Voice Microservices Private Edition Guide

Voice Registrar Service metrics and alerts

8/28/2025

Contents

- 1 Metrics
- 2 Alerts

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.