Voice Microservices Private Edition Guide

Observability in Voice Microservices

6/12/2026

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
```