Anthos GKE deployed on-prem includes multiple options for cluster logging and monitoring, including cloud-based managed services, open source tools, and validated compatibility with third-party commercial solutions. This page explains these options and provides some basic guidance on selecting the proper solution for your environment.

You have several logging and monitoring options for your GKE on-prem clusters:

- Stackdriver Logging and Stackdriver Monitoring, enabled by in-cluster agents deployed with GKE on-prem.
- Prometheus and Grafana, enabled by default in new clusters.
- Validated configurations with third-party solutions.

Stackdriver is the built-in observability solution for Google Cloud. It offers a fully managed logging solution, metrics collection, monitoring, dashboarding, and alerting. Stackdriver monitors GKE on-prem clusters in a similar way as cloud-based GKE clusters.

The Stackdriver agents can be configured with two different levels of logging and monitoring:

- System components only (the default).
- Stackdriver disabled (prior to disabling, see the support page (/gke-on-prem/docs/resources/support) for how Stackdriver is used for support purposes).

Stackdriver for GKE on-prem does not support collecting application logs or metrics at this time. This option will be a
future.

Stackdriver is an ideal solution for customers wanting a single, easy to configure, powerful cloud-based observability solution. Stackdriver is highly recommended when running workloads only on GKE on-prem, or workloads on GKE and GKE on-prem. For applications with components running on GKE on-prem and traditional on-premises infrastructure, you might consider other solutions for an end-to-end view those applications.

Currently, Stackdriver only collects cluster logs and system component metrics. The full Kubernetes Monitoring exper
available in a future release.

- See the Stackdriver section (#stackdriver_gkeop) for details on architecture, configuration, and what data is replicated to your Google Cloud project by default for GKE on-prem.
- See the Stackdriver Logging (/logging/docs/) and Stackdriver Monitoring (/monitoring/docs/) sections for more details on those services.

Prometheus and Grafana are two popular open source monitoring products:

- Prometheus (https://prometheus.io/docs/prometheus/latest) collects application and system metrics.
- Alertmanager (https://prometheus.io/docs/alerting/alertmanager/) handles sending alerts out with several different alerting mechanisms.
- Grafana (https://grafana.com/docs/) is a dashboarding tool.

Prometheus and Grafana runs on each admin cluster and user cluster by default. Prometheus and Grafana is recommended for application teams with prior experience with those products, or for operational teams who prefer to retain application metrics within the cluster and for troubleshooting issues when network connectivity is lost.

Google has worked with several third-party logging and monitoring solution providers to help their products work well with GKE on-prem. These include Datadog, Elastic, and Splunk. Additional validated third parties will be added in the future.

The following solution guides are available for using third party solutions with GKE on-prem:

- Monitoring GKE on-prem with the Elastic Stack (/solutions/partners/monitoring-gke-on-prem-with-the-elastic-stack)

Stackdriver Logging and Stackdriver Monitoring is installed and activated in each cluster when you create a new admin or user cluster.

The Stackdriver agents include several components on each cluster:

- *Stackdriver Operator* (`stackdriver-operator-*`), which manages the lifecycle for all other Stackdriver agents deployed onto the cluster.
- *Stackdriver Custom Resource* that is automatically created as part of the GKE on-prem installation process; users can change the custom resource to update values such as project ID, cluster name, and cluster location at any time.
- *Stackdriver Log Aggregator* (`stackdriver-log-aggregator-*`), a Fluentd StatefulSet that sends logs to the Stackdriver Logging API; if logs can't be sent, then the Log Aggregator buffers the log entries, up to 200GB, and tries to resend them for up to 24 hours. If the buffer gets full or if the Log Aggregator can't reach the Logging API for more than 24 hours, then logs will be dropped.
- *Stackdriver Log Forwarder* (`stackdriver-log-forwarder-*`), a Fluentbit daemonset that forwards logs from each machine to the *Stackdriver Log Aggregator*.
- *Stackdriver Metrics Collector* (`stackdriver-prometheus-k8s-`), a Prometheus and Stackdriver Prometheus Sidecar StatefulSet that sends Prometheus metrics to the Stackdriver Logging API.
- *Stackdriver Metadata Collector* (`stackdriver-metadata-agent-`), a deployment that sends metadata for Kubernetes resources such as pods, deployments, nodes, etc. to the Stackdriver Resource Metadata API; this data is used to enrich metric queries by enabling you to query by deployment name, node name, or even Kubernetes service name.

You can see all of the agents installed by Stackdriver by running the following command:

The output of this command is similar to the following:

The Stackdriver agents installed with GKE on-prem collect data about system components, subject to your settings and configuration, for the purposes of maintaining and troubleshooting issues with your GKE on-prem clusters, in one of the following modes:

Upon installation, Stackdriver agents are configured by default to collect logs and metrics, including performance details (for example, CPU and memory utilization), and similar metadata, for Google-provided system components including, all workloads in the admin cluster, and, for user clusters, workloads in the kube-system, gke-system, gke-connect, istio-system, and config-management-system namespaces.

Stackdriver agents can be disabled completely by deleting the Stackdriver custom resource. Before you disable Stackdriver, see the support page (/gke-on-prem/docs/support/getting-support) for details on how this affects Google Cloud Support's SLAs.

To disable Stackdriver for GKE on-prem:

Stackdriver agents capture data stored locally, subject to your storage and retention configuration. The data is replicated to the Google Cloud project specified at installation using a service account that is authorized to write data to that project. Stackdriver agents can be disabled at anytime, as described above, and data collected by Stackdriver agents can be managed and deleted like any other metric and log data, as described in Stackdriver documentation (/monitoring/docs/).

There are several configuration requirements to enable Stackdriver with GKE on-prem. These steps are included in Preparing to install (/gke-on-prem/docs/how-to/installation/preparing), and listed below.

1. A Stackdriver Monitoring Workspace must created within the Google Cloud project. This is accomplished by clicking **Monitoring** in Cloud Console and following the workflow.

2. You need to enable the following Stackdriver APIs:

   a. Stackdriver API (https://console.cloud.google.com/apis/api/stackdriver.googleapis.com/overview)

   b. Stackdriver Monitoring API (https://console.cloud.google.com/apis/api/monitoring.googleapis.com/overview)

   c. Stackdriver Logging API (https://console.cloud.google.com/apis/api/logging.googleapis.com/overview)

3. You need to assign the following Cloud IAM roles to the service account used by the Stackdriver agents:

   a. `logging.logWriter`

   b. `monitoring.metricWriter`

   c. `stackdriver.resourceMetadata.writer`

Anthos includes additional Stackdriver Logging free allotment above the standard Stackdriver Logging (/stackdriver/pricing#logging-costs) free allotment.

For more information, and to learn about credit for Stackdriver Logging metrics, please contact sales for pricing (/contact/?form=anthos).

Each GKE on-prem cluster is created with a Prometheus and Grafana instance deployed by default.

The Stackdriver Logging agents includes an embedded instance of Prometheus. This instance is separate from the theus and Grafana instance discussed in this section. You should not modify the Stackdriver Logging Prometheus ce.

The Prometheus Server is set up in a highly-available configuration with two replicas running on two separate nodes. Resource requirements are adjusted to support clusters running up to five nodes, with each handling up to 30 Pods that serve custom metrics. Prometheus has a dedicated

PersistentVolume with disk space preallocated to fit data for a retention period of four days plus an added safety buffer.

The admin control plane and each user cluster has a dedicated monitoring stack that you can configure independently. Each admin and user cluster includes a monitoring stack which delivers a full set of features: Prometheus Server for monitoring, Grafana for observability, and Prometheus Alertmanager for alerting.

All monitoring endpoints, transferred metric data, and monitoring APIs are secured with Istio components using mTLS and RBAC rules. Access to monitoring data is restricted only to cluster administrators.

Prometheus collects the following metrics and metadata from the admin control plane and user clusters:

- Resource usage, such as CPU utilization on Pods and nodes.
- Kubernetes control plane metrics.
- Metrics from add-ons and Kubernetes system components running on nodes, such as kubelet.
- Cluster state, such as health of Pods in a Deployment.
- Application metrics.
- Machine metrics, such as network, entropy, and inodes.

The Prometheus and Grafana instance installed on the admin cluster is specially configured to provide insight across the entire GKE on-prem instance, including the admin cluster and each user cluster. This enables you to:

- Use a Grafana dashboard to access metrics from all user clusters and admin clusters.
- View metrics from individual user clusters on Grafana dashboards; the metrics are available for direct queries in full resolution.
- Access user clusters' node-level and workload metrics for aggregated queries, dashboards and alerting (workload metrics are limited to workloads running in the kube-system namespace).
- Configure alerts for specific clusters.

Previous

← High availability (/gke-on-prem/docs/concepts/ha)