Voice Microservices Private Edition Guide

FrontEnd Service metrics and alerts

7/21/2025

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{nam 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.