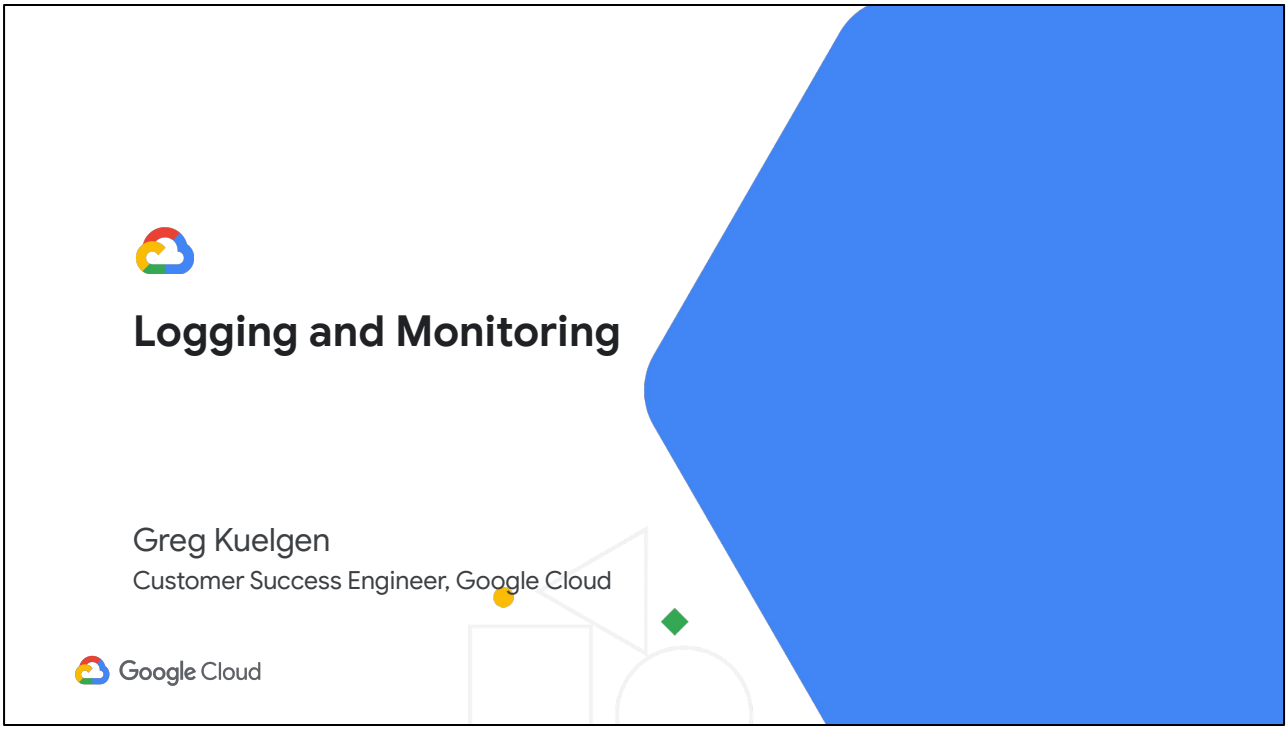


In this module, you will learn how Apigee hybrid logs system informational and error messages, You will also learn how you can use analytics and metrics data to monitor and troubleshoot the platform.

Finally, we will discuss the Apigee support model and how you can request support for issues related to Apigee hybrid.

Agenda

Logging

Metrics

Analytics

Monitoring and Troubleshooting

Lab

Apigee Support

Google Cloud

In this lecture, you will learn how Apigee hybrid logs messages from the runtime plane and how you can view those messages for troubleshooting purposes.

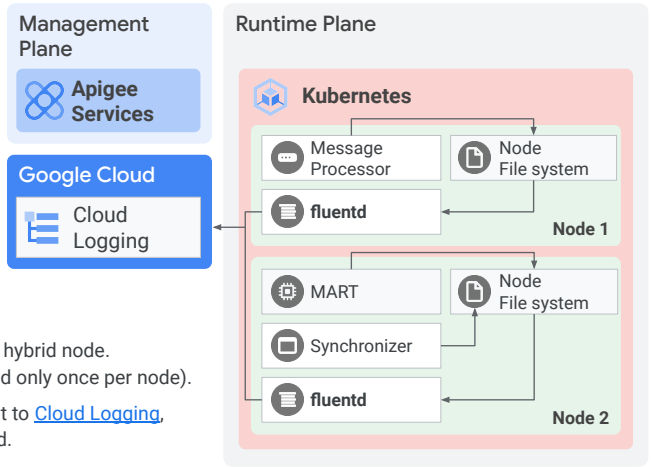
In later lectures, we will discuss how hybrid metrics and analytics are collected and reported and how you can use these capabilities to monitor Apigee hybrid.

We will then complete a lab on monitoring your runtime plane components.

And we will discuss how you can leverage the Apigee support process to get help with your Apigee hybrid installation.

Logging in hybrid

- All Apigee hybrid services in your Kubernetes cluster generate operational log information that can be used to help troubleshoot failures.
- If a pod's status indicates a problem, you can look at the log files for that pod to gain insight into the issue.
- Each runtime plane service writes log data to stdout/stderr, and Kubernetes writes this data to the host filesystem.
- A logs collector for hybrid runs on each hybrid node. The collector is a DaemonSet (replicated only once per node).
  - It collects the log data and exports it to Cloud Logging, where it can be viewed and analyzed.



Google Cloud

All of the Apigee hybrid services that run in your Kubernetes cluster generate log information.

This log information is useful for troubleshooting and debugging a given service or pod.

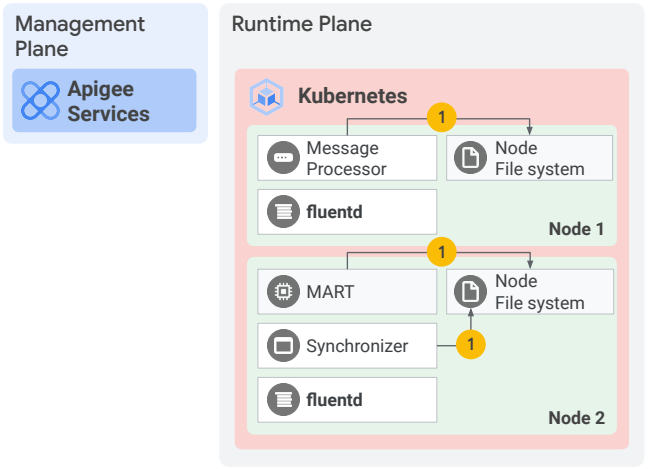
If a pod's status indicates a problem, you can look at the log files for that pod to gain insight into the issue.

Apigee support may request you to provide this log information to diagnose and solve a problem.

Let’s review how logging works in Apigee hybrid.

Logging operation

1. Hybrid runtime services write log data to stdout/stderr on their node's file system.

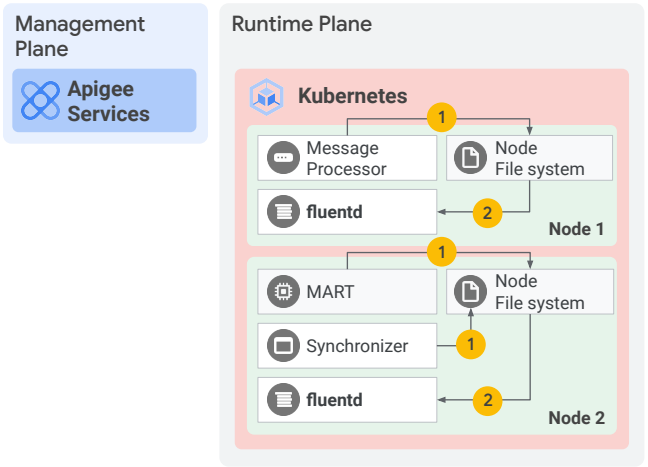


Google Cloud

Each runtime plane service in Apigee hybrid writes log data to stdout/stderr, which Kubernetes saves to the host filesystem.

Logging operation

1. Hybrid runtime services write log data to stdout/stderr on their node's file system.
2. The logs collector fluentd extracts the log data.



Google Cloud

A logs collector for hybrid runs on each node in the cluster.

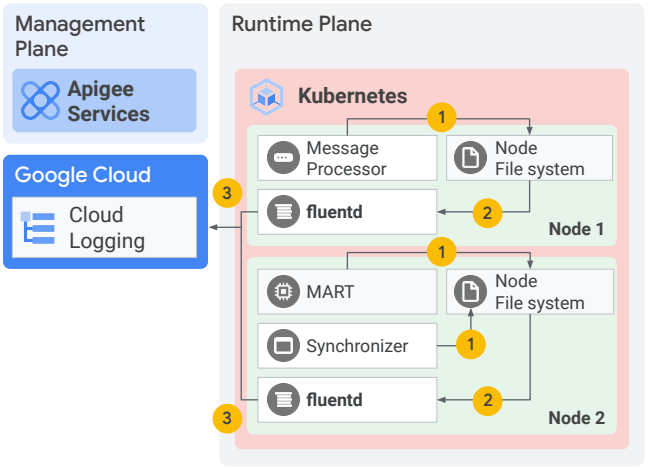
The collector is a Kubernetes DaemonSet that has one pod replica per node.

Apigee hybrid uses fluentd, which is an open source data collector application. Fluentd enables you to unify data collection and consumption.

Fluentd collects the written log data from the host file system.

Logging operation

1. Hybrid runtime services write log data to stdout/stderr on their node's file system.
2. The logs collector fluentd extracts the log data.
3. Fluentd sends the log data to Cloud Logging associated with your cloud account.

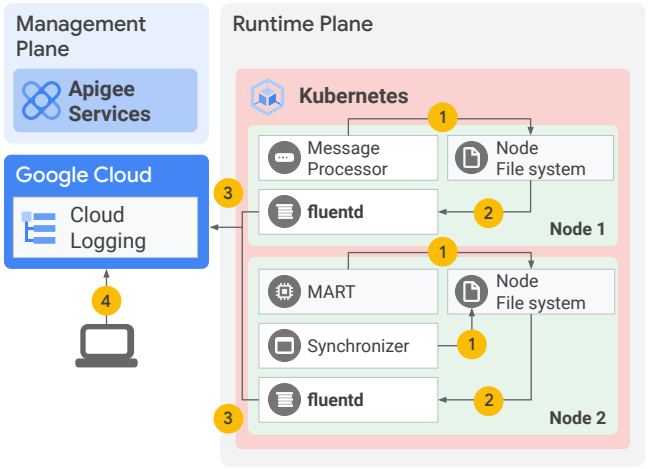


Google Cloud

The fluentd collector exports the logs to the Cloud Logging instance associated with your account on Google Cloud.

Logging operation

1. Hybrid runtime services write log data to stdout/stderr on their node's file system.
2. The logs collector fluentd extracts the log data.
3. Fluentd sends the log data to Cloud Logging associated with your cloud account.
4. You use the Google Cloud Console to access the log data.



Google Cloud

You access the log data via the Cloud Logging UI in the Google Cloud Console.

Viewing logs

You can view log data for any hybrid runtime component by using the kubectl logs command or on Cloud Logging in Google Cloud Console.

- Viewing logs directly:

  You can view the logs that are written to each pod's file system directly using the kubectl logs command:

  ```
  $ kubectl logs {pod_name} -n {namespace}
  ```

  For example: `$kubectl logs apigee-mp-hybrid-docs-test-blue-6fb96f5b9-2k8hp -n apigee`

- Viewing logs in Cloud Console:
  - In the Query builder resource drop-down list, select Kubernetes Container, the name of your cluster, your namespace, and the hybrid component container to view logs for that component.
  - Add and then run the query to view log entries in the query results.

Google Cloud

On GKE, Apigee hybrid uses the default logging mechanism for all runtime plane components.

You can view log data by using the kubectl logs command to directly query the hybrid runtime pod.

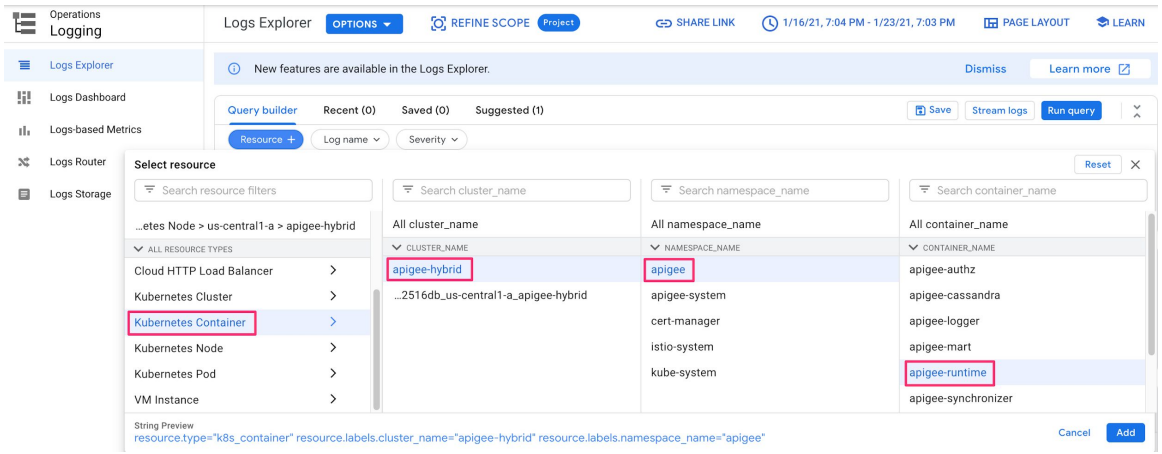
You must provide the pod name and pod namespace to the command.

You can also view logs in the Cloud Logging UI section of the Cloud Console.

Default retention period for log files is 30 days. This can be changed.

Log rotation by default occurs when the log file exceeds 10 MB.

Viewing logs in Cloud Console



Google Cloud

To view hybrid runtime component logs in Cloud Logging, you select the Kubernetes Container as the audited resource type.

You then select your cluster, the namespace in the cluster that contains the hybrid component, and the specific component whose logs you want to view.

In the example shown, the apigee-runtime or message processor component logs will be shown in the logs viewer.

Access logs

- Access logging is a feature provided with Istio ingress.
- By enabling access logging, you can get information about traffic that passes through the ingress gateway.
- Access logging is disabled by default in hybrid. You can enable access logging by adding the relevant entries to your overrides file (as shown).
- To view the access logs, run the kubectl logs command on the istio-ingressgateway pods:

```
$ kubectl logs {istio-ingressgatway-pod-name}
-n {namespace}
```

    - Access logs are returned in JSON format.

```
...
  ingress:
    enableAccesslog: true
...
```

Google Cloud

You can get access log information about API requests from client apps by enabling access logging on the Istio ingress gateway.

By default, access logging is disabled in Apigee hybrid. You can enable it by updating your overrides.yaml configuration file as shown, and applying the change to your cluster.

To view access logs, run the kubectl logs command and provide the name of the istio ingress gateway pod. The ingress gateway runs in the istio-system namespace in your cluster.

Audit logs

- Apigee hybrid leverages Cloud Audit Logs to create audit log entries.
- Apigee hybrid writes Admin Activity audit logs, which record operations that modify the configuration or metadata of a resource.
  - Admin Activity audit logs cannot be disabled.
- If enabled, Apigee hybrid writes Data Access audit logs. These logs contain API calls that read the configuration or metadata of resources within the project.
  - Data access audit logs are disabled by default.
- Apigee hybrid does not write System Event audit logs.
- Audit log entries can be viewed in Cloud Logging, the Cloud Logging API, or the gcloud command-line tool.

Google Cloud

Audit logs help you track activity on the resources in your project. Apigee hybrid uses Cloud Audit Logs in Google Cloud to write admin activity logs.

The admin activity audit logs record operations that modify the configuration or metadata of a resource and cannot be disabled.

You can also enable data access audit logs in Apigee hybrid. These logs record API calls that read the configuration or metadata of resources or update user-provided resource data.

Data access audit logs are disabled by default because they can be very large.

Both the admin activity and data access audit logs (if enabled), can be viewed by using Cloud Logging or the gcloud command line tool.

Audit logging

Agenda

Logging

Metrics

Analytics

Monitoring and Troubleshooting

Lab

Apigee Support

Google Cloud

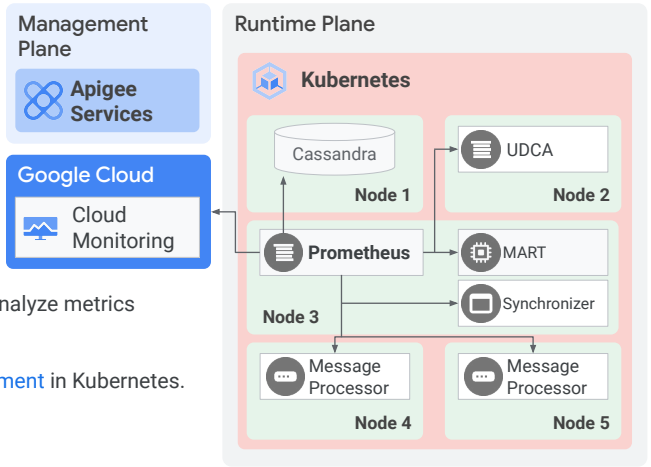
Metrics are an important feature used to monitor the health of a system.

In this lecture, you will learn how metrics are generated by the Apigee hybrid runtime plane components.

Later in this module, you will learn how to use these metrics for monitoring purposes.

Metrics collection

- Apigee hybrid collects operations metrics that you use to monitor the health of hybrid services.
- Apigee hybrid uses the Prometheus add-on for metrics collection. After collection, the metrics data is sent to Cloud Monitoring.
- Using the Cloud Monitoring console, you can view, search, and analyze metrics and manage alerts.
- Metrics is implemented as a Deployment in Kubernetes.



Google Cloud

Apigee hybrid uses Prometheus to collect metrics from all the runtime plane components in the cluster.

Prometheus is an open source monitoring toolkit that is deployed in your Kubernetes cluster as an operator add-on when you install Apigee hybrid.

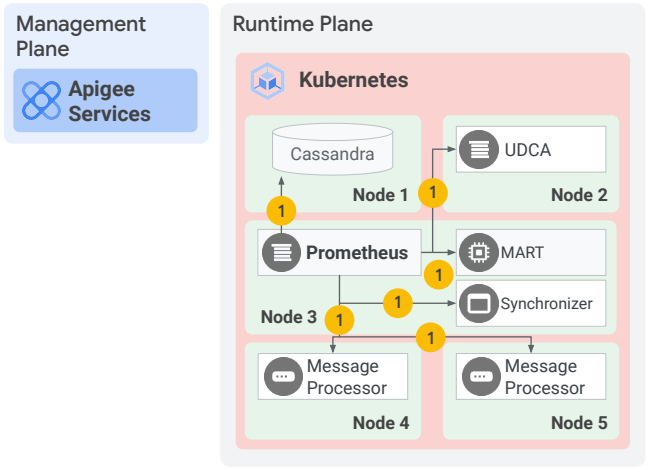
Prometheus scrapes application metrics from all hybrid services and sends the metrics to Cloud Monitoring.

There is one Prometheus server running per cluster.

Let's review how metrics collection works in Apigee hybrid.

Metrics operation

1. Prometheus scrapes metrics data every few seconds from all runtime services via HTTP and stores it.

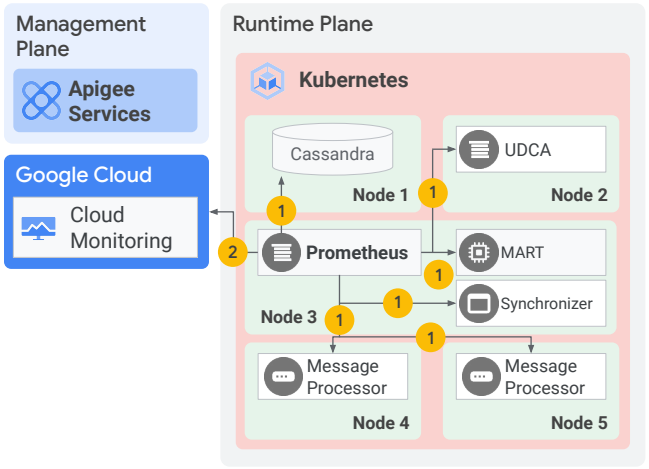


Google Cloud

Prometheus scrapes metrics data via HTTP endpoints every few seconds from all the runtime services in the cluster and then stores it.

Metrics operation

1. Prometheus scrapes metrics data every few seconds from all runtime services via HTTP and stores it.
2. It periodically sends the collected metrics data to the Cloud Monitoring service associated with your account in Google Cloud.



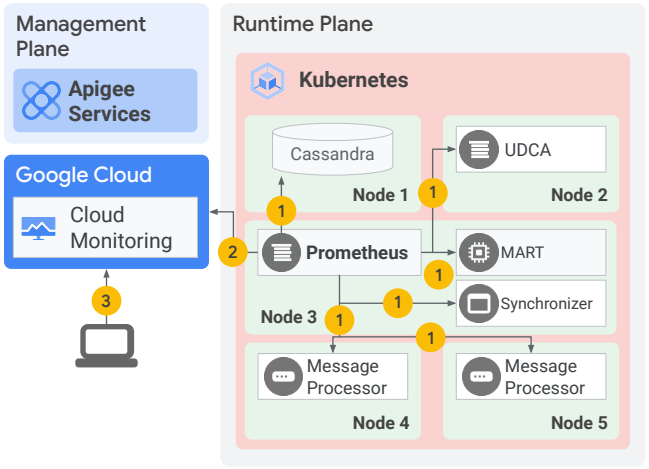
Google Cloud

It then periodically sends the collected metrics data to the Cloud Monitoring service running in Google Cloud.

The metrics data is associated with your Google Cloud account and project.

Metrics operation

1. Prometheus scrapes metrics data every few seconds from all runtime services via HTTP and stores it.
2. It periodically sends the collected metrics data to the Cloud Monitoring service associated with your account in Google Cloud.
3. You access the metrics data via the Cloud Monitoring UI in the Google Cloud Console.



Google Cloud

You can then access the metrics data via the Cloud Monitoring UI in the Cloud Console.

Configuring hybrid for metrics collection

- To configure hybrid for metrics collection, specify the following in the overrides.yaml config file:
  - project-id is the Google Cloud project ID.
  - region identifies the Google Cloud region where the apigee-metrics service sends data.
  - service_account_file is the path to the service account key file. The service account associated with the key must have the Monitoring Metrics Writer role.
- To disable metrics collection, set the enabled property to false in the overrides.yaml configuration file.

```
...
gcp:
  region: region
  projectID: project-id
metrics:
  serviceAccountPath: service_account_file
  enabled: true
...
```

Google Cloud

To configure metrics collection for your hybrid runtime components, edit your overrides.yaml file and provide the Google Cloud project ID associated with your hybrid project, and the Google Cloud region where metrics should be collected. This is also the region where the Apigee hybrid runtime component logs are sent.

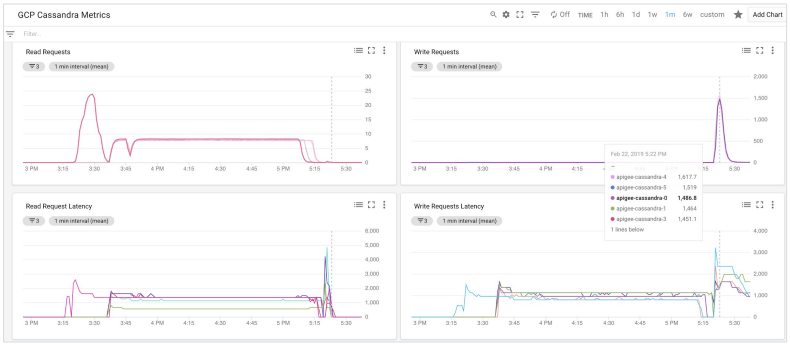
You must also provide the path to the service account key file that was created for the metrics component during hybrid installation.

Metrics collection is enabled by default. To disable it, set the enabled property to false.

After these changes are made, use the apigeectl command with the telemetry flag to apply them to your cluster.

Viewing metrics

- You can use the Cloud Monitoring Metrics Explorer in Cloud Console to select metrics to view.
- The selected metrics over various time periods can then be viewed in a dashboard.
- Apigee hybrid provides various metrics for Cassandra, UDCA, and other runtime components.



Google Cloud

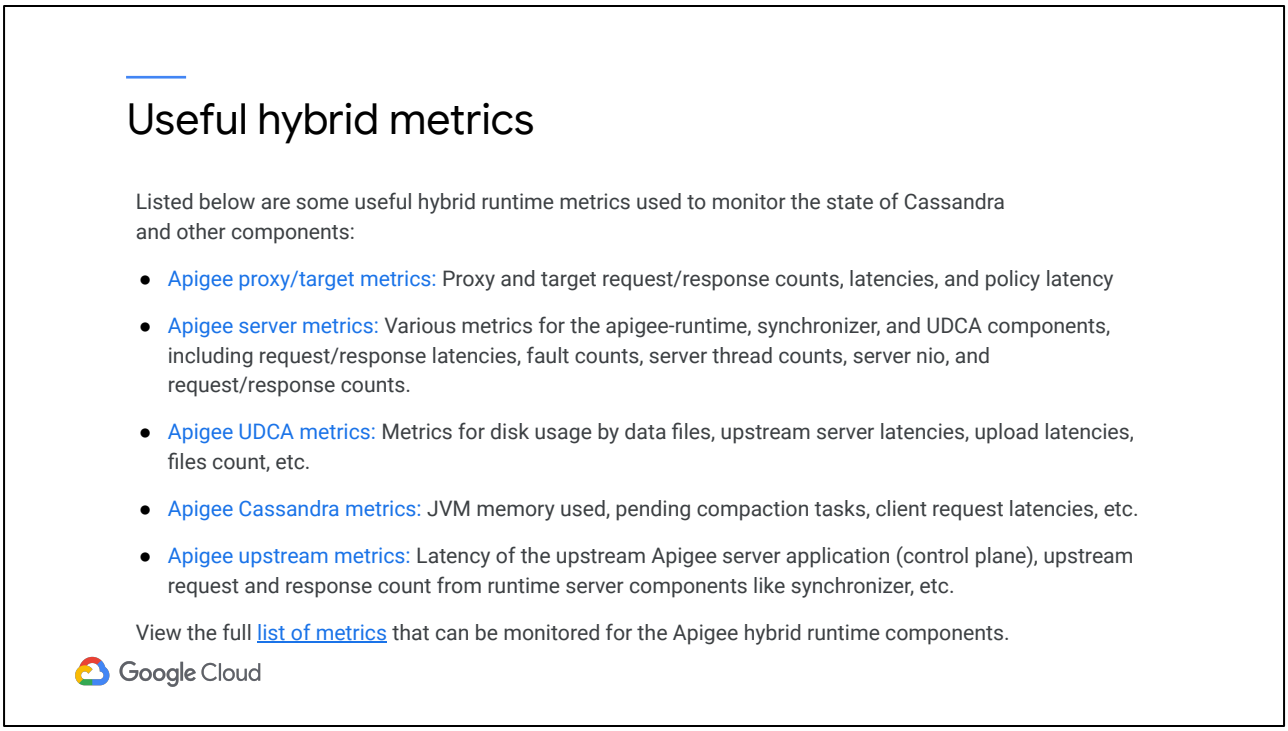
You can view the metrics generated by Apigee hybrid by using the Cloud Monitoring metrics explorer in the Google Cloud Console.

Using the metrics explorer, you can create dashboards to plot the various metrics for each hybrid runtime component.

You identify the runtime component by specifying the resource type and name.

Cloud Monitoring then generates a dashboard for the selected metric for that resource or component.

Viewing metrics



Apigee hybrid generates many useful metrics that you can use to monitor the health of your runtime plane.

These include API proxy and target metrics that provide data on API request and response latencies and traffic counts.

Metrics are also available for the individual hybrid runtime components.

For the Cassandra component, metrics on JVM memory usage, compaction tasks, etc. are all collected.

The full list of metrics is available on the Apigee hybrid documentation website.

Agenda

Logging

Metrics

Analytics

Monitoring and Troubleshooting

Lab

Apigee Support

Google Cloud

In this lecture, you will learn about Apigee analytics and the reporting capabilities available in Apigee hybrid.

Analytics

- Apigee hybrid analytics collects and calculates information that flows through API proxies.
- You can visualize this data with graphs and charts in the hybrid UI or use the Apigee APIs to download the data for offline analysis.
- Hybrid analytics collects a broad spectrum of API metadata, such as response time, request latency, target errors, and application information.
- Data retention: The length of time that the analytics data is retained; differs by offering.

Google Cloud

Apigee Analytics collects and aggregates a large amount of information that flows through your API proxies.

You can visualize this data with graphs and charts in the Apigee hybrid UI, or you can use the Apigee APIs to download the data for offline analysis.

Apigee API Analytics collects and analyzes a broad spectrum of data that flows across API proxies, including response time, request latency, request size, and target errors.

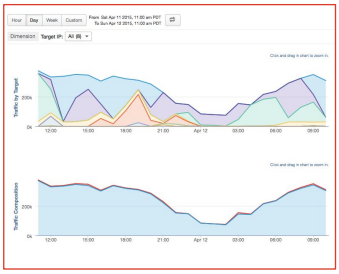
The complete list of data collected by analytics is available on the Apigee documentation website.

For a complete listing of data collected by analytics, see Analytics metrics, dimensions, and filters reference.

Visibility into API performance and usage

View trends on your API traffic, proxy, and target performance, top consuming developers and applications, API response times, error analysis, etc.

Proxy Performance



Target Performance

Error Analysis

Google Cloud

The Apigee hybrid UI provides a set of predefined dashboards that you can use to view analytics data.

Here are 3 sample dashboards.

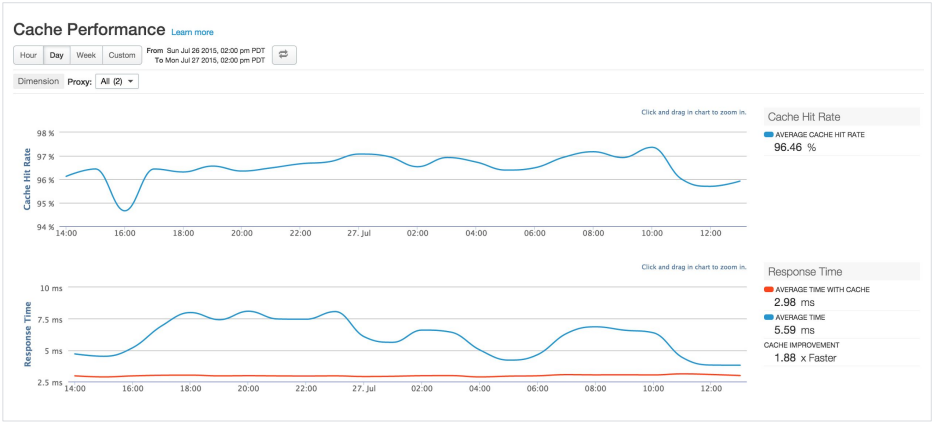
The proxy performance dashboard includes charts for various metrics such as total traffic received and average response time.

The target performance dashboard includes charts for metrics such as traffic by target server and target response times.

The error analysis dashboard includes charts for metrics such as proxy errors, target errors, and errors by response code.

Improve API response latency

Measure cache performance using the cache performance dashboard.



Google Cloud

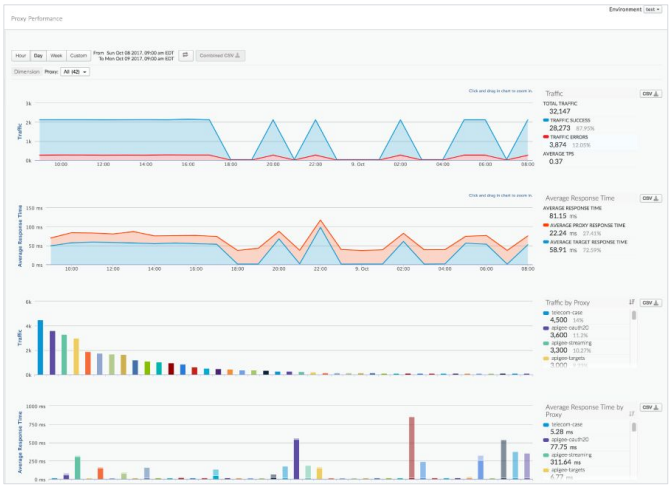
To improve API response times and reduce latency, your API proxies can use the cache policies available in Apigee hybrid.

The cache performance dashboard enables you to see how well your cache strategy is working.

It includes charts that plot various metrics such as the cache hit rate, total cache hits, and API response time with cache.

Accessing analytics data

- The hybrid UI provides a set of predefined dashboards that can be used to view analytics data.
- An example is the Proxy performance dashboard as shown.
- The following dashboards are available:
  - API Proxy Performance
  - Cache Performance
  - Error Code Analysis
  - Latency Analysis
  - Target Performance
  - Developer Engagement
  - Traffic Composition
  - Devices
  - GeoMap



Google Cloud

Apigee hybrid analytics currently includes 9 dashboards that you can use to track the performance of your APIs.

In addition to the performance dashboards, analytics also includes developer engagement and traffic composition dashboards that plot the number of active app developers, as well as top 10 apps, developers, API products, and proxies.

The error code analysis dashboard gives you a view into errors from your proxies, target servers, errors by response code etc., and provides valuable insight into the source of errors from your APIs.

The Devices dashboard plots information about the devices and user agents that are being used to access your APIs. It lets you spot trends in the device types being used to access your APIs.

The Geomap dashboard tracks traffic and error patterns across geographical locations. This dashboard helps you analyze API trends by location and focus on locations and the errors that may be occurring.

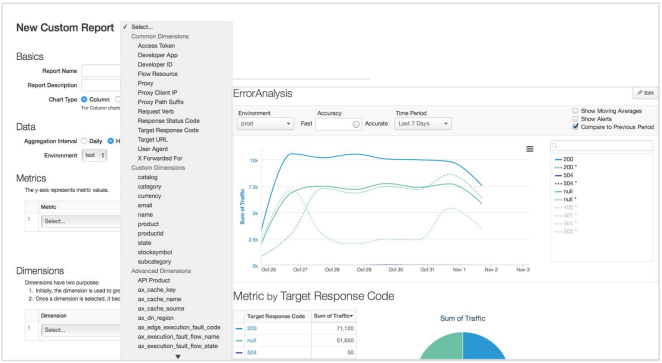
Analytics Overview

Custom reports

Custom reports let you drill down into specific API request/response metadata.

You create a custom report by selecting any of the dimensions and metrics built into hybrid by:

- Selecting the data you want to see, using metrics such as transactions per second or response time.
- Grouping the data in meaningful ways using dimensions such as API proxy, API product, or developer.
- Optionally limiting the data returned using filters.



Google Cloud

Apigee hybrid supports the creation of custom reports. Custom reports enable you to drill down into specific API metrics data for analysis.

Use the Apigee hybrid UI to create a custom report that includes any of the built-in Apigee dimensions and metrics.

Metrics API

- Apigee hybrid exposes a RESTful metrics API that can be used to download analytics data.
- You can use this API to automate certain analytics functions, such as retrieving data periodically using an automation client or script.
- You can also use the API to build your own visualizations in the form of custom widgets that can then be embedded in portals or custom apps.

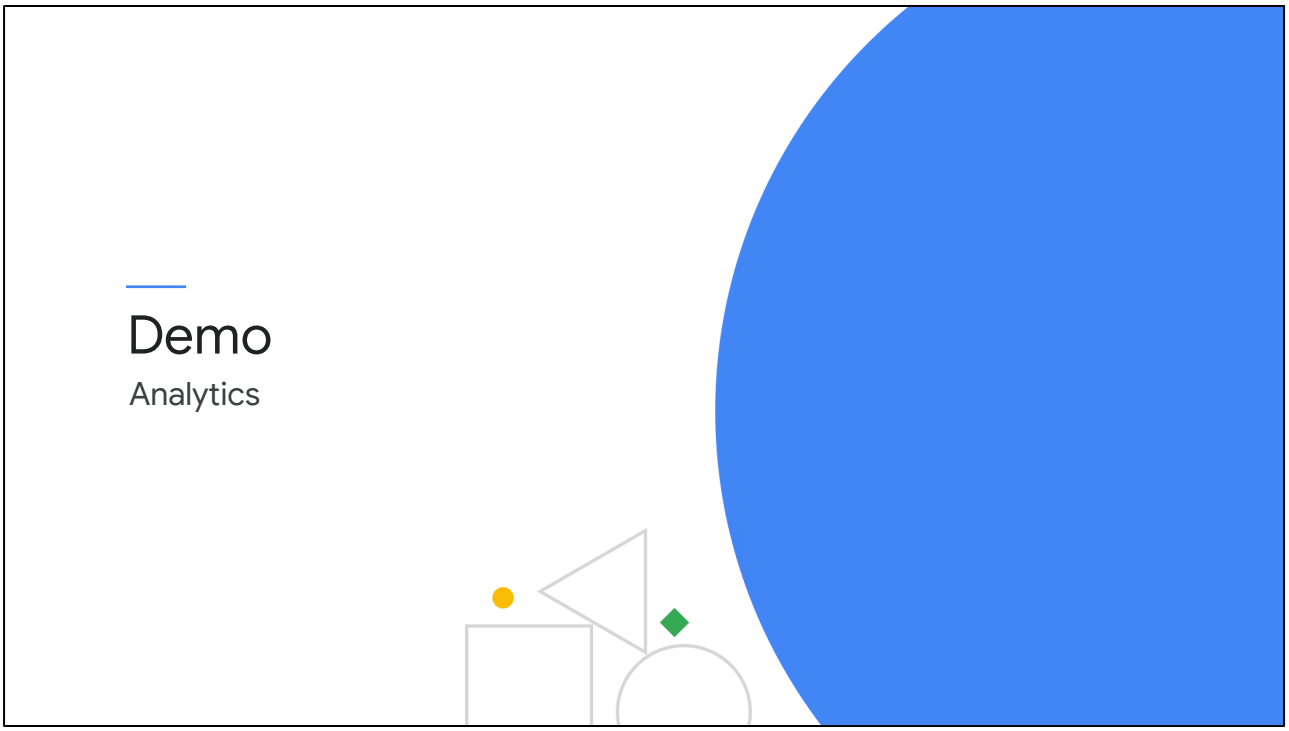
```
$ curl
https://apigee.googleapis.com/v1/organizatio
ns/{org_name}/environments/{env_name}/stats/
apiproxy?select=sum(message_count)&timeRange
=6/24/2020%2000:00~6/24/2020%2023:59&timeUni
t=hour -H "Authorization:Bearer $TOKEN"
```

Example API call returns the sum of requests (message count) per API proxy, in a 24-hour period, grouped by hour.

Google Cloud

Apigee hybrid provides a Metrics API that can be used to automate certain analytics functions, such as retrieving metrics periodically using an automation client or script.

You can also use the API to build your own visualizations in the form of custom widgets that you can embed in portals or custom apps.



Let’s review the Analytics capabilities of Apigee hybrid in this demo.

This demo uses the Apigee hybrid UI to showcase the various dashboard reports that are available.

You can use the dashboards to view metrics for your API proxes as they receive and process API traffic.

Agenda

Logging

Metrics

Analytics

Monitoring and Troubleshooting

Lab

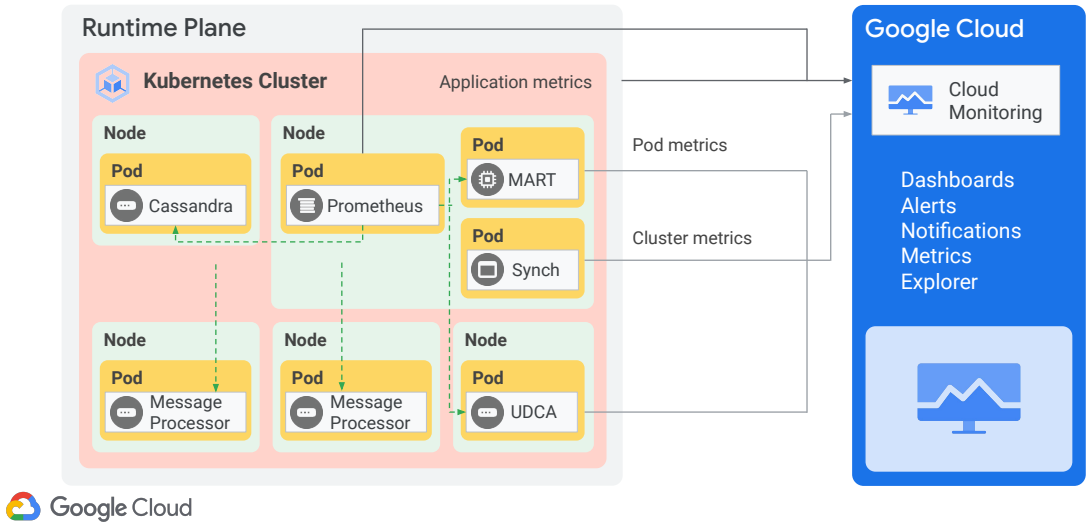
Apigee Support

Google Cloud

In this lecture, we will discuss monitoring and troubleshooting Apigee hybrid.

You will learn how to troubleshoot issues in the runtime plane. You will also complete a lab to monitor a hybrid runtime component using Cloud Monitoring in Google Cloud.

Monitoring context



Monitoring the Apigee hybrid runtime plane involves checking the health of your Kubernetes cluster and the health of the pods that are running in the cluster.

There are several metrics that can be used for cluster and pod monitoring.

The metrics generated by the cluster and the runtime pods are sent to Google Cloud, where you can view and monitor them using dashboards in the Cloud Monitoring UI.

Monitoring metrics

Several metrics of the Kubernetes hybrid runtime can be monitored.
They can generally be separated into 2 main groups: Cluster monitoring and Pod monitoring.

Cluster Monitoring

- Monitor the health of the entire cluster to make sure that nodes are operating normally.
- Determine resource usage of the cluster and gauge cluster capacity.

To monitor cluster health, Kubernetes provides some useful metrics to measure node resource utilization, including:

- CPU utilization: The fraction of allocatable CPU that is currently in use on the instance, and also request and limit utilization
- Memory utilization: The fraction of the allocatable memory that is currently in use on the instance
- Storage: Local ephemeral storage bytes used by the node
- Network bandwidth

Google Cloud

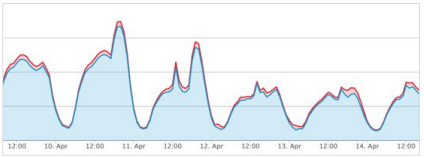
When monitoring the overall health of your Kubernetes cluster, you determine whether the nodes in the cluster are operating normally and whether there is sufficient capacity.

Many cluster metrics are available to gauge node resource utilization, such as CPU, memory, disk and network bandwidth utilization.

Pod monitoring metrics

Metrics for monitoring pods can be separated into three categories:

- Kubernetes metrics
  - Pod count: Actual/desired number of pods
  - Pod volume utilization: The fraction of the volume that is currently being used by the instance
  - Pod request latency
- Container metrics
  - CPU utilization: The fraction of CPU request and limit utilization
  - Memory limit utilization: The fraction of the memory limit that is currently in use on the instance
  - Restart count: Number of times the container has restarted
- Application metrics
  - Hybrid generates many metrics that can be used to monitor the runtime components.



Google Cloud

To monitor the pods in your cluster, you can use Kubernetes metrics that track the number of pods in the cluster and their storage volume utilization.

This can help you determine whether the number of pods running does not match the desired count, which indicates node or cluster resource constraints.

Container metrics like CPU utilization provide the fraction of requested CPU and the amount currently in use by the container.

Similar container metrics are also available for memory utilization. Both of these metrics can be used to gauge the resource usage, capacity, and health of your cluster.

As discussed in a previous lecture, Apigee hybrid generates many application metrics for each of the runtime components in the cluster.

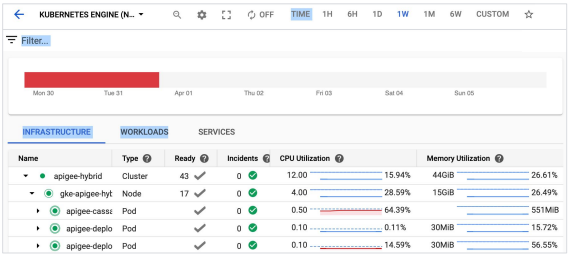
You can use these metrics to monitor the health of those components and of the proxies that process the API traffic flowing through the cluster.

Cloud Monitoring

Metrics generated and collected by the hybrid runtime are sent to Cloud Monitoring, where you can visualize them and monitor the health of the system.

Monitoring Dashboards, Alerts, Notifications:

- View and analyze metric data using predefined dashboards for the resources and services that you use.
- Create a custom dashboard to analyze Apigee hybrid metrics by creating charts for each metric.
- Create alerts using policies with hybrid runtime metrics based on threshold conditions.
- Create notifications based on alerts to take action when they are triggered.



Google Cloud

Apigee hybrid collects various metrics from the runtime components and sends them to Cloud Monitoring on Google Cloud.

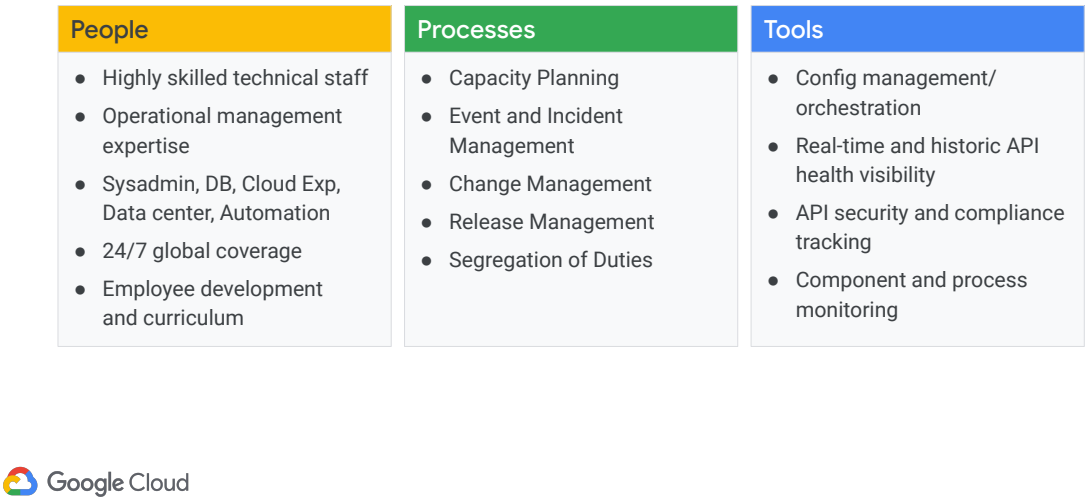
Cloud Monitoring is a service that collects and ingests metrics, events, and metadata. It generates insights into this data via dashboards and alerts.

You can use these predefined dashboards or create custom ones to view and analyze the metrics data from your Kubernetes cluster.

Using custom dashboards, you can analyze metrics from the Apigee hybrid runtime components and create alerts and notifications based on thresholds you define.

An alert notification enables you to review and take action to resolve the condition that triggered the alert.

Monitoring strategy: 3 pillars



Google Cloud

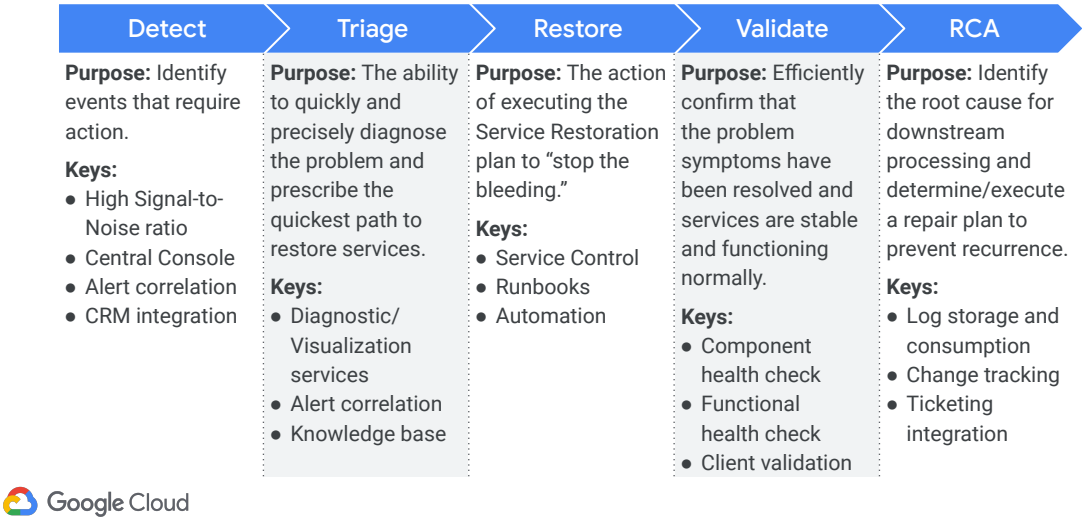
A well-defined monitoring strategy relies on trained personnel, well documented processes, and having the right tools in place.

Personnel are typically system administrators with experience in cloud operations, database management, devOps automation, and general operations.

Processes to manage your hybrid operations should include event and incident management, change and release management, and capacity planning.

To support your Apigee hybrid operations tasks, you need tools for configuration management, system observability and monitoring, and security and compliance tracking.

Support incident workflow



As part of your monitoring and troubleshooting process, adopting a well-defined support and incident workflow is critical.

The process should cover various phases in troubleshooting an issue, from detection, triage, service restoration and validation to root cause analysis.

Each phase incorporates distinctive capabilities.

It is important to have a central tool or console to detect and triage issues with the platform. A well-designed tool will have alert generation and integration with CRM tools to manage customer reporting and also visualization and diagnostic capabilities.

Using runbooks and automation is a repeatable best practice when restoring services. Component and functional health checks are essential pieces of a service validation strategy after the service has been restored.

A root cause analysis must always be performed for service outages in production environments. It helps document the cause and effect of the problem and helps to implement procedures to prevent recurrence.

Standardized actionable alerts and automation

Google Cloud

An alert that is generated by your monitoring system must have sufficient information that makes it actionable.

Details in the alert notification should include the metric that triggered the alert, service information, and other details that help the operator take corrective action.

A best practice is to create playbooks that document the problem, any triggers, and steps to resolve or escalate the issue. This helps to create a well defined, repeatable process for operators to follow when similar issues arise in the future.

In general, sending a notification of problems is not enough. Defining actionable alerts and playbooks to resolve the problems with automation is required and should be part of your monitoring and troubleshooting strategy.

Troubleshooting APIs

Some API calls are failing, or you are experiencing intermittent errors or latency.

Google Cloud

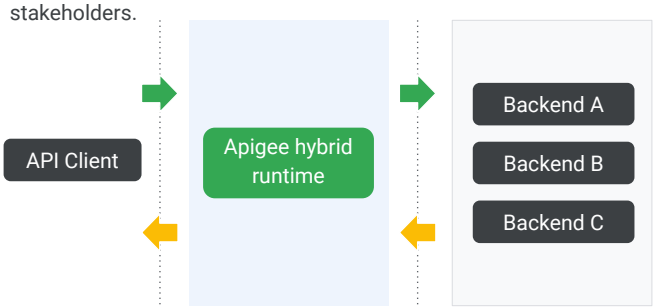
Let’s review some common troubleshooting scenarios.

Assume there are some requests to your API proxy that are failing intermittently or have increased latency.

Start with Apigee analytics

When issues arise, it is important to effectively diagnose the problem in order to solve it.

Apigee Analytics allows you to determine whether the problem is located within the hybrid's boundaries. You can quickly identify the proxies and target endpoints that are having problems and expedite an escalation to specific stakeholders.



Google Cloud

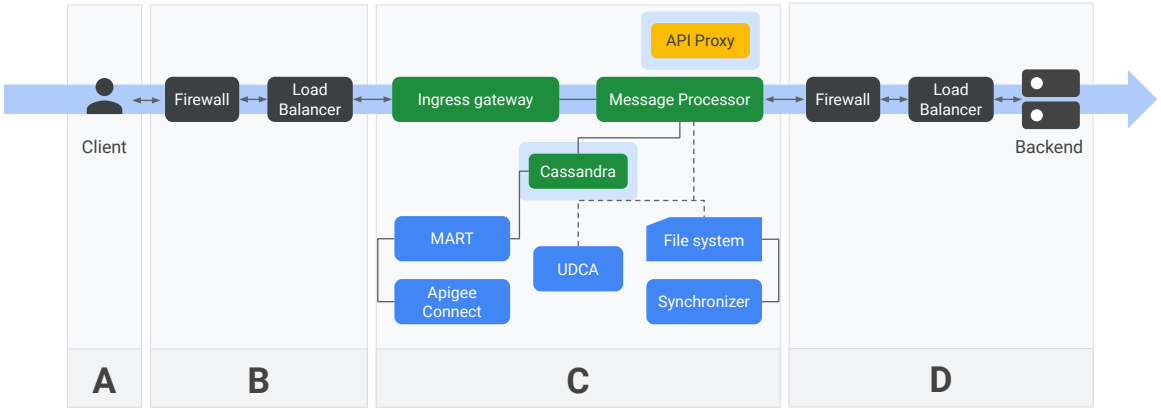
Apigee analytics is one of the tools that you can use to troubleshoot issues with your API proxies.

It is easy to access and use the various analytics dashboards in the Apigee hybrid UI.

You can review the error analysis, the API proxy and target performance, and latency dashboards to troubleshoot errors or latency issues with your API proxy.

Using the charts in these dashboards, you can determine the source of the errors or latency to determine whether the issue is from the target backend or the API proxy itself.

The runtime critical path



Google Cloud

When monitoring your API management platform, it is important to focus on the runtime critical path.

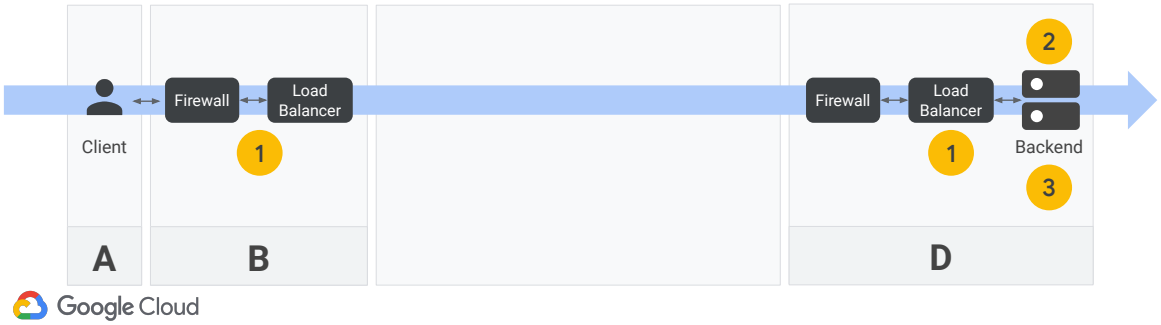
This includes the calling client application, any firewall and load balancers in front of the ingress gateway external to the cluster, the runtime message processor component that processes the API request, any firewall and load balancers in your network, and your backend system.

The components that you manage within your infrastructure should support a high level of uptime SLA and must be continuously monitored.

API runtime traffic interruption or high latency

API response time can be affected by multiple factors. Consider these key items when determining the source of the problem:

1. Northbound or southbound connectivity problems
2. Backend connectivity latency and target servers response time
3. Large payload



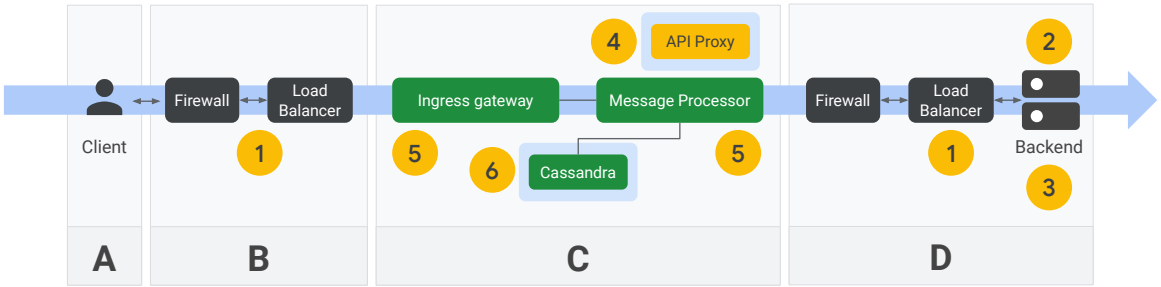
Issues with high latency or traffic interruption can be investigated at various places in the runtime critical path.

For connectivity issues between the client applications and the hybrid runtime plane, check any firewall or load balancer configuration you may have in front of the ingress gateway.

For connectivity and latency issues between the API proxy and the target backend servers, perform the check on any network objects that may also lie within your backend network.

API runtime traffic interruption or high latency

4. API Proxy complexity, processing time, or implementation issues:
   - Poor-performing code deployed to environment
   - Excessive logic or responsibility being executed in the Gateway
   - Capacity (CPU, RAM, Disk or number of R, MP, CS) to meet API traffic and throughput requirements
5. Ingress gateway/MP down or unresponsive
6. Cassandra ring down or response latency



Google Cloud

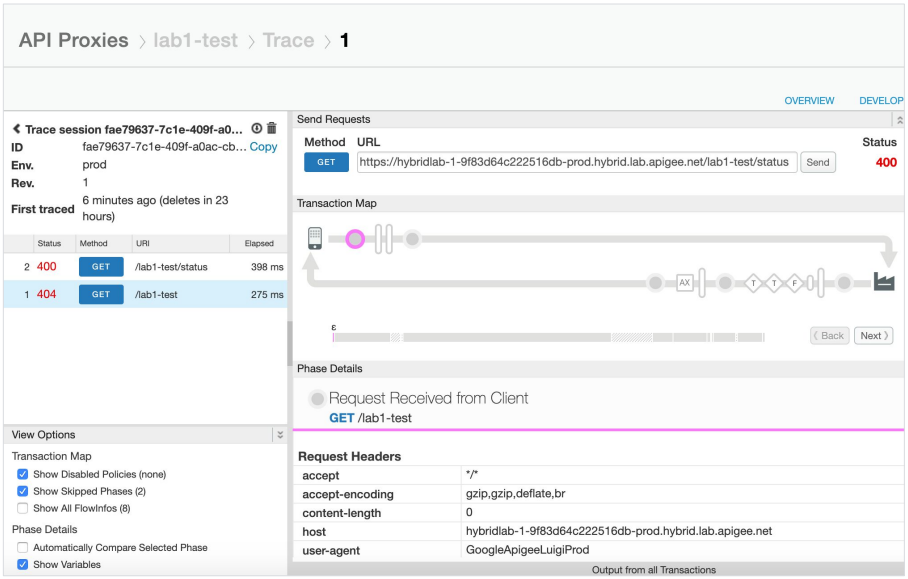
Of course there may be other issues outside of the network that cause traffic interruption or latency.

An unexpectedly large payload could cause the backend service to react unpredictably. You can mitigate this issue by implementing payload validation in your API proxy.

Other issues related to API proxy complexity or runtime resource capacity can be investigated by examining the API logic and current resource allocations in your cluster.

Trace

Use the trace feature in hybrid to debug issues in the API proxy.



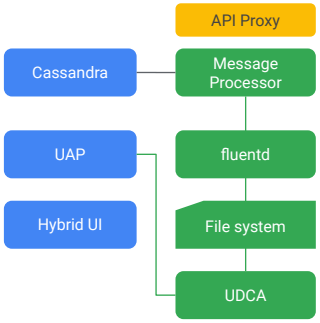
Google Cloud

The trace feature in Apigee hybrid is a useful tool that enables you to debug issues in your API proxy logic.

Using trace, you can debug the policies in your API proxy as they process the request and response data that flows through the proxy.

The trace tool displays the value of request and response metadata and any data you store in flow variables in your proxy. This enables you to view how this data is modified by your proxy logic and aids in debugging issues with the API.

Troubleshooting analytics



Consider these factors when troubleshooting analytic reports visualization and data capture:

- Connectivity between the Message Processor and UDCA data collection pods
- Connectivity between UDCA and UAP in the management plane
- Streaming of Analytics data to fluentd
- Pod file system capacity
- Permissions issues during creation of a directory on the file system
- Error during upload of datasets to UAP

For a list of UDCA log errors, review this page: UDCA logs

Google Cloud

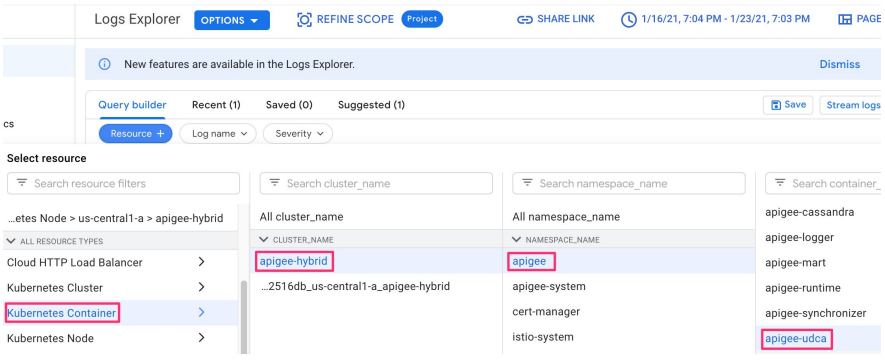
Troubleshooting the analytics service involves checking connectivity between some of the runtime components that are used to stream and collect analytics data.

The connections involved are between the runtime message processor and the Universal Data Collection Agent or UDCA components in the runtime plane and between UDCA and the Unified Analytics Platform or UAP component in the management plane.

Pod file system capacity and directory permissions can also contribute to issues with the analytics service in Apigee hybrid.

Troubleshooting UDCA

- To troubleshoot UDCA in Apigee hybrid, view the pod log files.
- Use the kubectl logs command to view log entries:
  - `$ kubectl logs {udca_pod_name} -n {namespace}`
- UDCA logs can also be viewed in Cloud Logging Logs Viewer.



Kubernetes Container -> {hybrid cluster name} -> {namespace} -> apigee-udca

Google Cloud

To troubleshoot UDCA operation, you can view the pod logs directly or within the Cloud Logging UI in Google Cloud Console.

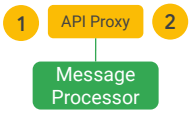
In the Cloud Logging UI, use the Kubernetes container filter to select the resource by selecting your hybrid cluster, namespace and the apigee-udca runtime component.

You can then view the log entries generated by the UDCA runtime component.

Troubleshooting API proxy deployments

Consider the following items when troubleshooting API proxy deployment failures:

1. Error in API proxy implementation
2. API proxy bundle size



Google Cloud

Let's discuss a few troubleshooting scenarios.

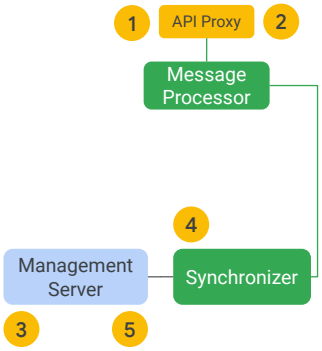
One of the common issues that arise is an error during API proxy deployment. Errors can be due to the proxy implementation logic or the size of the API proxy bundle.

The size and other product limits are documented on the Apigee hybrid resources website.

Troubleshooting API proxy deployments

Consider the following items when troubleshooting API proxy deployment failures:

1. Error in API proxy implementation
2. API proxy bundle size
3. Management server availability
4. Connectivity issues between Management Server and Synchronizer, or service account misconfiguration
5. Missing API proxy dependencies, including KVMs and target servers



Google Cloud

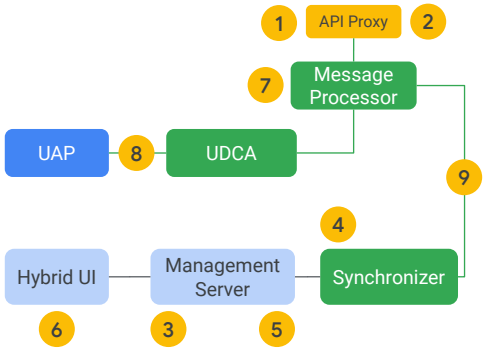
Other errors due to management service availability, or connectivity issues between the management service and the runtime plane components, although rare, are possible.

Errors can also be due to missing API proxy configuration dependencies such as key value maps and target servers. Make sure to configure these objects as part of your proxy development lifecycle in all the hybrid runtime environments where they are needed.

Troubleshooting API proxy deployments

Consider the following items when troubleshooting API proxy deployment failures:

6. Runtime pods sync status error in the hybrid UI
7. MP state: Unavailable, restarting, pod sync state not yet updated, or in error state
8. Connectivity between UDCA and UAP component in the management plane
9. MP-to-synchronizer connectivity issues or synchronizer failure.



Google Cloud

Another possible source of errors is that the runtime Synchronizer, Message Processor, or UDCA pods are not successfully running or there are connectivity issues between the synchronizer or UDCA components and the management plane.

To troubleshoot, you need to look at the individual component pod log files to determine the cause of the error.

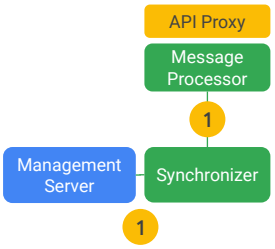
Troubleshooting the Message Processors (1)

Consider the following items when troubleshooting Message Processor (MP) errors:

1. One or more apigee-runtime (MP) pods are not in Ready state:

   ```
   $ kubectl describe pod -n {namespace} {runtime-pod-name}
   Readiness probe failed: HTTP probe failed with statuscode: 500
   ```

- The error is typically because no data is available to the MP
- Usually caused by connectivity issues with the synchronizer, or between the synchronizer and the management plane.



Google Cloud

The message processor runtime component deploys API proxies and executes proxy logic.

A message processor pod can sometimes indicate that it is not in a ready state. This can be caused by connectivity issues with the synchronizer runtime component or between the synchronizer component and the management plane.

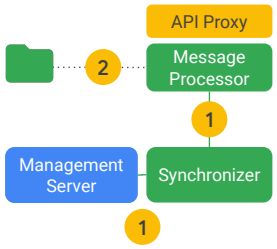
To troubleshoot, make sure that the synchronizer is up and running and that the service account used to configure the synchronizer for the environment is valid.

Troubleshooting the Message Processors (1)

2. Readiness probe fails due to invalid encryption key. Running the describe command above returns:
   Readiness probe failed: Probe hybrid-encryption-key-validation-probe failed

- Supported encryption key lengths are 16 or 24 or 32 bytes, and the key's value must be base64-encoded.
- Make sure this is properly configured in the overrides.yaml file.

Additional details and log entries for troubleshooting MP issues are documented here.



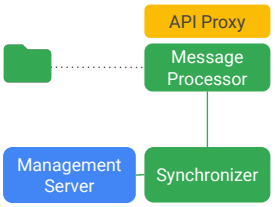
Google Cloud

For errors due to the validation of encryption keys, make sure that the configured key is base64-encoded and that the key length is supported by Apigee hybrid.

Troubleshooting the Message Processors (2)

Test the runtime pod operation using port-forwarding:

- Get the list of apigee-runtime (MP) pods in the production environment:
  `$ kubectl get pods -n apigee -l env=prod,app=apigee-runtime`
- For a given runtime pod, set up port forwarding:
  `$ kubectl port-forward -n apigee {podname} 8443:8443`
- In another terminal window, use a utility such as ***curl*** to send a request to any deployed API proxy in the *prod* environment, and check the response:
  `$ curl -k -v https://0:8443/lab1-test/200`



Google Cloud

To help isolate a problem with API proxies, check whether you can make an API proxy call directly from inside the apigee-runtime pod bypassing the ingress gateway.

To accomplish this, first get the list of all the runtime pods in the apigee namespace for your environment.

Set up port forwarding to the specific pod by providing the name of the runtime pod.

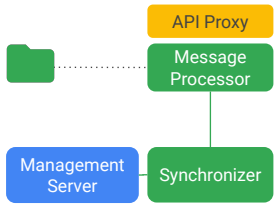
Call a deployed API proxy by using the curl command line utility on the forwarded port and using the basepath URL of the proxy.

The response from the proxy will indicate whether it is functioning as expected.

Troubleshooting the Message Processors (2)

Test the runtime pod operation using port-forwarding:

- Set up port forwarding on port 8843 to call the management API on the MP to list information about the deployed proxies:
  `$ curl -k https://0:8843/v1/classification/tree`
- Enable DEBUG mode for the runtime pod to include additional information in pod logs:
  `$ curl "https://0:8843/v1/logsessions?sessions=debug" -X POST -v -k`
- Check runtime pod logs:
  `$ kubectl logs -f -n apigee {podname}`



Google Cloud

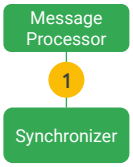
You can also call the management API on the runtime message processor pod to send a request to the classification tree API. The response from this API will list information about all the deployed proxies.

To help with further troubleshooting, you can enable DEBUG mode to include more detailed information in the apigee-runtime message processor pod logs by making an API call to the logsessions management API.

After you finish debugging an issue, you should reset the logging mode back to INFO, using the same API call with the sessions parameter set to info.

Troubleshooting Synchronizer

Consider the following items when troubleshooting synchronizer errors:



1. MP to synchronizer connectivity issues, or synchronizer failures.

   Check runtime pod logs:

   ```
   $ kubectl logs -f -n {namespace} {runtime-pod-name}
   ```

   Check synchronizer pod logs:

   ```
   $ kubectl logs -f -n {namespace} {synchronizer-pod-name}
   ```

   If the synchronizer for a hybrid environment is not running, restart it using the following command:

   ```
   $ apigeectl apply -f overrides/overrides.yaml --env {env}
   ```

Additional details and log entries for troubleshooting Synchronizer issues are documented here.

Google Cloud

To troubleshoot issues with the synchronizer component, you can view the synchronizer pod logs for the hybrid environment in which the issue occurs.

Use the kubectl logs command with the pod name.

Troubleshooting Synchronizer

Consider the following items when troubleshooting synchronizer errors:

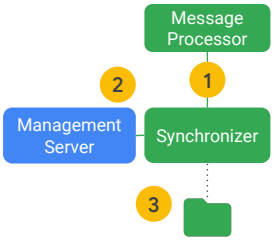
2. Connectivity issues between Management Server and Synchronizer.

   Check the version.properties for potential mismatch with the contract folder name:

   ```
   $ kubectl exec -it -n {namespace} {synchronizer-pod-name} --
   /bin/bash
   $ cat /opt/apigee/var/log/apigee-synchronizer/versions.properties
   ```

   If the value of the active.version property does not match the contract folder name, check synchronizer logs to troubleshoot further.

3. Make sure the required synchronizer configuration properties are provided with correct values.



Google Cloud

To troubleshoot synchronization issues with the management plane, check the versions.properties file to confirm that the current active version matches the folder name that contains the downloaded contract data.

To do this, use the kubectl command to start a shell on the synchronizer pod and provide the name of the pod.

Then compare the active.version property with the contract folder name to confirm that they match. If they do not match, check the synchronizer logs for details to debug the issue further.

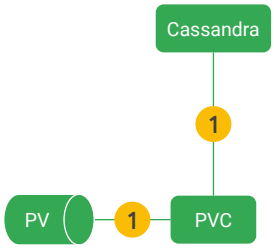
Also confirm that the configuration properties for the synchronizer component have correct values as per the hybrid configuration reference documentation.

Troubleshooting Cassandra (1)

Consider the following items when troubleshooting Cassandra errors:

1. During startup, one or more Cassandra pods stuck in Pending state indicate that the pod could not be created: `$ kubectl get pods -n {namespace}`

```
NAME                  READY   STATUS    RESTARTS   AGE
apigee-cassandra-0    0/1     Pending   0          10m
```



Google Cloud

In Apigee hybrid, Cassandra is a stateful component in the runtime plane with external storage attached and uses Persistent Volume Claims, or PVCs.

You may sometimes see that a Cassandra pod is stuck in a pending state during startup. This indicates that Kubernetes cannot schedule the pod on a node.

This may happen because the node doesn't have sufficient resources, such as CPU or memory, available to create the pod.

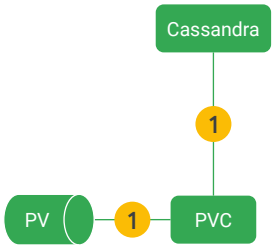
Another possible reason is that the pod is waiting for the persistent volume to be created.

Troubleshooting Cassandra (1)

By running the `kubectl describe pod` command, you can get more details on the possible causes:

- Insufficient CPU or memory: Resize the cluster to add additional nodes, or re-create the Cassandra nodepool with a larger machine type.
- Persistent Volume not created: Check the status of the persistent volume claim to determine the cause of the error, using these commands:

```
$ kubectl -n {namespace} get pvc
$ kubectl -n {namespace} describe pvc {pvc-name}
```



If the error is due to the underlying *StorageClass*, follow instructions here to change the default storage class.

Google Cloud

You can get more details on the source of the error by running the kubectl describe command, providing the pod name and namespace.

For errors due to insufficient resources, you can resize the runtime data nodepool in the cluster or use larger machine types for the Cassandra pods.

For errors due to persistent volumes, use the kubectl describe command on the PersistentVolumeClaim resource to further investigate the cause of the issue.

Troubleshooting Cassandra (2)

2. One or more Cassandra pods is in CrashLoopBackoff state, which indicates that the pod could not be created:

```
$ kubectl get pods -n {namespace}
```

NAME	READY	STATUS	RESTARTS	AGE
apigee-cassandra-0	0/1	CrashLoopBackoff	0	0m

Check the Cassandra error log to diagnose the problem:

```
$ kubectl logs {cassandra-pod} -n {namespace}
```

- If errors indicate previous data center differences, try deleting the old or stale PVCs from the cluster:
  `$ kubectl -n {namespace} delete pvc {cassandra-pvc-name}`
- If errors indicate that the Truststore directory is not found, verify the path to the Cassandra key and certificate files in the overrides.yaml file.

Google Cloud

During startup, the Cassandra pods are sometimes stuck in the CrashLoopBackoff state.

You can determine this by running the kubectl get pods command. This indicates that Kubernetes cannot create the pod.

Use the kubectl logs command to check the Cassandra pod error log for more details on the error.

If an error indicates data center differences, the pod has a stale persistent volume attached.

To resolve this issue, delete the old or stale PersistentVolume claims. For new installations, delete all the PVCs and retry the Cassandra setup.

For errors that indicate that the truststore directory is not found, verify that the key and certificates that are configured in your overrides.yaml configuration file are valid.

Troubleshooting Cassandra (2)

3. Underlying node failure could prevent the Cassandra pod from getting to a *Running* state.

   Check the node status:

```
$ kubectl get nodes -n {namespace}
```

```
NAME                                          STATUS     ROLES      AGE      VERSION
Gke-apigee-hybrid-default-pool-a7cb264c-rlwf   NotReady   <none>     8d       v1.13.2
```

Google Cloud

If a node that runs a Cassandra pod fails, the pod status will stay in a pending state and the node will indicate a status of NotReady.

You can determine the status of the node by using the kubectl get nodes command.

Troubleshooting Cassandra (3)

To fix underlying Cassandra node issues, perform these steps:

1. Using nodetool, remove the failed instance from the corresponding Cassandra pod:
   ```
   $ kubectl exec -it {cassandra-pod} -- nodetool status
   $ kubectl exec -it {cassandra-pod} -- nodetool removenode deadnode_hostID
   ```

2. The Cassandra pod has node affinity, so to prevent it from starting back up on the failed node, delete the persistent volume claim:
   ```
   $ kubectl delete pvc {volumeClaim_name} -n {namespace}
   ```

Additional details and log entries for troubleshooting Cassandra issues are documented here.

Google Cloud

To fix a failed Cassandra node, use the nodetool command and provide the name of the cassandra pod to check the node status and remove the node.

Next, remove the pod's PersistentVolumeClaim to prevent the Cassandra pod from attempting to start up on the failed node because of node affinity.

To do this, run the kubectl delete pvc command and provide the name of the PVC.

Troubleshooting Cassandra (3)

3. Update the volume template with the newly added hostname value.

   Apply the changes to create the PersistentVolume for the new node:

   ```
   $ kubectl apply -f volume-template.yaml
   ```

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: cassandra-data-3
spec:
  capacity:
    storage: 100Gi
  accessModes:
  - ReadWriteOnce
  persistentVolumeReclaimPolicy: Retain
  storageClassName: local-storage
  local:
    path: /apigee/data
  nodeAffinity:
    "required":
      "nodeSelectorTerms":
      - "matchExpressions":
        - "key": "kubernetes.io/hostname"
          "operator": "In"
          "values":
["gke-apigee-hybrid-default-pool-a7cb264c-rfz2"]
```

Google Cloud

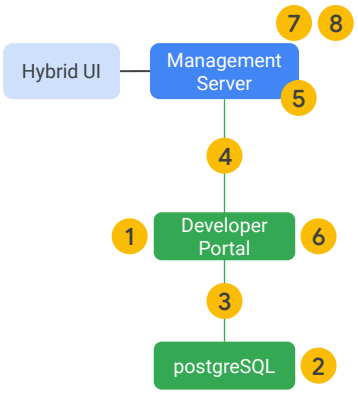
Update the volume template in your hybrid configuration file to use the new node hostname, and apply the configuration to your cluster by using the kubectl apply command and providing the name of the volume template configuration file.

Troubleshooting the developer portal

Apigee hybrid supports an Integrated Developer portal hosted in the management plane and a Drupal 8–based portal hosted by the customer.

Some of the common areas of focus for the D8 developer portal:

1. Developer Portal unavailable
2. Developer Portal data store unavailable
3. Developer Portal unable to connect to the database
4. Developer Portal unable to connect to Management Server
5. Management Server unavailable (hybrid management plane)
6. Developer Portal username/password used to connect to the Management Server does not work
7. Apps, Developer, or related data not found
8. Changes or deprecation of APIs or API products



Google Cloud

The Apigee Drupal 8 developer portal is installed and managed by you; therefore, you must maintain and troubleshoot any issues with the portal.

The Apigee Integrated developer portal is hosted and maintained by Google in the management plane on Google Cloud.

You need to monitor the Drupal 8 developer portal and troubleshoot any issues that may cause the service to become unavailable.

Any misconfiguration with the credentials that the portal uses to access its local database or connectivity issues to the database can cause the portal to become unavailable.

The developer portal also makes management API calls to the Apigee hybrid management server in the hybrid management plane. Make sure that there are no connectivity issues and that the portal is configured with the correct credentials to access the management plane.

Data synchronization issues between the portal and the management plane can cause Apgiee entities to be unavailable. Verify that the synchronization jobs are correctly configured in the developer portal.

Agenda

Logging

Metrics

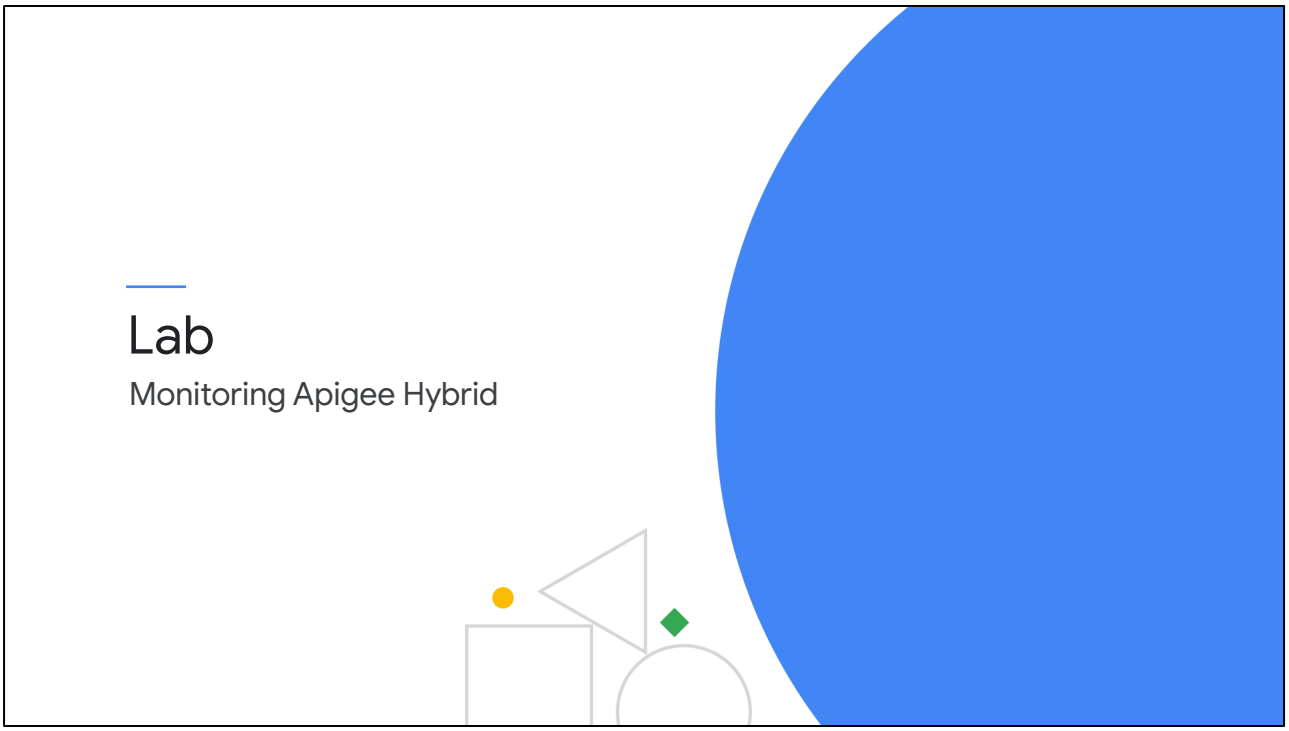
Analytics

Monitoring and Troubleshooting

Lab

Apigee Support

In this lab, you will learn about logging and monitoring for one of the components that run in the Apigee hybrid runtime plane.



You will learn how to use command line tools and Cloud Logging to view logs for the runtime components installed in your Apigee hybrid cluster.

You will also enable monitoring and alerting in Cloud Monitoring for the Cassandra runtime component.

In this lab, you learned how to use command line tools and Cloud Logging to view logs for the runtime components installed in your Apigee hybrid cluster.

You also enabled monitoring and alerting in Cloud Monitoring for the Cassandra runtime component.

Agenda

Logging

Metrics

Analytics

Monitoring and Troubleshooting

Lab

Apigee Support

Google Cloud

In this lecture we discuss the role of Apigee support and how you can request and receive support on Apigee hybrid.

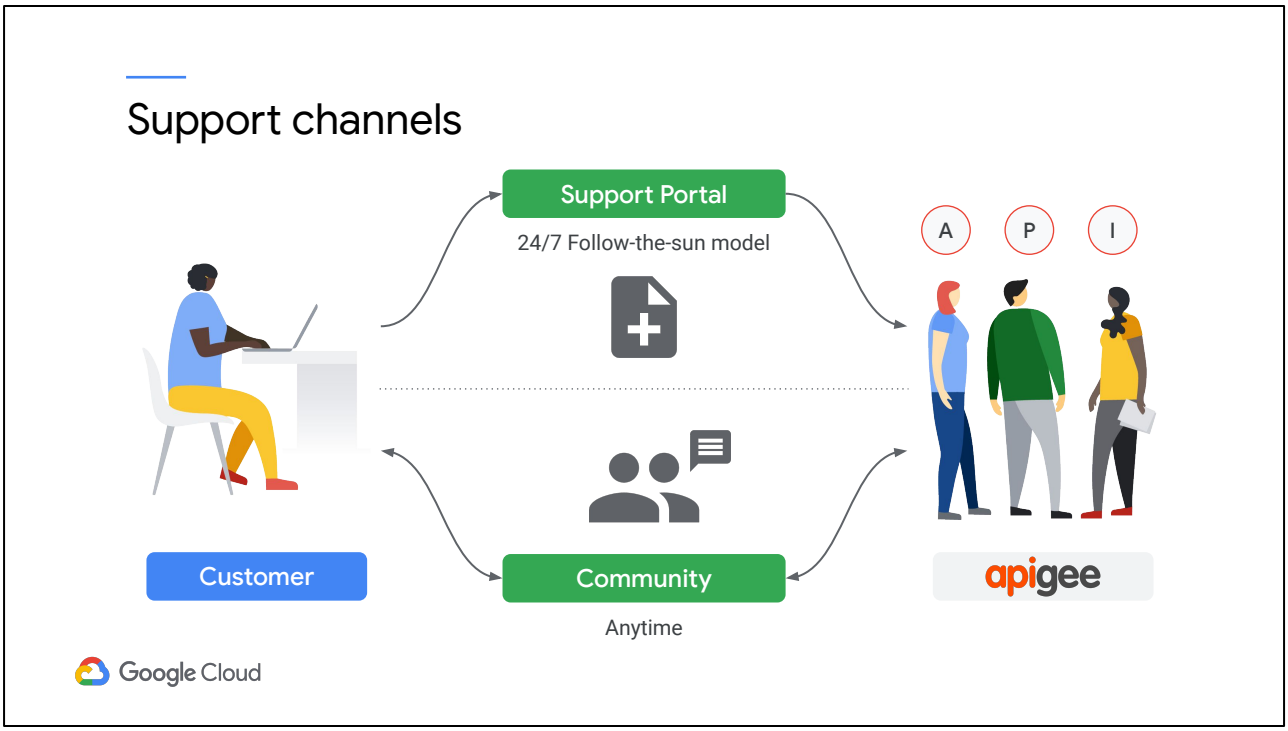
Global support and Cloud operations presence

- 7 Global Service Centers
  - US: Sunnyvale, Denver, Chicago
  - EU: London
  - APAC: Bangalore, Sydney, Tokyo
- 24/7 Follow-the-sun model

Google Cloud

Google Cloud has multiple service centers that provide continuous support for Apigee products.

You can receive support from any of these teams based on your location and time zone.

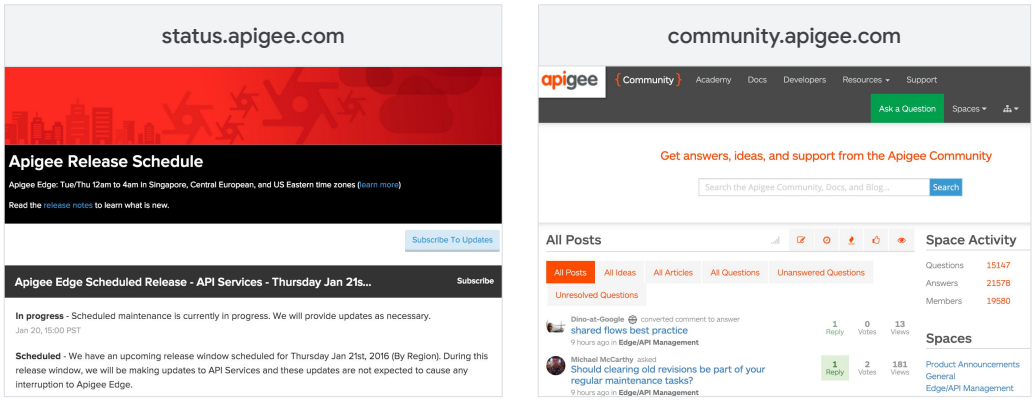


You can initiate support requests in the Apigee support portal by filing a support ticket.

In the widely used Apigee community portal, you can get answers to your technical questions on Apigee products and services.

Links to access these resources are provided in upcoming slides in this lecture.

Openness and transparency



Google Cloud

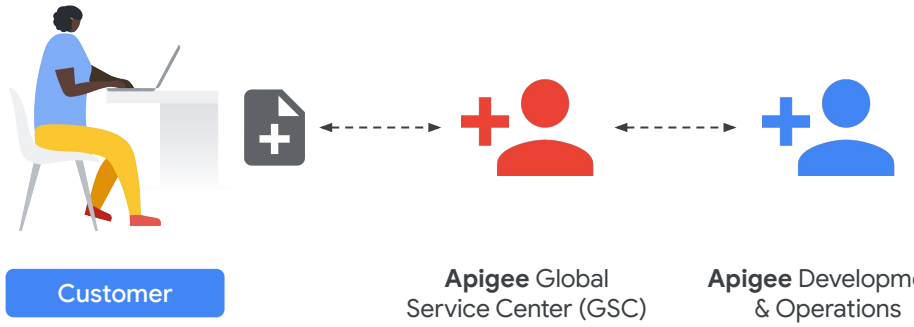
The Apigee status portal provides continuous updates on product releases, scheduled maintenance, and any service outages.

You can subscribe to receive email notifications from the status portal whenever Apigee creates, updates, or resolves an incident.

The Apigee community portal is an excellent resource for answers to technical questions on Apigee products and services.

You can also ask questions and receive answers from Apigee solution architects and other members of the community.

Speed of execution

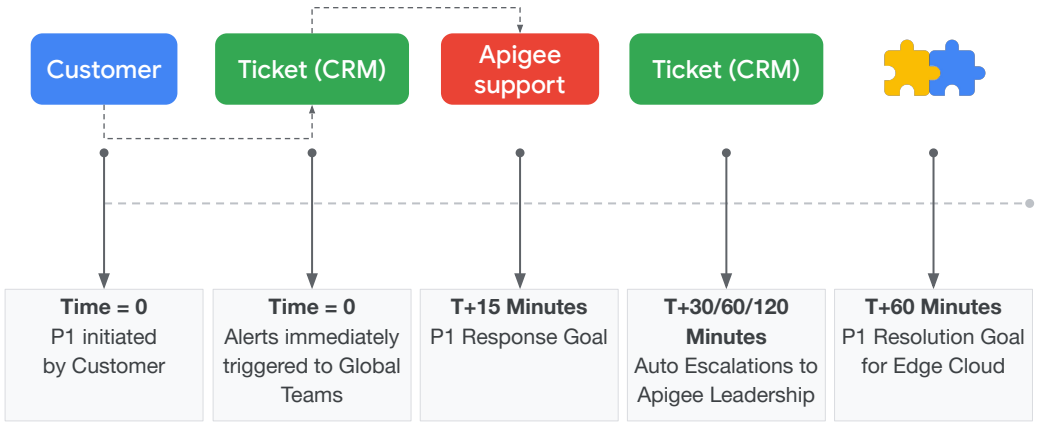


Google Cloud

When handling a support request, Apigee support personnel provide timely responses based on the priority of the request and your support plan.

They engage with Apigee product engineering and operations teams as needed to help resolve your issue.

Mission-critical support

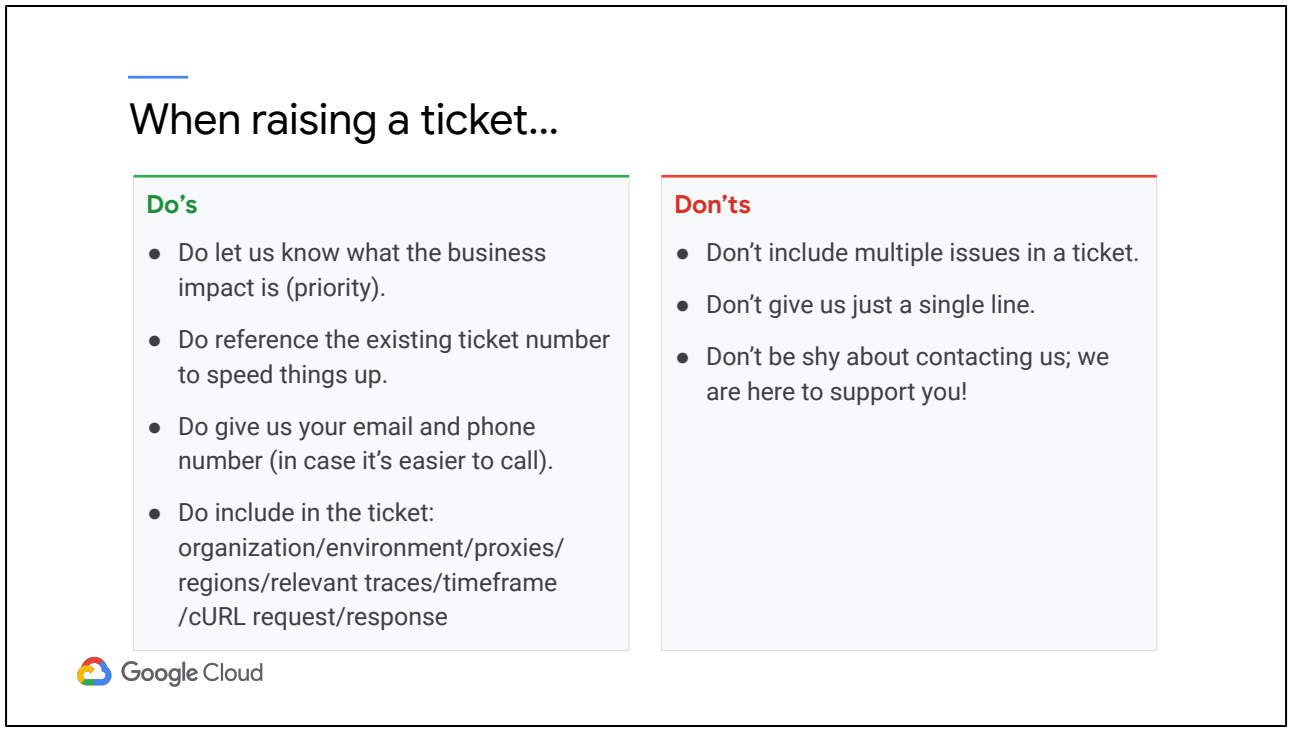


Google Cloud

Here is a sample timeline for a P1 priority support request with an enterprise support plan.

The initial response goal when a support ticket is created is 15 minutes, with a resolution goal of 60 minutes.

The enterprise and mission-critical support plans also include escalation of the support request to Apigee leadership.



Here are some best practices to follow when creating a support request.

You should always include the business impact of the issue, because it helps determine the priority of the request.

Reference any existing request ticket numbers and provide your contact information in the request.

Include all relevant technical information that will help the support team triage the issue.

Try not to combine multiple issues in the same request. Create separate support requests and provide relevant information as needed.

Apigee support services

Service	Apigee Enterprise Support
Scope of Support	Apigee product-specific break/fix questions
Product and Documentation Updates	Included
Severity 1 Issues Standard Hours of Operation	24/7 Follow-the-sun model
Severity 2 issues Standard Hours of Operation	Monday through Friday, 8 am to 5 pm (in customer's local time zone)
Method of Contact Support	Portal/Web
Root Cause Analysis for Severity 1 Incidents	Included*

- Apigee support spec-sheet: Apigee Edge Cloud Support Services
- Apigee support portal: Support | Apigee
- Apigee release notes: Apigee hybrid release notes
- Apigee release process: Apigee hybrid release process
- Supported software: Apigee hybrid supported platforms

* Not guaranteed for single-customer events and impact to non-Enterprise+ customers

Google Cloud

Apigee support has well-defined plans that include support services and performance goals.

This includes the scope of support provided, initial response and resolution goals, service request delivery, support channels, and methods of contact.

The links provided have full details on the scope of Apigee support services.

Apigee hybrid entitlements

- Apigee hybrid depends on the Kubernetes platform version as listed in the upgrade documentation.

Pricing plan	Hybrid	Max Orgs	Max Environments per Org	Anthos vCPUs
Standard	Not included	N/A	N/A	N/A
Enterprise	Included	25	75	300
Enterprise Plus	Included	25	75	800

- For the full list of Apigee entitlements by pricing plan, see https://cloud.google.com/apigee/pricing.
- In addition, refer to the Apigee hybrid product limits.

Google Cloud

Based on your pricing plan, Apigee currently includes hybrid entitlements.

Note that these entitlements can change with future product releases.

Apigee hybrid is only included in the Enterprise and Enterprise Plus pricing plans.

In addition, hybrid product limits are documented on the Apigee website.

These are technical limits on various hybrid entities and resources that are either currently enforced or planned to be enforced on the hybrid platform.

Links

Apigee Homepage
cloud.google.com/apigee

Documentation
https://cloud.google.com/apigee/docs

Resources
cloud.google.com/apigee/resources

Community
community.apigee.com
cloud.google.com/community

Training
cloud.google.com/training

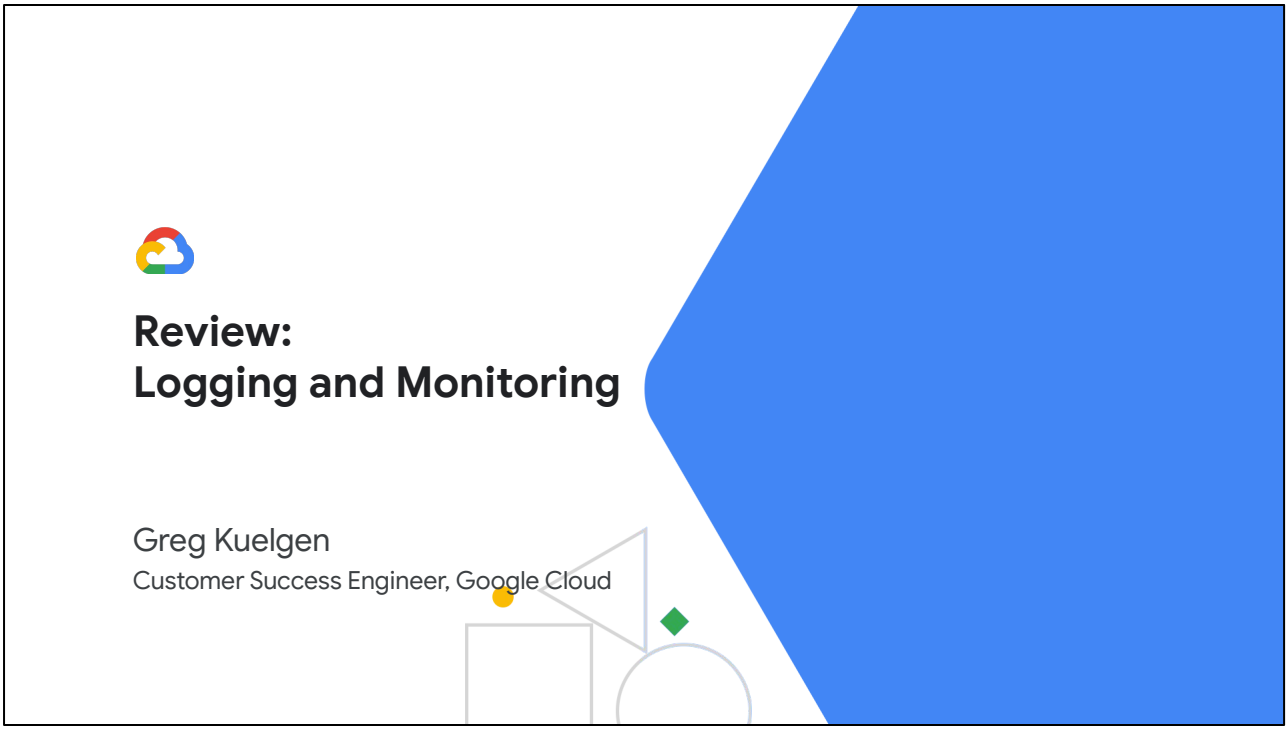
Events
cloud.google.com/events

Apigee API Platform Learning Guide
apigeek.page.link/learning-guide

Google Cloud

Many resources are available to you as you work with Apigee services.

Follow these links to learn more about Apigee products and services, read documentation, access training and certification materials, and learn about upcoming Google Cloud events.



In this module, you learned how Apigee hybrid logs system informational and error messages and how you can use analytics and metrics to monitor and troubleshoot Apigee hybrid.

You completed a lab on monitoring the hybrid runtime components using Cloud Monitoring.

You also learned about the Apigee hybrid support model and how you can request support for issues related to Apigee hybrid.