

# **GENESYS**

This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

Operations

## Table of Contents

Observability	
Monitoring overview and approach	6
Enabling monitoring in GKE Platform	10
System metrics	14
Summary of monitoring support	17
Sample Prometheus queries	22
Handling alerts	26
Grafana configuration	32
Monitoring Dashboards API	41
Logging	
Logging overview and approaches	44
Kubernetes-supported structured logging	53
Sidecar processed logging	56
RWX logging	60
Sample Kibana queries	65
Sample Logs Explorer queries	68

## Contents

- 1 Observability
- 2 Logging

Learn how to use your own logging and monitoring tools to maintain optimal performance of Genesys Multicloud CX private edition services.

### **Related documentation:**

•

### **RSS**:

• For private edition

This guide provides the instructions and details for you to use your own logging and monitoring tools for Genesys Multicloud CX private edition services. It provides information on how the cluster administrators, developers, and other users specify how services and pods are monitored in projects. It covers details on how to deploy application alerts and customize them, as required. The guide also explains the logging approaches that the Genesys Multicloud CX private edition services use.

### Observability

Learn about your monitoring tools, metrics, and handling alerts.

- Monitoring overview and approach
- Enabling monitoring in GKE Platform
- System metrics
- Summary of monitoring support
- Sample Prometheus queries
- Handling alerts
- Grafana configuration
- Monitoring Dashboards API

### Logging

Find out the approaches of logging used by Genesys Multicloud CX services to write log files that contain the important diagnostic information for various issues that may arise.

Logging overview and approaches

- Kubernetes-supported structured logging
- Sidecar processed logging
- RWX (unstructured) logging
- Sample Kibana queries
- Sample Logs Explorer queries

## Monitoring overview and approach

## Contents

- 1 Metrics, alerts, and monitoring approach for services
  - 1.1 Approach
  - 1.2 GKE monitoring
  - 1.3 AKS Monitoring
- 2 Enabling monitoring for your services

Learn about the types of metrics, and the monitoring approach for your Genesys Multicloud CX services that are used in private edition.

### **Related documentation:**

•

### RSS:

• For private edition

### Metrics, alerts, and monitoring approach for services

Services provide the necessary interface to use your own monitoring and logging tools, Prometheusbased metrics, and the endpoint that the Prometheus platform can scrape for alerting and monitoring. The default operators do not scrape user workload or user-defined applications like Genesys services. You must enable Prometheus to scrape user workload. Once enabled, Prometheus scrapes all metrics from endpoints exposed by services.

Some services optionally use Pushgateway to push metrics from jobs that cannot be scraped.

### Approach

In general, the monitoring approach in a private edition deployment is Prometheus-based. Through Prometheus support, the metrics that are generated by Genesys services are made available for visualization (using tools like Grafana). For more details, see the respective sections based on your cloud platform.

### Important

If you are not using Prometheus or an APM tool that supports Prometheus CRDs and PodMonitor or ServiceMonitor objects, then you must build your own solution until Genesys includes the Prometheus annotation support.

There are two types of metrics: system and service.

• System metrics contain data pertaining to cluster performance and status such as CPU usage, memory usage, network I/O pressure, disk usage, and so on. When Prometheus is deployed, by default the system metrics are automatically collected. They provide monitoring of cluster components and ship

with a set of alerts to immediately notify the cluster administrator about any occurring problems

• Service metrics contain data pertaining to Genesys services. For most services, you must enable 'user workload monitoring', and then create ServiceMonitor or PodMonitor per your requirement. However, services that do not use CRD or annotation, run the Pushgateway (Cron job) to collect metrics and push them into the Prometheus gateway.

**GKE** monitoring

GKE monitoring enables you to identify issues related to the performance of your services, and acquire visibility into containers, nodes, and pods within your GKE environment. There are two approaches in GKE for monitoring: Google Cloud operations suite and Prometheus-based approach. For more details, refer to the following sections:

#### **Google Cloud operations suite**

By default, GKE clusters are natively integrated with monitoring. When you create a GKE cluster, monitoring is enabled by default. Cloud Monitoring collects metrics, events, and metadata from Google Cloud. Refer to the following for more details:

- https://cloud.google.com/stackdriver/docs
- https://cloud.google.com/monitoring/docs

#### **Prometheus-based approach**

Prometheus is the monitoring tool that is often used with Kubernetes. Prometheus covers a full stack of Kubernetes cluster components, deployed microservices, alerts, and dashboards. If you configure Cloud Operations for GKE and include Prometheus support, then the metrics that are generated by services using the Prometheus exposition format can be exported from the cluster and made visible as external metrics in Cloud Monitoring. To know more about Prometheus toolkit, refer to the following:

https://prometheus.io/docs/introduction/overview

Click here to learn about deploying Prometheus.

### **AKS Monitoring**

Azure Monitor is the native monitoring service for AKS. You can setup and use Container insights feature in Azure Monitor to monitor the system and workloads.

Refer Genesys monitoring github for more detailed instructions.

## Enabling monitoring for your services

To set up monitoring for the cluster and your private edition services in cloud platforms, find instructions here:

• Enabling monitoring in GKE Platform

## Enabling monitoring in GKE Platform

## Contents

- 1 Setting up monitoring for your private edition services in GKE Platform
  - 1.1 Google Cloud operations suite Cloud Monitoring
  - 1.2 Google Cloud Monitoring API

Learn how to enable monitoring in GKE Platform for the cluster and your private edition services.

### **Related documentation:**

•

### **RSS**:

• For private edition

## Setting up monitoring for your private edition services in GKE Platform

This section describes how to use Cloud Monitoring to monitor your Google Kubernetes Engine (GKE) clusters. It also describes how to enable and authorize use of the Monitoring API

### Google Cloud operations suite - Cloud Monitoring

Google Cloud's operations suite (formerly Stackdriver) enables a centralized capability of receiving events, logs, metrics, and traces from your GKE platform resources.

**Cloud Monitoring** tracks metrics, events, and metadata from GKE platform, uptime probes, and services. **Stackdriver** ingests that data and makes it available via dashboards, charts, and alerts.

For more details, refer to https://cloud.google.com/monitoring/docs.

~	Monitoring	← GKE Dashboard • SE	ND FEEDBACK					
*	Metrics Scope >	= ADD FILTER						
<b></b>	Overview							
Ei	Dashboards	Timeline 0 Alerts 22 Info events	7 Warning events Time selection	is Dec 5 2:03 PM - 3:0	03 PM			
e°.	Services	> 0 alerts						
ıh	Metrics explorer	29 Kubernetes 21					_	
	Alerting							
<u>*</u>	Uptime checks	Dec 5, 2:03 PM	2:10 PM	2:20 PM		2:30 PM	2:40 PM	
(ii)	Groups	Clusters No active alerts 0 clusters wi	h active alerts					
۵	Managed Prometheus	Name Alerts 🖓	Labels		Container restarts	Error logs	CPU utilization 💡	
\$	Settings	gke1 0	Location: us-west1-a Project: g	cpe0001	1	12,160	20.53% of 17.97 C	

### Enable cloud monitoring

Supported values for the --logging flag for the create and update commands.

Source	Value	Logs collected
System	SYSTEM	Metrics from essential system components required for Kubernetes functionality. See a complete list of these Kubernetes metrics.
Workload	WORKLOAD	Enable a fully managed pipeline capable of collecting Prometheus-style metrics exposed by any GKE workload. You must configure which metrics to collect by deploying a PodMonitor custom resource.

### Console UI

To enable cloud monitoring through console UI, follow these steps:

- 1. Navigate to Console UI.
- 2. Select **Clusters** and then select the cluster instance.
- 3. Under **Features > Cloud Monitoring**, click the **Edit** icon.
- 4. Select Enable Cloud Monitoring and then select System and Workflow from the drop-down list.
- 5. Click **SAVE CHANGES**.

This section explains setting up Prometheus on a Kubernetes cluster for monitoring the Kubernetes cluster.

### GCloud CLI

To enable cloud monitoring through GCloud UI, follow these steps:

1. Log on to the existing cluster.

gcloud container clusters get-credentials --zone --project

2. Configure the logs to be sent to Cloud Monitoring by updating a comma-separated list of values to the gcloud container clusters update with --monitoring flag. Here is an example:

```
gcloud container clusters update gkel \
    --zone=us-westl-a \
    --monitoring==SYSTEM,WORKLOAD
```

### Google Cloud Monitoring API

Google Cloud Monitoring API refers to the API that is provided with Google Cloud operations suite to customize your Monitoring solution inside GKE platform.

Stackdriver reads this configuration to prescribe how it processes, manages, and responds to monitored events generated in the cluster.

For more details, refer to Introduction to the Cloud Monitoring API.

## System metrics

## Contents

- 1 Kubernetes and Node metrics
- 2 Kubernetes metrics
- 3 Node metrics

Find useful metrics provided by Kubernetes and other system resources to monitor the status and performance of the cluster and nodes.

### **Related documentation:**

- •
- •

### **RSS:**

• For private edition

## Kubernetes and Node metrics

In addition to the service-defined metrics described in the service-level guides (see links here), standard Kubernetes and other system metrics are obviously important for monitoring the status and performance of your cluster(s), nodes, and services.

- Kubernetes metrics
- Node metrics

### Kubernetes metrics

For full information about all the cluster metrics Kubernetes provides, see the Kubernetes documentation. Genesys recommends that you pay attention to the following cluster-related metrics in particular.

Metric	Prometheus formula	Indicator of
Pod Restarts	increase(kube_pod_container_statu pod=~"\$service.*"})[1m]	us_restarts_total{namespace="\$name
The cgroup's total memory	<pre>sum(container_memory_usage_byt .*", container!=""}) by (pod)</pre>	es{namespace="\$namespace",pod= Memory
The cgroup's CPU usage	<pre>sum (rate (container_cpu_usage_seconds_tot .*", container!="POD"}[1m])) by (pod) * 100</pre>	al{pamespace="\$namespace",pod=~
Bytes transmitted over the network by the container	rate(container_network_transmit_b .*", container!=""}[1m])	ytes_total{namespace="\$namespace
Bytes received over the network by the container	rate(container_network_receive_by .*", container!=""}[1m])	<pre>/tes_total{namespace="\$namespace"</pre>

## Node metrics

Genesys recommends that you pay attention to the following node-related metrics in particular.

Metric	Prometheus formula	Indicator of
Process HEAP All	{SERVICE_NAME}_process_heap_by	y <b>teតត្poda</b> tម\$\$pod",service="\$servic
Process CPU All	<pre>sum(rate({SERVICE_NAME}_proces * 100) by (pod)</pre>	s cpu seconds total{pod=~"\$pod", EPU utilization
Process Memory All: resident memory	{SERVICE_NAME}_process_residen	t_Mmemmomyy_bytes{pod=~"\$pod",serv
Process Memory All: virtual memory	{SERVICE_NAME}_process_virtual_	m <b>@emooyy</b> bytes{pod=~"\$pod",servic

## Summary of monitoring support

Find information about enabling monitoring for your respective services.

### **Related documentation:**

.

### **RSS:**

• For private edition

The service-level guides provide information about enabling monitoring for the respective services. Click the link in the "Included service" column in the summary below to go to the applicable page for service-specific information.

Service	Included service	CRD or annotations?	Port	Endpoint/ Selector	Metrics update interval
		Both — ServiceMonitor and annotations	4004	nexus.nexus.svc metrics	cluster.local/ 15 seconds
CX Contact	CX Contact API Aggregator	ServiceMonitor	9102	/metrics	15 seconds
CX Contact	CX Contact Campaign Manager	ServiceMonitor	3106	/metrics	15 seconds
CX Contact	CX Contact Compliance Manager	ServiceMonitor	3107	/metrics	15 seconds
CX Contact	CX Contact Dial Manager	ServiceMonitor	3109	/metrics	15 seconds
CX Contact	CX Contact Job Scheduler	ServiceMonitor	3108	/metrics	15 seconds
CX Contact	CX Contact List Builder	ServiceMonitor	3104	/metrics	15 seconds
CX Contact	CX Contact List Manager	ServiceMonitor	3105	/metrics	15 seconds

Service	Included service	CRD or annotations?	Port	Endpoint/ Selector	Metrics update interval
Designer	Designer Application Server	ServiceMonitor	8081	See selector details on the DAS metrics and alerts page	10 seconds
Designer	Designer	ServiceMonitor	8888	See selector details on the DES metrics and alerts page	10 seconds
Email Service	Email Service	Both or either, depends on harvester	Default is 4024 (overridden by values)	/iwd-email/v3/ metrics	15 sec recommended, depends on harvester
Genesys Authentication	Authentication Service	Annotations	8081	/prometheus	Real-time
Genesys Authentication	Environment Service	Annotations	8081	/prometheus	Real-time
Genesys Customer Experience Insights	Genesys CX Insights	ServiceMonitor	8180	See selector details on the GCXI metrics and alerts page	15 minutes
Genesys Customer Experience Insights	Reporting and Analytics Aggregates	PodMonitor and PrometheusRule	metrics: 9100, health: 9101	See selector details on the RAA metrics and alerts page	metrics: several seconds, health: up to 3 minutes
Genesys Info Mart	GIM Config Adapter	PodMonitor	9249	See selector details on the GCA metrics and alerts page	30 seconds
Genesys Info Mart	GIM	PodMonitor	8249	See selector details on the GIM metrics and alerts page	30 seconds
Genesys Info Mart	GIM Stream Processor	PodMonitor	9249	See selector details on the GSP metrics and alerts page	30 seconds
Genesys Pulse	Tenant Data Collection Unit (DCU)	PodMonitor	9091	See selector details on the Tenant Data Collection Unit (DCU) metrics and alerts page	30 seconds
Genesys Pulse	Tenant Load Distribution Server (LDS)	PodMonitor	9091	See selector details on the Tenant Load Distribution Server (LDS)	30 seconds

Service	Included service	CRD or annotations?	Port	Endpoint/ Selector	Metrics update interval
				metrics and alerts page	
Genesys Pulse	Pulse Web Service	ServiceMonitor	8090	See selector details on the Pulse metrics and alerts page	30 seconds
Genesys Pulse	Tenant Permissions Service				
Genesys Voice Platform	Voice Platform Configuration Server	Service/Pod Monitoring Settings	Not applicable	See selector details on the Voice Platform Configuration Server metrics and alerts page	
Genesys Voice Platform	Voice Platform Media Control Platform	Service/Pod Monitoring Settings	9116, 8080, 8200	See selector details on the Voice Platform Media Control Platform metrics and alerts page	
Genesys Voice Platform	Voice Platform Service Discovery	Automatic	9090	See selector details on the Voice Platform Service Discovery metrics and alerts page	
Genesys Voice Platform	Voice Platform Reporting Server	ServiceMonitor / PodMonitor	9116	See selector details on the Voice Platform Reporting Server metrics and alerts page	
Genesys Voice Platform	Voice Platform Resource Manager	ServiceMonitor / PodMonitor	9116, 8200	See selector details on the Voice Platform Resource Manager metrics and alerts page	
Interaction Server (IXN)	Interaction Server (IXN)	PodMonitor	13131, <sup>13133,</sup> 13139	option <b>ixnServer.ports</b> - default port 13131 - Endpoint: "/health/ prometheus/ all"	<b>.health</b> Default

Service	Included service	CRD or annotations?	Port	Endpoint/ Selector	Metrics update interval
				option <b>ixnNode.ports.defa</b> - default port 13133 - Endpoint: "/metrics" option <b>ixnVQNode.ports.h</b> - default port 13139 - Endpoint: "/metrics" <b>Note</b> : The above options are references to ports that match endpoints. Use these options to perform the associated query	ealth
Tenant Service	Tenant Service	PodMonitor	15000	/metrics (http://:15000/ metrics)	30 seconds (Applicable for any metric(s) that Tenant Service exposes. The update interval is not a property of the metric; it is a property of the optional PodMonitor that you can create.)
Voice Microservices	Agent State Service	PodMonitor	11000	http://:11000/ metrics	30 seconds
Voice Microservices	Call State Service	Supports both CRD and annotations	11900	http://:11900/ metrics	30 seconds
Voice Microservices	Config Service	Supports both CRD and annotations	9100	http://:9100/ metrics	30 seconds
Voice Microservices	Dial Plan Service	Supports both CRD and annotations	8800	http://:8800/ metrics	30 seconds
Voice Microservices	FrontEnd Service	Supports both CRD and annotations	9101	http://:9101/ metrics	30 seconds
Voice	ORS	Supports both	11200	http://:11200/	30 seconds

Service	Included service	CRD or annotations?	Port	Endpoint/ Selector	Metrics update interval
Microservices		CRD and annotations		metrics	
Voice Microservices	Voice Registrar Service	Supports both CRD and annotations	11500	http://:11500/ metrics	30 seconds
Voice Microservices	Voice RQ Service	Supports both CRD and annotations	12000	http://:12000/ metrics	30 seconds
Voice Microservices	Voice SIP Cluster Service	Supports both CRD and annotations	11300	http://:11300/ metrics	30 seconds
Voice Microservices	Voice SIP Proxy Service	Supports both CRD and annotations	11400	http://:11400/ metrics	30 seconds
Voice Microservices	Voicemail	Supports both CRD and annotations	8081	http://:8081/ metrics	30 seconds
WebRTC Media Service	WebRTC Gateway Service	PodMonitor	10052	/metrics	30s

## Sample Prometheus queries

Sample Prometheus queries to collect metrics.

### **Related documentation:**

.

### **RSS:**

• For private edition

Here are some sample Prometheus queries to collect metrics. The result of each query in Prometheus can either be shown as a graph or viewed as console output.

Query1: kubelet\_http\_requests\_total

### **Output:**

Graph

### Sample Prometheus queries

Prome													
O Enable	query history												
kubel	et_http_request	ts_total									6	Load time: Resolution	: 758ms n: 14s
Execu	te kubelet	_http_reques	sts_tc ¢									rotai time	series; o
Graph	Console												
	- 1h	+	<b>₩</b> Until	*	Res. (s)	🕑 stacked							
500	× -												
400													
											_		
300	k -												
200	< -												
100	k												
				05:20			05.45		08400			08.45	
0	) -	p_requests_total	al{addon_gke_io_node_lo	al_dns_ds_ready="true",b	eta_kubernetes_io_arch="	amd64",beta_kubernetes_i	o_instance_type="e2-standar	d-16*,beta_kubernetes_io_o	s="linux",cloud_google_com_	ike_boot_disk="pd-standar	d",cloud_google_com_gki	e_container_runtime="container	nerd",clou
	<pre>kubelet_http w = kubelet_http kubelet_http kubelet_http</pre>	tp_requests_tota tp_requests_tota	al{addon_gke_io_node_lo al{addon_gke_io_node_lo al(addon_gke_io_node_lo	al_dns_ds_ready="true",b al_dns_ds_ready="true",b al_dns_ds_ready="true",b	eta_kubernetes_io_arch=" eta_kubernetes_io_arch=" eta_kubernetes_io_arch="	amd64*,beta_kubernetes_i amd64*,beta_kubernetes_i amd64*,beta_kubernetes_i	o_instance_type="e2-standar o_instance_type="e2-standar	d-16*,beta_kubernetes_io_or d-16*,beta_kubernetes_io_or d-16*,beta_kubernetes_io_or	s="linux",cloud_google_com_ s="linux",cloud_google_com_ s="linux",cloud_google_com_	ke_boot_disk="pd-standar ke_boot_disk="pd-standar	d",cloud_google_com_gk d",cloud_google_com_gk d",cloud_google_com_gk	e_container_runtime="containe e_container_runtime="containe	nerd",clou nerd",clou
	kubelet_http kubelet_http kubelet_http	tp_requests_tota tp_requests_tota tp_requests_tota	al{addon_gke_io_node_loi al{addon_gke_io_node_loi al{addon_gke_io_node_loi	al_dns_ds_ready="true",b al_dns_ds_ready="true",b al_dns_ds_ready="true",b	eta_kubernetes_io_arch= eta_kubernetes_io_arch=" eta_kubernetes_io_arch="	amd64*,beta_kubernetes_i amd64*,beta_kubernetes_i amd64*,beta_kubernetes_i	o_instance_type="e2-standar o_instance_type="e2-standar o_instance_type="e2-standar	d-16",beta_kubernetes_io_or d-16",beta_kubernetes_io_or d-16",beta_kubernetes_io_or	s="linux",cloud_google_com_ s="linux",cloud_google_com_ s="linux",cloud_google_com_	ke_boot_disk="pd-standar ke_boot_disk="pd-standar jke_boot_disk="pd-standar	d ,cloud_google_com_gkr d",cloud_google_com_gkr d",cloud_google_com_gkr	e_container_runtime= containe e_container_runtime="containe e_container_runtime="containe	herd",clou herd",clou herd",clou
	<pre>kubelet_http://www.example.com/initialized initialized initia</pre>	p_requests_total p_requests_total in requests_total	al{addon_gke_io_node_lo al{addon_gke_io_node_lo al{addon_gke_io_node_lo	al_dns_ds_ready="true",b al_dns_ds_ready="true",b al_dns_ds_ready="true",b	eta_kubernetes_io_arch=" eta_kubernetes_io_arch=" eta_kubernetes_io_arch="	amd64",beta_kubernetes_i amd64",beta_kubernetes_i amd64",beta_kubernetes_i	o_instance_type="e2-standar o_instance_type="e2-standar o_instance_type="e2-standar	d-16",beta_kubernetes_io_o d-16",beta_kubernetes_io_o d-16",beta_kubernetes_io_o	s="linux",cloud_google_com_ s="linux",cloud_google_com_ s="linux",cloud_google_com_	ke_boot_disk="pd-standar ke_boot_disk="pd-standar ke_boot_disk="pd-standar	d",cloud_google_com_gk d",cloud_google_com_gk d" cloud_google_com_gk	e_container_runtime="containe e_container_runtime="containe e_container_runtime="containe	herd",clou herd",clou
	naborot_nig	p requests total	al/addon_gke_io_node_io	"_ano_ao_road) a ao ta	ona_nabonnotoo_no_anon	and a post_nabornated_	- instance_gpo on standar	d-16",beta_kubernetes_io_o	s="linux",cloud_google_com_	ke_boot_disk="pd-standar	d",cloud_google_com_gk d".cloud_google_com_gk	e_container_runtime="containe e_container_runtime="containe	herd",clou herd",clou
	<pre>www.www.www.www.www.www.www.www.www.ww</pre>	p_requests_total	al{addon_gke_io_node_lo	al_dns_ds_ready="true",b al_dns_ds_ready="true",b	eta_kubernetes_io_arch=* eta_kubernetes_io_arch=*	amd64",beta_kubernetes_i amd64",beta_kubernetes_i	o_instance_type="e2-standar o_instance_type="e2-standar	d-16",beta_kubernetes_io_o	s="linux",cloud_google_com_i	No_000(_0iak- pu-atanuai			
Con Prome	kubelet_http://www.subelet_http:	p_requests_total p_requests_total p_requests_total ts_Graph	jaddon_gke_io_node_loa al{addon_gke_io_node_loa al{addon_gke_io_node_loa al{addon_gke_io_node_loa	al_dns_ds_ready="true",br l_dns_ds_ready="true",br al_dns_ds_ready="true",br al_dns_ds_ready="true",br	ata_kubernetes_io_arch=" ta_kubernetes_io_arch=" ata_kubernetes_io_arch=" ta_kubernetes_io_arch="	amd54",beta_kubernetes_i amd64*,beta_kubernetes_i amd64*,beta_kubernetes_i amd64*,beta_kubernetes_i	o_instance_type="d2-standar o_instance_type="d2-standar o_instance_type="d2-standar o_instance_type="d2-standar	d-16*,beta_kubernetes_io_o: J-16*,beta_kubernetes_io_o: d-16*,beta_kubernetes_io_o:	s="linux",cloud_google_com_ =*"linux",cloud_google_com_ s="linux",cloud_google_com_	ke_boot_disk="pd-standar ke_boot_disk="pd-standar	d",cloud_google_com_gk d",cloud_google_com_gk	e_container_runtime≓'contain e_container_runtime="contain	herd",clou
Con Prome © Enable kubele	Kubele_Inty     Kubele_Inty     Kubele_Inty     Kubele_Inty     Kubele_Inty     Kubele_Inty     Kubele_Inty     Kubele_Inty     kubele_Inty	p_requests_total p_requests_total p_requests_total ts_Graph ts_total	sladdon gke io node jo sladdon gke io node jo sladdon gke io node jo Status ≁ Help	al one <u>G</u> ready="true",b   one <u>G</u> ready="true",b    one <u>G</u> ready="true",b    one <u>G</u> ready="true",b	ata (ubernetes jo arch=" tal (ubernetes jo arch=" ata (ubernetes jo arch=" ata (ubernetes jo_arch="	amd64 <sup>+</sup> , beta_kubernetes_i amd64 <sup>+</sup> , beta_kubernetes_i amd64 <sup>+</sup> , beta_kubernetes_i amd64 <sup>+</sup> , beta_kubernetes_i	C_mainto, ype=°2-standar C_mainto, ype=°2-standar C_mainto, ype=°2-standar C_mainto, ype=°2-standar	1-16",beta kubernetes, jo o 1-16",beta kubernetes, jo o 1-16",beta kubernetes, jo o	e=linux'cloud_google_com_ ==linux'cloud_google_com_ e=linux'cloud_google_com_	ko_ocu_alse_podusk="pd-standar ko_boot_disk="pd-standar	d".cloud_google_com_gik d".cloud_google_com_gik	e_container_runtime="containe e_container_runtime="containe Load time: Resolution Totat time:	ero",clou herd",clou :: 758ms h: 14s series: 6
Prome D Enable kubel	Kubele_htt Kubele_htt Kubele_htt Kubele_htt Kubele_htt kubele_htt kubele_htt kubele_http://www.example.com/ kubele_http://wwww.example.com/ kubele_http://wwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwww	p_requests_total p_requests_total p_requests_total ts_Graph ts_total _http_request	aljadon gka jo node jo aljadon gka jo node jo nijadon gka jo node jo Status ▼ Help sts_tc \$	(, and y-"thur β, an	ala, kubernetes (ja jaroh- tak, kubernetes (ja jaroh- tak, kubernetes jo jaroh- tak, kubernetes jo jaroh-	andG4: bala, Jubernetes, andG4: bala, Jubernetes, andG4: bala, Jubernetes, andG4: bala, Jubernetes, andG4: bala, Jubernetes,	o_nsanoo_type=to-sanoda nstanoo_type=to-sanoda nstanoo_type=to-sanoda nstanoo_type=to-sanoda	1-16", beta, kubernetes, jo o 1-16", beta, kubernetes, jo o 1-16", beta, kubernetes, jo o	e= inux cloud google.com_ e= inux'cloud google.com_ e= inux'cloud google.com	ka_boot_disk="pd-standar ka_boot_disk="pd-standar	d" doud google com gik d" doud google com gik	e_container_runtime="containe e_container_runtime="containe Load time: Resolution Total time	:: 758ms h: 14s series: 6
Prome D Enable kubele Execu Graph	theus Alert a query history et_http://exubel.ntt a query history et_http://equest te kubelet_t	p_requests_tota p_requests_tota p_requests_tota ts_Graph ts_total	alleddon jek je node je leddon jek je node je vljaddon jek je node je Status ₹ Help sta_tc \$	Lourd'-plane, 20, μηλ. (Annote: Annote: Annote:: Annote: Annote:: Annote:: Annote:: A	La Jubernetes (jo jardh- la Jubernetes (jo jardh- Ha Jubernetes (jo jardh- Ha Jubernetes (jo jardh-	andidé-bala, Jukametag Jandidé-bala, Jukametag Jandidé-bala, Jukametag Jandidé-bala, Jukametag Jandidé-bala, Jukametag	0_Islando, type-to-sandar Jostando, type-to-sandar Jostando, type-to-sandar Jostando, type-to-sandar Jostando, type-to-sandar	11° (bela yukeentes jo oo	e= inux cloud google.com_ e= inux cloud google.com_ e= inux cloud google.com_	ka bool, dika "po dianda ka bool, dika "po diandar	om given google.com give	e_container_runtime="containe e_container_runtime="containe Load time: Resolution Total time	n 758ms h: 758ms h: 14s series: 6
CON Prome © Enable kubel Execu Graph	theus Alert sole: theus Alert e query history et_http_request the kubelet_htt console for a console	p_requests_total p_requests_total ts_Graph ts_total	alidedon gike jo nodo jo alidedon gike jo nodo jo stadedon gike jo nodo lo Status ▼ Help sts_tc_tc ♥	עריידעיים, אין איין איין איין איין איין איין איין	La Jubarnetes (jo jardh- la Jubarnetes (jo jardh- fa Jubarnetes jo jardh- fa Jubarnetes jo jardh-	andidé-béla, Jukametang Jandidé-béla, Jukametang Jandidé-béla, Jukametang Jandidé-béla, Jukametang Jandidé-béla, Jukametang	0_statino_ype=ros ype=ros photos photoso 0_instance_ype=ros 0_instance_ype=ros 2_statinos_ype=ros instance_ype=ros	110" bela julioentes jo o 110" bela julioentes jo o 110" bela julioentes jo o	e= inix cloub google.com_ e= inix cloub google.com_ e= inix cloub google.com_	ka boot, dika-poteinning ka boot, dika-poteinning ka boot, dika-poteinning	an google com gik an google com gik google com gik	e_container_runtime="containe e_container_runtime="containe Load time: Resolution Total time	:: 758ms :: 758ms :: 14s : series: 6
CON Prome © Enable kubele Execu Graph	todale, the todale, the t	p_request_total p_request_total ts_Graph ts_Cotal ts_total	kleden pke in orde in o kleden pke in onde fo kleden pke in node fo Status ₹ Help sts_tc \$	(, und γ-mutu), (und γ-mutu),	Ja Jubernetes (jo jardh- ta Jubernetes (jo jardh- ta Jubernetes (jo jardh- ta Jubernetes (jo jardh- ta Jubernetes (jo jardh-	andok-bala, Jukamete, andok-bala, Jukamete, andok-bala, Jukamete, andok-bala, Jukamete, andok-bala, Jukamete,	o_neartro ypper 62 ynoar o ynaance, ypper 62 ynaach o _naance_typer 62 ynaach o _naance_typer 62 ylandar	110 (bel, subernates ), o 110 (bela, Jubernates ), o 110 (bela, Jubernates ), o	e= inix cloud, google, com_ e= inix cloud, google, com_ e= inix cloud, google, com_s	ka boot, diik="po-teinning ka boot, diik="po-teinning	dia	e_container_runtime="contain e_container_runtime="contain Load time: Resolution Total time	:: 758ms :: 758ms :: 14s : series: 6
CON Prome Enable kubel Execu Graph	Console     Kubelet, the     Console     Kubelet, the     Console     Kubelet, the	p_requests_tota p_requests_tota p_requests_tota ts_Graph ts_total _http_request tal(addon_gke, _m_kke_roneta _iinux*jong_rur	aleddon gike jo nodo jo aleddon gike jo nodo jo vleddon gike jo nodo jo Status * Help sts_tc * jo_node jocal_dns_ds_ aleg_runter="container" jo_node jocal_dns_ds_ ning_runter="container" ning="false",method- s-west1-a"	<pre>d_drs_d_ready="true"; b d_ord_d_ready="true"; b d_ord_d_ready="true"; b d_ord_d_ready="true"; b d_ord_d_ready="true"; b d_ord_d_ready="true"; b d_d_d_d_d_d_d_d_d_d_d_d_d_d_d_d_d_d_d_</pre>	Ja, Jubernetes, jo, arch- ta, Jubernetes, jo, arch- ta, Jubernetes, jo, arch- ta, Jubernetes, jo, arch- ta, Jubernetes, jo, arch- a, Jubernetes, jo, arch- and Kalundon, Jubernete, Johnson, Johns	anddk-bela, ukkennete anddk-bela, ukkennete anddk-bela, ukkennete anddk-bela, ukkennete bela, ukkennete bela, ukkennetes bela, ukkennetes bela	Stance_type="co-standard protocol type="co-standard protocol type="co-st	16 "bela "kubernetes jo 16 "bela "kubernetes jo o 19 "bela "kubernetes jo o 19 "bela "kubernetes jo 16 "bela "kubernetes jo 16 "bela "kubernetes jo "bela uter den jo o o o o o o o o o o o o o o o o o o	es intx doug google.com se intx: doug google.com es intx: doug google.com oss="linux".cloud.google.com oss="linux".cloud.google.com chine.family="e2".failure.do give.gke1-default-pool-8 o_zone="us-west1-a".topo	ke boot_disk="pd-tender jep boot_disk="pd-tender m_gke_boot_disk="pd- main_beta_lubernetes_ Ga337- oy_kubernetes_lo_regid	ie dead google com gie et dead google com gie dead google com gie lo, region="us- m="us-	e_container_runtime="containe e_container_runtime="containe Load time: Resolution Total time	<ul> <li>2758ms</li> <li>2758ms</li></ul>
CON Prome Enable kubeld Execu Graph	Console     Moment     Momen	p_requests_lota p_requests_lota p_request_lota ts_Graph ts_total http_request tal{addon_gke_ m_kbernets_ m_kbernet	Aligodon jako je nodo je Aligodon jako je nodo je Aligodon jako je nodo lo Status * Help Sta_tc * Jo_node_Jocal_dns_ds_ Join_node_Jocal_dns_ds_ Join_rumtime="contain join_cone"Josewathad Join_cone"Josewathad Join_cone"Josewathad Join_cone"Josewathad Join_cone"Josewathad Join_cone"Josewathad	addy="true",beta_kube datase="true",beta_kube datas	La Jubernetes (jo grah- sta Jubernetes (jo grah- grah) (jo grah) metes (jo grah- gram) (jo grah) (jo grah) (jo grah) (jo grah) (jo grah) (jo grah) (jo grah) (jo grah) (jo grah) (jo grah) (jo grah) (jo grah) (jo grah) (jo grah) (jo grah) (jo grah) (jo grah) (jo g	andok-bela, kubernete, andok-bela, kubernete, andok-bela, kubernete, andok-bela, kubernetes, andok-bela, kubernetes, bela, kubernetes, jo. jn -polo". Joba - kubernetes, jo. jn	Stance_type="e2-standard o_natance_oper-e2-standard o_natance_oper-e2-standard o_natance_oper-e2-standard o_natance_type="e2-standard n_gke_os_distribution="co- rnetes_io_arch="amd64", stance_type="re2-standard n_gke_os_distribution="co- rnetes_io_arch="amd64", thtn",server_type="readon	-16", beta, Judemetes, jo. of 10", beta, Judemetes, jo. 10", beta, Judemetes, jo. 10", beta, Judemetes, jo. 10", beta, Judemetes, jo. Jostametes Hemetes, jo. Jostametes, jo. 10", beta, kudemetes, jo. Jostametes, jo. 10", beta, kudemetes, jo. Jostametes, jo. Jo. Jostametes, jo. Jostametes, jo. Jostametes, jo. Jostametes, jo. Jostametes, jo. Jo. Jostametes, jo. Jo. Jostametes, jo. Jo. Jostametes, jo. Jostametes, jo. Jostametes, jo. Jostametes, jo. Jo. Jo. Jo. Jostametes, jo. Jo. Jo. Jo. Jo. Jo. Jo. Jo. Jostametes, jo. J	os="linux".cloud_google.com_ estimuc".cloud_google.com_ estimuc".cloud_google.com_ com_ com_ com_ com_ com_ com_ com_	ko Jooz, disk="pd-standar ko Jooz, disk="pd-standar majke Joot, disk="pd- main, Jota, Jubernetes, Ga397- emain, Jota, Jubernetes, Ga397- emetes, Jo, region="us-	io_region="us- io_region="us- io_region="us- west1",topology,kuberr	e_container_runtime="containe e_container_runtime="containe Resolution Resolution Total time	<ul> <li>T58ms</li> <li>T58ms</li> <li>T4s</li> <li>series: 6</li> <li>Value</li> <li>378</li> <li>6633</li> </ul>
CON Prome Enable kubelet standar Elemer kubelet standar west1; kubelet standar west1; kubelet standar west1; kubelet standar vest1; kubel	tobelet the second	p_request_lota p_request_lota p_request_lota p_request_lota ts_Graph ts_total 	Aligodon Jako Jonodo Jo Aligodon Jako Jonodo Jo Aligodon Jako Jonodo Jo Aligodon Jako Jonodo Jo Status V Help Sts_tc * Jonodo Jocal, dms. dds Jolen Jocal, dms. dds Jonodo Jocal, dms. dds Jocanod Jocanod Joca	adystrue", beta, kube darwet,	netes_io_arch="am666 bit_bit_bit_bit_bit_bit_bit_bit_bit_bit_	anddk-bela, ukoemete, anddk-bela, ukoemete, andk-bela, u	stance_type="e2-standard oranance_oper-b2-standard oranance_oper-b2-standard oranance_oper-b2-standard oranance_oper-b2-standard n_gke_os_distribution="cor- rnetes_lo_arch="amd64",k" stance_type="e2-standard n_gke_os_distribution="cor- rnetes_lo_arch="amd64",k" stance_type="e2-standard n_gke_os_distribution="cor- rnetes_lo_arch="amd64",k" is_"server_type="readon	-16" beta, kubernetes, jo. of 10" beta, ku	os="linux",cloud_google.com_ estimux",cloud_google.com_ estimux",cloud_google.com_ estimux",cloud_google.com_ estimux",cloud_google.com_ "gke-gke1-default-pool-8 "gke-gke1-default-pool-8 "gke-gke1-default-pool-8 "gke-gke1-default-pool-8 "gke-gke1-default-pool-8 "gke-gke1-default-pool-8 "gke-gke1-default-pool-8 "gke-gke1-default-pool-8 "gke-gke1-default-pool-8 "gke-gke1-default-pool-8 "se-west1-a",topology,kul	ko joot, dika "pd-danda ko joot, dika "pd-danda ko joot, dika "pd-danda main, beta, kubernetas, c0a307. ugka, boot, dikk "pd- main, beta, kubernetas, c0a307. majka, boot, dikk "pd- main, beta, kubernetas, c0a307. erretes, jo, region="us- erretes, jo, region="us-	or doud google com gik of doud google com gik of doud google com gik lo, region="us- in="us- lo_region="us- west1",topology_kuberr lo_region="us- west1",topology_kuberr	e_container_runtime="containe e_container_runtime="containe Load time: Resolution Total time netes_jo_zone="us-	177 (00) 1758ms 1758
Con Prome Denable kubele Standard Elemer kubelet standard west <sup>17</sup> , <sup>1</sup> kubelet standard west <sup>17</sup> , <sup>1</sup> kubelet standard standard west <sup>17</sup> , <sup>1</sup> kubelet standard standard standard west <sup>17</sup> , <sup>1</sup> kubelet standard sta	todale, this     is builder, this     is     is     is     is     is builder, this     i	p_request_lota p_request_lota p_request_stata p_request_stata p_request_stata ts_Craph ts_total ts_total _http_request _http_request _tal(addon_gke, _orm_gke_conta tal(addon_gke, _orm_gke_conta tal(addon_gke, _orm_gke_conta tal(addon_gke, _orm_gke_conta tal(addon_gke, _orm_gke_conta tal(addon_gke, _orm_gke_conta tal(addon_gke, _orm_gke_conta tal(addon_gke, _orm_gke_conta tal(addon_gke, _orm_gke_conta	Aligodon Jako Jonodo Jo Videdon Jako Jonodo Jo Videdon Jako Jonodo Jo Videdon Jako Jonodo Jo Status V Help Sts_tc + Jonodo Jocal Jans da, Jonodo Jocal Jans da, Jocan da Jocal Jans da, Jocan Jocan Jocal Jans da, Jocan Jocan Joc	ady-"true" beta kube darwa bar bar bar bar bar bar bar bar bar ba	metes jo arch="amd66 bit Notemetes jo arch=" ta Notemetes jo arch=" ta Notemetes jo arch=" ta Notemetes jo arch=" arch="arch="amd66 gize_nodepool="default fault-pool-default fault-pool	anddk-bela, uklemieta, anddk-bela, uklemieta, anddk-bela, uklemieta, anddk-bela, uklemieta, anddk-bela, uklemieta, anddk-bela, uklemieta, bela, uklemieta, pool-cloud, google, cor 15e*,job*-uklekt-kube standard-16*,path="me "beta, kubernetes, jo, jn "beta, kubernetes, jo, jn", jn "beta, kubernetes, jo, jn", jn", jn", jn", jn", jn", jn", jn"	stance_type="e2-standard des_od_distribution="co- rnetes_to_arch="amd64", tainer.og_type="e2-standard des_od_distribution="co- rnetes_to_arch="amd64", tainert.ogs",server_type=" stance_type="e2-standard des_od_distribution="co- rnetes_to_arch="amd64", the",server_type="readord stance_type="e2-standard des_od_distribution="co- rnetes_to_arch="amd64", tics",server_type="readord stance_type="e2-standard des_od_distribution="co- rnetes_to_arch="amd64", tics",server_type="readord	-16", beta, kubernetes, jo. of 16", beta, kubernetes, jo. 16", beta, kubernetes, jo. botamane- ter, topology, gke. jo. zone	es "inux" cloud google.com se "inux" cloud google.com es "inux" cloud google.com com com com com com com com	ko boot, disk="pd-atender ko boot, disk="pd-atender ko boot, disk="pd-atender main, beta, kubernetes, c9a397- gy, kubernetes, jo.regio main, beta, kubernetes, c9a397- main, beta, kubernetes, c9a397- mentes, jo.region="us- mentes, jo.region="us- main, beta, kubernetes, c9a397- main, beta, kubernetes, c9a397- bernetes, jo.region="us-	o, region="us- oor", doud google_com_giv oor", doud google_com_giv oor	e_container_runtime="containe e_container_runtime="containe Load time: Resolution Total time netes_jo_zone="us- netes_jo_zone="us-	<ul> <li>26548</li> <li>26548</li> <li>44</li> </ul>
Con Promee Labelet kubelet standard WestT <sup>1</sup> , kubelet standarwestT <sup>1</sup> , kubelet standarwestT <sup>1</sup> , kubelet standarwestT <sup>1</sup> , kubelet standarwestT <sup>1</sup> , kubelet standarwestT <sup>1</sup> , vestT	todale, this induced,	p_requests_tota p_requests_tota p_requests_tota ts_Craph ts_total 	Aligodon Jako Jonodo Jo Vieddon Jako Jonodo Jo Vieddon Jako Jonodo Jo Vieddon Jako Jonodo Jo Status V Help Sts_tc + Jonodo Jocal dns, ds, Jonodo Jocal dns, ds, Jocano Jocano Jocal dns, Jocano Jocano Joc	ady-"true", beta, kube daga-"true", beta, kube rdd, cloud, geogle, com, nstance-"gke-gkel-de GET", node, kubernetes ady-"true", beta, kube rdd cloud, geogle, com, nstance-"gke-gkel-de GET", node, kubernetes ady-"true", beta, kubernetes	metes_io_arch="amd64 https://www.commetes.io.arch=" https://ww	andd-belg, ukemeter, andd-belg, ukemeter, andd-belg, ukemeter, andd-belg, ukemeter, andd-belg, ukemeter, andd-belg, ukemeter, belg, ukemeter,	stance_type="e2-standard des_od_distribution="co- rnetes_to_arch="amd64", tainer.og_bype="e2-standard des_od_distribution="co- rnetes_to_arch="amd64", tainer.ogs",server_type="taindard stance_type="e2-standard des_od_distribution="co- rnetes_to_arch="amd64", the",server_type="readord stance_type="e2-standard des_od_distribution="co- rnetes_to_arch="amd64", trics",server_type="readord stance_type="e2-standard des_od_distribution="co- rnetes_to_arch="amd64", trics",server_type="readord stance_type="e2-standard des_od_distribution="co- rnetes_to_arch="amd64", trics",server_type="readord stance_type="e2-standard", trics",server_type="readord", server_type="taindard", trics",server_type="taindard"	<ul> <li>16" beta, kubernetes, jo. of 16" beta, kubernetes, jo. 16" beta, kubernetes, jo. Jostimere- te", foud, google, com, mac bernetes, jo. hostimere- te", foud, google, com, mac bernetes, jo. hostimere- te", foud, google, com, sol 16" beta, kubernetes, jo. Jostimere- te", foud, jo. Jostimere- "readoniy", fopology, gke 1000 beta, jo. Jostimere- "readoniy", fopology, gke</li> </ul>	es "inux" cloud google.com serinux" cloud google.com estinux" cloud google.com com com estinux" cloud google.com com estinux" cloud google.com estinux" cloud google.com estin	m, gke_boot_disk="pd-dended gia boot_disk="pd-dended gia boot_disk="pd- main_beta_lubernetes_ coa337- gy_kubernetes_lo_region="us- m_gke_boot_disk="pd- main_beta_lubernetes_ coa337- ternetes_lo_region="us- m_gke_boot_disk="pd- main_beta_lubernetes_ c3a37- ternetes_lo_region="us- m_gke_boot_disk="pd- main_beta_lubernetes_ c3a37- ternetes_lo_region="us- m_gke_boot_disk="pd- main_beta_lubernetes_ c3a37- ternetes_lo_region="us- m_gke_boot_disk="pd- main_beta_lubernetes_ c3a37-	d' cloud google com gié d' cloud google com gié de la google com gié lo region="us- n="us- lo_region="us- west1",topology,kuberr lo_region="us- west1",topology,kuberr lo_region="us- west1",topology,kuberr lo_region="us- topregion="us- topregion="us- lo_region="us- lo_region="us- lo_region="us- lo_region="us- lo_region="us- lo_region="us-	e_container_runtime="containe e_container_runtime="containe Resolution Total time netes_lo_zone="us- netes_lo_zone="us-	<ul> <li>258ms</li> <li>258ms</li> <li>14s</li> <li>series: 6</li> <li>6633</li> <li>26548</li> <li>44</li> <li>19910</li> </ul>



### Output

Graph:



Prometheus Alerts Graph Status - Help		
Enable query history		Load time 440mc
sum(irate(sipproxy_requests_processed_self_total{pod=~"voice-sipproxy-0"}[5m])) by (pod,method)		Resolution: 14s
Execute sipproxy_requests_proce +		iotai time series: 2
Graph Console		
✓ Moment		
Element	Value	
{method="OPTIONS",pod="voice-sipproxy-0"}	0.06666666666666666	
{method="REGISTER",pod="voice-sipproxy-0"}	0	
		Remove Graph
Add Graph		

Query 3: node\_cpu\_utilisation:avg1m

### Output

Graph:

Promet	neus Alerts Graph Status + Help	
O Enable c	uery history	Load time: 335ms
:node_c	pu_utilisation:avg1m	Resolution: 28s Total time series: 1
Execute	e :node_cpu_utilisation:avg +	
Graph	Console	
	<ul> <li>2h ➡ ➡ ➡ Until ➡ Res. (s) ➡ Stacked</li> </ul>	
0.05 =		
0.04		
0.04 -	والأستعمام والمرجع فالمتلب والمتلب والمتلية والمتلك المراجع المراجع والمتلاف والمتعالي والمتلا	
0.03 -		
0.02 -		
0.04		
0.01 -		
0 -	0500 0600	
	<pre>/ Inde_cpu_udilation.avg1m()</pre>	
Add Or		Remove Graph
Add Gh		

### Console:

Prometheus Alerts Graph Status - Help		
Enable query history		
:node_cpu_utilisation:avg1m		Load time: 211ms Resolution: 28s Total time series: 1
Execute :node_cpu_utilisation:avg +		
Graph Console		
Moment		
Element	Value	
	Tulde .	
:node_cpu_utilisation:avg1m{}	0.04255425367476107	
:node_cpu_utilisation:avg1m() Add Graph	0.04255425367476107	Remove Graph
:node_cpu_utilisation:avg1m() Add Graph	0.04255425367476107	Remove Graph

## Handling alerts

## Contents

- 1 Introduction
- 2 Alert rules
- 3 Prometheus / Alertmanager
  - 3.1 Alerting Rules
- 4 Customizing Alertmanager configuration for notifications
  - 4.1 Alertmanager configuration for Notifications
- 5 GKE platform
  - 5.1 Google Cloud operations suite Alerting
  - 5.2 Google Cloud Monitoring API Alert Policy

Learn about deploying service alerts.

### **Related documentation:**

٠

### RSS:

• For private edition

### Introduction

Alerts notify you when certain metrics exceed specified thresholds. In some services, alerting is enabled by default; in others, you must enable alerting when you deploy the service. See the respective service guides (listed here) for details about service-specific support for alerting.

### Alert rules

By default, most services define alerts for certain key operational parameters. The alerts are **PrometheusRule** objects that are defined in a YAML file. The metrics collected from the applicable service are evaluated based on the expression specified in the rule. An alert is triggered if the value of the expression is true.

Private edition does not support custom alerts triggered by rules you define yourself. However, some services — for example, Designer — enable you to modify certain parameters in the **values.yaml** file to customize the predefined alerts by modifying the values that trigger the alert. See the respective service-level guides for information about the limited customization each service might support.

### Prometheus / Alertmanager

Enable ServiceMonitor or PodMonitor to scrape metrics from the cluster. To import custom alerts or notification configurations, follow these steps.

### Alerting Rules

This section describes how to create alert rules and import custom rules.

1. Create alert rules. These rules triggers alerts based on the values.

```
apiVersion: "monitoring.coreos.com/v1"
kind: PrometheusRule
metadata:
  name: -alertrules
  labels:
    genesysengage/monitoring: prometheus
    service:
    servicename:
    tenant: --> Ex: shared
spec:
  groups:
   name: -alert
    rules:
    - alert:
      expr:
      for: For ex: 5m
      labels:
        severity: For ex: critical
        service:
        servicename:
      annotations:
summary: ""
```

2. Import the custom rule.

kubectl apply -f -n monitoring

### Customizing Alertmanager configuration for notifications

Alertmanager sends notifications to the notification provider such as email or Webhook (PagerDuty) when an alert is triggered.

### Alertmanager configuration for Notifications

Alertmanager sends notifications to the notification provider (such as email or PagerDuty) when an alert is triggered.

To add notification configuration, edit **alertmanager.yaml** using the following steps:

1. Load the configuration map into a file using the following command.

```
kubectl get configmap prometheus-alertmanager --namespace=monitoring -o yaml >
alertmanager.yaml
```

2. Add the configuration in alertmanager.yaml.

```
global:
    resolve_timeout: 5m
route:
    group_wait: 30s
    group_interval: 5m
    repeat_interval: 12h
    receiver: default
    routes:
    - match:
        alertname: Watchdog
        repeat_interval: 5m
```

```
receiver: watchdog
- match:
    service:
    routes:
    - match:
    receiver:
receivers:
- name: default
- name: watchdog
- name:
```

3. Save the changes in the file and replace the configuration map.

kubectl replace configmaps prometheus-alertmanager --namespace=monitoring -f
alertmanager.yaml

For more details about configuring receivers for alert notification and how the receiver types are created/configured, refer to Configuring alert notifications.

## GKE platform

### Google Cloud operations suite - Alerting

Google Cloud operations suite is backed by Stackdriver which ingests and processes alerts based on predefined policy configuration.

Stackdriver utilizes Google Cloud Monitoring API for management of metric and alert policies within the operation suite.

Here are some key features provided by Google Cloud operation suite:

- Google Cloud API supports over 1,500 Cloud Monitoring metrics.
- Alert policies are configured as a resource object in cloud monitoring API.
- Unlike Alert Manager, policies are defined directly through GCP Cloud Monitoring API via REST or GRCP request. There are no custom resource objects in Kubernetes for alert polices in GKE.
- Defining alert policies allows you to define specific conditions and actions to take in reaction to key metrics and other criteria.
- Notification channels are used to specify where alerts should be sent when an incident occurs. For example:
  - Webhook
  - Email
  - PagerDuty

For more details, refer to the following Google document pages:

Introduction to alerting

- Resource: AlertPolicy
- Resource: NotificationChannel
- Resource: UptimeCheckConfig

### Google Cloud Monitoring API - Alert Policy

### Alert Policy REST API

All API requests to Google Cloud Monitoring API require proper authentication before you query and apply configuration.

See Google authentication for further details.

Here are various functions that are available for creation of custom alert policy.

### projects.alertPolicies.create

POST https://monitoring.googleapis.com/v3/{name}/alertPolicies

### projects.alertPolicies.delete

DELETE https://monitoring.googleapis.com/v3/{name}

### projects.alertPolicies.get

GET https://monitoring.googleapis.com/v3/{name}

### projects.alertPolicies.list

GET https://monitoring.googleapis.com/v3/{name}/alertPolicies

### projects.alertPolicies.patch

PATCH https://monitoring.googleapis.com/v3/{alertPolicy.name}

### Alert Policy example

This example assumes you have created notification channel and uptime check prior to deployment.

### **AlertPolicy - NGINX Ingress Uptime Check**

```
{
   "displayName": "Uptime-Test uptime failure- Ingress",
   "documentation": {
      "content": "Indicates issue with NGINX Ingress availability. Check ingress-nginx-
controller-* in the 'ingress-nginx' namespace",
      "mimeType": "text/markdown"
   },
   "conditions": [
      {
        "displayName": "Failure of uptime check_id uptime-test",
        "conditionThreshold": {
        "aggregations": [
        {
        "displayName": [
        }
    }
}
```

```
"alignmentPeriod": "1200s",
"crossSeriesReducer": "REDUCE_COUNT_FALSE",
                    "groupByFields": [
"resource.label.*"
                    ],
                    "perSeriesAligner": "ALIGN_NEXT_OLDER"
                 }
             ],
"comparison": "COMPARISON_GT",
comparison : "COMPARISON_GI",
    "duration": "60s",
    "filter": "metric.type=\"monitoring.googleapis.com/uptime_check/check_passed\" AND
metric.label.check_id=\" pod " AND resource.type=\"k8s_service\"",
    "thresholdValue": 1,
    "thresholdValue": 1
             "trigger": {
    "count": 1
              }
          }
       }
   ],
    "combiner": "OR",
    "enabled": true,
    "notificationChannels": [
       "projects/gcpe0001/notificationChannels/"
    ]
}
```

## Grafana configuration

## Contents

- 1 Grafana in GKE
  - 1.1 Google Cloud Monitoring in Grafana
  - 1.2 Deploying Prometheus
  - 1.3 Deploying Grafana
  - 1.4 Grafana Plugins
  - 1.5 Creating Grafana Instance
  - 1.6 Connecting Prometheus to custom Grafana
- 2 Grafana dashboards
  - 2.1 Importing custom dashboards
  - 2.2 Creating Grafana dashboards

Learn about how to use Grafana to set up a monitoring solution for your services.

### **Related documentation:**

•

### RSS:

• For private edition

Grafana enables you to query, visualize, alert on, and understand your metrics.

### Important

Although some services have packaged dashboard configuration within their Helm charts, Genesys Multicloud CX private edition does not currently support monitoring dashboards. The following information is provided purely as guidance based on Genesys experimentation, and does not represent a supported configuration.

## Grafana in GKE

### Google Cloud Monitoring in Grafana

For details about cloud monitoring in Grafana, refer to https://grafana.com/docs/grafana/latest/ datasources/google-cloud-monitoring/.

### **Deploying Prometheus**

### Prerequisites

- Create a namespace for deploying Prometheus operator.
- Clone or download source from https://github.com/prometheus-operator/kube-prometheus.
- Make sure you remove the Grafana files. Grafana is deployed using the operator.

### Steps to deploy Prometheus

 Run the setup from the root of downloaded source. This deploys the Prometheus operator and CRDs. kubectl create -f manifests/setup 2. For Prometheus to scrape the cluster (all namespaces), edit prometheus-clusterRole.yaml.

```
metadata:
  labels:
    app.kubernetes.io/component: prometheus
    app.kubernetes.io/name: prometheus
    app.kubernetes.io/part-of: kube-prometheus
    app.kubernetes.io/version: 2.30.0
  name: prometheus-k8s
rules:
- apiGroups:
  resources:
  - nodes/metrics
  verbs:
  - get
- nonResourceURLs:
  - /metrics
  verbs:
  - aet
- apiGroups:
  resources:
  - services
  - pods
  - endpoints
  verbs:
  - get
  - list
  - watch
```

 After the setup is complete, execute the following command: kubectl create - f manifests/

This deploys the following components.

- Prometheus
- Alertmanager
- Prometheus node-exporter
- Prometheus Adapter for Kubernetes Metrics APIs
- kube-state-metrics
- Deploy required components kubectl create -f manifests/

### **Deploying Grafana**

#### Configuring Grafana

The community-powered Grafana is deployed in a new namespace (ex. monitoring) . Follow the instructions to deploy Grafana in GKE.

Installing using Command Line Interface

Download/clone the Grafana operator rom https://github.com/integr8ly/grafana-operator and change the working directory to **grafana-operator-xx**.

Steps to deploy Grafana operator manually

1. Create a new namespace or switch to a namespace (for example: monitoring) where Prometheus is deployed.

\$ kubectl create -f config/crd/bases

2. Create operator roles.

\$ kubectl create -f deploy/roles

3. Modify ClusterRoleBinding (cluster\_role\_binding\_grafana\_operator.yaml). The namespace must be updated with the current namespace where Grafana is deployed (for example: monitoring).

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
   name: grafana-operator
roleRef:
   name: grafana-operator
   kind: ClusterRole
   apiGroup: ""
subjects:
   - kind: ServiceAccount
```

```
name: grafana-operator
```

namespace: monitoring

4. Scan for dashboards in other namespaces you also need the cluster roles.

\$ kubectl create -f deploy/cluster\_roles

To scan dashboards deployed in all namespaces, --scan-all should be added to operator container as argument.

*--scan-all*: watch for dashboards in all namespaces. This requires the operator service account to have cluster wide permissions to get, list, update and watch dashboards.

5. Deploy the operator to that namespace you can use deploy/operator.yaml.

```
containers:
```

- name: grafana-operator

```
image: quay.io/integreatly/grafana-operator:vX.X.X
```

args:

- '--scan-all'

6. Deploy the operator to that namespace. You can use deploy/operator.yaml

\$ kubectl create -f deploy/operator.yaml -n

7. Check the status of the operator pod.

### **Grafana** Plugins

If a data source or dashboard requires a plugin, it can be added in the dashboard itself or it can be added as custom environment variable to the Grafana deployment.

Install plugins using Grafana environment variable

The operator allows you to pass custom environment variable to the Grafana deployment. This means that you can set the **GF\_INSTALL\_PLUGINS** flag, as described.

1. Create and deploy the secret kubectl create -f .yaml -n .

```
apiVersion: v1
kind: Secret
metadata:
name:
type: Opaque
stringData:
GF_INSTALL_PLUGINS:
Add the section to Grafana CR.
deployment:
envFrom:
'''-''' secretRef:
name:
```

### Creating Grafana Instance

1. Modify **Grafana.yaml** with the required values before creating Grafana instance. Update name and add hostname if ingress is enabled.

```
apiVersion: integreatly.org/vlalphal
kind: Grafana
metadata:
name: grafana-app
spec:
client:
preferService: true
```

```
ingress:
enabled: True
hostname: "grafana.gkel-uswestl.gcpe001.gencpe.com"
pathType: Prefix
path: "/"
config:
log:
mode: "console"
level: "error"
log.frontend:
enabled: true
auth:
disable_login_form: False
disable_signout_menu: True
auth.anonymous:
enabled: True
service:
name: "grafana-service"
labels:
app: "grafana"
type: "grafana-service"
dashboardLabelSelector:
- matchExpressions:
- { key: app, operator: In, values: [grafana] }
resources:
Optionally specify container resources
limits:
cpu: 200m
memory: 200Mi
requests:
cpu: 100m
```

- 2. Create a new Grafana instance in the namespace.
  - \$ kubectl create -f deploy/examples/Grafana.yaml -n
- 3. Retrieve the Grafana UI login admin credentials.

```
$ echo "User: admin"
$ echo "Password: $(oc get secret --namespace -o
jsonpath="{.data.GF SECURITY ADMIN PASSWORD}" | base64 --decode)"
```

### Connecting Prometheus to custom Grafana

Deploy Grafana data source kubectl create -f -n . If Grafana instance is deleted and redeployed, you must delete and redeploy Grafana data source as well.

```
apiVersion: integreatly.org/vlalphal
kind: GrafanaDataSource
metadata:
    name: grafana-datasource
    namespace: monitoring
spec:
    datasources:
        - access: proxy
        editable: true
        isDefault: true
        name: Prometheus
        type: prometheus
        url: 'http://prometheus-k8s.monitoring.svc:9090'
    name: grafana-datasource.yaml
```

### Grafana dashboards

### Importing custom dashboards

To import a custom Grafana dashboard from a JSON file within Grafana, click **Import** and then click **Upload Json file** as shown in the following screenshot:

<b>@</b> Q		Import Import dashboard from file or Grafana.com
+	Create	t, Unload JSON file
88	品 Dashboard	
Ø	E Folder	Import via grafana.com
¢		Grafana.com dashboard url or id Lood
ø		Import via panel ison
Ū		
₩.		
?		

### Creating Grafana dashboards

To create Grafana dashboard, use the following template:

```
apiVersion: integreatly.org/vlalphal
kind: GrafanaDashboard
metadata:
  name:
  namespace:
  labels:
    app: grafana --> label should match the dashboardLabelSelector defined in Grafana operator
spec:
  customFolderName: "folder name"
  json:
""
  configMapRef:
    name:
    key:
- - -
apiVersion: v1
kind: ConfigMap
metadata:
  name: voice-sips-dashboard-from-cm
data:
  : |-
```

### Important

Each product has a set of dashboards that come with the service for you to enable/ disable as per your choice.

You can deploy new customized dashboards. You can either deploy them as Grafana dashboard in the namespace or it can be directly loaded on to the Grafana UI. Refer to https://github.com/integr8ly/grafana-operator/tree/master/deploy/examples/dashboards for more details about different ways to deploy a dashboard.

## Monitoring Dashboards API

Learn about Cloud Monitoring API used to create dashboards, update existing dashboards or delete dashboards.

### **Related documentation:**

•

### **RSS**:

• For private edition

The Cloud Monitoring API provides a resource called projects.dashboards which offers a familiar set of methods: create, delete, get, list, and patch.

### Create

POST https://monitoring.googleapis.com/v1/{parent}/dashboards

### Delete

DELETE https://monitoring.googleapis.com/v1/{name}

### GET

GET https://monitoring.googleapis.com/v1/{name}

### List

GET https://monitoring.googleapis.com/v1/{parent}/dashboards

### Patch

PATCH https://monitoring.googleapis.com/v1/{dashboard.name}

Here is an example:

https://content-monitoring.googleapis.com/v1/projects//dashboards

### Errors in Logs Dashboard: Using this example, you can find errors in logs.

{

```
"category": "CUSTOM",
  "displayName": "Errors in Logs Dashboard",
"mosaicLayout": {
    "columns": 12,
    "tiles": [
      {
         "height": 4,
         "widget": {
           "alertChart": {
             "name": "projects//alertPolicies/1502724684856373513"
           }
         },
         "width": 6,
         "xPos": 0,
         "yPos": 0
      },
       {
         "height": 4,
         "widget": {
           "title": "logging/user/Kubernetes-container-error-logs [SUM]",
           "xyChart": {
              "chartOptions": {
               "mode": "COLOR"
             },
             "dataSets": [
                {
                  "minAlignmentPeriod": "60s",
                  "plotType": "STACKED_BAR",
"targetAxis": "Y1",
                  "timeSeriesQuery": {
                    "apiSource": "DEFAULT CLOUD",
                    "timeSeriesFilter": {
                      "aggregation": {
                         "alignmentPeriod": "60s"
                        "crossSeriesReducer": "REDUCE_NONE",
"perSeriesAligner": "ALIGN_RATE"
                      },
                      "filter": "metric.type=\"logging.googleapis.com/user/Kubernetes-container-
error-logs\" resource.type=\"k8s_container\"",
                       "secondaryAggregation": {
                         "alignmentPeriod": "60s"
                         "crossSeriesReducer": "REDUCE SUM",
                         "groupByFields": [
                           "resource.label.\"pod_name\""
                         ],
                         "perSeriesAligner": "ALIGN NONE"
                      }
                    }
                 }
               }
             ],
             "timeshiftDuration": "0s",
             "yAxis": {
               "label": "y1Axis",
"scale": "LINEAR"
             }
           }
        },
         "width": 6,
         "xPos": 6,
         "yPos": 0
      },
```

```
"height": 4,
          "widget": {
             "timeSeriesTable": {
               "dataSets": [
                  {
                    "minAlignmentPeriod": "60s",
                    "tableDisplayOptions": {},
                    "timeSeriesQuery": {
    "timeSeriesFilter": {
                          "aggregation": {
                            "alignmentPeriod": "60s",
"crossSeriesReducer": "REDUCE_NONE",
"perSeriesAligner": "ALIGN_RATE"
                         },
"filter": "metric.type=\"logging.googleapis.com/user/Kubernetes-container-
"filter": "metric.type=\"logging.googleapis.com/user/Kubernetes-container-
error-logs\" resource.type=\"k8s_container\" resource.label.\"namespace_name\"!=\"kube-
system\"",
                          "secondaryAggregation": {
                            "alignmentPeriod": "60s",
                            "crossSeriesReducer": "REDUCE_MAX",
                            "groupByFields": [
                               "resource.label.\"pod_name\""
                            ],
                             "perSeriesAligner": "ALIGN MAX"
                         }
                      }
                   }
                 }
               ]
            },
"title": "logging/user/Kubernetes-container-error-logs (filtered) [99TH PERCENTILE]"
          },
          "width": 6,
          "xPos": 0,
"yPos": 4
      }
    ]
 }
}
```

## Logging overview and approaches

## Contents

- 1 Overview and approaches
- 2 Solution-level logging approaches
  - 2.1 AKS logging approach
- 3 GKE logging
  - 3.1 Enable cloud logging
  - 3.2 Accessing logs
  - 3.3 Cloud Monitoring Console
  - 3.4 GKE Console
  - 3.5 Command-Line

Learn about the structured, unstructured, and Sidecar logging methods that Genesys Multicloud CX private edition services use.

### **Related documentation:**

•

### **RSS:**

• For private edition

### Overview and approaches

Application log files contain the important diagnostic information for various issues that may arise. Support of Genesys services rely on access to these application logs. In Genesys Multicloud CX private edition, the Genesys Multicloud CX services write these log files using different methods and formats. Some services write to a standard out/standard error (stdout/stderr) console while others write directly into an RWX shared storage. This data must be accessible outside of the cluster environment for shipping diagnostic logs for further review.

By default, GKE clusters are natively integrated with Cloud Logging. When you create a GKE cluster, Cloud Logging is enabled by default.

## Solution-level logging approaches

Private edition services use one of the following approaches:

- **Kubernetes-supported structured logging** The services write structured logs. These logs are written in the standard stdout/stderr console and supported by Kubernetes. Fluentd collects these logs from multiple nodes and formats them by appending Kubernetes pod and project metadata. For more information, see Kubernetes-supported structured logging.
- **Sidecar processed logging** The services write their logs in a log file. A sidecar container processes these log files and then writes them to the stdout/stderr console. A log aggregator such as Fluentd collects these logs from stdout/stderr and formats them by appending Kubernetes pod and project metadata. For more information, see Sidecar processed logging.
- **RWX logging (unstructured)** The services write unstructured logs. These unstructured logs can neither be directly processed by a sidecar container nor be collected by Fluentd. These services write their logs in a mounted Persistent Volume Claim (PVC) bound to Persistent Volume (PV) which is backed by an RWX shared storage such as NFS or NAS for ease of access. For more information, see RWX (unstructured) logging.

### Important

A Cluster Administrator must create appropriate PVCs and RWX shared storage path for the services that use the RWX logging method. For more information about creating the log-specific storage, refer to the related Genesys Multicloud CX private edition services.

RWX logging is deprecated. It will be phased out with the use of sidecars to facilitate legacy logging behavior.

### AKS logging approach

In Azure, the Log Analytics workspace feature in the Azure Monitor service collects log data from multiple services and system. You can create a single or multiple workspaces and feed the application logs into them.

For more detailed instructions, refer Genesys logging github.

### GKE logging

Google Cloud's operations suite is backed by Google Stackdriver which controls logging, monitoring, and alerting within Google Cloud Platform. System and user workload logs are captured using Google's own Fluentd DaemonSet called Google-Fluentd that runs on each node in your cluster. The Daemon set parses container logs and pipes them to the stackdriver for processing.

Stackdriver provides built-in log metric capabilities that allows you to monitor specific log events for building dashboards and alert policies.

By default, GKE clusters are natively integrated with cloud logging. When you create a GKE cluster, cloud logging is enabled by default.

You can create a cluster with Logging enabled, or enable Logging in an existing cluster.



### Enable cloud logging

The following table provides the supported values for the --logging flag for the create and update commands.

Source	Value	Logs collected
System	SYSTEM	<ul> <li>Collects logs from:</li> <li>Pods running in namespaces kube-system, istio-system, knative-serving, gke-system, and config-management-system.</li> <li>Key services that are not containerized including docker/containerd runtime, kubelet, kubelet-monitor, node-problem-detector,</li> </ul>

Source	Value	Logs collected
		<ul> <li>and kube-container- runtime-monitor.</li> <li>The node's serial ports output, if the VM instance metadata serial-port-logging- enable is set to <b>true</b>.</li> </ul>
Workload	WORKLOAD	All logs generated by non-system containers running on user nodes.

#### Console UI

To enable cloud logging through console UI, follow these steps:

- 1. Navigate to Console UI using: https://console.cloud.google.com/kubernetes/list/ overview?project=gcpe0001
- 2. Select **Clusters** and then select the cluster name.
- 3. Under Features, select Cloud Logging, and then click Edit.
- 4. Select **Enable Cloud Logging** and then select **System and Workflow** from drop-down.
- 5. Save the changes.

•	Clusters	Security		
54	Workloads	Binary authorization	Disabled	i
-		Shielded GKE nodes	Enabled	1
A	Services & Ingress	Confidential GKE Nodes Beta	Disabled	â
	Applications	Application-layer secrets encryption	Disabled	1
Ħ	Configuration	Workload Identity	Disabled	1
_	,	Google Groups for RBAC	Disabled	*
0	Storage	Legacy authorization	Disable	Edit Cloud Logging
1	Object Browser	Basic authentication	Disable	
A	Migrate to containers	Client certificate	Disable	Cloud Logging is a Google Kubernetes Engine (GKE) addon that collects logs emitted by
				your applications and by GKE infrastructure. Learn more
	Config Management	Metadata		Enable Cloud Logging
		Description	None	
		Labels	None	System and Workloads
		Features		
		Cloud Run for Anthos	Disable	CANCEL SAVE CHANGES
		Cloud Logging	System,	
			View Logs	

### GCloud CLI

To enable cloud logging through GCloud CLI, follow these steps:

1. Log on to the existing GCloud cluster.

gcloud container clusters get-credentials gkel --zone us-westl-a --project gcpe0001

2. Configure the logs to be sent to Cloud Logging by updating a comma-separated list of values to the gcloud container clusters update with --logging flag.

```
gcloud container clusters update gkel \
    --zone=us-west1-a \
    --logging=SYSTEM,WORKLOAD
```

### Accessing logs

Log Explorer

Log explorer is Google's central Logging UI. You can access logs for your Google cloud resources from this console, including GKE, Cloud SQL, VM instances and so on. You can then use logging filters to select the Kubernetes resources, such as cluster, node, namespace, pod, or container logs.

For more details about the console, click here.



≡	Google Cloud Platform	<b>\$•</b> gcpe0001 →	Search products and resources		
E	Operations Logging	Logs Explorer Options 🗸	🖓 REFINE SCOPE 🕐 Reget avout 🗢 Least 1 Hour 🖽 Page Lavout 🗢 Lea	ARN	
Ξ	Logs Explorer	Query Recent (59) Saved (0)	Suggested (1) Stream logs Run query	÷	
52	Logs Dashboard	resource.type="k8s_container" re	ource.labels.project_id="gcpe0001" resource.labels.location="us-west1-a" resource.labels.cluster_name="gke1" 🖉 Editgue	ery	
ılı	Logs-based Metrics	Log fields	X Histogram Q Q	×	
N\$	Logs Router		40		
E	Logs Storage	▲ RESOURCE TYPE		>	
		Kubernetes Container Clear	X (Sep 29, 1:03 PM) 1:30 PM (Sep 29, 2:04 PM)	29. 2:04 PM	
		<ul> <li>SEVERITY</li> </ul>			
		i Info	237 Query results [] Jump to now Actions V Configure	~	
		LOG NAME	SEVERITY TIMESTAMP ADT - SUMMARY		
		stdout	237 > 1 2021-09-29 14:02:36.077 ADT 📪 voice-sipproxy ("date":"2021-09-29717:02:36.07700000002", "level":"debug", "module_		
		▲ PROJECT ID	> i 2021-09-29 14:02:39.519 ADT 📪 voice-sipproxy {"date":"2021-09-29T17:02:39.51800000002", "level":"debug", "module		
		Scheen Clear	x) 2021-09-29 14:02:39.519 ADT 📪 voice-sipproxy ("date":"2021-09-29T17:02:39.5190000012", "level":"debug", "module		
		∧ LOCATION	> i 2021-09-29 14:02:45.672 ADT 📪 voice-sipproxy {"date":"2021-09-29T17:02:45.6720000002", "level":"debug", "module		
		🖌 us-west1-a Clea	x 2021-09-29 14:02:46.065 ADT 📪 voice-sipproxy {"date":"2021-09-29T17:02:46.06500000002", "level":"debug", "module		
		CLUSTER NAME	> i 2021-09-29 14:02:46.065 ADT 📪 voice-sipproxy {"date":"2021-09-29T17:02:46.0650000002Z", "level":"debug", "module_		
			> (i 2021-09-29 14:02:52.595 ADT  voice-sipproxy {"date":"2021-09-29T17:02:52.5950000002", "level":"debug", "module		
		V gkel	> i 2021-09-29 14:02:52.595 ADT  voice-sipproxy {"date":"2021-09-29T17:02:52.5950000002Z", "level":"debug", "module		
<b>C</b> <sup>2</sup>	Poloaso Notos	▲ NAMESPACE NAME	> i 2021-09-29 14:02:55.679 ADT voice-sipproxy {"date":"2021-09-29T17:02:55.6780000002", "level":"debug", "module		
Ē	Nelease NULES	Clear	X > 1 2021-09-29 14:02:59.154 ADT		
4		▲ POD NAME	> 1 2021-09-29 14:02:59.155 ADT (# voice-sipprox) {"date":"2021-09-29717:02:59.1540000022", "level":"debug", "module	- I.	
, CI		-		- 1	

### Cloud Monitoring Console

Cloud Monitoring Console allows you to track metrics of resources within your GCP/GKE environment. This console allows you to access your logs from a particular Cluster, Namespace, Node, and Pod.

	55 5			
<u> </u>	Monitoring	>	Overview	1
PROD	UCTS 🔨		Dashboards	an
	Network services	>	Services	
·I]·	Hybrid Connectivity	>	Metrics explorer Alerting	ain
0	Network Service Tiers		Uptime checks	
4	Network Security	>	Groups	
	Network Intelligence	>	Settings	

≡	Google Cloud Platform	\$• gcpe0001 👻	Namespace details Sep 30 8:04 AM - 9:04 AM SEND FE	EDBACK X
<u>~</u>	Monitoring	← GKE Dashboard - SEND I	₩ FILTER BY RESOURCE	
	Metrics Scope >		gauth	^
~	Overview	Namespaces No active alerts 0 namespa	- System labels	
2011	overview .	Name Alerts 😧	name : "gauth" cluster_name : "gke1" location : "us-west1-a" monitoring_service : "monitoring.googleapis.com/kubernet	es"
	Dashboards	cert-manager 0		
•0	Services	default 0	ALERTS SLOS EVENTS METRICS LOG	s
th	Metrics explorer	elastic-system 0	Severity	
	Alerting	epiphone 0	Logs Default  Finter Finter Finter logs	
	Uptime checks	gauth 0	Scanned up to 9/30/21, 9:02 AM. Scanned 1.1 MB.	
Ŧ			2821-89-30T12:02:59.694426333Z 2021/09/30 12:02:59 [error] 22#22: *42273 directory index of "/var/www/gws/" is forbid	iden,
(ii)	Groups		2021-09-30T12:02:59.694433630Z 2021-09-30T12:02:59+00:00 [-] 403 "GET / HTTP/1.1" 118 [-] [-] [-] [GoogleHC/1.0] 130.	.211.2 🖸
\$	Settings	No active alerts 0 nodes with active	ew log details 21-09-30712:03:05.3432043082 2021/09/30 12:03:05 [error] 22#22: +42274 directory index of "/var/www/yms/" is forbin 1 2021-09-30712:03:05.343263716Z 2021-09-30712:03:05+00:00 [-] 403 "GET / HTTP/1.1" 118 [-] [-] [GoogleHC/1.0] 130	Jden,
		Name Alerts 🚱	2821-89-30T12:03:14.393601259Z 2021-09-30T12:03:14+00:00 [-] 403 "GET / HTTP/1.1" 118 [-] [-] [-] [GoogleHC/1.0] 130	.211.2
		gke-gke1-default-pool-8d 0	12:021-09-30T12:03:14.393604762Z 2021/09/30 12:03:14 [error] 16#16: *42275 directory index of "/var/www/gws/" is forbid	dden,
		gke-gke1-default-pool-8d 0	II 2021-09-30T12:03:14.696021781Z 2021/09/30 12:03:14 [error] 22#22: *42276 directory index of "/var/www/gws/" is forbid	iden,
		ake-ake1-default-pool-8d 0	▶ 🚦 2021-09-30T12:03:14.696101381Z 2021-09-30T12:03:14+00:00 [-] 403 "GET / HTTP/1.1" 118 [-] [-] [-] [GoogleHC/1.0] 130	.211.2
			▶ 🖪 2021-09-30T12:03:20.345176365Z 2021-09-30T12:03:20+00:00 [-] 403 "GET / HTTP/1.1" 118 [-] [-] [-] [GoogleHC/1.0] 130	.211.2
Ē	Release Notes	gke-gke i -default-pool-8d 0	12 2021-09-30T12:03:20.345187858Z 2021/09/30 12:03:20 [error] 22#22: *42277 directory index of "/var/www/gws/" is forbid	iden,
		gke-gke1-default-pool-8d 0	12 2021-09-30T12:03:29.395286461Z 2021/09/30 12:03:29 [error] 22#22: *42278 directory index of "/var/www/gws/" is forbid	iden,
<۱			2021-09-30T12:03:29.395319582Z 2021-09-30T12:03:29+00:00 [-] 403 "GET / HTTP/1.1" 118 [-] [-] [-] [GoogleHC/1.0] 130	.211.2

### GKE Console

GKE web console enables you to access to logs on individual pods actively running within a workload.

There is a filter option available to filter specific events, and a drop-down field to target specific severity of log events.

Logs provide a link to access **Logs Explorer** from a given pod to access the main logs explorer page for enhanced querying capabilities and other features.

### Logging overview and approaches

← Deployment details C REFRESH ✓ EDIT T DELETE III ACTIONS ▼ T KUBECTL ▼	SHOW INFO PANEL
S wehrte-gateway-blue	
Test to gatemay side	
OVERVIEW DETAILS REVISION HISTORY EVENTS LOGS YAML	
Souaritu	
Container logs Showing 23 log entries Default - Filter Fil	
Scanned up to 11/26/21, 3:59 AM. Scanned 44.6 KB.	
10.198	64.1 X-Forwarded-X-Forwarded-F
10 2021-11-26T15:02:43.890335376Z HTTP session is created	
11 2021-11-26T15:02:43.914271711Z >>> HTTP/1.1 200 Added Server: WEBRTCGW-100.0.016.0000 Cache-Control: no-cache Expires: 0 Connection: close Content-Type:	text/plain Content-Length: 11
11 2021-11-26T15:02:43.914401356Z 200 Added	
🕨 🗓 2021-11-26T15:02:44.025108583Z <<< GET /blue/wait HTTP/1.1 Host: webrtc.gke1-uswest1.gcpe002.gencpe.com X-Request-ID: f9e101f52526718c9686c418fb7d674f X	-Real-IP: 10.198.64.1 X-Forwar
11 2021-11-26T15:03:14.013783259Z >>> HTTP/1.1 200 OK Server: WEBRTCGW-100.0.016.0000 Cache-Control: no-cache Expires: 0 Connection: keep-alive Content-Typ	e: text/plain Content-Length:
11 2021-11-26T15:83:14.131020590Z <<< GET /blue/wait HTTP/1.1 Host: webrtc.gke1-uswest1.gcpe002.gencpe.com X-Request-ID: 43dd21f483fe5c499ea716a3b3493933 X	-Real-IP: 10.198.64.1 X-Forwar
11 2021-11-26T15:03:44.122467741Z >>> HTTP/1.1 200 OK Server: WEBRTCGW-100.0.016.0000 Cache-Control: no-cache Expires: 0 Connection: keep-alive Content-Typ	e: text/plain Content-Length:
11 2021-11-26T15:83:44.235827814Z <<< GET /blue/wait HTTP/1.1 Host: webrtc.gke1-uswest1.gcpe002.gencpe.com X-Request-ID: 0c24a3791840f48367405e869417192f X	-Real-IP: 10.198.64.1 X-Forwar
11 2021-11-26T15:04:14.233171256Z >>> HTTP/1.1 200 OK Server: WEBRTCGW-100.0.016.0000 Cache-Control: no-cache Expires: 0 Connection: keep-alive Content-Typ	e: text/plain Content-Length:
11 2021-11-26T15:04:14.359002048Z <<< GET /blue/wait HTTP/1.1 Host: webrtc.gke1-uswest1.gcpe002.gencpe.com X-Request-ID: d2879de3bd8739fcde291f26a71fd9a9 X	-Real-IP: 10.198.64.1 X-Forwar
III 2021-11-26T15:04:44.356324020Z >>> HTTP/1.1 200 OK Server: WEDRTCGW-100.0.016.0000 Cache-Control: no-cache Expires: 0 Connection: keep-alive Content-Typ	e: text/plain Content-Length:
11 2021-11-26T15:04:44.469342809Z <<< GET /blue/wait HTTP/1.1 Host: webrtc.gke1-uswest1.gcpe002.gencpe.com X-Request-ID: 3a710c48c7271ccf18d2110798d345f0 X	-Real-IP: 10.198.64.1 X-Forwar
11 2021-11-26T15:05:14.458473441Z >>> HTTP/1.1 200 OK Server: WEBRTCGW-100.0.016.0000 Cache-Control: no-cache Expires: 0 Connection: keep-alive Content-Typ	e: text/plain Content-Length:
🕨 🔢 2021-11-26T15:85:14.571867859Z <<< GET /blue/wait HTTP/1.1 Host: webrtc.gke1-uswest1.gcpe802.gencpe.com X-Request-ID: 6afb35648e149b88f657e97c651bab07 X	-Real-IP: 10.198.64.1 X-Forwar

### Command-Line

The standard **kubectl** logs commands are supported in GKE. They provide actively running stdout logs from containers.

### Example:

kubebctl logs gvp-mcp-0 -n gvp -c fluentbit | more

mcooke@GEN_3HJD0F3:/mmt/c/Users/mcooke/PAT\$ kubectl logs -n gvp gvp-mcp-0 -c fluentbit   more
[0] MCP.genesys.log.gvp.mcp.gvp.mcp.0.WCP.20211109_104055_624.log: [1637216845.829599642, {"log">>"2021-11-18T06:27:25.051 Int 50019 00640219-10000EA1 140580807077184 prompt_end done"}]
[1] MCP.genesys.log.gvp.mcp.gvp.mcp-0.MCP.20211109_104055_624.log: [1637216845.829604394, {"log"=>"2021-11-18T06:27:25.051 Int 50036 00640219-10000EA1 140588090100032 appl_end "}]
[2] MCP.genesys.log.gvp.mcp.gvp.mcp.0.WCP.20211109_104055_624.log: [1637216845.829685409, {"log"=>"2021-11-18T06:27:25.053 Int 50152 00640219-10000EA1 140588403956032 rtp_stats RTP 38190: RX 0/0 lost 0 dropped 0 dec_err
0 jitter 0, Tx 182/31304 enc_err 0"}]
[3] MCP.genesys.log.gvp.mcp.gvp.mcp.0.WCP.20211109_104055_624.log: [1637216845.829606269, {"log"=>"2021-11-18T06:27:25.053 Int 50001 00640219-10000EA1 140588225337664 incall_end aplend"}]
[4] MCP.genesys.log.gvp.mcp.gvp.mcp-0.MCP.20211109_104055_624.log: [1637216845.829607118, {"log"=>"2021-11-18T06:27:25.222 Int 50052 00640219-10000EA2 140588225337664 incall_initiated 0:0"}]
[5] MCP.genesys.log.gvp.mcp.gvp.mcp.0.WCP.20211109_104055_624.log: [1637216845.829607867, {"log">>"2021-11-18T06:27:25.231 Int 50056 00640219-10000EA2 140588225337664 call_reference 1-38137@10.198.68.76]3F05FE9E-9123-7065
B-837B-944998C6DB65[N/A N/A N/A"}]
[6] MCP.genesys.log.gvp.mcp.gvp.mcp.0.WCP.20211109_104055_624.log: [1637216845.829608641, {"log"=>"2021-11-18T06:27:25.231 Int 50000 00640219-10000EA2 140588225337664 incall_begin sip:msml@127.0.0.1:5070 sip:vleg1@10.198
.68.76:1236[20211118216845593]N/A[N/A[N/A]W/A"}]
[0] MCP.genesys.log.gvp.mcp.gvp.mcp-0.MCP.20211109_104055_624.log: [1637216851.840976950, {"log"=>>"2021-11-18T06:27:30.251 Int 50152 00640219-10000E42 140588402948416 rtp_stats RTP 38192: Rx 0/0 lost 0 dropped 0 dec_err
0 jitter 0, Tx 249/42828 enc_err 0"}]
[1] MCP.genesys.log.gvp.mcp.gvp.mcp.0.WCP.20211109_104055_624.log: [1637216851.840980903, {"log">>"2021-11-18T06:27:30.251 Int 50001 00640219-10000EA2 140588225337664 incall_end usrend"}]
[0] MCP.genesys.log.gvp.mcp.gvp.mcp-0.MCP.20211109_104055_624.log: [1637216855.045891724, {"log"=>"2021-11-18T06:27:35.263 Int 50052 00640219-10000EA3 140588225337664 incall_initiated 0:0"}]
[1] MCP.genesys.log.gvp.mcp.gvp.mcp.0.MCP.20211109_104055_624.log: [1637216855.845896570, {"log"=>"2021-11-18T06:27:35.263 Int 50056 00640219-10000EA3 140588225337664 call_reference 1-21321@10.198.65.79 B298697A-E371-062
3-FA67-0987F67110EE[Environment 1VRAppDefault[N/A"}]
[2] MCP.genesys.log.gvp.mcp.gvp.mcp-0.MCP.20211109_104055_624.log: [1637216855.045897577, {"log"=>"2021-11-18T06:27:35.263 Int 50000 00640219-10000EA3 140588225337664 incall_begin sip:msml-dn@127.0.0.1:5060 sip:sip@10.1
98.65.79:1236[20211118216855594[N/A N/A N/A"}]
[3] MCP.genesys.log.gvp.mcp.gvp.mcp.0.WCP.20211109_104055_624.log: [1637216855.845898481, {"log">>"2021-11-18T06:27:35.263 Int 50130 00640219-10000EA3 140588089092416 appl_begin INIT_URL=file:///samples/ulaw/helloworld
.vxml  DEFAULTS=file:///usr/local/genesys/mcp/config/defaults-ng-dev.vxml  ANI=sip:sipp@10.198.65.79!1236 DNIS=sip:msml-dn@127.0.0.1:5060 PROTOCOLVAME=sip PROTOCOLVERSION=2.0 CALLIDREF=1-21321@10.198.65.79 VXMLI_TYPE=NGI"]

## Kubernetes-supported structured logging

## Contents

• 1 GKE logging

A secondary method of logging required for standard stdout/stderr structured logging.

### **Related documentation:**

### RSS:

• For private edition

This logging method that is required for standard stdout/stderr structured logs that are generated by containers within the Kubernetes environment. Therefore, this method is also called Kubernetes-supported logging. Here, the container is writes stdout/stderr logs to a **-** *var/log/containers* directory.



You will be given the option to choose the external log aggregator to implement the aggregation. Services that use Kubernetes structured logging:

• Genesys Authentication

- Web Services and Applications
- Genesys Engagement Services
- Designer

### Important

Some services (such as Genesys Info Mart) use the Kubernetes logging approach with an exception that the logs are written in an unstructured format.

## GKE logging

Click here for details about GKE logging.

## Sidecar processed logging

## Contents

- 1 What does a sidecar container with a logging agent (like Fluent Bit) require?
  - 1.1 Services support for Sidecar logging

Learn about the Sidecar processing of the structured logging to Stdout/Stderr that is available as an option for private edition services.

### **Related documentation:**

•

### **RSS:**

• For private edition

Some Genesys service containers write logs to log files. This method is similar to that of the structured logging in terms of the the log aggregation. Here, a sidecar to be applied to a sidecar container that is applied to the service. The sidecar container processes this data and sends it to stdout/stderr. Any log aggregator (such as Fluentd) picks up this data and applies the same operations as that of standard structured logs.

Services that can log on to stdout/stderr can be ingested into Elasticsearch by using sidecar container for processing the logs. The service writes the logs to **EmptyDir** and sidecar container collects and processes the output to the **/var/log/pods** directory.

A log aggregator will scrape directory and post log data to the Elasticsearch index.



What does a sidecar container with a logging agent (like Fluent Bit) require?

You require a ConfigMap that contains the configuration to configure Fluent Bit. For more information on configuring Fluent Bit, refer to Fluent Bit Documentation.

Here is a ConfigMap sample with Fluent Bit version 1.8.x:

```
## FluenbtBit Configmap
apiVersion: v1
kind: ConfigMap
metadata:
  name: fluent-bit-config
  labels:
    k8s-app: fluent-bit
data:
  fluent-bit.conf: |
    [SERVICE]
        Flush
                       1
        Log_Level
                       debug
                       off
        Daemon
        Parsers_File parsers.conf
        HTTP Server
                       0n
        HTTP_Listen
HTTP_Port
                       0.0.0.0
                       2020
```

```
@INCLUDE input-kubernetes.conf
 @INCLUDE output-stdout.conf
input-kubernetes.conf: |
  [INPUT]
      Name
                        tail
                       kube.*
     Tag
     Path
     Parser
                       docker
                        /var/log/flb_kube.db
     DB
     Mem_Buf_Limit
                        5MB
      Skip_Long_Lines
                       0n
     Refresh_Interval 10
output-stdout.conf: |
  [OUTPUT]
      Name
                      stdout
     Match
```

You also require a pod that has a sidecar container running Fluentd. The pod mounts a volume where Fluentd can pick up its configuration data. Here is an example:

### Services support for Sidecar logging

These services have the option use the Sidecar processed logging approach:

- Genesys Customer Experience Insights GCXI
- Genesys Voice Platform GVP
- Voice Microservice
- Voice Tenant Service
- Web Based Real-Time Reporting (Pulse)
- WebRTC Media Service
- Gplus WFM

## RWX logging

## Contents

- 1 RWX logging
- 2 Storage Prerequisites
  - 2.1 Direct NFS Persistent Storage
  - 2.2 Azure-Files Persistent Storage for ARO (NFS Backed)

Learn about the legacy logging method of writing logs to an RWX storage such as NFS or NAS server.

### **Related documentation:**

•

### **RSS**:

• For private edition

### RWX logging

### Important

RWX logging is deprecated. It will be phased out with the use of sidecars to facilitate legacy logging behavior.

Some Genesys Multicloud CX services neither write structured logs in Kubernetes format nor do they write to the stdout/stderr console. These services use RWX logging, which is the legacy logging method of writing logs to an RWX storage such as NFS or NAS server.

Legacy Genesys Multicloud CX applications are not structured to be supported by logging capabilities offered in Kubernetes, nor do they write to sufficient detail in stdout/stderr. To accommodate this type of logging behavior, deployments must be provisioned to support mounting PVC/PV to NFS storage for the application to write its logs. Each Service mounts to its own PV which is backed by an external NFS share. After the logs are written to NFS share, the application controls the size and retention of the file and files can be accessed externally from NFS share directly to package and provide to care.



The method of logging unstructured logs is not suitable for kubernetes-supported logging aggregators such as Elasticsearch.

The sample procedures provided in the following section, help in setting up the RWX storage of your choice.

Services that use the RWX logging approach:

- WebRTC
- GVP
- GCXI
- Voice Microservices
- Genesys Pulse
- Interaction Server
- Tenant Services

### Storage Prerequisites

### **Direct NFS Persistent Storage**

With Direct NFS approach, shares are mounted using NFS IP/FQDN and share path is mounted using NFS-subdir-external-provisioner.

For more details about this provisioner, refer to NFS Subdir External Provisioner.

**Prerequisite:** You must have a dedicated NFS server to create NFS persistent storage.

Create StorageClass for NFS Retained Storage

Here is a sample configuration to create StorageClass for NFS persistent storage. The following configuration is suitable for a bare metal server.

#### bare-metal-sc.yaml

```
provisioner: cluster.local/nfs-vce-c00ds-voll-nfs-subdir-external-provisioner
mountOptions:
    nfsvers=3
    uid=500
    gid=500
parameters:
    archiveOnDelete: 'false'
    volumeBindingMode: Immediate metadata
    name:
    kind: StorageClass
    reclaimPolicy: Retain
    allowVolumeExpansion: true
    apiVersion: storage.k8s.io/v1
    oc apply -f bare-metal-sc.yaml
```

Create PVC to dynamically create and bind to PV

#### create-pvc.yaml

apiVersion: v1
kind: PersistentVolumeClaim
metadata:
 name: -pvc
 namespace:
 spec:
 accessModes:
 - ReadWriteOnce
 resources:
 requests:
 storage: 10Gi
 storageClassName:
 volumeMode: Filesystem

### Azure-Files Persistent Storage for ARO (NFS Backed)

For ARO type deployments you can map NFS directly. Therefore, you can create NFS share within

Azure using Azure-files. You need to create storage class of type Azure-Files:

- "recalimPolicy" set to "Retain"
- "parameters" set based on your specific Azure deployment

For more details, refer to:

- How to create an NFS share
- · Dynamically create and use a persistent volume with Azure Files in AKS

Create Storage Class for retained Azure-File NFS storage

#### azure-file-retain-sc.yaml

```
allowVolumeExpansion: true
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
    name: azure-files-retain
annotations:
    description: azure-files-retain
provisioner: kubernetes.io/azure-file
parameters:
    location: westus2
    skuName: Standard_LRS
reclaimPolicy: Retain
volumeBindingMode: Immediate
```

oc apply -f azure-file-retain-sc.yaml

Create PVC to dynamically create and bind to PV

#### create-pvc.yaml

apiVersion: v1
kind: PersistentVolumeClaim
metadata:
name: -pvc
namespace:
spec:
accessModes:
- ReadWriteOnce
resources:
requests:
storage: 10Gi
storageClassName:
volumeMode: Filesystem

oc apply -f create-pvc.yaml

## Sample Kibana queries

Sample Kibana queries to find logs

### **Related documentation:**

.

### **RSS:**

• For private edition

Here are some sample queries for you to understand what information could be searched for in Kibana. The search is specific to the values in the query. It returns the log messages that matches the query.

### Query 1: To return podname , namespace , and container name

kubernetes.pod\_name:"t100-0" AND kubernetes.namespace\_name:"voice" AND kubernetes.container name.raw:"tenant"

### **Output:**



### Query 2: Any string and logs listed with the string

#### "Voice-sip"

#### Output



## Query 3: Any combination with service, service name, instance , name , version, any value available in the logs

"service=ixn" AND "servicename=ixn-vqnode"

#### **Output:**



#### **Query 4: CallUUID is consistent across interactions**

kubernetes.pod\_name."voice-sip-0" AND kubernetes.namespace\_name."voice" AND kubernetes.container\_name.raw."voice-sip" AND "CallUUID.00E5IM3D048KVE8LFG1862LAES0001SP"

Output:																		
25 hits											New	Save Op	en Share	Inspect	C Auto-refresh	<	🖸 Last 1w	>
>_ kubernetes.pod_name:"voice	-sip-0" /	AND ku	ibernet	es.namespace_name	e:"voice" A	ND kubernetes.cont	ainer_name	e.raw:"voice-sip" Al	ND "CallUUID: 00E5II	//3D048KVE8LFG1	862LAES000	1SP"			Option	ns	C Refrest	
Add a filter +																		
*		0					Septerr	iber 22nd 2021, 10:3	6:41.758 - September :	9th 2021, 10:36:41.	758 — Auto	~						
Selected fields																		
t kubernetes.container_name			25															
t message		t	15 -															
Available fields	۰	Col	10 -															
⊘ @timestamp			5															
t _id			0	2021-09-22 21:00		2021-09-23 21:00		2021-09-24 21:00	2021-09-2 ©timestan	21:00 Ip per 3 hours	2021-09-26	21:00	2021	-09-27 21:00	2021	1-09-28 2	1:00	
t _index																		
# _score			Time			kubernetes.conta	iner_name	20050300										
t_type			TIME	- <del>-</del>				message										
t docker.container_id		,	Septe	ember 29th 2021, 10:2	26:25.726	voice-sip		{"date":"2021-0 4c65-8d40-1f75a ES0001SP","Othe	89-29T13:26:25.726000 a2e080dd","ConnID":" arDN":"+16504662188",	1000Z","level":"de 175f032490c78799", "TenantID":"9350e	bug","module " <mark>CallUUID</mark> ":" 2fc-aldd-4c6	":"sip_node@a 00E5IM3D048K\ 5-8d40-1f75a2	pi","id":"E <mark>E8LFG1862LA</mark> e080dd"}	ventAbandon ES0001SP","	ed","ThisDN":"5550 CallThreadID":"003	0012348 ISS0JDF0	9350e2fc-aldd 8KVE8LFG1862L	A

## Sample Logs Explorer queries

## Contents

- 1 Sample queries to find important logs using the Logs Explorer
  - 1.1 A sample query for int-id in MCP
  - 1.2 A sample query for UUID in voice-sip
  - 1.3 A sample query for UUID in ORS
  - 1.4 A sample query for Call-ID for sip proxy
  - 1.5 A sample query for GAUTH for user authentication

Learn about the interface for analyzing logs data - the Logs Explorer, and take a look at some sample Logs Explorer queries to find logs.

### **Related documentation:**

•

### **RSS:**

• For private edition

### Sample queries to find important logs using the Logs Explorer

The Cloud Logging interface, the Logs Explorer, enables you to quickly and efficiently retrieve, view, and analyze logs from your queries.

Here are some sample queries for you to understand how to find important logs using the Logs Explorer in the Google Cloud Console.

#### A sample query for int-id in MCP

resource.type="k8s\_container"

resource.labels.project\_id=""

resource.labels.location=""

resource.labels.cluster\_name=""

resource.labels.namespace\_name="gvp"

labels.k8s-pod/app\_kubernetes\_io/instance="gvp-mcp"

labels.k8s-pod/app\_kubernetes\_io/log-monitor-name="gvp-mcp-log"

labels.k8s-pod/app\_kubernetes\_io/name="gvp-mcp"

00640220-10003852

#### A sample query for UUID in voice-sip

resource.type="k8s\_container"

resource.labels.project\_id=""

resource.labels.location=""

resource.labels.cluster\_name=""

resource.labels.namespace\_name="voice"

labels.k8s-pod/app\_kubernetes\_io/instance="voice-sip"

### A sample query for UUID in ORS

resource.type="k8s\_container"
resource.labels.project\_id=""
resource.labels.location=""
resource.labels.cluster\_name=""
resource.labels.namespace\_name="voice"
labels.k8s-pod/app\_kubernetes\_io/instance="voice-ors"
labels.k8s-pod/app\_kubernetes\_io/name="voice-ors"
00ES0D0T04905B2SK13CC2LAES00002Q

### A sample query for Call-ID for sip proxy

resource.type="k8s\_container"
resource.labels.project\_id=""
resource.labels.location=""
resource.labels.cluster\_name=""
resource.labels.namespace\_name="voice"
labels.k8s-pod/app\_kubernetes\_io/instance="voice-sipproxy"
labels.k8s-pod/app\_kubernetes\_io/name="voice-sipproxy"
00154DB6-1D01-1202-AC5C-A046C60AAA77-9483@10.198.70.160

### A sample query for GAUTH for user authentication

resource.type="k8s\_container"
resource.labels.project\_id="project ID"
resource.labels.location=""
resource.labels.cluster\_name=""
resource.labels.namespace\_name="gauth"
labels.k8s-pod/app\_kubernetes\_io/instance="gauth"
labels.k8s-pod/app\_kubernetes\_io/name="gauth"