml
```

Agent State Service metrics and alerts

Contents

Find the metrics Agent State Service exposes and the alerts defined for Agent State Service.

Service	CRD or annotations?	Port	Endpoint/Selector	Metrics update interval
Agent State Service	PodMonitor	11000	http://:11000/metrics	30 seconds

See details about:

- Agent State Service metrics
- Agent State Service alerts

Metrics

Voice Agent State Service exposes Genesys-defined, Agent State Service–specific metrics as well as some standard Kafka metrics. You can query Prometheus directly to see all the metrics that the Agent State Service exposes. The following metrics are likely to be particularly useful. Genesys does not commit to maintain other currently available Agent State Service metrics not documented on this page.

Metric and description	Metric details	Indicator of
agent_redis_state Current Redis connection state: -1 - error 0 - disconnected 1 - connected 2 - ready	**Unit:** N/A **Type:** gauge **Label:** location, redis_cluster_name **Sample value:** 2	
agent_stream_redis_state Current Tenant Redis connection state: 0 - disconnected 1 - connected	**Unit:** N/A **Type:** gauge **Label:** location, redis_cluster_name **Sample value:** 1	
agent_total_sessions Total number of agent sessions.	**Unit:** N/A **Type:** gauge **Label:** tenant **Sample value:**	Saturation
agent_callevents Total number of received call events.	**Unit:** N/A **Type:** counter **Label:** tenant	Traffic

Metric and description	Metric details	Indicator of
	Sample value:	
agent_logged_in_agents Number of logged-in agents.	**Unit:** N/A **Type:** gauge **Label:** tenant **Sample value:**	Saturation
agent_health_level Health level of the agent node: -1 - error 0 - fail 1 - degraded 2 - pass	**Unit:** N/A **Type:** gauge **Label:** tenant **Sample value:** 2	Traffic
agent_envoy_proxy_status Status of the Envoy proxy: -1 - error 0 - disconnected 1 - connected	**Unit:** N/A **Type:** gauge **Label:** **Sample value:** 1	
agent_config_node_status Status of the config node connection: 0 - disconnected 1 - connected	**Unit:** N/A **Type:** gauge **Label:** **Sample value:** 1	
http_client_request_duration_seconds HTTP client time from request to response, in seconds.	**Unit:** seconds **Type:** histogram **Label:** target_service_name **Sample value:**	
http_client_response_count HTTP client responses received.	**Unit:** N/A **Type:** counter **Label:** target_service_name, tenant, status **Sample value:**	Traffic
kafka_consumer_recv_messages_total Number of messages received from Kafka.	**Unit:** N/A **Type:** counter **Label:** topic, tenant, kafka_location **Sample value:**	Traffic
kafka_consumer_error_total Number of Kafka consumer errors.	**Unit:** N/A **Type:** counter **Label:** topic, kafka_location **Sample value:**	Errors
kafka_consumer_latency Consumer latency is the time difference between when the message is produced	**Unit:** **Type:** histogram **Label:** topic, tenant, kafka_location	Latency

Metric and description	Metric details	Indicator of
and when the message is consumed. That is, the time when the consumer received the message minus the time when the producer produced the message.	**Sample value:**	
kafka_consumer_rebalance_total Number of Kafka consumer re-balance events.	**Unit:** N/A **Type:** counter **Label:** topic, kafka_location **Sample value:**	
kafka_consumer_state Current state of the Kafka consumer.	**Unit:** N/A **Type:** gauge **Label:** topic, kafka_location **Sample value:**	
kafka_producer__messages_total Number of messages received from Kafka.	**Unit:** N/A **Type:** counter **Label:** topic, tenant, kafka_location **Sample value:**	
kafka_producer_queue_depth Number of Kafka producer pending events.	**Unit:** N/A **Type:** gauge **Label:** kafka_location **Sample value:**	Saturation
kafka_producer_queue_age_seconds Age of the oldest producer pending event in seconds.	**Unit:** seconds **Type:** gauge **Label:** kafka_location **Sample value:**	
kafka_producer_error_total Number of Kafka producer errors.	**Unit:** N/A **Type:** counter **Label:** kafka_location **Sample value:**	
kafka_producer_state Current state of the Kafka producer.	**Unit:** N/A **Type:** gauge **Label:** kafka_location **Sample value:**	
log_output_bytes_total Total amount of log output, in bytes.	**Unit:** bytes **Type:** counter **Label:** level, format, module **Sample value:**	

Alerts

The following alerts are defined for Agent State Service.

Alert	Severity	Description	Based on	Threshold
Kafka events latency is too high	Warning	Actions: • If the alarm is triggered for multiple topics, ensure there are no issues with Kafka (CPU, memory, or network overload). • If the alarm is triggered only for topic {{ $labels.topic }}, check if there is an issue with the service related to the topic (CPU, memory, or network overload).	kafka_consumer_latency_bucket	Latency for more than 5% of messages is more than 0.5 seconds for topic {{ $labels.topic }}.
Possible messages lost	Critical	Actions: • Check Kafka and {{ $labels.job }} service overload, network degradation.	kafka_consumer_recv_messages_total, kafka_producer_sent_messages_total	Number of sent requests is two times higher than received for topic {{ $labels.topic }}.
Too many Kafka consumer failed health checks	Warning	Actions: • If the alarm is triggered for multiple services, ensure there are no issues with Kafka. Restart Kafka. • If the alarm is triggered only for container {{ $labels.container }}, check if there is an issue with the service.	kafka_consumer_error_total	Health check failed more than 10 times in 5 minutes for Kafka consumer for topic {{ $labels.topic}}.

Alert	Severity	Description	Based on	Threshold
Too many Kafka consumer request timeouts	Warning	Actions: • If the alarm is triggered for multiple services, ensure there are no issues with Kafka. Restart Kafka. • If the alarm is triggered only for container {{ $labels.container }}, check if there is an issue with the service.	kafka_consumer_error_total	More than 10 request timeouts appeared in 5 minutes for Kafka consumer for topic {{ $labels.topic}}.
Too many Kafka consumer crashes	Critical	Actions: • If the alarm is triggered for multiple services, ensure there are no issues with Kafka. Restart Kafka. • If the alarm is triggered only for container {{ $labels.container }}, check if there is an issue with the service.	kafka_consumer_error_total	More than 3 Kafka consumer crashes in 5 minutes for service {{ $labels.container }}.
Pod status Failed	Warning	Actions: • Restart the pod. Check if there are any issues with the pod after restart.	kube_pod_status_phase	Pod {{ $labels.pod }} is in Failed state.
Pod status Unknown	Warning	Actions: • Restart the pod. Check if	kube_pod_status_phase	Pod {{ $labels.pod }} is in Unknown state for 5 minutes.

Alert	Severity	Description	Based on	Threshold
		there are any issues with pod after restart.		
Pod status Pending	Warning	Actions: • Restart the pod. Check if there are any issues with the pod after restart.	kube_pod_status_phase	Pod {{ $labels.pod }} is in Pending state for 5 minutes.
Pod status NotReady	Critical	Actions: • Restart the pod. Check if there are any issues with the pod after restart.	kube_pod_status_ready	Pod {{ $labels.pod }} is in NotReady status for 5 minutes.
Container restarted repeatedly	Critical	Actions: • Check if the new version of the image was deployed. • Check for issues with the Kubernetes cluster.	kube_pod_container_status_restarts_total	Container {{ $labels.container }} was restarted 5 or more times within 15 minutes.
Max replicas is not sufficient for 5 mins	Critical	The desired number of replicas is higher than the current available replicas for the past 5 minutes.	kube_statefulset_replicas, kube_statefulset_status_replicas	The desired number of replicas is higher than the current available replicas for the past 5 minutes.
Kafka not available	Critical	Actions: • If the alarm is triggered for multiple services, ensure there are no issues with Kafka. Restart Kafka.	kafka_producer_state, kafka_consumer_state	Kafka is not available for pod {{ $labels.pod }} for 5 consecutive minutes.

Alert	Severity	Description	Based on	Threshold
		• If the alarm is triggered only for pod {{ $labels.pod }}, check if there is an issue with the pod.		
Redis not available	Critical	Actions: • If the alarm is triggered for multiple services, ensure there are no issues with Redis. Restart Redis. • If the alarm is triggered only for pod {{ $labels.pod }}, check if there is an issue with the pod.	agent_redis_state, agent_stream_redis_state	Redis is not available for pod {{ $labels.pod }} for 5 consecutive minutes.
Agent service fail	Critical	Actions: • Check if there is any problem with pod {{ $labels.pod }}, then restart the pod.	agent_health_level	Agent health level is Fail for pod {{ $labels.pod }} for 5 consecutive minutes.
Config node fail	Warning	Actions: • Check if there is any problem with pod {{ $labels.pod }} and the config node.	http_client_response_count	Requests to the config node fail for 5 consecutive minutes.
Pod CPU greater than 65%	Warning	High CPU load for pod {{ $labels.pod }}.	container_cpu_usage_seconds_total, container_spec_cpu_period	Container {{ $labels.container }} CPU usage exceeded 65% for 5 minutes.
Pod CPU greater	Critical	Critical CPU load	container_cpu_usage_seconds_total,	Container {{

Alert	Severity	Description	Based on	Threshold
than 80%		for pod {{ $labels.pod }}.	container_spec_cpu_period	$labels.container }} CPU usage exceeded 80% for 5 minutes.
Pod memory greater than 65%	Warning	High memory usage for pod {{ $labels.pod }}.	container_memory_working_set_bytes, kube_pod_container_resource_requests_memory_bytes	Container {{ $labels.container }} memory usage exceeded 65% for 5 minutes.
Pod memory greater than 80%	Critical	Critical memory usage for pod {{ $labels.pod }}.	container_memory_working_set_bytes, kube_pod_container_resource_requests_memory_bytes	Container {{ $labels.container }} memory usage exceeded 80% for 5 minutes.
Too many Kafka pending events	Critical	Actions: • Ensure there are no issues with Kafka or {{ $labels.pod }} pod's CPU and network.	kafka_producer_queue_depth	Too many Kafka producer pending events for pod {{ $labels.pod }} (more than 100 in 5 minutes).

Call State Service metrics and alerts

Contents

Find the metrics Call State Service exposes and the alerts defined for Call State Service.

Service	CRD or annotations?	Port	Endpoint/Selector	Metrics update interval
Call State Service	Supports both CRD and annotations	11900	http://:11900/metrics	30 seconds

See details about:

- Call State Service metrics
- Call State Service alerts

Metrics

Voice Call State Service exposes Genesys-defined, Call State Service–specific metrics as well as some standard Kafka metrics. You can query Prometheus directly to see all the metrics that the Call State Service exposes. The following metrics are likely to be particularly useful. Genesys does not commit to maintain other currently available Call State Service metrics not documented on this page.

Metric and description	Metric details	Indicator of
callthread_call_threads Number of monitored call threads.	**Unit:** N/A **Type:** counter **Label:** **Sample value:**	Saturation
callthread_envoy_proxy_status Status of the envoy proxy: -1 - error 0 - disconnected 1 – connected	**Unit:** N/A **Type:** gauge **Label:** **Sample value:**	
callthread_health_level Health level of the agent node: -1 - error 0 - fail 1 - degraded 2 - pass	**Unit:** N/A **Type:** gauge **Label:** **Sample value:**	
callthread_healthcheck_generic_exception Generic error during health check.	**Unit:** N/A **Type:** gauge	

Metric and description	Metric details	Indicator of
	Label: **Sample value:**	
callthread_redis_state Current Redis connection state: -1 – error 0 – disconnected 1 – connected 2 – ready	**Unit:** N/A **Type:** gauge **Label:** **Sample value:**	Errors
http_client_request_duration_seconds HTTP client time from request to response, in seconds.	**Unit:** seconds **Type:** histogram **Label:** target_service_name **Sample value:**	
http_client_response_count The number of HTTP client responses received.	**Unit:** N/A **Type:** counter **Label:** target_service_name, tenant, status **Sample value:**	
kafka_consumer_recv_messages_total Number of messages received from Kafka.	**Unit:** N/A **Type:** counter **Label:** topic, tenant, kafka_location **Sample value:**	Traffic
kafka_consumer_error_total Number of Kafka consumer errors.	**Unit:** N/A **Type:** counter **Label:** topic, kafka_location **Sample value:**	Errors
kafka_consumer_latency Consumer latency is the time difference between when the message is produced and when the message is consumed. That is, the time when the consumer received the message minus the time when the producer produced the message.	**Unit:** **Type:** histogram **Label:** topic, tenant, kafka_location **Sample value:**	Latency
kafka_consumer_rebalance_total Number of Kafka consumer re-balance events.	**Unit:** N/A **Type:** counter **Label:** topic, kafka_location **Sample value:**	
kafka_consumer_state Current state of Kafka consumer.	**Unit:** N/A **Type:** gauge **Label:** topic, kafka_location **Sample value:**	
kafka_producer__messages_total Number of messages received from	**Unit:** N/A **Type:** counter	Traffic

Metric and description	Metric details	Indicator of
Kafka.	**Label:** topic, tenant, kafka_location **Sample value:**	
kafka_producer_queue_depth Number of Kafka producer pending events.	**Unit:** N/A **Type:** gauge **Label:** kafka_location **Sample value:**	Saturation
kafka_producer_queue_age_seconds Age of the oldest producer pending event, in seconds.	**Unit:** seconds **Type:** gauge **Label:** kafka_location **Sample value:**	
kafka_producer_error_total Number of Kafka producer errors.	**Unit:** N/A **Type:** counter **Label:** kafka_location **Sample value:**	Errors
kafka_producer_state Current state of the Kafka producer.	**Unit:** N/A **Type:** gauge **Label:** kafka_location **Sample value:**	
log_output_bytes_total Total amount of log output, in bytes.	**Unit:** bytes **Type:** counter **Label:** level, format, module **Sample value:**	

Alerts

The following alerts are defined for Call State Service.

Alert	Severity	Description	Based on	Threshold
Kafka events latency is too high	Critical	Actions: • If the alarm is triggered for multiple topics, ensure there are no issues with Kafka (CPU, memory, or network overload). • If the alarm is triggered only for topic {{	kafka_consumer_latency_bucket	Latency for more than 5% of messages is more than 0.5 seconds for topic {{ $labels.topic }}.

Alert	Severity	Description	Based on	Threshold
		$labels.topic }}, check if there is an issue with the service related to the topic (CPU, memory, or network overload).		
Too many Kafka consumer failed health checks	Warning	Actions: • If the alarm is triggered for multiple services, make sure there are no issues with Kafka, and then restart Kafka. • If the alarm is triggered only for {{ $labels.container }}, check if there is an issue with the service.	kafka_consumer_error_total	Health check failed more than 10 times in 5 minutes for Kafka consumer for topic {{ $labels.topic }}.
Too many Kafka consumer request timeouts	Warning	Actions: • If the alarm is triggered for multiple services, make sure there are no issues with Kafka, and then restart Kafka. • If the alarm is triggered only for {{ $labels.container }}, check if there is an issue with the service.	kafka_consumer_error_total	More than 10 request timeouts appeared in 5 minutes for Kafka consumer for topic {{ $labels.topic }}.
Too many Kafka consumer crashes	Critical	Actions:	kafka_consumer_error_total	More than 3 Kafka consumer crashes in 5 minutes for

Alert	Severity	Description	Based on	Threshold
		• If the alarm is triggered for multiple services, make sure there are no issues with Kafka, and then restart Kafka. • If the alarm is triggered only for {{ $labels.container }}, check if there is an issue with the service.		topic {{ $labels.topic }}.
Pod status Failed	Warning	Actions: • Restart the pod. Check if there are any issues with the pod after restart.	kube_pod_status_phase	Pod {{ $labels.pod }} is in Failed state.
Pod status Unknown	Warning	Actions: • Restart the pod. Check if there are any issues with pod after restart.	kube_pod_status_phase	Pod {{ $labels.pod }} is in Unknown state for 5 minutes.
Pod status Pending	Warning	Actions: • Restart the pod. Check if there are any issues with the pod after restart.	kube_pod_status_phase	Pod {{ $labels.pod }} is in Pending state for 5 minutes.
Pod status NotReady	Critical	Actions: • Restart the pod. Check if there are any issues with the pod after restart.	kube_pod_status_ready	Pod {{ $labels.pod }} is in NotReady status for 5 minutes.

Alert	Severity	Description	Based on	Threshold
Container restarted repeatedly	Critical	Actions: • Check if the new version of the image was deployed. • Check for issues with the Kubernetes cluster.	kube_pod_container_status_restarts_total	Container {{ $labels.container }} was restarted 5 or more times within 15 minutes.
Max replicas is not sufficient for 5 mins	Critical	The desired number of replicas is higher than the current available replicas for the past 5 minutes.	kube_statefulset_replicas, kube_statefulset_status_replicas	The desired number of replicas is higher than the current available replicas for the past 5 minutes.
Kafka not available	Critical	Actions: • If the alarm is triggered for multiple services, make sure there are no issues with Kafka, and then restart Kafka. • If the alarm is triggered only for pod {{ $labels.pod }}, check if there is an issue with the pod.	kafka_producer_state, kafka_consumer_state	Kafka is not available for pod {{ $labels.pod }} for 5 consecutive minutes.
Redis not available	Critical	Actions: • If the alarm is triggered for multiple services, make sure there are no issues with Redis, and then restart Redis. • If the alarm is triggered only for pod {{	callthread_redis_state	Redis is not available for pod {{ $labels.pod }} for 5 consecutive minutes.

Alert	Severity	Description	Based on	Threshold
		$labels.pod }}, check if there is an issue with the pod.		
Pod CPU greater than 65%	Warning	High CPU load for pod {{ $labels.pod }}.	container_cpu_usage_seconds_total, container_spec_cpu_period	Container {{ $labels.container }} CPU usage exceeded 65% for 5 minutes.
Pod CPU greater than 80%	Critical	Critical CPU load for pod {{ $labels.pod }}.	container_cpu_usage_seconds_total, container_spec_cpu_period	Container {{ $labels.container }} CPU usage exceeded 80% for 5 minutes.
Pod memory greater than 65%	Warning	High memory usage for pod {{ $labels.pod }}.	container_memory_working_set_bytes, kube_pod_container_resource_requests_memory_bytes	Container {{ $labels.container }} memory usage exceeded 65% for 5 minutes.
Pod memory greater than 80%	Critical	Critical memory usage for pod {{ $labels.pod }}.	container_memory_working_set_bytes, kube_pod_container_resource_requests_memory_bytes	Container {{ $labels.container }} memory usage exceeded 80% for 5 minutes.
Too many Kafka pending events	Critical	Actions: • Ensure there are no issues with Kafka or {{ $labels.container }} service's CPU and network.	kafka_producer_queue_depth	Too many Kafka producer pending events for service {{ $labels.container }} (more than 100 in 5 minutes).

Config Service metrics and alerts

Contents

Find the metrics Config Service exposes and the alerts defined for Config Service.

Service	CRD or annotations?	Port	Endpoint/Selector	Metrics update interval
Config Service	Supports both CRD and annotations	9100	http://:9100/metrics	30 seconds

See details about:

- Config Service metrics
- Config Service alerts

Metrics

You can query Prometheus directly to see all the metrics that the Voice Config Service exposes. The following metrics are likely to be particularly useful. Genesys does not commit to maintain other currently available Config Service metrics not documented on this page.

Metric and description	Metric details	Indicator of
config_device_response Number of device responses for each request.	**Unit:** N/A **Type:** counter **Label:** location, tenant, request_type, status **Sample value:** 2	Traffic
config_tenant_response Number of Tenant responses for each request.	**Unit:** N/A **Type:** counter **Label:** location, request_type, status **Sample value:** 2	Traffic
config_node_get_response Number of Get responses for each request.	**Unit:** N/A **Type:** counter **Label:** **Sample value:**	Traffic
config_node_agent_response Number of agent responses for each request.	**Unit:** N/A **Type:** counter **Label:** **Sample value:**	Traffic
config_redis_state	**Unit:** N/A	Errors

Metric and description	Metric details	Indicator of
Current Redis connection state: -1 – error 0 – disconnected 1 – connected 2 – ready	**Type:** gauge **Label:** location, redis_cluster_name **Sample value:** 2	
service_version_info Displays the version of Voice Config Service that is currently running. In the case of this metric, the labels provide the important information. The metric value is always 1 and does not provide any information.	**Unit:** N/A **Type:** gauge **Label:** version **Sample value:** service_version_info{version="100.0.1000006"} 1	
config_health_level Health level of the config node: -1 – error 0 – fail 1 – degraded 2 – pass	**Unit:** N/A **Type:** gauge **Label:** **Sample value:** 2	Errors
config_healthcheck_generic_exception Generic error during health check.	**Unit:** N/A **Type:** gauge **Label:** **Sample value:** 0	

Alerts

The following alerts are defined for Config Service.

Alert	Severity	Description	Based on	Threshold
Redis disconnected for 5 minutes	Warning	Actions: • If the alarm is triggered for multiple services, make sure there are no issues with Redis, then restart Redis. • If the alarm is triggered only for the pod {{ $labels.pod }}, check to see if there is an issue with the	redis_state	Redis is not available for pod {{ $labels.pod }} for 5 minutes.

Alert	Severity	Description	Based on	Threshold
		pod.		
Redis disconnected for 10 minutes	Critical	Actions: • If the alarm is triggered for multiple services, make sure there are no issues with Redis, then restart Redis. • If the alarm is triggered only for the pod {{ $labels.pod }}, check to see if there is an issue with the pod.	redis_state	Redis is not available for the pod {{ $labels.pod }} for 10 minutes.
Pod Failed	Warning	Actions: • One of the containers in the pod has entered a failed state. Check the Kibana logs for the reason.	kube_pod_status_phase	Pod failed {{ $labels.pod }}.
Pod Unknown state	Warning	Actions: • If the alarm is triggered for multiple services, make sure there are no issues with the Kubernetes cluster. • If the alarm is triggered only for the pod {{ $labels.pod }}, check to see whether the image is correct and if the container is starting up.	kube_pod_status_phase	Pod {{ $labels.pod }} is in Unknown state for 5 minutes.

Alert	Severity	Description	Based on	Threshold
Pod Pending state	Warning	Actions: • If the alarm is triggered for multiple services, make sure the Kubernetes nodes where the pod is running are alive in the cluster. • If the alarm is triggered only for the pod {{ $labels.pod }}, check the health of the pod.	kube_pod_status_phase	Pod {{ $labels.pod }} is in Pending state for 5 minutes.
Pod Not ready for 10 minutes	Critical	Actions: • If this alarm is triggered, check whether the CPU is available for the pods. • Check whether the port of the pod is running and serving the request.	kube_pod_status_ready	Pod {{ $labels.pod }} is in NotReady state for 10 minutes.
Container restarted repeatedly	Critical	Actions: • One of the containers in the pod has entered a failed state. Check the Kibana logs for the reason.	kube_pod_container_status_restarts_total	Container {{ $labels.container }} was restarted 5 or more times within 15 minutes.
Pod memory greater than 65%	Warning	High memory usage for pod {{ $labels.pod }}. Actions: • Check whether	container_memory_working_set_bytes kube_pod_container_resource_requests_memory_bytes	Container {{ $labels.container }} memory usage exceeded 65% for 5 minutes.

Alert	Severity	Description	Based on	Threshold
		the horizontal pod autoscaler has triggered and if the maximum number of pods has been reached. • Check Grafana for abnormal load. • Collect the service logs; raise an investigation ticket.		
Pod memory greater than 80%	Critical	Critical memory usage for pod {{ $labels.pod }}. Actions: • Check whether the horizontal pod autoscaler has triggered and if the maximum number of pods has been reached. • Check Grafana for abnormal load. • Restart the service. • Collect the service logs; raise an investigation ticket.	container_memory_working_set_bytes kube_pod_container_resource_requests_memory_bytes	Container {{ $labels.container }} memory usage exceeded 80% for 5 minutes.
Pod CPU greater than 65%	Warning	High CPU load for pod {{ $labels.pod }}. Actions: • Check whether the horizontal	container_cpu_usage_seconds_total container_spec_cpu_period	Container {{ $labels.container }} CPU usage exceeded 65% for 5 minutes.

Alert	Severity	Description	Based on	Threshold
		pod autoscaler has triggered and if the maximum number of pods has been reached. • Check Grafana for abnormal load. • Collect the service logs; raise an investigation ticket.		
Pod CPU greater than 80%	Critical	Critical CPU load for pod {{ $labels.pod }}. Actions: • Check whether the horizontal pod autoscaler has triggered and if the maximum number of pods has been reached. • Check Grafana for abnormal load. • Restart the service. • Collect the service logs; raise an investigation ticket.	container_cpu_usage_seconds_total, container_spec_cpu_period	Container {{ $labels.container }} CPU usage exceeded 80% for 5 minutes.

Dial Plan Service metrics and alerts

Contents

Find the metrics Dial Plan Service exposes and the alerts defined for Dial Plan Service.

Service	CRD or annotations?	Port	Endpoint/Selector	Metrics update interval
Dial Plan Service	Supports both CRD and annotations	8800	http://:8800/metrics	30 seconds

See details about:

- Dial Plan Service metrics
- Dial Plan Service alerts

Metrics

You can query Prometheus directly to see all the metrics that the Voice Dial Plan Service exposes. The following metrics are likely to be particularly useful. Genesys does not commit to maintain other currently available Dial Plan Service metrics not documented on this page.

Metric and description	Metric details	Indicator of
dialplan_health_level Aggregated health level of the dialplan node for dependent services such as Redis and the Envoy sidecar connection: -1 – fail 0 – starting 1 – degraded 2 – pass	**Unit:** N/A **Type:** gauge **Label:** **Sample value:** 2	Health
dialplan_redis_state Current Redis connection state: 0 – disconnected 1 – connecting 2 – connected	**Unit:** N/A **Type:** gauge **Label:** redis_cluster_name **Sample value:** 2	Health
dialplan_total_request Number of dialplan requests received.	**Unit:** N/A **Type:** counter **Label:** tenant, pod, operation_type **Sample value:**	Traffic
dialplan_failure_response	**Unit:** N/A	Traffic

Metric and description	Metric details	Indicator of
The number of Dial Plan failure responses.	**Type:** counter **Label:** tenant, pod, operation_type, status, reason **Sample value:**	
dialplan_response_time Dialplan request processing duration histogram, in ms.	**Unit:** milliseconds **Type:** histogram **Label:** **Sample value:**	Latency
dialplan_redis_cache_latency_msec Redis fetch latency, measured in milliseconds.	**Unit:** milliseconds **Type:** histogram **Label:** tenant **Sample value:**	Latency

Alerts

The following alerts are defined for Dial Plan Service.

Alert	Severity	Description	Based on	Threshold
DialPlan processing time > 0.5 seconds	Warning	Actions: • If the alarm is generated for all dialplan pods, then Redis or network delay might be the most probable cause. • If the alarm is generated in a single dialplan pod, then it might be due to Envoy or a network issue.	dialplan_response_time	When the latency for 95% of the dial plan messages is more than 0.5 seconds for a duration of 5 minutes, then this warning alarm is raised for the {{ $labels.container }}.
DialPlan processing time > 2 seconds	Critical	Actions: • If the alarm is generated for all dialplan pods, then Redis or network delay might be the	dialplan_response_time	If the latency for 95% of the dial plan messages is more than 2 seconds for a duration of 5 minutes, then this warning alarm is raised for the {{ $labels.container

Alert	Severity	Description	Based on	Threshold
		most probable cause. • If the alarm is generated in a single dialplan pod, then it might be due to Envoy or a network issue.		}}.
Aggregated service health failing for 5 minutes	Critical	Actions: • Check the dialplan dashboard for Aggregated Service Health errors and, in case of a Redis error, first check for any issues/crashes in the pod and then restart Redis. • In the case of an Envoy error, the dialplan container will be restarted by the liveness probe. If the issue still exists after that, restart the pod.	dialplan_health_level	Dependent services or the Envoy sidecar is not available for 5 minutes in the pod {{ $labels.pod }}.
Redis disconnected for 5 minutes	Warning	Actions: • If the alarm is triggered for multiple services, make sure there are no issues with Redis and then restart Redis. • If the alarm is triggered only for the pod {{ $labels.pod }}, check to see if	redis_state	Redis is not available for the pod {{ $labels.pod }} for 5 minutes.

Alert	Severity	Description	Based on	Threshold
		there is an issue with the pod.		
Redis disconnected for 10 minutes	Critical	Actions: • If the alarm is triggered for multiple services, make sure there are no issues with Redis and then restart Redis. • If the alarm is triggered only for the pod {{ $labels.pod }}, check to see if there is an issue with the pod.	redis_state	Redis is not available for the pod {{ $labels.pod }} for 10 minutes.
Pod Failed	Warning	Actions: • One of the containers in the pod has entered a failed state. Check the Kibana logs for the reason.	kube_pod_status_phase	Pod {{ $labels.pod }} failed.
Pod Unknown state	Warning	Actions: • If the alarm is triggered for multiple services, make sure there are no issues with the Kubernetes cluster. • If the alarm is triggered only for the pod {{ $labels.pod }}, check whether the image is correct and if the container is	kube_pod_status_phase	Pod {{ $labels.pod }} is in Unknown state for 5 minutes.

Alert	Severity	Description	Based on	Threshold
		starting up.		
Pod Pending state	Warning	Actions: • If the alarm is triggered for multiple services, make sure the Kubernetes nodes where the pod is running are alive in the cluster. • If the alarm is triggered only for the pod {{ $labels.pod }}, check the health of the pod.	kube_pod_status_phase	Pod {{ $labels.pod }} is in the Pending state for 5 minutes.
Pod Not ready for 10 minutes	Critical	Actions: • If this alarm is triggered, check whether the CPU is available for the pods. • Check whether the port of the pod is running and serving the request.	kube_pod_status_ready	Pod {{ $labels.pod }} is in the NotReady state for 10 minutes.
Pod memory greater than 65%	Warning	High memory usage for pod {{ $labels.pod }}. Actions: • Check whether the horizontal pod autoscaler has triggered and if the maximum number of pods has been reached.	container_memory_working_set_bytes kube_pod_container_resource_limits	Container {{ $labels.container }} memory usage exceeded 65% for 5 minutes.

Alert	Severity	Description	Based on	Threshold
		• Check Grafana for abnormal load. • Collect the service logs; raise an investigation ticket		
Pod memory greater than 80%	Critical	Critical memory usage for pod {{ $labels.pod }}. Actions: • Check whether the horizontal pod autoscaler has triggered and if the maximum number of pods has been reached. • Check Grafana for abnormal load. • Restart the service. • Collect the service logs; raise an investigation ticket.	container_memory_working_set_bytes kube_pod_container_resource_limits	Container {{ $labels.container }} memory usage exceeded 80% for 5 minutes.
Pod CPU greater than 65%	Warning	High CPU load for pod {{ $labels.pod }}. Actions: • Check whether the horizontal pod autoscaler has triggered and if the maximum number of pods has been reached. • Check Grafana	container_cpu_usage_seconds_total kube_pod_container_resource_limits	Container {{ $labels.container }} CPU usage exceeded 65% for 5 minutes.

Alert	Severity	Description	Based on	Threshold
		for abnormal load. • Collect the service logs; raise an investigation ticket.		
Pod CPU greater than 80%	Critical	Critical CPU load for pod {{ $labels.pod }}. Actions: • Check whether the horizontal pod autoscaler has triggered and if the maximum number of pods has been reached. • Check Grafana for abnormal load. • Restart the service. • Collect the service logs; raise an investigation ticket.	container_cpu_usage_seconds_total, kube_pod_container_resource_limits	Container {{ $labels.container }} CPU usage exceeded 80% for 5 minutes.

FrontEnd Service metrics and alerts

Contents

Find the metrics FrontEnd Service exposes and the alerts defined for FrontEnd Service.

Service	CRD or annotations?	Port	Endpoint/Selector	Metrics update interval
FrontEnd Service	Supports both CRD and annotations	9101	http://:9101/metrics	30 seconds

See details about:

- FrontEnd Service metrics
- FrontEnd Service alerts

Metrics

Voice FrontEnd Service exposes Genesys-defined, FrontEnd Service–specific metrics as well as some standard Kafka metrics. You can query Prometheus directly to see all the metrics that the FrontEnd Service exposes. The following metrics are likely to be particularly useful. Genesys does not commit to maintain other currently available FrontEnd Service metrics not documented on this page.

Metric and description	Metric details	Indicator of
kafka_producer_queue_depth Number of Kafka producer pending events.	**Unit:** N/A **Type:** gauge **Label:** kafka_location **Sample value:** 0	
kafka_producer_queue_age_seconds Age of the oldest producer pending event, in seconds.	**Unit:** seconds **Type:** gauge **Label:** kafka_location **Sample value:**	
kafka_producer_error_total Number of Kafka producer errors.	**Unit:** N/A **Type:** counter **Label:** kafka_location **Sample value:**	
kafka_producer_state Current state of the Kafka producer.	**Unit:** N/A **Type:** gauge **Label:** kafka_location **Sample value:**	
kafka_producer_biggest_event_size	**Unit:**	

Metric and description	Metric details	Indicator of
Biggest event size so far.	**Type:** gauge **Label:** kafka_location, topic **Sample value:** 515	
kafka_max_request_size Exposed config to compare with biggest event size.	**Unit:** **Type:** gauge **Label:** kafka_location **Sample value:**	
log_output_bytes_total Total amount of log output, in bytes.	**Unit:** bytes **Type:** counter **Label:** level, format, module **Sample value:**	
sipfe_requests_total Number of requests.	**Unit:** N/A **Type:** counter **Label:** tenant **Sample value:**	Traffic
sipfe_responses_total Number of responses for the requests.	**Unit:** N/A **Type:** counter **Label:** tenant **Sample value:**	Traffic
sipfe_sip_nodes_total Number of SIP nodes that are alive.	**Unit:** N/A **Type:** gauge **Label:** **Sample value:**	
sipfe_sip_node_requests_total Number of requests to the SIP nodes.	**Unit:** N/A **Type:** counter **Label:** sip_node_id, tenant **Sample value:**	
sipfe_sip_node_responses_total Number of responses from the SIP nodes for the requests.	**Unit:** N/A **Type:** counter **Label:** sip_node_id, tenant, status **Sample value:**	
sipfe_sip_node_request_duration_seconds The duration of time between the SIP node request and the response, measured in seconds.	**Unit:** seconds **Type:** histogram **Label:** le, sip_node_id, tenant, status **Sample value:**	Latency
service_version_info Displays the version of Voice FrontEnd Service that is currently running. In the case of this metric, the labels provide the important information. The metric value is always 1 and does not provide any information.	**Unit:** **Type:** gauge **Label:** version **Sample value:** service_version_info{version="100.0.1000006"} 1	

Metric and description	Metric details	Indicator of
sipfe_health_level Health level of the sipfe node: -1 – fail 0 – starting 1 – degraded 2 – pass	**Unit:** N/A **Type:** gauge **Label:** **Sample value:** 2	Errors
sipfe_health_check_error Health check errors for the sipfe node: 1 – has error 0 – no error	**Unit:** N/A **Type:** gauge **Label:** reason **Sample value:** 0	Errors

Alerts

The following alerts are defined for FrontEnd Service.

Alert	Severity	Description	Based on	Threshold
Too many Kafka pending producer events	Critical	Actions: • Make sure there are no issues with Kafka or {{ $labels.pod }} pod's CPU and network.	kafka_producer_queue_depth	Too many Kafka producer pending events for pod {{ $labels.pod }} (more than 100 in 5 minutes).
Too many received requests without a response	Critical	Actions: • Collect the service logs for pod {{ $labels.pod }}; raise an investigation ticket. • Restart the service.	sipfe_requests_total	For too many requests, the Front End service at pod {{ $labels.pod }} did not send any response (more than 100 requests without a response, measured over 5 minutes).
SIP Cluster Service response latency is too high	Critical	Actions: • If the alarm is triggered for multiple pods, make sure there are no	sipfe_sip_node_request_duration_seconds_bucket	Latency for 95% of messages is more than 0.5 seconds for service {{ $labels.container }}.

Alert	Severity	Description	Based on	Threshold
		issues with the SIP Cluster Service (CPU, memory, or network overload). • If the alarm is triggered only for pod {{ $labels.pod }}, check if there is an issue with the pod (CPU, memory, or network overload).		
No requests received	Critical	Absence of received requests for pod {{ $labels.pod }}. Actions: • For pod {{ $labels.pod }}, make sure there are no issues with Orchestration Service and Tenant Service or the network to them.	sipfe_requests_total	increase(sipfe_requests_total{pod .+"}[5m]) 100
Too many failure responses sent	Critical	Too many failure responses are sent by the Front End service at pod {{ $labels.pod }}. Actions: • For pod {{ $labels.pod }}, make sure received requests are valid.	sipfe_responses_total	More than 100 failure responses in 5 consecutive minutes.
Too many Kafka producer errors	Critical	Kafka responds with errors at pod {{ $labels.pod }}.	kafka_producer_error_total	More than 100 errors in 5 consecutive minutes.

Alert	Severity	Description	Based on	Threshold
		Actions: • For pod {{ $labels.pod }}, make sure there are no issues with Kafka.		
Too many SIP Cluster Service error responses	Critical	SIP Cluster Service responds with errors at pod {{ $labels.pod }}. Actions: • If the alarm is triggered for multiple pods, make sure there are no issues with the SIP Cluster Service (CPU, memory, or network overload). • If the alarm is triggered only for pod {{ $labels.pod }}, check if there are issues with requests sent by the pod.	sipfe_sip_node_responses_total	More than 100 errors in 5 consecutive minutes.
Kafka not available	Critical	Kafka is not available for pod {{ $labels.pod }}. Actions: • If the alarm is triggered for multiple services, make sure there are no issues with Kafka, and then restart Kafka. • If the alarm is triggered only	kafka_producer_state	Kafka is not available for pod {{ $labels.pod }} for 5 consecutive minutes.

Alert	Severity	Description	Based on	Threshold
		for pod {{ $labels.pod }}, check if there is an issue with the pod.		
SIP Node(s) is not available	Critical	No available SIP Nodes for pod {{ $labels.pod }}. Actions: • If the alarm is triggered for multiple services, make sure there are no issues with SIP Nodes, and then restart SIP Nodes. • If the alarm is triggered only for pod {{ $labels.pod }}, check if there is an issue with the pod or the network to SIP Nodes.	sipfe_sip_nodes_total	No available SIP Nodes for pod {{ $labels.pod }} for 5 consecutive minutes.
Pod status Failed	Warning	Pod {{ $labels.pod }} is in Failed state. Actions: • Restart the pod. Check to see if there are any issues with the pod after restart.	kube_pod_status_phase	Pod {{ $labels.pod }} is in Failed state.
Pod status Unknown	Warning	Pod {{ $labels.pod }} is in Unknown state for 5 minutes. Actions: • Restart the pod. Check to see if there are	kube_pod_status_phase	Pod {{ $labels.pod }} is in Unknown state for 5 minutes.

Alert	Severity	Description	Based on	Threshold
		any issues with the pod after restart.		
Pod status Pending	Warning	Pod {{ $labels.pod }} is in Pending state for 5 minutes. Actions: • Restart the pod. Check to see if there are any issues with the pod after restart.	kube_pod_status_phase	Pod {{ $labels.pod }} is in Pending state for 5 minutes.
Pod status NotReady	Critical	Pod {{ $labels.pod }} is in the NotReady state for 5 minutes. Actions: • Restart the pod. Check to see if there are any issues with the pod after restart.	kube_pod_status_ready	Pod {{ $labels.pod }} is in the NotReady state for 5 minutes.
Container restarted repeatedly	Critical	Container {{ $labels.container }} was restarted 5 or more times within 15 minutes. Actions: • Check if a new version of the image was deployed. • Check for issues with the Kubernetes cluster.	kube_pod_container_status_restarts_total	Container {{ $labels.container }} was restarted 5 or more times within 15 minutes.
Max replicas is not sufficient for 5 mins	Critical	For the past 5 minutes, the desired number of replicas is higher than the number	kube_statefulset_replicas, kube_statefulset_status_replicas	Desired number of replicas is higher than current available replicas for the past 5

Alert	Severity	Description	Based on	Threshold
		of replicas currently available. Actions: • Check resources available for Kubernetes. Increase resources, if necessary.		minutes.
Pods scaled up greater than 80%	Critical	For the past 5 minutes, the desired number of replicas is greater than the number of replicas currently available. Actions: • Check resources available for Kubernetes. Increase resources, if necessary.	kube_hpa_status_current_replicas, kube_hpa_spec_max_replicas	(kube_hpa_status_current_replicas{node-hpa"} * 100) / kube_hpa_spec_max_replicas{node-hpa"} > 80 for: 5m
Pods less than Min Replicas	Critical	The current number of replicas is lower than the minimum number of replicas that should be available. Actions: • Check if Kubernetes cannot deploy new pods or if pods are failing in their status to be active/ read.	kube_hpa_status_current_replicas, kube_hpa_spec_min_replicas	For the past 5 minutes, the current number of replicas is lower than the minimum number of replicas that should be available.
Pod CPU greater than 65%	Warning	High CPU load for pod {{ $labels.pod }}. Actions:	container_cpu_usage_seconds_total, container_spec_cpu_period	Container {{ $labels.container }}. CPU usage exceeded 65% for 5 minutes.

Alert	Severity	Description	Based on	Threshold
		• Check whether the horizontal pod autoscaler has triggered and if the maximum number of pods has been reached. • Check Grafana for abnormal load. • Collect the service logs for pod {{ $labels.pod }}; raise an investigation ticket.		
Pod CPU greater than 80%	Critical	Critical CPU load for pod {{ $labels.pod }}. Actions: • Check whether the horizontal pod autoscaler has triggered and if the maximum number of pods has been reached. • Check Grafana for abnormal load. • Restart the service.	container_cpu_usage_seconds_total, container_spec_cpu_period	Container {{ $labels.container }} CPU usage exceeded 80% for 5 minutes.
Pod memory greater than 65%	Warning	High memory usage for pod {{ $labels.pod }}. Actions: • Check whether the horizontal pod autoscaler has triggered	container_memory_working_set_bytes, kube_pod_container_resource_requests_memory_bytes	Container {{ $labels.container }} memory usage exceeded 65% for 5 minutes.

Alert	Severity	Description	Based on	Threshold
		and if the maximum number of pods has been reached. • Check Grafana for abnormal load. • Collect the service logs for pod {{ $labels.pod }}; raise an investigation ticket.		
Pod memory greater than 80%	Critical	Critical memory usage for pod {{ $labels.pod }}. Actions: • Check whether the horizontal pod autoscaler has triggered and if the maximum number of pods has been reached. • Check Grafana for abnormal load. • Restart the service for pod {{ $labels.pod }}.	container_memory_working_set_bytes / kube_pod_container_resource_requests_memory_bytes	Container {{ $labels.container }} memory usage exceeded 80% for 5 minutes.

ORS metrics and alerts

Contents

Find the metrics ORS exposes and the alerts defined for ORS.

Service	CRD or annotations?	Port	Endpoint/Selector	Metrics update interval
ORS	Supports both CRD and annotations	11200	http://:11200/metrics	30 seconds

See details about:

- ORS metrics
- ORS alerts

Metrics

You can query Prometheus directly to see all the metrics that the Voice Orchestration Service exposes. The following metrics are likely to be particularly useful. Genesys does not commit to maintain other currently available Orchestration Service metrics not documented on this page.

Metric and description	Metric details	Indicator of
orsnode_callevents Total number of received call events.	**Unit:** N/A **Type:** counter **Label:** **Sample value:**	Traffic
orsnode_ha_writes The number of HA writes to Redis.	**Unit:** N/A **Type:** counter **Label:** **Sample value:**	Traffic
orsnode_ha_reads The number of HA reads from Redis.	**Unit:** N/A **Type:** counter **Label:** **Sample value:**	Traffic
orsnode_interactions The number of active interactions.	**Unit:** N/A **Type:** gauge **Label:** **Sample value:**	Traffic
orsnode_total_interactions The total number of interactions that	**Unit:** N/A **Type:** counter	Traffic

Metric and description	Metric details	Indicator of
have been created.	**Label:** **Sample value:**	
orsnode_cleared_interactions The total number of call interactions that have been cleared.	**Unit:** N/A **Type:** counter **Label:** **Sample value:**	
orsnode_strategies The number of strategies that are running.	**Unit:** N/A **Type:** gauge **Label:** **Sample value:**	Traffic
orsnode_total_strategies The total number of strategies that have been created.	**Unit:** N/A **Type:** counter **Label:** **Sample value:**	Traffic
orsnode_load_errors The total number of strategy load errors.	**Unit:** N/A **Type:** counter **Label:** **Sample value:**	Errors
orsnode_fetch_errors The total number of errors encountered when a strategy tried to fetch data from a Designer Application Server (DAS).	**Unit:** N/A **Type:** counter **Label:** **Sample value:**	Errors
orsnode_config_errors The total number of strategy configuration errors.	**Unit:** N/A **Type:** counter **Label:** **Sample value:**	Errors
orsnode_invoke_errors The total number of strategy invoke errors.	**Unit:** N/A **Type:** counter **Label:** **Sample value:**	Errors
orsnode_treatments The total number of strategy treatments.	**Unit:** N/A **Type:** counter **Label:** **Sample value:**	Traffic
orsnode_failed_treatments The total number of failed strategy treatments.	**Unit:** N/A **Type:** counter **Label:** **Sample value:**	Errors
orsnode_userdata_updates The total number of times that a strategy	**Unit:** N/A **Type:** counter	Traffic

Metric and description	Metric details	Indicator of
updated user data.	**Label:** **Sample value:**	
orsnode_scxml_transitions The total number of SCXML transitions.	**Unit:** N/A **Type:** counter **Label:** **Sample value:**	Traffic
orsnode_scxml_events The total number of SCXML events.	**Unit:** N/A **Type:** counter **Label:** **Sample value:**	Traffic
orsnode_scxml_error_events The total number of SCXML `error.*` events.	**Unit:** N/A **Type:** counter **Label:** **Sample value:**	Errors
orsnode_http_fetch_requests The total number of HTTP `fetch` requests.	**Unit:** N/A **Type:** counter **Label:** **Sample value:**	Errors
orsnode_http_fetch_duration The HTTP fetch time, measured in milliseconds (ms).	**Unit:** milliseconds **Type:** histogram **Label:** **Sample value:**	Latency
orsnode_http_fetch_errors The total number of HTTP `fetch` errors.	**Unit:** N/A **Type:** counter **Label:** **Sample value:**	Errors
orsnode_http_fetch_error_status Status of the HTTP `fetch` error.	**Unit:** **Type:** histogram **Label:** **Sample value:**	Errors
orsnode_urs_rlib_latency_msec The Universal Routing Server (URS) `rlib` latency, measured in milliseconds (ms).	**Unit:** milliseconds **Type:** histogram **Label:** **Sample value:**	Latency
orsnode_urs_rlib_errors The total number of URS `rlib` errors.	**Unit:** N/A **Type:** counter **Label:** **Sample value:**	Errors
orsnode_urs_rlib_requests The total number of URS `rlib` requests.	**Unit:** N/A **Type:** counter	

Metric and description	Metric details	Indicator of
	Label: **Sample value:**	
orsnode_urs_rlib_events The total number of URS `rlib` events.	**Unit:** N/A **Type:** counter **Label:** **Sample value:**	
orsnode_urs_rlib_timeouts The total number of URS `rlib` timeouts.	**Unit:** N/A **Type:** counter **Label:** **Sample value:**	
orsnode_redis_state Current Redis connection state.	**Unit:** N/A **Type:** gauge **Label:** redis_cluster_name **Sample value:**	
orsnode_redis_disconnect The number of times that the ORS node disconnected from Redis.	**Unit:** N/A **Type:** counter **Label:** **Sample value:**	
orsnode_sdr_messages_sent The number of SDR messages that have been sent.	**Unit:** N/A **Type:** counter **Label:** **Sample value:**	
orsnode_rq_latency_msec Redis queue latency, measured in milliseconds (ms).	**Unit:** milliseconds **Type:** histogram **Label:** le, service **Sample value:**	Latency
orsnode_routing_latency_msec Routing latency, measured in milliseconds (ms).	**Unit:** milliseconds **Type:** histogram **Label:** **Sample value:**	Latency
orsnode_rstream_latency_msec Redis stream latency, measured in (ms).	**Unit:** milliseconds **Type:** histogram **Label:** le, node **Sample value:**	Latency
orsnode_digital_latency_msec Digital stream latency, measured in milliseconds (ms).	**Unit:** milliseconds **Type:** histogram **Label:** **Sample value:**	Latency
orsnode_sip_health_check ORS health check.	**Unit:** N/A **Type:** gauge	

Metric and description	Metric details	Indicator of
	Label: node **Sample value:**	
orsnode_ixn_health_check Interaction health check.	**Unit:** N/A **Type:** gauge **Label:** **Sample value:**	
orsnode_rq_state Current Redis queue connection state.	**Unit:** N/A **Type:** gauge **Label:** **Sample value:**	
orsnode_ixn_events Total number of interaction stream events received.	**Unit:** N/A **Type:** counter **Label:** **Sample value:**	
orsnode_rq_disconnect Number of times the ORS node disconnected from the RQ Service.	**Unit:** N/A **Type:** counter **Label:** **Sample value:**	
service_version_info Displays the version of Voice Orchestration Service that is currently running. In the case of this metric, the labels provide the important information. The metric value is always 1 and does not provide any information.	**Unit:** N/A **Type:** gauge **Label:** version **Sample value:** service_version_info{version="100.0.1000006"} 1	
orsnode_route_redirected Total number of EventRouteUsed events without a ReferenceID.	**Unit:** N/A **Type:** counter **Label:** **Sample value:**	
orsnode_balancer_stream_state The state of the voice balancer stream.	**Unit:** N/A **Type:** gauge **Label:** balancer_stream_type **Sample value:**	
orsnode_high_memory Indicates when the ORS node is using a lot of memory.	**Unit:** N/A **Type:** gauge **Label:** **Sample value:**	
orsnode_urs_rlib_state Indicates a Tenant `rlib` request timeout.	**Unit:** N/A **Type:** gauge **Label:** **Sample value:**	
orsnode_stuck_interactions	**Unit:** N/A	

Metric and description	Metric details	Indicator of
The number of stuck interactions.	**Type:** gauge **Label:** **Sample value:**	
orsnode_urs_scxml_submit_requests The total number of URS `SCXMLSubmit` requests.	**Unit:** N/A **Type:** counter **Label:** **Sample value:**	
orsnode_urs_scxml_cancel_requests The total number of URS `SCXMLQueueCancel` requests.	**Unit:** N/A **Type:** counter **Label:** **Sample value:**	
orsnode_urs_queue_submit_done_events Total number of URS `queue.submit.done` events.	**Unit:** N/A **Type:** counter **Label:** **Sample value:**	
orsnode_health_level Summarized health level of the ORS node: -1 – fail 0 – starting 1 – degraded 2 – pass	**Unit:** N/A **Type:** gauge **Label:** **Sample value:**	
orsnode_health_check_error Health check errors for the ORS node: 1 – has error 0 – no error	**Unit:** N/A **Type:** gauge **Label:** reason **Sample value:**	Errors
orsnode_running_applications The number of active sessions for each Designer application.	**Unit:** N/A **Type:** gauge **Label:** **Sample value:**	
orsnode_failed_applications The number of failed sessions for each Designer application.	**Unit:** N/A **Type:** gauge **Label:** **Sample value:**	
orsnode_total_applications The total number of sessions created for each Designer application.	**Unit:** N/A **Type:** gauge **Label:** **Sample value:**	
orsnode_failed_scripts The number of scripts that failed to load in the Tenant Service configuration	**Unit:** N/A **Type:** gauge **Label:**	

Metric and description	Metric details	Indicator of
management environment.	**Sample value:**	
orsnode_session_load_time_msec The time it takes for the strategy to be compiled and go through its initial states.	**Unit:** milliseconds **Type:** histogram **Label:** **Sample value:**	
orsnode_service_started Timestamp when the ORS node started.	**Unit:** N/A **Type:** gauge **Label:** started **Sample value:**	
orsnode_total_terminal_requests Total number of terminal requests (like Deliver, PlaceInQueue, StopProcessing for Digital and RequestClearCall, RequestRouteCall for Voice).	**Unit:** N/A **Type:** counter **Label:** **Sample value:**	
orsnode_total_non_terminal_requests Total number of non-terminal requests to the Interaction Server (for Digital) or SIP Server (for Voice).	**Unit:** N/A **Type:** counter **Label:** **Sample value:**	
orsnode_sip_post_errors Total number of errors encountered in POST requests to the SIP node.	**Unit:** N/A **Type:** counter **Label:** **Sample value:**	Errors
orsnode_pending_tlib_requests Total number of pending TLib requests.	**Unit:** N/A **Type:** counter **Label:** **Sample value:**	
orsnode_sips_rest_connections The number of active REST connections with SIP Cluster Service.	**Unit:** N/A **Type:** gauge **Label:** **Sample value:**	
orsnode_number_compiled_applications The number of compiled applications in the cache.	**Unit:** N/A **Type:** counter **Label:** **Sample value:**	
orsnode_cached_applications_size The sum of the sizes of the cached applications.	**Unit:** **Type:** gauge **Label:** **Sample value:**	
orsnode_tlib_latency_msec The TLib Rest API request latency,	**Unit:** milliseconds **Type:** histogram	Latency

Metric and description	Metric details	Indicator of
measured in (ms).	**Label:** le **Sample value:**	
orsnode_application_size The compiled size of the Designer application.	**Unit:** **Type:** gauge **Label:** **Sample value:**	
orsnode_application_microstep_count The number of microsteps while executing the Designer application.	**Unit:** N/A **Type:** gauge **Label:** **Sample value:**	
orsnode_application_run_time_msec The length of time the Designer application was running, measured in milliseconds (ms).	**Unit:** milliseconds **Type:** gauge **Label:** **Sample value:**	
orsnode_application_compiled_date The date on which the Designer application was compiled.	**Unit:** N/A **Type:** gauge **Label:** **Sample value:**	
orsnode_application_last_invoked_date The date when the Designer application was last invoked.	**Unit:** N/A **Type:** gauge **Label:** **Sample value:**	

Alerts

The following alerts are defined for ORS.

Alert	Severity	Description	Based on	Threshold
Number of running strategies is too high	Warning	Too many active sessions. Actions: • Check whether the horizontal pod autoscaler has triggered and if the maximum number of pods has been reached.	orsnode_strategies	More than 400 strategies running in 5 consecutive seconds.

Alert	Severity	Description	Based on	Threshold
		• Check the number of voice, digital, and callback calls in the system.		
Number of running strategies is critical	Critical	Too many active sessions. Actions: • Check whether the horizontal pod autoscaler has triggered and if the maximum number of pods has been reached. • Check the number of voice, digital, and callback calls in the system.	orsnode_strategies	More than 600 strategies running in 5 consecutive seconds.
Redis disconnected for 5 minutes	Warning	Actions: • If the alarm is triggered for multiple services, make sure there are no issues with Redis, and then restart Redis. • If alarm is triggered only for pod {{ $labels.pod }}, check if there is an issue with the pod.	redis_state	Redis is not available for the pod {{ $labels.pod }} for 5 minutes.
Redis disconnected for 10 minutes	Critical	Actions: • If the alarm is triggered for multiple services, make	redis_state	Redis is not available for the pod {{ $labels.pod }} for 10 minutes.

Alert	Severity	Description	Based on	Threshold
		sure there are no issues with Redis, and then restart Redis. • If the alarm is triggered only for pod {{ $labels.pod }}, check if there is an issue with the pod.		
Pod status Failed	Warning	Pod {{ $labels.pod }} failed. Actions: • One of the containers in the pod has entered a Failed state. Check the Kibana logs for the reason.	kube_pod_status_phase	Pod {{ $labels.pod }} is in Failed state.
Pod in Unknown state	Warning	Pod {{ $labels.pod }} is in Unknown state. Actions: • If the alarm is triggered for multiple services, make sure there are no issues with the Kubernetes cluster. • If the alarm is triggered only for pod {{ $labels.pod }}, check whether the image is correct and if the container is starting up.	kube_pod_status_phase	Pod {{ $labels.pod }} is in Unknown state for 5 minutes.
Pod in Pending state	Warning	Pod {{ $labels.pod }} is in Pending state.	kube_pod_status_phase	Pod {{ $labels.pod }} is in Pending state for 5

Alert	Severity	Description	Based on	Threshold
		Actions: • If the alarm is triggered for multiple services, make sure the Kubernetes nodes where the pod is running are alive in the cluster. • If the alarm is triggered only for the pod {{ $labels.pod }}, check the health of the pod.		minutes.
Pod Not ready for 10 minutes	Critical	Pod {{ $labels.pod }} in NotReady state. Actions: • If this alarm is triggered, check whether the CPU is available for the pods. • Check whether the port of the pod is running and serving the request.	kube_pod_status_ready	Pod {{ $labels.pod }} in NotReady state for 10 minutes.
Container restored repeatedly	Critical	Actions: • One of the containers in the pod has entered a failed state. Check the Kibana logs for the reason.	kube_pod_container_status_restarts_total	Container {{ $labels.container }} was restarted 5 or more times within 15 minutes.
Pod memory greater than 65%	Warning	High memory usage for pod {{ $labels.pod }}.	container_memory_working_set_bytes, kube_pod_container_resource_requests_memory_bytes	Container {{ $labels.container }} memory usage

Alert	Severity	Description	Based on	Threshold
		Actions: • Check whether the horizontal pod autoscaler has triggered and if the maximum number of pods has been reached. • Check Grafana for abnormal load. • Collect the service logs; raise an investigation ticket.		exceeded 65% for 5 minutes.
Pod memory greater than 80%	Critical	Critical memory usage for pod {{ $labels.pod }}. Actions: • Check whether the horizontal pod autoscaler has triggered and if the maximum number of pods has been reached. • Check Grafana for abnormal load. • Restart the service. • Collect the service logs; raise an investigation ticket.	container_memory_working_set_bytes, kube_pod_container_resource_requests_memory_bytes	Container {{ $labels.container }} memory usage exceeded 80% for 5 minutes.
Pod CPU greater than 65%	Warning	High CPU load for pod {{ $labels.pod }}. Actions:	container_cpu_usage_seconds_total, container_spec_cpu_period	Container {{ $labels.container }} CPU usage exceeded 65% for 5 minutes.

Alert	Severity	Description	Based on	Threshold
		• Check whether the horizontal pod autoscaler has triggered and if the maximum number of pods has been reached. • Check Grafana for abnormal load. • Collect the service logs; raise an investigation ticket.		
Pod CPU greater than 80%	Critical	Critical CPU load for pod {{ $labels.pod }}. Actions: • Check whether the horizontal pod autoscaler has triggered and if the maximum number of pods has been reached. • Check Grafana for abnormal load. • Restart the service. • Collect the service logs; raise an investigation ticket.	container_cpu_usage_seconds_total, container_spec_cpu_period	Container {{ $labels.container }} CPU usage exceeded 80% for 5 minutes.

Voice Registrar Service metrics and alerts

Contents

Find the metrics Voice Registrar Service exposes and the alerts defined for Voice Registrar Service.

Service	CRD or annotations?	Port	Endpoint/Selector	Metrics update interval
Voice Registrar Service	Supports both CRD and annotations	11500	http://:11500/metrics	30 seconds

See details about:

- Voice Registrar Service metrics
- Voice Registrar Service alerts

Metrics

Voice Registrar Service exposes Genesys-defined, Registrar Service–specific metrics as well as some standard Kafka metrics. You can query Prometheus directly to see all the metrics that the Registrar Service exposes. The following metrics are likely to be particularly useful. Genesys does not commit to maintain other currently available Voice Registrar Service metrics not documented on this page.

Metric and description	Metric details	Indicator of
registrar_register_count Number of registrations.	**Unit:** N/A **Type:** counter **Label:** location, tenant **Sample value:**	Traffic
registrar_health_level Health level of the registrar node: -1 – fail 0 – starting 1 – degraded 2 – pass	**Unit:** N/A **Type:** gauge **Label:** **Sample value:**	Errors
registrar_request_latency Time taken to process the request (ms).	**Unit:** milliseconds **Type:** histogram **Label:** le, location, tenant **Sample value:**	Latency
registrar_active_sip_registrations Number of active SIP registrations.	**Unit:** N/A **Type:** gauge **Label:** tenant **Sample value:**	Traffic

Metric and description	Metric details	Indicator of
kafka_consumer_latency Consumer latency is the time difference between when the message is produced and when the message is consumed. That is, the time when the consumer received the message minus the time when the producer produced the message.	**Unit:** **Type:** histogram **Label:** tenant, topic **Sample value:**	Latency
kafka_consumer_state Current Kafka consumer connection state: 0 – disconnected 1 – connected	**Unit:** **Type:** gauge **Label:** **Sample value:**	

Alerts

The following alerts are defined for Voice Registrar Service.

Alert	Severity	Description	Based on	Threshold
Kafka events latency is too high	Warning	Actions: • If the alarm is triggered for multiple topics, make sure there are no issues with Kafka (CPU, memory, or network overload). • If the alarm is triggered only for topic {{ $labels.topic }}, check if there is an issue with the service related to the topic (CPU, memory, or network overload).	kafka_consumer_latency_bucket	Latency for more than 5% of messages is more than 0.5 seconds for topic {{ $labels.topic }}.
Too many Kafka consumer failed health checks	Warning	Actions: • If the alarm is triggered for multiple	kafka_consumer_error_total	Health check failed more than 10 times in 5 minutes for Kafka consumer for topic {{$labels.topic}}.

Alert	Severity	Description	Based on	Threshold
		services, make sure there are no issues with Kafka, and then restart Kafka. • If the alarm is triggered only for {{ $labels.container }}, check if there is an issue with the service.		
Too many Kafka consumer request timeouts	Warning	Actions: • If the alarm is triggered for multiple services, make sure there are no issues with Kafka, and then restart Kafka. • If the alarm is triggered only for {{ $labels.container }}, check if there is an issue with the service.	kafka_consumer_error_total	There were more than 10 request timeouts within 5 minutes for the Kafka consumer for topic {{$labels.topic}}.
Too many Kafka consumer crashes	Critical	Actions: • If the alarm is triggered for multiple services, make sure there are no issues with Kafka, and then restart Kafka. • If the alarm is triggered only for {{ $labels.container }}, check if	kafka_consumer_error_total	There were more than 3 Kafka consumer crashes within 5 minutes for service {{ $labels.container }}.

Alert	Severity	Description	Based on	Threshold
		there is an issue with the service.		
Kafka not available	Critical	Kafka is not available for pod {{ $labels.pod }}. Actions: • If the alarm is triggered for multiple services, make sure there are no issues with Kafka, and then restart Kafka. • If the alarm is triggered only for pod {{ $labels.pod }}, check if there is an issue with the pod.	kafka_producer_state, kafka_consumer_state	Kafka is not available for pod {{ $labels.pod }} for 5 consecutive minutes.
Redis disconnected for 5 minutes	Warning	Actions: • If the alarm is triggered for multiple services, make sure there are no issues with Redis, and then restart Redis. • If the alarm is triggered only for pod {{ $labels.pod }}, check if there is an issue with the pod.	redis_state	Redis is not available for pod {{ $labels.pod }} for 5 minutes.
Redis disconnected for 10 minutes	Critical	Actions: • If the alarm is triggered for multiple services, make sure there are	redis_state	Redis is not available for pod {{ $labels.pod }} for 10 minutes.

Alert	Severity	Description	Based on	Threshold
		no issues with Redis, and then restart Redis. • If the alarm is triggered only for pod {{ $labels.pod }}, check if there is an issue with the pod.		
Pod Failed	Warning	Pod {{ $labels.pod }} failed. Actions: • One of the containers in the pod has entered a failed state. Check the Kibana logs for the reason.	kube_pod_status_phase	Pod {{ $labels.pod }} is in Failed state.
Pod Unknown state	Warning	Pod {{ $labels.pod }} is in Unknown state. Actions: • If the alarm is triggered for multiple services, make sure there are no issues with the Kubernetes cluster. • If the alarm is triggered only for pod {{ $labels.pod }}, check whether the image is correct and if the container is starting up.	kube_pod_status_phase	Pod {{ $labels.pod }} is in Unknown state for 5 minutes.
Pod Pending state	Warning	Pod {{ $labels.pod }} is in Pending state.	kube_pod_status_phase	Pod {{ $labels.pod }} is in Pending state for 5 minutes.

Alert	Severity	Description	Based on	Threshold
		Actions: • If the alarm is triggered for multiple services, make sure the Kubernetes nodes where the pod is running are alive in the cluster. • If the alarm is triggered only for pod {{ $labels.pod }}, check the health of the pod.		
Pod Not ready for 10 minutes	Critical	Actions: • If this alarm is triggered, check whether the CPU is available for the pods. • Check whether the port of the pod is running and serving the request.	kube_pod_status_ready	Pod {{ $labels.pod }} is in the NotReady state for 10 minutes.
Container restarted repeatedly	Critical	Actions: • One of the container in the pod has entered a Failed state. Check the Kibana logs for the reason.	kube_pod_container_status_restarts_total	Container {{ $labels.container }} was restarted 5 or more times within 15 minutes.
Pod CPU greater than 65%	Warning	High CPU load for pod {{ $labels.pod }}. Actions: • Check whether	container_cpu_usage_seconds_total kube_pod_container_resource_limits	Container {{ $labels.container }} CPU usage exceeded 65% for 5 minutes.

Alert	Severity	Description	Based on	Threshold
		the horizontal pod autoscaler has triggered and if the maximum number of pods has been reached. • Check Grafana for abnormal load. • Collect the service logs; raise an investigation ticket.		
Pod memory greater than 65%	Warning	High memory usage for pod {{ $labels.pod }}. Actions: • Check whether the horizontal pod autoscaler has triggered and if the maximum number of pods has been reached. • Check Grafana for abnormal load. • Collect the service logs; raise an investigation ticket.	container_memory_working_set_bytes, kube_pod_container_resource_limits	Container {{ $labels.container }} memory usage exceeded 65% for 5 minutes.
Pod memory greater than 80%	Critical	Critical memory usage for pod {{ $labels.pod }}. Actions: • Check whether the horizontal pod autoscaler has triggered	container_memory_working_set_bytes, kube_pod_container_resource_limits	Container {{ $labels.container }} memory usage exceeded 80% for 5 minutes.

Alert	Severity	Description	Based on	Threshold
		and if the maximum number of pods has been reached. • Check Grafana for abnormal load. • Restart the service. • Collect the service logs: raise an investigation ticket.		
Pod CPU greater than 80%	Critical	Critical CPU load for pod {{ $labels.pod }}. Actions: • Check whether the horizontal pod autoscaler has triggered and if the maximum number of pods has been reached. • Check Grafana for abnormal load.	container_cpu_usage_seconds_total, kube_pod_container_resource_limits	Container {{ $labels.container }} CPU usage exceeded 80% for 5 minutes.

Voice RQ Service metrics and alerts

Contents

Find the metrics Voice RQ Service exposes and the alerts defined for Voice RQ Service.

Service	CRD or annotations?	Port	Endpoint/Selector	Metrics update interval
Voice RQ Service	Supports both CRD and annotations	12000	http://:12000/metrics	30 seconds

See details about:

- Voice RQ Service metrics
- Voice RQ Service alerts

Metrics

You can query Prometheus directly to see all the metrics that the Voice RQ Service exposes. The following metrics are likely to be particularly useful. Genesys does not commit to maintain other currently available Voice RQ Service metrics not documented on this page.

Metric and description	Metric details	Indicator of
rqnode_clients Number of clients connected.	**Unit:** N/A **Type:** gauge **Label:** **Sample value:**	Traffic
rqnode_streams Number of active streams present.	**Unit:** N/A **Type:** gauge **Label:** **Sample value:**	Traffic
rqnode_xreads Number of XREAD requests received.	**Unit:** N/A **Type:** counter **Label:** **Sample value:**	Traffic
rqnode_xadds Number of XADD requests received.	**Unit:** N/A **Type:** counter **Label:** **Sample value:**	Traffic
rqnode_redis_state Current Redis connection state.	**Unit:** N/A **Type:** gauge	Errors

Metric and description	Metric details	Indicator of
	Label: **Sample value:**	
rqnode_redis_disconnects The number of Redis disconnects that occurred for the RQ node.	**Unit:** **Type:** counter **Label:** **Sample value:**	Errors
rqnode_consul_leader_error Number of errors received from Consul during the leadership process.	**Unit:** N/A **Type:** counter **Label:** **Sample value:**	Errors
rqnode_active_master Service master role is active.	**Unit:** N/A **Type:** gauge **Label:** **Sample value:**	Saturation
rqnode_active_backup Service backup role is active.	**Unit:** N/A **Type:** gauge **Label:** **Sample value:**	Saturation
rqnode_read_latency RQ latency; that is, the time duration between when an event is added to Redis and when it's read via XREAD.	**Unit:** **Type:** histogram **Label:** le, healthcheck **Sample value:**	Latency
rqnode_add_latency RQ latency; that is, the time duration between when a message is received and when it's added to the list.	**Unit:** **Type:** histogram **Label:** le, healthcheck **Sample value:**	Latency
rqnode_redis_latency Latency caused by Redis read/write.	**Unit:** **Type:** histogram **Label:** le **Sample value:**	Latency

Alerts

The following alerts are defined for Voice RQ Service.

Alert	Severity	Description	Based on	Threshold
Number of Redis streams is too high	Warning	Too many active sessions. Actions:	rqnode_streams	More than 10000 active streams running.

Alert	Severity	Description	Based on	Threshold
		• Check whether the horizontal pod autoscaler has triggered and if the maximum number of pods has reached. • Check the number of voice, digital, and callback calls in the system.		
Redis disconnected for 5 minutes	Warning	Redis is not available for the pod {{ $labels.pod }}. Actions: • If the alarm is triggered for multiple services, make sure there are no issues with Redis, restart Redis. • If the alarm is triggered only for the pod {{ $labels.pod }}, check to see if there is any issue with the pod.	redis_state	Redis is not available for the pod {{ $labels.pod }} for 5 minutes.
Redis disconnected for 10 minutes	Critical	Redis is not available for the pod {{ $labels.pod }}. Actions: • If the alarm is triggered for multiple services, make sure there are no issues with	redis_state	Redis is not available for the pod {{ $labels.pod }} for 10 minutes.

Alert	Severity	Description	Based on	Threshold
		Redis, and then restart Redis. • If the alarm is triggered only for the pod {{ $labels.pod }}, check to see if there is any issue with the pod.		
Pod failed	Warning	Pod {{ $labels.pod }} failed. Actions: • One of the containers in the pod has entered a Failed state. Check the Kibana logs for the reason.	kube_pod_status_phase	Pod {{ $labels.pod }} is in Failed state.
Pod Unknown state	Warning	Pod {{ $labels.pod }} in Unknown state. Actions: • If the alarm is triggered for multiple services, make sure there are no issues with the Kubernetes cluster. • If the alarm is triggered only for the pod {{ $labels.pod }}, check whether the image is correct and if the container is starting up.	kube_pod_status_phase	Pod {{ $labels.pod }} in Unknown state for 5 minutes.
Pod Pending state	Warning	Pod {{ $labels.pod }} is in the Pending state.	kube_pod_status_phase	Pod {{ $labels.pod }} is in the Pending state for 5 minutes.

Alert	Severity	Description	Based on	Threshold
		Actions: • If the alarm is triggered for multiple services, make sure the Kubernetes nodes where the pod is running are alive in the cluster. • If the alarm is triggered only for the pod {{ $labels.pod }}, check the health of the pod.		
Pod not ready for 10 minutes	Critical	Pod {{ $labels.pod }} in NotReady state. Actions: • If this alarm is triggered, check whether the CPU is available for the pods. • Check whether the port of the pod is running and serving the request.	kube_pod_status_ready	Pod {{ $labels.pod }} in NotReady state for 10 minutes.
Container restored repeatedly	Critical	Container {{ $labels.container }} was repeatedly restarted. Actions: • One of the containers in the pod has entered a failed state. Check the Kibana logs for	kube_pod_container_status_restarts_total	Container {{ $labels.container }} was restarted 5 or more times within 15 minutes.

Alert	Severity	Description	Based on	Threshold
		the reason.		
Pod memory greater than 65%	Warning	High memory usage for pod {{ $labels.pod }}. Actions: • Check whether the horizontal pod autoscaler has triggered and if the maximum number of pods has been reached. • Check Grafana for abnormal load. • Collect the service logs; raise an investigation ticket.	container_memory_working_set_bytes kube_pod_container_resource_requests_memory_bytes	Container {{ $labels.container }} memory usage exceeded 65% for 5 minutes.
Pod memory greater than 80%	Critical	Critical memory usage for pod {{ $labels.pod }}. Actions: • Check whether the horizontal pod autoscaler has triggered and if the maximum number of pods has been reached. • Check Grafana for abnormal load. • Restart the service. • Collect the service logs; raise an investigation ticket.	container_memory_working_set_bytes kube_pod_container_resource_requests_memory_bytes	Container {{ $labels.container }} memory usage exceeded 80% for 5 minutes.

Alert	Severity	Description	Based on	Threshold
Pod CPU greater than 65%	Warning	High CPU load for pod {{ $labels.pod }}. Actions: • Check whether the horizontal pod autoscaler has triggered and if the maximum number of pods has been reached. • Check Grafana for abnormal load. • Collect the service logs; raise an investigation ticket	container_cpu_usage_seconds_total, container_spec_cpu_period	Container {{ $labels.container }} CPU usage exceeded 65% for 5 minutes.
Pod CPU greater than 80%	Critical	Critical CPU load for pod {{ $labels.pod }}. Actions: • Check whether the horizontal pod autoscaler has triggered and if the maximum number of pods has been reached. • Check Grafana for abnormal load. • Restart the service. • Collect the service logs; raise an investigation ticket.	container_cpu_usage_seconds_total, container_spec_cpu_period	Container {{ $labels.container }} CPU usage exceeded 80% for 5 minutes.

Voice SIP Cluster Service metrics and alerts

Contents

Find the metrics Voice SIP Cluster Service exposes and the alerts defined for Voice SIP Cluster Service.

Service	CRD or annotations?	Port	Endpoint/Selector	Metrics update interval
Voice SIP Cluster Service	Supports both CRD and annotations	11300	http://:11300/metrics	30 seconds

See details about:

- Voice SIP Cluster Service metrics
- Voice SIP Cluster Service alerts

Metrics

Voice SIP Cluster Service exposes Genesys-defined, SIP Cluster Service–specific metrics as well as some standard Kafka metrics. You can query Prometheus directly to see all the metrics that the SIP Cluster Service exposes. The following metrics are likely to be particularly useful. Genesys does not commit to maintain other currently available SIP Cluster Service metrics not documented on this page.

Metric and description	Metric details	Indicator of
http_client_request_duration_seconds HTTP client time from request to response, measured in seconds.	**Unit:** seconds **Type:** histogram **Label:** le, target_service_name **Sample value:**	Latency
http_client_response_count Number of received HTTP client responses.	**Unit:** N/A **Type:** counter **Label:** target_service_name **Sample value:**	Traffic
kafka_producer_queue_depth Number of Kafka producer pending events.	**Unit:** N/A **Type:** gauge **Label:** kafka_location **Sample value:**	Traffic
kafka_producer_queue_age_seconds Age of the oldest producer pending event, measured in seconds.	**Unit:** seconds **Type:** gauge **Label:** kafka_location **Sample value:**	Traffic

Metric and description	Metric details	Indicator of
kafka_producer_error_total Number of Kafka producer errors.	**Unit:** N/A **Type:** counter **Label:** kafka_location **Sample value:**	Errors
log_output_bytes_total Total amount of log output in bytes.	**Unit:** bytes **Type:** counter **Label:** level, format, module **Sample value:**	Traffic
sipnode_requests_total Number of processed requests.	**Unit:** N/A **Type:** counter **Label:** tenant, request **Sample value:**	Traffic
sipnode_pending_requests_current Number of pending requests.	**Unit:** N/A **Type:** gauge **Label:** tenant, request **Sample value:**	Traffic
sipnode_requests_queue_size Number of postponed requests.	**Unit:** N/A **Type:** gauge **Label:** **Sample value:**	Saturation
sipnode_sips_request_duration_seconds Duration of the request processed by SIP Cluster Service, measured in seconds.	**Unit:** seconds **Type:** histogram **Label:** le, tenant, request **Sample value:**	Traffic
sipnode_events_total Call events streamed to Redis.	**Unit:** N/A **Type:** counter **Label:** tenant, event **Sample value:**	Traffic
sipnode_ha_writes_total Number of HA writes to Redis.	**Unit:** N/A **Type:** counter **Label:** **Sample value:**	Traffic
sipnode_ha_reads_total Number of HA reads from Redis.	**Unit:** N/A **Type:** counter **Label:** **Sample value:**	Traffic
sipnode_monitoring_events_total Number of monitoring events submitted to Kafka.	**Unit:** N/A **Type:** counter **Label:** tenant **Sample value:**	Traffic
sipnode_redis_restored_calls_total	**Unit:** N/A	Traffic

Metric and description	Metric details	Indicator of
Total number of restored calls from Redis cache.	**Type:** counter **Label:** **Sample value:**	
sipnode_sips_restarts_total Total number of SIP Server restarts.	**Unit:** N/A **Type:** counter **Label:** **Sample value:**	Errors
sipnode_sips_disconnects_total Total number of SIP Cluster Service disconnections from SIP Server.	**Unit:** N/A **Type:** counter **Label:** **Sample value:**	Errors
sipnode_redis_state Current Redis connection state.	**Unit:** N/A **Type:** gauge **Label:** redis_cluster_name **Sample value:**	Errors
sipnode_ors_tlib_latency_msec T-Library latency from Orchestration Service to SIP Cluster, measured in milliseconds.	**Unit:** milliseconds **Type:** histogram **Label:** le, ors **Sample value:**	Latency
sipnode_ors_health_check SIP Cluster Service to Orchestration Service health check.	**Unit:** N/A **Type:** gauge **Label:** **Sample value:**	Traffic
service_version_info Displays the version of Voice SIP Cluster Service that is currently running. In the case of this metric, the labels provide the important information. The metric value is always 1 and does not provide any information.	**Unit:** N/A **Type:** gauge **Label:** version **Sample value:** service_version_info{version="100.0.1000006"} 1	
sipnode_treatment_not_applied Number of unsuccessful treatments.	**Unit:** N/A **Type:** counter **Label:** tenant **Sample value:**	Errors
sipnode_default_routing_total Total number of default routed calls.	**Unit:** N/A **Type:** counter **Label:** tenant **Sample value:**	Traffic
sipnode_envoy_proxy_status Status of the Envoy proxy: -1 – error 0 – disconnected	**Unit:** N/A **Type:** gauge **Label:** **Sample value:** 1	Health

Metric and description	Metric details	Indicator of
1 – connected		
sipnode_config_node_status Status of the config node connection: 0 – disconnected 1 – connected	**Unit:** N/A **Type:** gauge **Label:** **Sample value:** 1	Health
sipnode_health_level Health level of the SIP node (SIP Cluster Service): -1 – fail 0 – starting 1 – degraded 2 – pass	**Unit:** N/A **Type:** gauge **Label:** **Sample value:** 2	Traffic
sipnode_call_state_health_check SIP Cluster Service to Call State Service health check.	**Unit:** N/A **Type:** gauge **Label:** memberId **Sample value:**	Health
sips_hastate Current HA state of SIP Server: 0 – Unknown 1 – backup 2 – primary	**Unit:** N/A **Type:** gauge **Label:** **Sample value:** 2	
sips_calls Current number of calls.	**Unit:** N/A **Type:** gauge **Label:** **Sample value:**	Traffic
sips_call_rate Call rate.	**Unit:** N/A **Type:** gauge **Label:** **Sample value:**	Traffic
sips_cpu_usage_sips SIP Server CPU usage.	**Unit:** N/A **Type:** gauge **Label:** **Sample value:**	Saturation
sips_cpu_usage_main SIP Server main thread CPU usage.	**Unit:** N/A **Type:** gauge **Label:** **Sample value:**	Saturation
sips_cpu_usage_cm CPU usage of the call manager thread.	**Unit:** N/A **Type:** gauge **Label:**	Saturation

Metric and description	Metric details	Indicator of
	Sample value:	
sips_calls_created Total number of created calls.	**Unit:** N/A **Type:** gauge **Label:** **Sample value:**	Traffic
sips_abandoned_calls Total number of abandoned calls.	**Unit:** N/A **Type:** gauge **Label:** **Sample value:**	Errors
sips_rejected_calls Total number of rejected calls.	**Unit:** N/A **Type:** gauge **Label:** **Sample value:**	Errors
sips_dialogs_created Total number of created SIP dialogs.	**Unit:** N/A **Type:** gauge **Label:** **Sample value:**	Traffic
sips_call_recording_failed Number of failed call recording sessions.	**Unit:** N/A **Type:** gauge **Label:** **Sample value:**	Errors
sips_urs_response_1_to_5_sec Number of URS responses from 1 to 5 seconds.	**Unit:** N/A **Type:** gauge **Label:** **Sample value:**	Latency
sips_urs_response_more_5_sec Number of URS responses more than 5 seconds.	**Unit:** N/A **Type:** gauge **Label:** **Sample value:**	Latency
sips_user_data_updates Number of UserData updates.	**Unit:** N/A **Type:** gauge **Label:** **Sample value:**	Traffic
sips_routing_timeouts Number of routing timeouts.	**Unit:** N/A **Type:** gauge **Label:** **Sample value:**	Errors
sips_trequest_rate T-Requests rate.	**Unit:** N/A **Type:** gauge **Label:**	Traffic

Metric and description	Metric details	Indicator of
	Sample value:	
sips_treatment_rate TApplyTreatment requests rate.	**Unit:** N/A **Type:** gauage **Label:** **Sample value:**	Traffic
sips_userdata_rate UserData change rate.	**Unit:** N/A **Type:** gauge **Label:** **Sample value:**	Traffic
sips_sips_memory_usage Memory usage of the SIP Server process.	**Unit:** N/A **Type:** gauge **Label:** **Sample value:**	Saturation
sips_stat_fetch_total Number of successful SIP Server statistic fetches.	**Unit:** N/A **Type:** counter **Label:** **Sample value:**	Other
sips_sip_response_time_ms SIP Server metric of response time, measured in milliseconds.	**Unit:** milliseconds **Type:** histogram **Label:** le **Sample value:**	Latency
sips_trunk_in_service Trunk devices that are in service.	**Unit:** N/A **Type:** gauge **Label:** device_name, tenant **Sample value:**	Traffic
sips_trunk_ncallscreated Number of created calls per trunk.	**Unit:** N/A **Type:** gauge **Label:** device_name, tenant **Sample value:**	Traffic
sips_trunk_noos_detected Number of trunks that are out of service.	**Unit:** N/A **Type:** gauge **Label:** device_name, tenant **Sample value:**	Errors
sips_trunk_n4xx_received Number of received 4xx messages.	**Unit:** N/A **Type:** gauge **Label:** device_name, tenant **Sample value:**	Errors
sips_trunk_n5xx_received Number of received 5xx messages.	**Unit:** N/A **Type:** gauge **Label:** device_name, tenant	Errors

Metric and description	Metric details	Indicator of
	Sample value:	
sips_trunk_n6xx_received Number of received 6xx messages.	**Unit:** N/A **Type:** gauge **Label:** device_name, tenant **Sample value:**	Errors
sips_softswitch_in_service Softswitch devices that are in service.	**Unit:** N/A **Type:** gauge **Label:** device_name, tenant **Sample value:**	Traffic
sips_softswitch_ncallscreated Number of created calls per softswitch device.	**Unit:** N/A **Type:** gauge **Label:** device_name, tenant **Sample value:**	Traffic
sips_softswitch_noos_detected Number of softswitch devices that are out of service.	**Unit:** N/A **Type:** gauge **Label:** device_name, tenant **Sample value:**	Errors
sips_softswitch_n4xx_received Number of received 4xx messages.	**Unit:** N/A **Type:** gauge **Label:** device_name, tenant **Sample value:**	Errors
sips_softswitch_n5xx_received Number of received 5xx messages.	**Unit:** N/A **Type:** gauge **Label:** device_name, tenant **Sample value:**	Errors
sips_softswitch_n6xx_received Number of received 6xx messages.	**Unit:** N/A **Type:** gauge **Label:** device_name, tenant **Sample value:**	Errors
sips_msml_in_service MSML devices that are in service.	**Unit:** N/A **Type:** gauge **Label:** device_name **Sample value:**	Traffic
sips_msml_ncallscreated Number of created calls per MSML device.	**Unit:** N/A **Type:** gauge **Label:** device_name **Sample value:**	Traffic
sips_msml_noos_detected Number of MSML devices that are out of service.	**Unit:** N/A **Type:** gauge **Label:** device_name	Errors

Metric and description	Metric details	Indicator of
	Sample value:	
sips_msml_n4xx_received Number of received 4xx messages.	**Unit:** N/A **Type:** gauge **Label:** device_name **Sample value:**	Errors
sips_msml_n5xx_received Number of received 5xx messages.	**Unit:** N/A **Type:** gauge **Label:** device_name **Sample value:**	Errors
sips_msml_n6xx_received Number of received 6xx messages.	**Unit:** N/A **Type:** gauge **Label:** device_name **Sample value:**	Errors
sips_dp_state Dial Plan Service state: 0 – Out-Of-Service 1 – In-Service	**Unit:** N/A **Type:** gauge **Label:** **Sample value:** 1	Traffic
sips_dp_queue_size Size of the request queue to Dial Plan Service.	**Unit:** N/A **Type:** gauge **Label:** **Sample value:**	Traffic
sips_dp_avg_queue_time Average queue time (msec) of requests to Dial Plan Service.	**Unit:** milliseconds **Type:** gauge **Label:** **Sample value:**	Latency
sips_dp_connections Number of connections to Dial Plan Service per URL.	**Unit:** N/A **Type:** gauge **Label:** **Sample value:**	Traffic
sips_dp_active_connections Number of active connections to Dial Plan Service.	**Unit:** N/A **Type:** gauge **Label:** **Sample value:**	Traffic
sips_dp_req_rate Request rate to Dial plan Service.	**Unit:** N/A **Type:** gauge **Label:** **Sample value:**	Traffic
sips_dp_400_errors Dial Plan Service 400 type of errors.	**Unit:** N/A **Type:** gauge	Errors

Metric and description	Metric details	Indicator of
	Label: **Sample value:**	
sips_dp_404_errors Dial Plan Service 404 type of errors.	**Unit:** N/A **Type:** gauge **Label:** **Sample value:**	Errors
sips_dp_4xx_errors Dial Plan Service 4xx type of errors.	**Unit:** N/A **Type:** gauge **Label:** **Sample value:**	Errors
sips_dp_500_errors Dial Plan Service 500 type of errors.	**Unit:** N/A **Type:** gauge **Label:** **Sample value:**	Errors
sips_dp_501_errors Dial Plan Service 501 type of errors.	**Unit:** N/A **Type:** gauge **Label:** **Sample value:**	Errors
sips_dp_5xx_errors Dial Plan Service 5xx type of errors.	**Unit:** N/A **Type:** gauge **Label:** **Sample value:**	Errors
sips_dp_timeouts Dial Plan Service timeouts.	**Unit:** N/A **Type:** gauge **Label:** **Sample value:**	Errors
sips_dp_average_response_latency Dial Plan Service average response latency.	**Unit:** **Type:** gauge **Label:** **Sample value:**	Latency
sips_sipproxy_in_service SIP Proxy Service state: 0 – Out-Of-Service 1 – In-Service	**Unit:** N/A **Type:** gauge **Label:** **Sample value:** 1	Traffic
trunk_config_synced_count Number of trunks synchronized with SIP Server.	**Unit:** N/A **Type:** gauge **Label:** **Sample value:**	
trunk_config_cached_count	**Unit:** N/A	

Metric and description	Metric details	Indicator of
Number of trunks obtained from the config node.	**Type:** gauge **Label:** **Sample value:**	
trunk_config_cfg_node_error_count Number of failed attempts to read from the config node.	**Unit:** N/A **Type:** counter **Label:** **Sample value:**	
trunk_config_tlib_connection Number of trunks with the T-Library connection.	**Unit:** N/A **Type:** gauge **Label:** **Sample value:**	

Alerts

The following alerts are defined for Voice SIP Cluster Service.

Alert	Severity	Description	Based on	Threshold
Too many Kafka pending events	Critical	Too many Kafka producer pending events for pod {{ $labels.pod }}. Actions: • Ensure there are no issues with Kafka, {{ $labels.pod }} pod's CPU, and network.	kafka_producer_queue_depth	Too many Kafka producer pending events for service {{ $labels.container }} (more than 100 in 5 minutes).
Dial Plan node is overloaded	Critical	Dial Plan node is overloaded as the response latency increases. Actions: • Check that the inbound call rate to SIP Server is not too high. • Check the Dial Plan node CPU and memory usage.	sips_dp_average_response_latency	Dial Plan node is overloaded as the response latency increases (more than 1000).

Alert	Severity	Description	Based on	Threshold
		• Check the network connection between SIP Server and Dial Plan nodes.		
Dial Plan Queue Increase	Critical	Because Dial Plan requests are huge in size or there is a connection issue with the Dial Plan node, the processing queue size increases in size. Actions: • Check SIP Server inbound call rate. • Check the connection between SIP Server and the Dial Plan node.	sips_dp_queue_size	The processing queue size is greater than 10 requests for 1 minute.
SIP Proxy overloaded	Critical	SIP Proxy is overloaded. Actions: • Check SIP Proxy nodes for CPU and memory usage. • If SIP Proxy nodes have acceptable CPU and memory usage, then check for errors or a "hang-up" state which could delay SIP Proxy in forwarding. • Check the SBC side for network	sips_sip_response_time_ms_sum, sips_sip_response_time_ms_count	Response time is greater than 20 milliseconds for 1 minute.

Alert	Severity	Description	Based on	Threshold
		delays.		
SIP Node HealthCheck Fail	Critical	SIP Node health level fails for pod {{ $labels.pod }}. Actions: • Check for failure of dependent services (Redis/ Kafka/SIP Proxy/GVP/Dial Plan). • Check for Envoy proxy failure, then restart the pod.	sipnode_health_level	SIP Node health level fails for pod {{ $labels.pod }} for 5 minutes.
Kafka not available	Critical	Kafka is not available for pod {{ $labels.pod }}. Actions: • If the alarm is triggered for multiple services, ensure there are no issues with Kafka. Restart Kafka. • If the alarm is triggered only for pod {{ $labels.pod }}, check if there is an issue with the pod.	kafka_producer_state	Kafka is not available for pod {{ $labels.pod }} for 5 minutes.
Pod Status Error	Warning	Actions: • Restart the pod. Check if there are any issues with the pod after restart.	kube_pod_status_phase	Pod {{ $labels.pod }} is in Failed, Unknown, or Pending state.
Pod Status NotReady	Warning	Pod {{ $labels.pod }} is in NotReady	kube_pod_status_ready	Pod {{ $labels.pod }} is in NotReady

Alert	Severity	Description	Based on	Threshold
		state. Actions: • Restart the pod. Check if there are any issues with the pod after restart.		state for 5 minutes.
Container Restarted Repeatedly	Critical	Container {{ $labels.container }} was repeatedly restarted. Actions: • Check if the new version of the image was deployed. • Check for issues with the Kubernetes cluster.	kube_pod_container_status_restarts_total	Container {{ $labels.container }} was restarted 5 or more times within 15 minutes.
Ready Pods below 60%	Critical	The number of statefulset {{ $labels.statefulset}} pods in the Ready state has dropped below 60%. Actions: • Check if the new version of the image was deployed. • Check for issues with the Kubernetes cluster.	kube_statefulset_status_replicas_ready, kube_statefulset_status_replicas_current	For the last 5 minutes, fewer than 60% of the currently available statefulset {{ $labels.statefulset}} pods have been in the Ready state.
Pods scaled up greater than 80%	Critical	The current number of replicas is more than 80% of the maximum number of replicas. Actions:	kube_hpa_status_current_replicas, kube_hpa_spec_max_replicas	For 5 consecutive minutes, the number of replicas is more than 80% of the maximum number of replicas.

Alert	Severity	Description	Based on	Threshold
		• Check if max replicas must be modified based on load.		
Pods less than Min Replicas	Critical	The current number of replicas is less than the minimum replicas that should be available. This might be because Kubernetes cannot deploy a new pod or pods are failing to be active/ready. Actions: • If all services have the same issue, then check Kubernetes nodes and Consul health. • If the issue is only with the SIP Cluster service, then check pod logs or the deployment manifest/helm errors.	kube_hpa_status_current_replicas, kube_hpa_spec_min_replicas	For 5 consecutive minutes, the number of replicas is less than the minimum replicas that should be available.
Pod CPU greater than 80%	Critical	Critical CPU load for pod {{ $labels.pod }}. Actions: • Check whether the horizontal pod autoscaler has triggered and if the maximum number of pods has been reached. • Check Grafana for abnormal	container_cpu_usage_seconds_total, container_spec_cpu_period	Container {{ $labels.container }} CPU usage exceeded 80% for 5 minutes.

Alert	Severity	Description	Based on	Threshold
		load. • Collect the service logs for pod {{ $labels.pod }}; raise an investigation ticket.		
Pod CPU greater than 65%	Warning	High CPU load for pod {{ $labels.pod }}. Actions: • Check whether the horizontal pod autoscaler has triggered and if the maximum number of pods has been reached. • Check Grafana for abnormal load. • Collect the service logs for pod {{ $labels.pod }}; raise an investigation ticket.	container_cpu_usage_seconds_total, container_spec_cpu_period	Container {{ $labels.container }} CPU usage exceeded 65% for 5 minutes.
Pod memory greater than 80%	Critical	Critical memory usage for pod {{ $labels.pod }}. Actions: • Check whether the horizontal pod autoscaler has triggered and if the maximum number of pods has been reached. • Check Grafana	container_memory_working_set_bytes, kube_pod_container_resource_requests_memory_bytes	Container {{ $labels.container }} memory usage exceeded 80% for 5 minutes.

Alert	Severity	Description	Based on	Threshold
		for abnormal load. • Restart the service for pod {{ $labels.pod }}.		
Pod memory greater than 65%	Warning	High memory usage for pod {{ $labels.pod }}. Actions: • Check whether the horizontal pod autoscaler has triggered and if the maximum number of pods has been reached. • Check Grafana for abnormal load. • Collect the service logs for pod {{ $labels.pod }}; raise an investigation ticket.	container_memory_working_set_bytes, kube_pod_container_resource_requests_memory_bytes	Container {{ $labels.container }} memory usage exceeded 65% for 5 minutes.
Redis not available	Critical	Redis is not available for pod {{ $labels.pod }}. Actions: • If the alarm is triggered for multiple services, ensure there are no issues with Redis. Restart Redis. • If the alarm is triggered only for pod {{ $labels.pod }},	redis_state	Redis is not available for pod {{ $labels.pod }} for 5 consecutive minutes.

Alert	Severity	Description	Based on	Threshold
		check if there is an issue with the pod.		
Too many Kafka producer errors	Critical	Kafka responds with errors at pod {{ $labels.pod }}. Actions: • For pod {{ $labels.pod }}, ensure there are no issues with Kafka.	kafka_producer_error_total	More than 100 errors for 5 consecutive minutes.
SIP Server main thread consuming more than 65% CPU for 5 mins	Warning	Main thread consumes too much CPU. Actions: • Collect SIP Server Main thread logs; that is, log files without index in the file name (appname_date.log files). Raise an investigation ticket.	sips_cpu_usage_main	Main thread consumes too much CPU (more than 65% for 5 consecutive minutes).
Calls activity drop	Warning	A noticeable reduction in the number of active calls on a specific SIP Server and no new calls are arriving for processing. Actions: • If a problematic SIP Server is primary, do a switchover, and then restart the former primary server. • If a problematic SIP Server is	sips_calls, sips_calls_created	The absolute value of active calls on a specific SIP Server dropped by more than 30 calls in 2 minutes and no new calls are arriving at the SIP Server for processing.

Alert	Severity	Description	Based on	Threshold
		backup, restart the backup server. Collect SIP Server Main thread logs; that is, log files without index in the file name (appname_date.log files). Raise an investigation ticket.		
Dial Plan Node Down	Critical	No Dial Plan nodes are reachable from SIP Server and all connections to Dial Plan nodes are down. Actions: • Check the network connection between SIP Server and the Dial Plan node host. • Check the Dial Plan node CPU and memory usage.	sips_dp_active_connections	All connections to Dial Plan nodes are down.
Dialplan Node problem	Warning	Dial Plan node rejects requests with an error or it doesn't respond to requests and requests are timed out. Actions: • Check the network connection between SIP Server and the Dial Plan host. • Check that Dial Plan nodes are running.	sips_dp_timeouts	During 1 minute, the Dial Plan node rejects more than 5 requests with an error or more than 5 requests time out because the Dial Plan node fails to respond.

Alert	Severity	Description	Based on	Threshold
Routing timeout counter growth	Warning	The trigger detects that routing timeouts are increasing. Actions: • Check the URS_RESPONSE_MORE5SEC stat value. If it's increasing, then investigate why URS doesn't respond to SIP Server in time. • Check SIPS-to-URS network connectivity.	sips_routing_timeouts	The absolute value of NROUTINGTIMEOUTS on a specific SIP Server increased by more than 20 in 2 minutes.

SIP trunk is out of service	Critical	SIP trunk is out of service.
		Actions:
		• For Primary and Secondary trunks:
		  • Troubleshoot SIP Server-to-SBC network connection. Collect network stats and escalate to the Network team to resolve network issues, if necessary.
		  • Troubleshoot the SBC. For Inter-SIP Server trunks: troubleshoot the SIP Server-to-
			sips_trunk_in_service	SIP trunk is out of service for more than 1 minute.

Alert	Severity	Description	Based on	Threshold
		SIP Server network connection. Collect network stats and escalate to the Network team to resolve network issues, if necessary.		
Media service is out of service	Critical	Media service is out of service. Actions: • Troubleshoot the SIP Server-to-Resource Manager (RM) network connection. Collect network stats and escalate to the Network team to resolve network issues, if necessary. • Troubleshoot RM, consider RM restart. • After 5 minutes, redirect traffic to another site.	sips_msml_in_service	Media service is out of service for more than 1 minute.
SIP softswitch is out of service	Critical	Actions: • Troubleshoot the SIP Server-to-SBC network connection. Collect network stats and escalate to the Network team to resolve network issues,	sips_softswitch_in_service	SIP softswitch is out of service.

Alert	Severity	Description	Based on	Threshold
		if necessary. • Troubleshoot the SBC.		
SIP Proxy is out of service	Critical	Actions: • Troubleshoot the SIP Server-to-SIP Proxy nodes network connections. Collect network stats and escalate to the Network team to resolve network issues, if necessary. • Troubleshoot SIP Proxy nodes.	sips_sipproxy_in_service	SIP Proxy is out of service.

Voice SIP Proxy Service metrics and alerts

Contents

Find the metrics Voice SIP Proxy Service exposes and the alerts defined for Voice SIP Proxy Service.

Service	CRD or annotations?	Port	Endpoint/Selector	Metrics update interval
Voice SIP Proxy Service	Supports both CRD and annotations	11400	http://:11400/metrics	30 seconds

See details about:

- Voice SIP Proxy Service metrics
- Voice SIP Proxy Service alerts

Metrics

Voice SIP Proxy Service exposes Genesys-defined, SIP Proxy Service–specific metrics as well as some standard Kafka metrics. You can query Prometheus directly to see all the metrics that the SIP Proxy Service exposes. The following metrics are likely to be particularly useful. Genesys does not commit to maintain other currently available SIP Proxy Service metrics not documented on this page.

Metric and description	Metric details	Indicator of
sipproxy_requests_total Total number of received requests.	**Unit:** N/A **Type:** counter **Label:** method **Sample value:**	Traffic
sipproxy_rejected_requests_total The total number of rejected requests.	**Unit:** N/A **Type:** counter **Label:** **Sample value:**	Errors
sipproxy_requests_processed_self_total The total number of received requests that were processed by SIP Proxy itself.	**Unit:** N/A **Type:** counter **Label:** method **Sample value:**	Traffic
sipproxy_requests_forwarded_total The total number of forwarded requests.	**Unit:** N/A **Type:** counter **Label:** method, request_direction, sip_node_id **Sample value:**	Traffic
sipproxy_requests_sip_node_resolved_total	**Unit:** N/A	Errors

Metric and description	Metric details	Indicator of
Total count of sip-node reselection.	**Type:** counter **Label:** **Sample value:**	
sipproxy_responses_forwarded_total Total count of forwarded responses.	**Unit:** N/A **Type:** counter **Label:** method, sip_node_id, request_direction **Sample value:**	Traffic
sipproxy_response_latency SIP response latency.	**Unit:** **Type:** histogram **Label:** le, sip_node_id, request_direction, target, node_in_cache **Sample value:**	Latency
sipproxy_register_processed_total Total number of REGISTER requests that SIP Proxy received for processing.	**Unit:** N/A **Type:** counter **Label:** **Sample value:**	Traffic
sipproxy_register_rejected_total Total number of REGISTER requests for processing that were rejected.	**Unit:** N/A **Type:** counter **Label:** **Sample value:**	Errors
sipproxy_calls_per_second_count Current calculated calls per second.	**Unit:** N/A **Type:** gauge **Label:** **Sample value:**	Saturation
sipproxy_active_sip_nodes_count Current number of active SIP nodes.	**Unit:** N/A **Type:** gauge **Label:** **Sample value:**	
sipproxy_sip_nodes_count Current number of discovered SIP nodes.	**Unit:** N/A **Type:** gauge **Label:** **Sample value:**	
sipproxy_tenants_count Current count of discovered tenants.	**Unit:** N/A **Type:** gauge **Label:** **Sample value:**	
sipproxy_consul_record_processing_errors_count Current number of errors while processing records got from Consul.	**Unit:** N/A **Type:** counter **Label:** **Sample value:**	
sipproxy_consul_errors_count	**Unit:** N/A	

Metric and description	Metric details	Indicator of
Current number of Consul errors.	**Type:** counter **Label:** **Sample value:**	
sipproxy_sip_node_is_capacity_available Indicates whether SIP node has available capacity or not.	**Unit:** **Type:** gauge **Label:** sip_node_id **Sample value:**	
service_version_info Displays the version of Voice SIP Proxy Service that is currently running. In the case of this metric, the labels provide the important information. The metric value is always 1 and does not provide any information.	**Unit:** N/A **Type:** gauge **Label:** version **Sample value:** service_version_info{version="100.0.1000006"} 1	
sipproxy_health_level Health level of the SIP Proxy node: -1 – fail 0 – starting 1 – degraded 2 – pass	**Unit:** N/A **Type:** gauge **Label:** **Sample value:**	
sipproxy_envoy_proxy_status Status of the Envoy proxy: -1 – error 0 – disconnected 1 – connected	**Unit:** N/A **Type:** gauge **Label:** **Sample value:** 1	
sipproxy_config_node_status Status of the Config node connection: 0 – disconnected 1 – connected	**Unit:** N/A **Type:** gauge **Label:** **Sample value:** 1	
sip_server_transactions_created_total Total number of created server transactions.	**Unit:** N/A **Type:** counter **Label:** **Sample value:**	Traffic
sip_client_transactions_created_total Total number of created client transactions.	**Unit:** N/A **Type:** counter **Label:** **Sample value:**	Traffic
sip_server_transactions_deleted_total Total number of deleted server transactions.	**Unit:** N/A **Type:** counter **Label:** **Sample value:**	Traffic
sip_client_transactions_deleted_total	**Unit:** N/A	Traffic

Metric and description	Metric details	Indicator of
Total number of deleted client transactions.	**Type:** counter **Label:** **Sample value:**	
sip_client_transactions_count Current number of client transactions.	**Unit:** N/A **Type:** gauge **Label:** **Sample value:**	Saturation
sip_server_transactions_count Current number of server transactions.	**Unit:** N/A **Type:** gauge **Label:** **Sample value:**	Saturation
sip_server_transactions_rejected_total Total number of server transactions rejected for internal reasons.	**Unit:** N/A **Type:** counter **Label:** **Sample value:**	Errors
sip_proxy_contexts_count Current number of active SIP Proxy forwarding contexts.	**Unit:** N/A **Type:** gauge **Label:** **Sample value:**	Saturation
sip_received_bytes_total Total traffic received, measured in bytes.	**Unit:** bytes **Type:** counter **Label:** transport **Sample value:**	Traffic
sip_sent_bytes_total Total traffic sent, measured in bytes.	**Unit:** bytes **Type:** counter **Label:** transport **Sample value:**	Traffic
sip_transport_errors_total Total number of transport errors.	**Unit:** N/A **Type:** counter **Label:** transport, address **Sample value:**	Errors
sip_stream_transport_wait_drain_total Total number of requests to wait for drain events on stream transports.	**Unit:** N/A **Type:** counter **Label:** **Sample value:**	
sip_stream_transport_flood_total Total number of flood events on the stream transports.	**Unit:** N/A **Type:** counter **Label:** **Sample value:**	
http_client_request_duration_seconds	**Unit:** seconds	Latency

Metric and description	Metric details	Indicator of
The time duration between the HTTP client request and the response, measured in seconds.	**Type:** histogram **Label:** le, target_service_name **Sample value:**	
http_client_response_count The number of HTTP client responses received.	**Unit:** N/A **Type:** counter **Label:** target_service_name, status **Sample value:**	Traffic

log_output_bytes_total
The total amount of log output, measured in bytes.
	Unit: bytes
	Type: counter
	Label: level, format, module
	Sample value:
	`log_output_bytes_total{level="info",format="txt",module="sipproxy_node@config-manager"} 3175`
	`log_output_bytes_total{level="info",format="txt",module="sipproxy_node@sipproxy-node"} 96`
	`log_output_bytes_total{level="info",format="txt",module="sipproxy_node@sipproxy@sip"} 181`
	`log_output_bytes_total{level="info",format="json",module="sipproxy_node@config-manager"} 4184`
	`log_output_bytes_total{level="info",format="json",module="sipproxy_node@sipproxy-node"} 135`
	`log_output_bytes_total{level="info",format="json",module="sipproxy_node@sipproxy@sip"} 259`	

kafka_consumer_recv_messages_total Number of messages received from Kafka.	**Unit:** **Type:** counter **Label:** **Sample value:**	Traffic
kafka_consumer_error_total Number of Kafka consumer errors.	**Unit:** **Type:** counter **Label:** **Sample value:**	Errors
kafka_consumer_latency Consumer latency is the time difference between when the message is produced and when the message is consumed. That is, the time when the consumer received the message minus the time when the producer produced the message.	**Unit:** **Type:** histogram **Label:** **Sample value:**	Latency
kafka_consumer_rebalance_total Number of Kafka consumer rebalance events.	**Unit:** **Type:** counter **Label:** **Sample value:**	
kafka_consumer_state Current state of the Kafka consumer.	**Unit:** **Type:** gauge **Label:** **Sample value:**	
kafka_producer__messages_total Number of messages received from	**Unit:** **Type:** counter	Traffic

Metric and description	Metric details	Indicator of
Kafka.	**Label:** **Sample value:**	
kafka_producer_queue_depth Number of Kafka producer pending events.	**Unit:** **Type:** gauge **Label:** kafka_location **Sample value:**	Saturation
kafka_producer_queue_age_seconds Age of the oldest producer pending event in seconds.	**Unit:** seconds **Type:** gauge **Label:** kafka_location **Sample value:**	
kafka_producer_error_total Number of Kafka producer errors.	**Unit:** **Type:** counter **Label:** kafka_location **Sample value:**	Errors
kafka_producer_state Current state of the Kafka producer.	**Unit:** **Type:** gauge **Label:** kafka_location **Sample value:**	
kafka_producer_biggest_event_size Biggest event size so far.	**Unit:** **Type:** gauge **Label:** kafka_location, topic **Sample value:** 231	
kafka_max_request_size Exposed config to compare with biggest event size.	**Unit:** **Type:** gauge **Label:** kafka_location **Sample value:** 1000000	
kafka_producer_dropped_event_number Number of dropped events.	**Unit:** **Type:** gauge **Label:** **Sample value:**	

Alerts

The following alerts are defined for Voice SIP Proxy Service.

Alert	Severity	Description	Based on	Threshold
Too many Kafka pending events	Critical	Too many Kafka producer pending events for pod {{ $labels.pod }}.	kafka_producer_queue_depth	Too many Kafka producer pending events for service {{

Alert	Severity	Description	Based on	Threshold
		This alert means there are issues with SIP REGISTER processing on this voice-sipproxy. Actions: • Make sure there are no issues with Kafka or with the {{ $labels.pod }} pod's CPU and network.		$labels.container }} (more than 100 in 5 minutes).
SIP server response time too high	Warning	Actions: • If the alarm is triggered for multiple sipproxy-nodes, make sure there are no issues on {{ $labels.sip_node_id }}. • If the alarm is triggered only for sipproxy-node {{ $labels.pod }}, check to see if there is an issue with the service related to the topic (CPU, memory, or network overload).	sipproxy_response_latency_bucket	SIP response latency for more than 95% of messages forwarded to {{ $labels.sip_node_id }} is more than 1 second for sipproxy-node {{ $labels.pod }}.
Pod status failed	Warning	Actions: • Restart the pod and check to see if there are any issues with the pod after restart.	kube_pod_status_phase	Pod {{ $labels.pod }} is in Failed state.
Pod status Unknown	Warning	Pod {{ $labels.pod }} is in Unknown state.	kube_pod_status_phase	Pod {{ $labels.pod }} is in Unknown state for 5 minutes.

Alert	Severity	Description	Based on	Threshold
		Actions: • Restart the pod and check to see if there are any issues with the pod after restart.		
Pod status Pending	Warning	Pod {{ $labels.pod }} is in Pending state. Actions: • Restart the pod and check to see if there are any issues with the pod after restart.	kube_pod_status_phase	Pod {{ $labels.pod }} is in Pending state for 5 minutes.
Pod status NotReady	Critical	Pod {{ $labels.pod }} is in NotReady state. Actions: • Restart the pod and check to see if there are any issues with the pod after restart.	kube_pod_status_ready	Pod {{ $labels.pod }} is in NotReady state for 5 minutes.
Container restarted repeatedly	Critical	Container {{ $labels.container }} was repeatedly restarted. Actions: • Check to see if a new version of the image was deployed. Also check for issues with the Kubernetes cluster.	kube_pod_container_status_restarts_total	Container {{ $labels.container }} was restarted 5 or more times within 15 minutes.
No sip-nodes available for 2 minutes	Critical	No sip-nodes are available for the pod {{ $labels.pod }}.	sipproxy_active_sip_nodes_count	No sip-nodes are available for the pod {{ $labels.pod }} for 2 minutes.

Alert	Severity	Description	Based on	Threshold
		Actions: • If the alarm is triggered for multiple services, make sure there are no issues with sip-nodes. • If the alarm is triggered only for pod {{ $labels.pod }}, check to see if there is any issues with the pod.		
sip-node capacity limit reached	Warning	The sip-node {{ $labels.sip_node_id }} hit capacity limit on {{ $labels.pod }}. Actions: • If alarm is triggered for multiple services make sure there is no issues with sip-node {{ $labels.sip_node_id }}. • If alarm is triggered only for pod {{ $labels.pod }} check if there is any issue with the pod	sipproxy_sip_node_is_capacity_available	The sip-node {{ $labels.sip_node_id }} hit capacity limit on {{ $labels.pod }} for 3 consecutive minutes.
Pod CPU greater than 80%	Critical	Critical CPU load for pod {{ $labels.pod }}. Actions: • Check whether the horizontal pod autoscaler has triggered	container_cpu_usage_seconds_total container_spec_cpu_period	Container {{ $labels.container }} CPU usage exceeded 80% for 5 minutes.

Alert	Severity	Description	Based on	Threshold
		and the maximum number of pods has been reached. • Check Grafana for abnormal load. • Collect the service logs for pod {{ $labels.pod }} and raise an investigation ticket.		
Pod CPU greater than 65%	Warning	High CPU load for pod {{ $labels.pod }}. Actions: • Check whether the horizontal pod autoscaler has triggered and the maximum number of pods has been reached. • Check Grafana for abnormal load. • Collect the service logs for pod {{ $labels.pod }} and raise an investigation ticket.	container_cpu_usage_seconds_total, container_spec_cpu_period	Container {{ $labels.container }} CPU usage exceeded 65% for 5 minutes.
Pod memory greater than 80%	Critical	Critical memory usage for pod {{ $labels.pod }}. Actions: • Check whether the horizontal pod autoscaler	container_memory_working_set_bytes, kube_pod_container_resource_requests_memory_bytes	Container {{ $labels.container }} memory usage exceeded 80% for 5 minutes

Alert	Severity	Description	Based on	Threshold
		has triggered and the maximum number of pods has been reached. • Check Grafana for abnormal load. • Restart the service for pod {{ $labels.pod }}.		
Pod memory greater than 65%	Warning	Pod {{ $labels.pod }} has high memory usage. Actions: • Check whether the horizontal pod autoscaler has triggered and the maximum number of pods has been reached. • Check Grafana for abnormal load. • Collect the service logs for pod {{ $labels.pod }} and raise an investigation ticket	container_memory_working_set_bytes kube_pod_container_resource_requests_memory_bytes	Container {{ $labels.container }} memory usage exceeded 65% for 5 minutes.
Config node fail	Warning	The request to the config node failed. Action: • Check if there is any problem with pod {{ $labels.pod }} and config node.	http_client_response_count	Requests to the config node fail for 5 consecutive minutes.

Voicemail metrics and alerts

Contents

Find the metrics Voicemail exposes and the alerts defined for Voicemail.

Service	CRD or annotations?	Port	Endpoint/Selector	Metrics update interval
Voicemail	Supports both CRD and annotations	8081	http://:8081/metrics	30 seconds

See details about:

- Voicemail metrics
- Voicemail alerts

Metrics

You can query Prometheus directly to see all the metrics that the Voice Voicemail Service exposes. The following metrics are likely to be particularly useful. Genesys does not commit to maintain other currently available Voicemail Service metrics not documented on this page.

Metric and description	Metric details	Indicator of
voicemail_access_call_rate The voicemail access call rate.	**Unit:** **Type:** gauge **Label:** **Sample value:**	Traffic
voicemail_deposit_call_rate The voicemail deposit call rate.	**Unit:** **Type:** gauge **Label:** **Sample value:**	Traffic
voicemail_gws_request_total The total number of requests sent to GWS.	**Unit:** N/A **Type:** counter **Label:** **Sample value:**	Traffic
voicemail_redis_request_total The total number of requests sent to Redis.	**Unit:** N/A **Type:** counter **Label:** **Sample value:**	Traffic
voicemail_config_request_total The total number of requests sent to the	**Unit:** N/A **Type:** counter	Traffic

Metric and description	Metric details	Indicator of
Config node.	**Label:** **Sample value:**	
voicemail_config_request_failed_total The total number of requests sent to the config node that failed.	**Unit:** N/A **Type:** counter **Label:** response code; for example, `Internal Server Error` or `Service Unavailable` **Sample value:**	Errors
voicemail_redis_request_failed_total The total number of Message Waiting Indicator (MWI) notification requests sent to the Redis stream that failed.	**Unit:** N/A **Type:** counter **Label:** **Sample value:**	Errors
voicemail_gws_request_failed_total The total number of authentication errors when the Voicemail API is accessing via GWS SSO.	**Unit:** N/A **Type:** counter **Label:** response code; for example, `Internal Server Error` or `Service Unavailable` **Sample value:**	Errors
voicemail_service_health_check Status of the service health check: 2 – The service health check is alive 1 – The service health check is degraded 0 – Initializing -1 – The service health check has failed The service health check takes the status of all the dependencies into consideration. The overall Voicemail Service health is updated every two minutes.	**Unit:** N/A **Type:** gauge **Label:** **Sample value:**	Aggregated health check
voicemail_envoy_proxy_status The status of the Envoy proxy: 1 – The Envoy proxy is alive 0 – The Envoy proxy is down	**Unit:** N/A **Type:** gauge **Label:** **Sample value:**	Aggregated health check
voicemail_gws_status The status of GWS: 1 – GWS is alive 0 – GWS is down	**Unit:** N/A **Type:** gauge **Label:** **Sample value:**	Aggregated health check
voicemail_config_node_status Config node status: 1 – the Config node is alive 0 – The Config node is down	**Unit:** N/A **Type:** gauge **Label:** **Sample value:**	Aggregated health check
voicemail_redis_state	**Unit:**	Aggregated health check

Metric and description	Metric details	Indicator of
Indicator of redis_state: 2 – redis_state is ready 1 – redis_state is degraded 0 – redis_state is failed	**Type:** gauge **Label:** **Sample value:**	

Alerts

The following alerts are defined for Voicemail.

Alert	Severity	Description	Based on	Threshold
voicemail_storage_failed_account	Critical	The Storage account is down and, as a result, the service will not be able to fetch the data.	voicemail_storage_failed_account	The Storage account is down.
VoicemailConfigRequestFailure	Critical	Voicemail Service {{$labels.pod}} unable to connect to Config Node.	voicemail_config_request_failed_total	At least 6 requests failed per minute for the past 10 minutes.
VoicemailRedisConnectionDown	Critical	Voicemail Service {{$labels.pod}} unable to connect to the Redis cluster.	voicemail_redis_connection_failure	At least 6 requests failed per minute for the past 10 minutes.
voicemail_node_memory_usage_80	Critical	Critical memory usage for pod {{ $labels.pod }}.	container_memory_working_set_bytes, kube_pod_container_resource_requests_memory_bytes	The Voicemail pod exceeded 80% memory usage for 5 minutes.
voicemail_node_cpu_usage_80	Critical	Critical CPU load for pod {{ $labels.pod }}.	container_cpu_usage_seconds_total, kube_pod_container_resource_requests_cpu_cores	The Voicemail pod exceeded 80% CPU usage for 5 minutes.
PodStatusNotReadyforCritical	Critical	The Voicemail pod is down.	kube_pod_status_ready	The Voicemail pod is down for more than 10 minutes.

Alert	Severity	Description	Based on	Threshold
ContainerRestartedRepeatedly	Critical	The Voicemail pod restarts repeatedly.	kube_pod_container_status_restarts_total	Container {{ $labels.container }} was restarted 5 or more times within 15 minutes.
VoicemailEnvoyHealthFailed	Critical	Voicemail Service {{$labels.pod}} Envoy service is not available.	voicemail_envoy_proxy_status	Voicemail Service {{$labels.pod}} Envoy service is not available for 10 minutes.
VoicemailConfigHealthFailed	Critical	Voicemail Service {{$labels.pod}} GWS service is not available.	voicemail_config_node_status	Voicemail Service {{$labels.pod}} GWS service is not available for 10 minutes.
VoicemailGWSHealthFailed	Critical	Voicemail Service {{$labels.pod}} GWS service is not available.	voicemail_gws_status	Voicemail Service {{$labels.pod}} GWS service is not available for 15 minutes.

Feature support and known limitations

Contents

- Administrator

Understand the differences between the Voice Microservices features and functionality and on-premises voice architecture.

Related documentation:

-
-
-

RSS:

- For private edition

This page provides high-level information about functionality that is not supported or is only partially supported in Voice Microservices architecture in the cloud compared with the legacy on-premises deployments.

Unsupported functionality in Voice Microservice architecture

The following functionality was supported in legacy, on-premises voice architecture, but is not supported in the cloud-based Voice Microservices architecture:

- ACD queues
- Alternate routing for stranded calls
- Inter-Server Call Control (ISCC); that is, multi-site support
- "Nailed-up" connections
- Associating an ACD queue with a routing point
- Asterisk-based voicemail integration
- Call park/retrieve
- Call pickup
- NETANN-based call recording
- Media Server reliability NETANN
- Supervision of Routing Points (IVR supervision is supported instead)
- IP Multimedia Subsystem (IMS) integration
- Instant Messaging
- Presence from switches and endpoints
- Smart **OtherDN** handling
- Trunk capacity control
- Find Me Follow Me functionality
- Hunt Groups feature
- Preview interactions functionality
- E911 emergency gateway
- Remote supervision
- Class of Service (COS) functionality
- "Dummy" media session parameters (SDP)
- `P-Access-Network-Info` private header
- Remote server registration
- Shared Call Appearance (SCA)
- Do Not Disturb (DND) feature
- Opt-out call recording
- SDP codec filtering

- SIP TCP keep-alive mechanism (not supported at the SIP Proxy Service level)
- SIP-based agent login
- SIP phone Device Management functionality

Limitations and constraints in Voice Microservice architecture

The following Voice Microservices features and functionality are supported in the cloud, but with some limitations or constraints:

- A consult call is always supervised.
- The only supported **consult-user-data** model is `inherited`.
- **No Answer Supervision** timeouts are fixed and not configurable.
- **Wrap-up-time** option configuration at the Route Point level is not supported.
- Authentication of outbound predictive calls (falls under the SIP Authentication feature).
- The Reason Code is not set when placing an agent in the Not Ready state on No-Answer Supervision.
- The Customer-on-Hold Privacy feature is disabled.
- Support for Mute/Unmute for two-party calls (disabled in static configuration).