Welcome to “Introduction to Google Cloud Monitoring Tools.” In this module, we will take some time to do a high-level overview of the various products that constitute Google Cloud’s logging, monitoring, and observability suite.

Agenda

Overview of Google Cloud Monitoring Tools

Operations-Based Tools

Application Performance Management Tools

The core operations tools in Google Cloud break down into two major categories:

The operations-focused components—including Logging, Monitoring, Error Reporting, and Service Monitoring—tend to be more for personnel who are primarily interested in infrastructure, and keeping that infrastructure up, running, and error-free.

The application performance management tools, including Debugger, Trace, and Profiler, in contrast, tend to be more for developers who are trying to perfect or troubleshoot applications that are running in one of the Google Cloud compute products.

But it isn't fair to think of these tools as belonging purely to either of these two groups. A developer would, of course, sometimes need access to logs or monitoring metrics, just like an operation team member might need to trace latency.

For reference, the homepage for documentation related to this course can be found at: cloud.google.com/stackdriver/docs

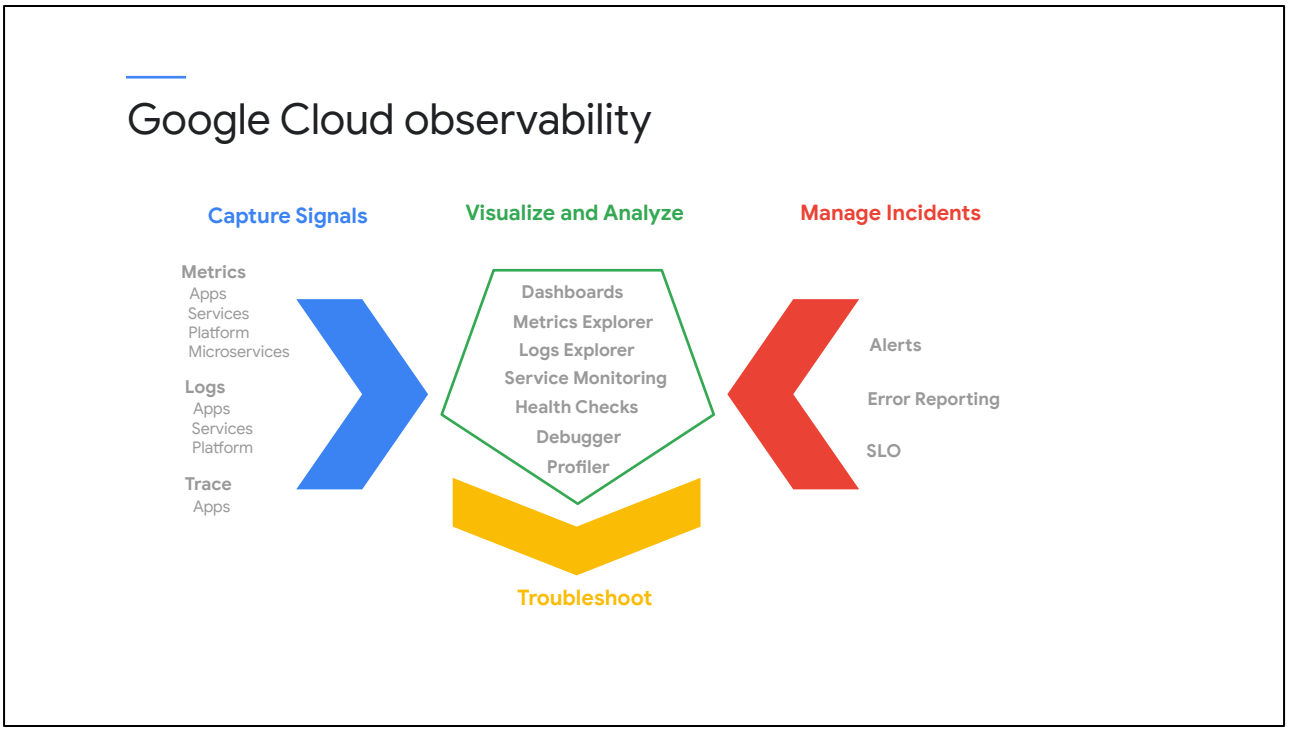
Agenda

Overview of Google Cloud Monitoring Tools

Operations-Based Tools

Application Performance Management Tools

Let’s start with an overview of why we need these tools, and then we’ll spend a little time getting to know both the operations and the application performance management products.



If you've ever worked with on-premises environments, you know that you, or someone in your organization, can actually physically touch any of your servers. If an application becomes unresponsive, someone can walk in and physically investigate.

In the cloud though, the servers aren't yours, they're Google's, and you won't be able to inspect them physically. So the question becomes, how do you know what's happening with your server, or database, or application? The answer is the tools discussed in this course.

It all starts with signals. Metric, logging, and trace data capturing is integrated into Google products from the hardware layer up.

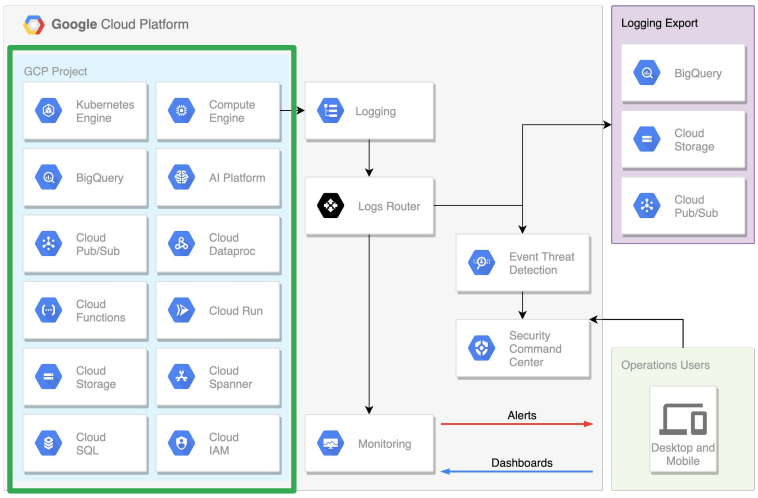
From those products, the signal data flows into the Google Cloud operation's tools, where you can visualize it in Dashboards and through the Metrics Explorer. You can dissect and analyze automated and custom logs in Logs Explorer. Monitor services for compliance with Service Level Objectives (SLOs), and track error budgets. Use Health Checks to check uptime and latency for external-facing sites and services. You can also debug and profile running applications.

When indicators pick up possible problems, signal data can generate notifications to code or, through various information channels, to key personnel. Error Reporting can help operations and developer teams spot, count, and analyze crashes in cloud-based services.

The visualization and analysis tools can then help troubleshoot what exactly is happening in Google Cloud.

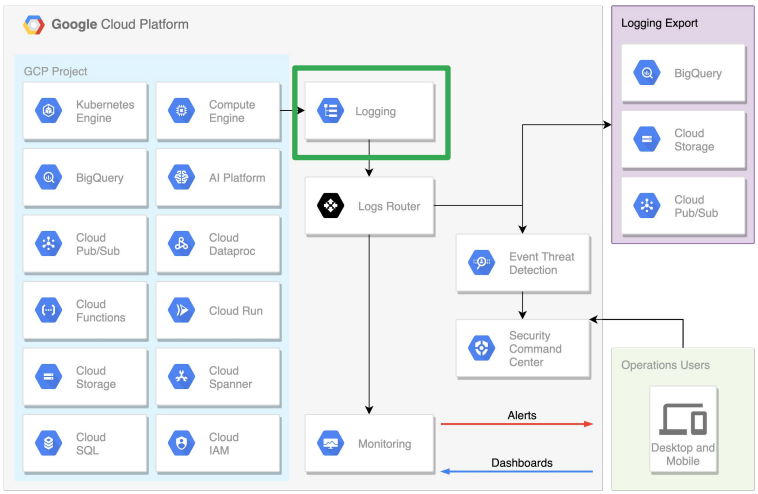
Ultimately, you won't miss that easy server access, because Google is going to allow you more precise insights into your Cloud install than you ever had on-premises.

Application and infrastructure observability



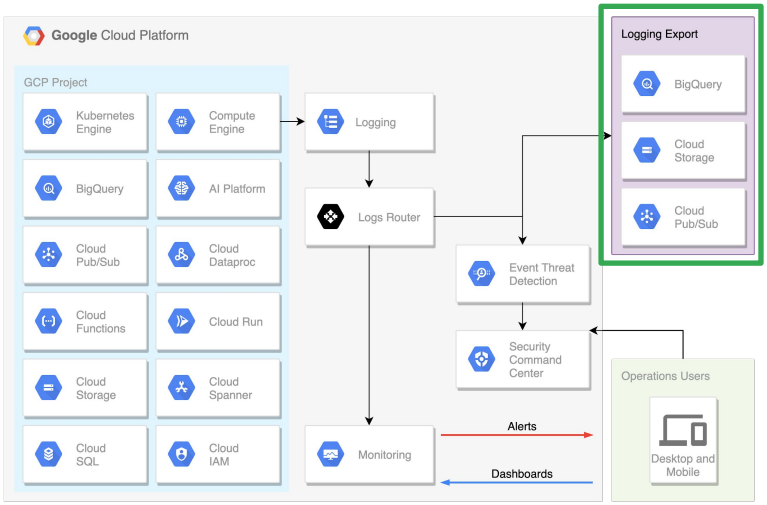
Google Cloud has many products, from Kubernetes, to BigQuery, to Spanner, and they all stream metrics and logs into Google's Cloud Logging and Cloud Monitoring components.

Application and infrastructure observability



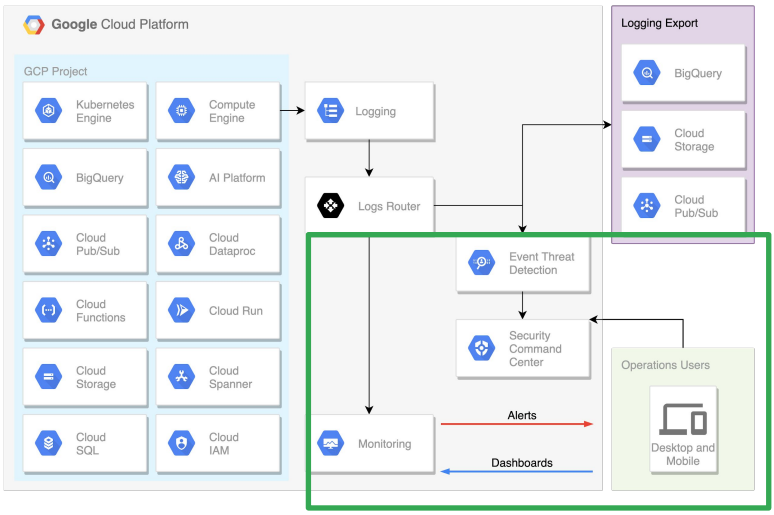
The Logs Router determines where the data goes and can be used to exclude some types of entries,

Application and infrastructure observability



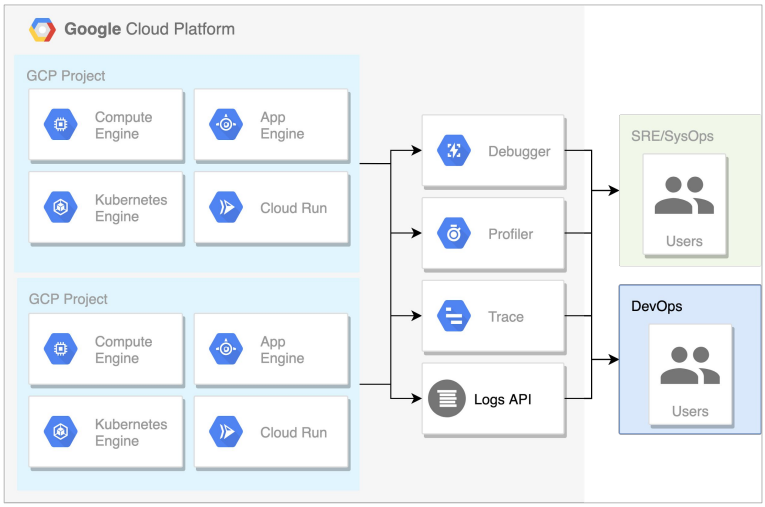
or to route logs to external locations like Pub/Sub or BigQuery, perhaps for automated handling and/or long term storage and analysis.

Application and infrastructure observability



An auditor might inspect Logs Viewer to see when a given Spanner instance was created and by whom. Security personnel could use Threat Detection or the Security Command Center to spot and analyze intrusion attempts. A network engineer might run SQL queries in BigQuery to better understand network flow.

Application performance management tools



In addition to raw monitoring metrics and logging entries, Google Cloud also helps SysOps/SRE and DevOps personnel analyze and improve application performance.

Take, as an example, an HTTP-based Java service running inside a Compute Engine VM.

Debugger would allow the inspection of the service's code state without stopping or degrading its performance. It helps answer the question, "What was happening in the code when this particular line executed?"

Similarly, Profiler can be used to examine CPU and memory use to help spot bottlenecks and improve algorithmic performance.

Trace is all about analyzing latency in a multi-layer, microservice application.

And the Logs API can be used by developers to write directly to Google Cloud logs.

Agenda

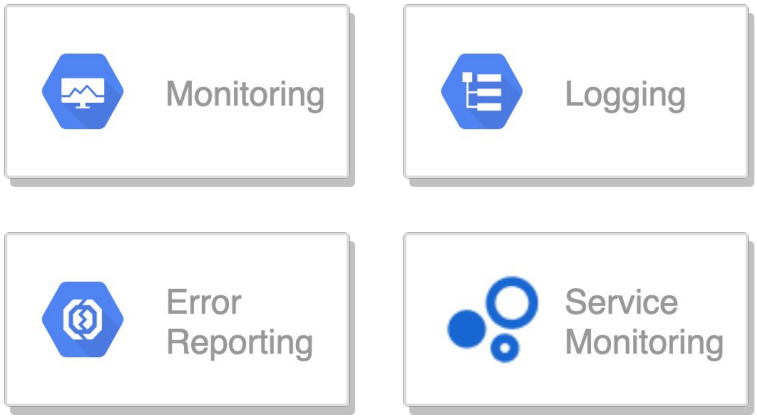
Overview of Google Cloud Monitoring Tools

Operations-Based Tools

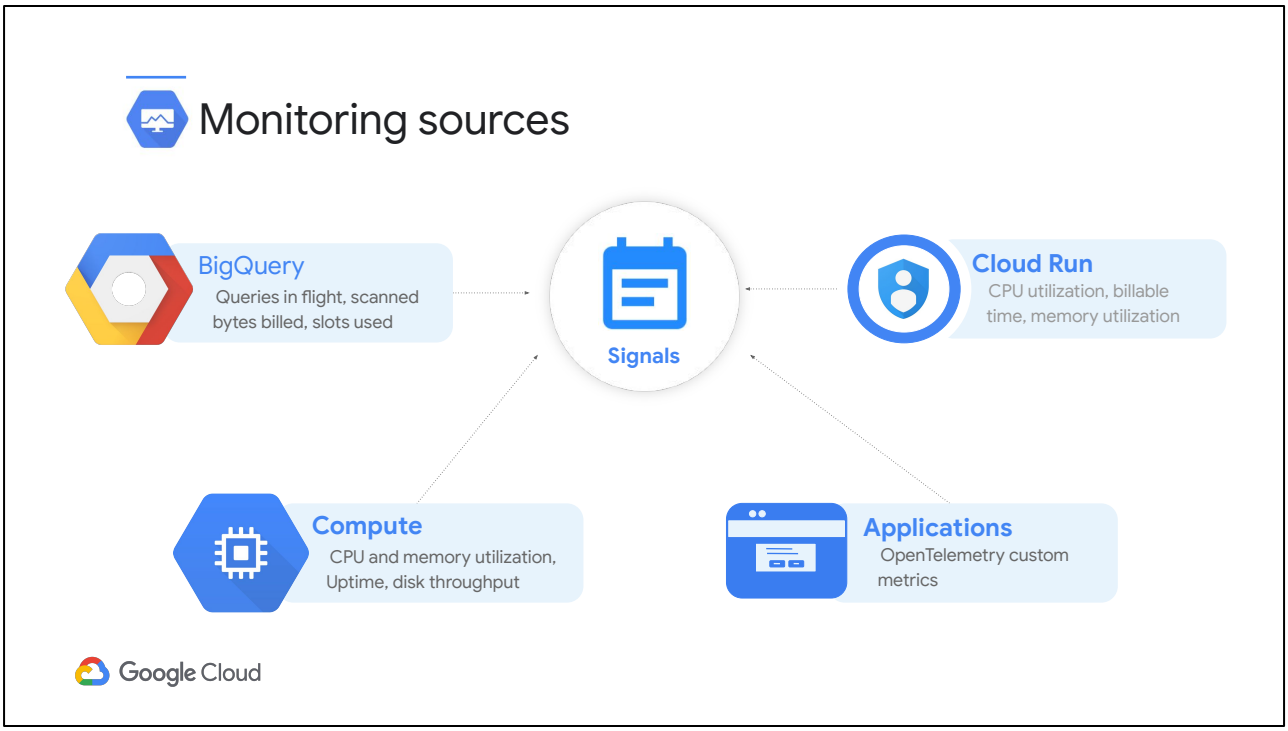
Application Performance Management Tools

Now that we've introduced the products in Google Cloud's logging and monitoring tool suite, let's take a closer look.

Operations-based tools



Let's start with the products that tend to be of interest for the operations folk: Monitoring, Logging, Error Reporting, and Service Monitoring.



When DevOps personnel think about tracking exactly what's happening inside Google Cloud Projects, Monitoring is often the first consideration. It's mentioned first on the documentation home page, just like the first product in the operations section of the Google Cloud navigation menu.

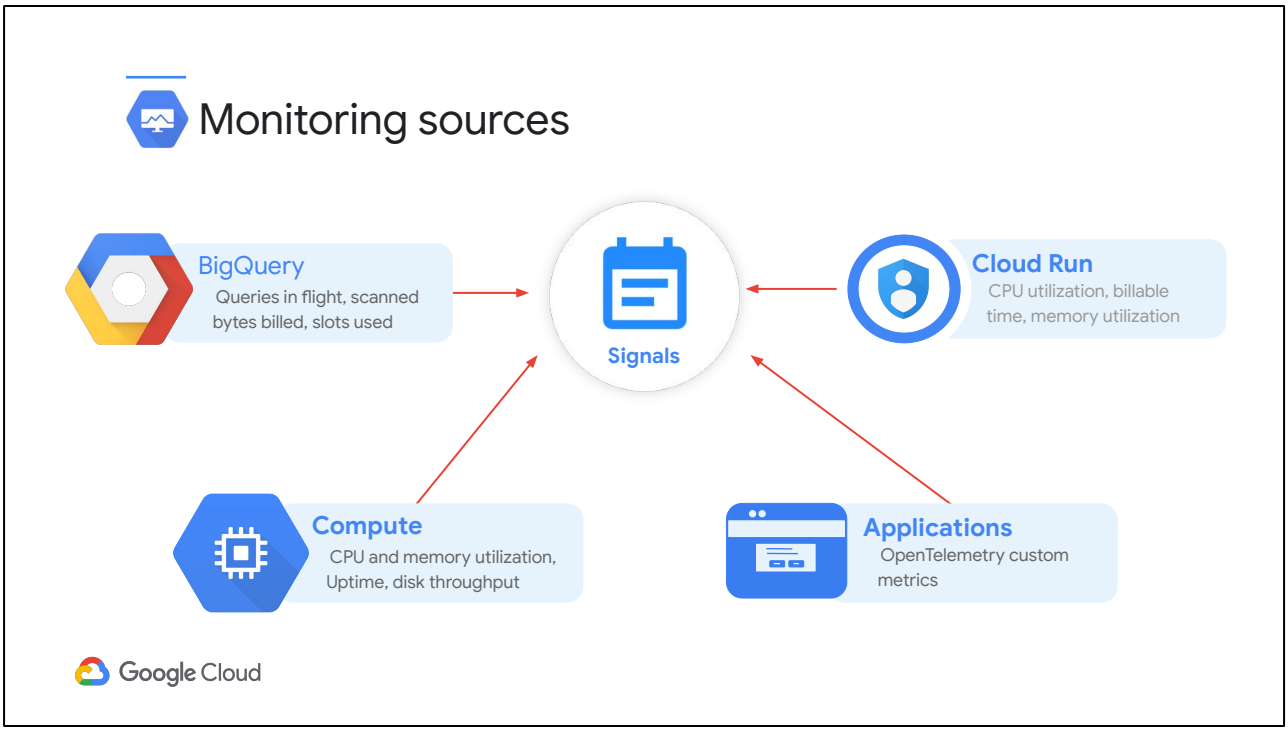
As we stated previously, monitoring starts with signal data.

When the data scientists are running massively scalable queries in BigQuery, knowing how many queries are currently in flight, how many bytes have been scanned and added to the bill, and data slot usage patterns, all will be important.

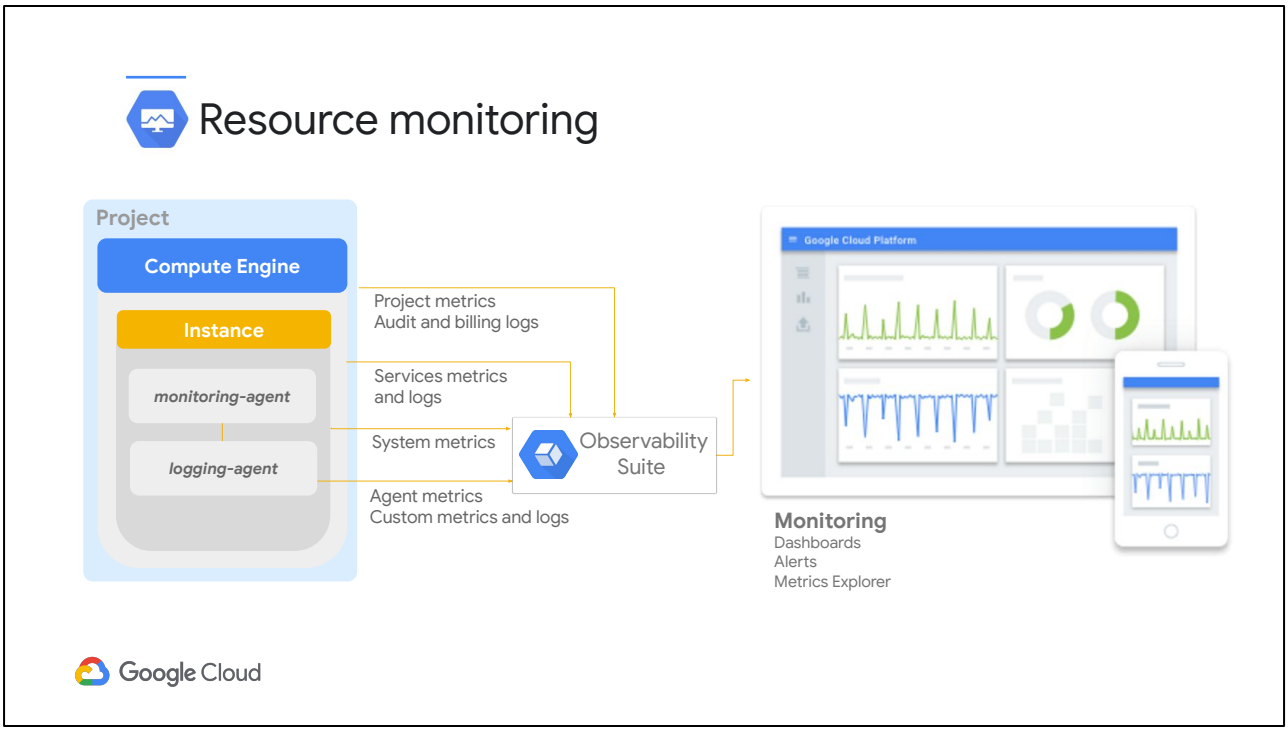
It could also be critical to DevOps teams running containerized applications in Cloud Run to know CPU and memory utilization, and app bill time.

Workloads on Compute Engine will benefit from CPU and memory utilization data, along with uptime, disk throughput, and scores of others.

And if those same DevOps teams want to augment the signal metrics coming out of their custom application wherever it's running, they could use the open-source OpenTelementry and create their own metrics.



Google Cloud, by default, collects more than a thousand different streams of metric data, which can be incorporated into dashboards, alerts, and a number of other key tools.

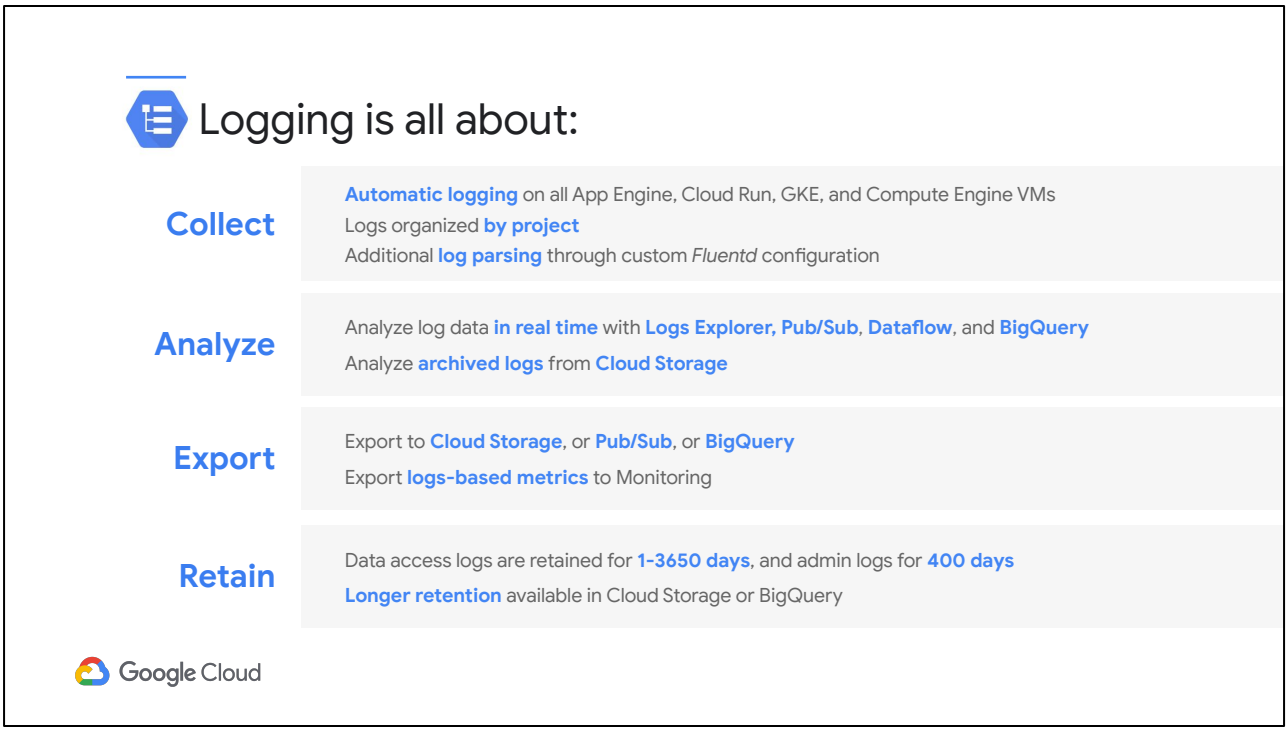


Here we see a project running a Compute Engine VM instance with the logging and monitoring agents installed.

Cloud Monitoring provides visibility into the performance, uptime, and overall health of cloud-powered applications.

It collects metrics, events, and metadata from projects, logs, services, systems, agents, custom code, and various common application components, including Cassandra, Nginx, Apache Web Server, and Elasticsearch.

Monitoring ingests that data and generates insights via dashboards, Metrics Explorer charts, and automated alerts.



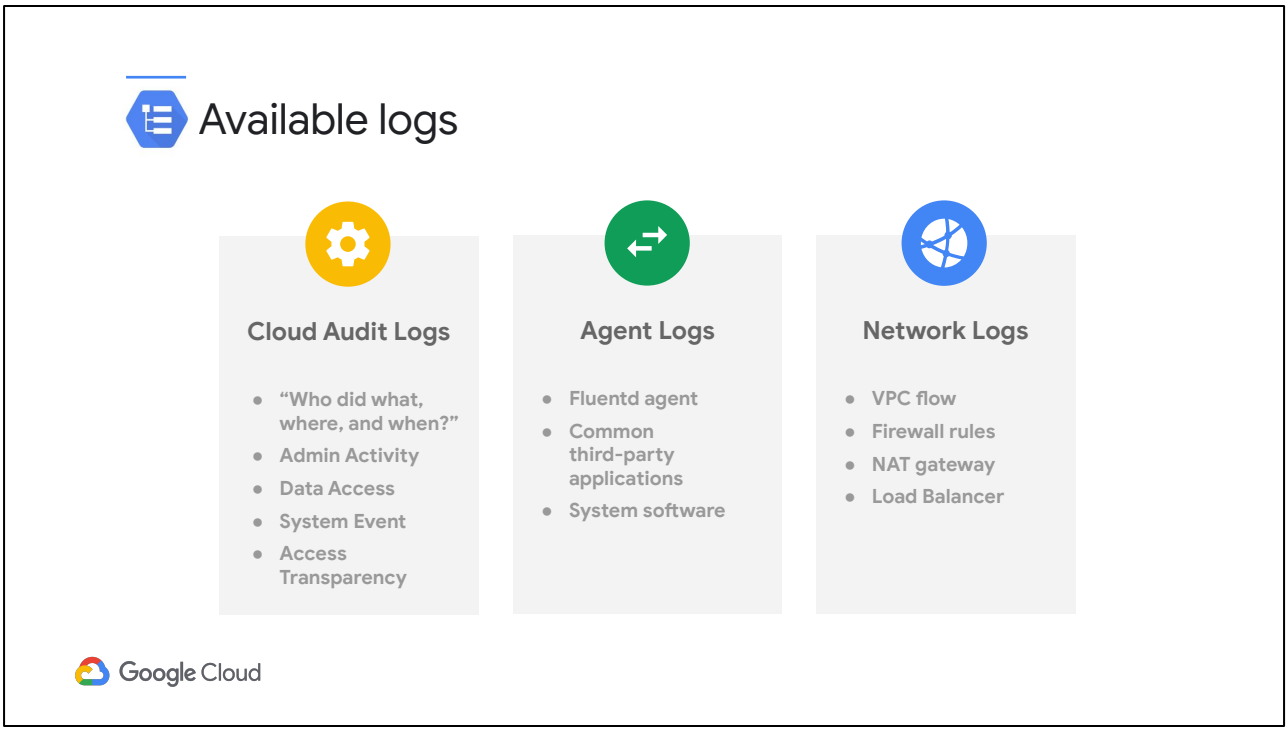
Google's Cloud Logging is all about collecting, storing, searching, analyzing, monitoring, and alerting on log entries and events.

Automated logging is integrated into Google Cloud products like App Engine, Cloud Run, Compute Engine VMs that run the logging agent, and GKE.

Export log data as files to Cloud Storage as messages through Pub/Sub or into BigQuery tables. Logs-based metrics may be created and integrated into Cloud Monitoring dashboards, alerts and service SLOs.

Pub/Sub messages can be analyzed in near real time using custom code or stream processing technologies like Dataflow. BigQuery allows analysts to examine logging data through SQL queries, and archive log files in Cloud Storage can be analyzed with several tools and techniques.

Default log retention in Cloud Logging depends on the log type. Data access logs are retained by default for 30 days, but this is configurable up to a max of 3650 days. Admin logs are stored by default for 400 days. You can export logs to Cloud Storage or BigQuery to extend retention periods.

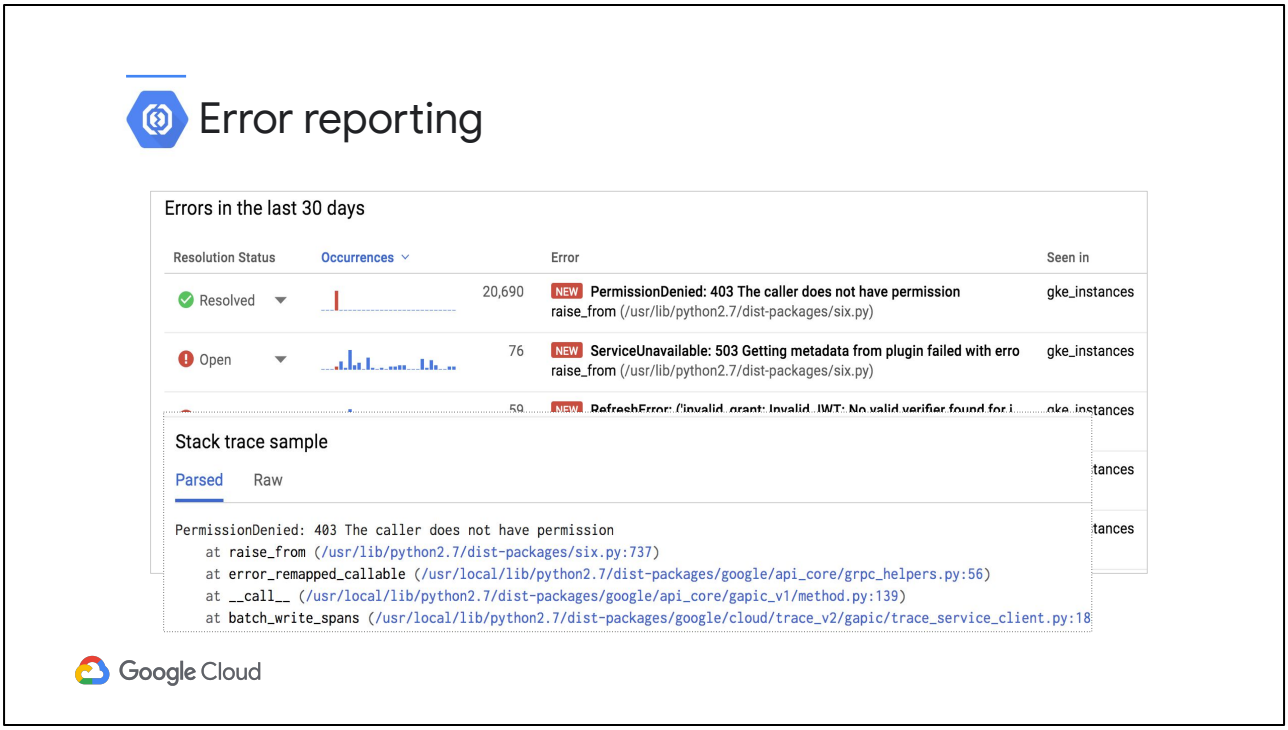


The Google Cloud platform logs that are visible to you in Cloud Logging vary, depending on which Google Cloud resources you're using in your Google Cloud project or organization.

Three key log categories are audit logs, agent logs, and network logs.

1. Cloud Audit Logs help answer the question, "Who did what, where, and when?" Admin Activity tracks configuration changes. Data Access tracks calls that read the configuration or metadata of resources, as well as user-driven calls that create, modify, or read user-provided resource data. System Events are non-human Google Cloud administrative actions that change the configuration of resources. Access Transparency provides you with logs that capture the actions Google personnel take when accessing your content.
2. Agent logs use a Google-customized and packaged Fluentd agent that can be installed on any AWS or Google Cloud VM to ingest log data from Google Cloud instances (for example, Compute Engine, Managed VMs, or Containers) as well as AWS EC2 instances.
3. Network logs provide both network and security operations with in-depth network service telemetry. VPC Flow Logs record samples of VPC network flow and can be used for network monitoring, forensics, real-time security analysis, and expense optimization. *Firewall Rules Logging* allows you to audit, verify, and analyze the effects of your firewall rules. NAT Gateway logs

1. capture information on NAT network connections and errors.



Error Reporting counts, analyzes, and aggregates the crashes in your running cloud services. Its management interface displays the results with sorting and filtering capabilities. A dedicated view shows the error details: time chart, occurrences, affected user count, first- and last-seen dates, and a cleaned exception stack trace. You can also create alerts to receive notifications on new errors.

Service monitoring

- Understand and troubleshoot intra-service dependencies
- Current support for App Engine, Anthos Service Mesh, and Istio
- Know when you're meeting or breaking SLOs
- Know when you have error budget to spend

Google Cloud

Service Monitoring helps organizations understand and troubleshoot intra-service dependencies.

With current support for App Engine, Anthos Service Mesh, and Istio, Service Monitoring streamlines the creation of latency and error-based SLOs, to make it easy for organizations to spot customer pain.

When SLOs are created, it also tracks and reports error budget burn, to make it clear to DevOps team members when there's budget to spend, and when things should be left alone.

Agenda

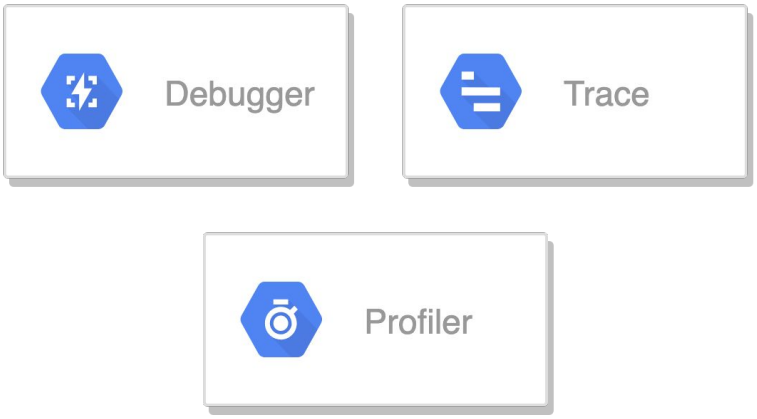
Overview of Google Cloud Monitoring Tools

Operations-Based Tools

Application Performance Management Tools

Now that we've explored the operations-based tools, let's spend a little time on the tools designed to help with application performance management, namely...

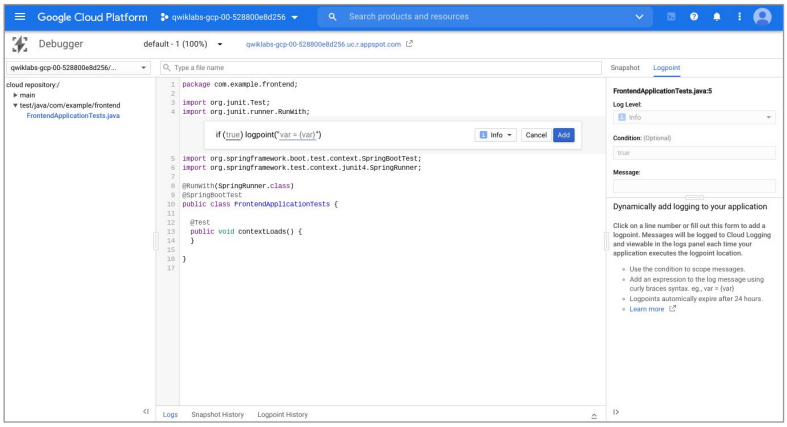
Application performance management tools



Debugger, Trace, and Profiler.

Debugger

- Real-time app debugging
- Increased collaboration by sharing debug sessions
- Debug snapshots, logpoints, conditional debugging
- Integrations with popular IDEs
- Multiple version control sources (GitHub, Cloud Source Repositories, Bitbucket, GitLab)



Google Cloud

Google Cloud's Debugger lets you debug your applications while they are running in production, without stopping them or slowing them down, so you can examine your code's function and performance under actual production conditions.

Easily collaborate with other team members by sharing your debug session. Sharing a debug session is as easy as sending the Console URL.

Capture the state of your application in production at a specific line location with snapshots. Use logpoints to inject a new logging statement on-demand at a specific line location. Capture a snapshot or write a logpoint message only when you need it, using a simple conditional expression written in your application's language.

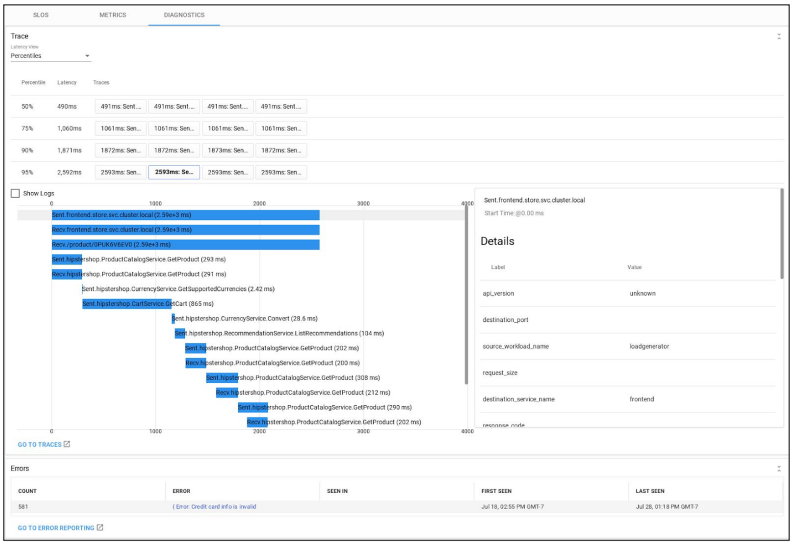
Cloud Debugger is easily integrated into existing developer workflows. Launch Debugger and take snapshots directly from Cloud Logging, error reporting, dashboards, IDEs, and the gcloud command-line interface.

And Debugger knows how to display the correct version of the source code because it easily integrates with version control systems, such as Cloud Source Repositories, GitHub, Bitbucket, or GitLab.

Trace

- Distributed latency analysis
- Support for App Engine, Compute Engine, and Google Kubernetes Engine
- Near-real-time performance insights
- In-depth latency reports
- Find performance degradations in apps
- Automatic issue detection

Google Cloud

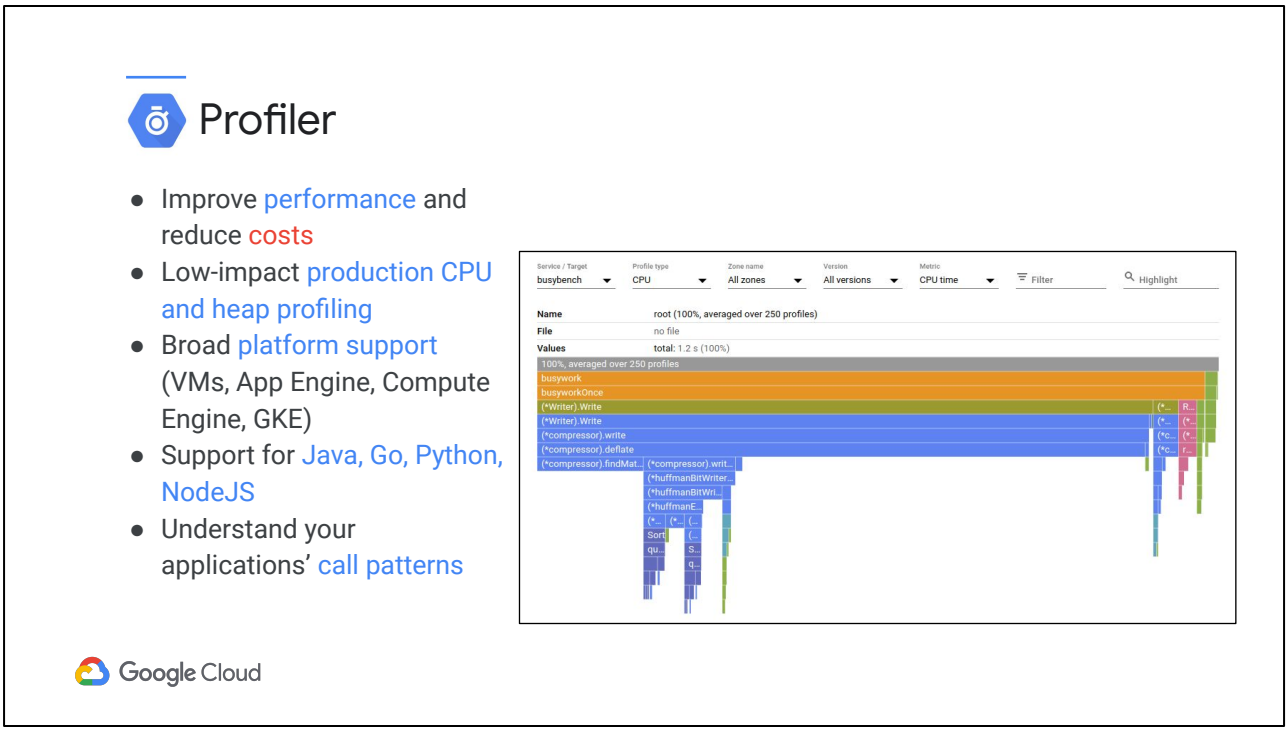


Based on the tools Google uses on its production services, Cloud Trace is a tracing system that collects latency data from your distributed applications and displays it in the Google Cloud Console.

Trace can capture traces from applications deployed on App Engine, Compute Engine VMs, and Google Kubernetes Engine containers.

Performance insights are provided in near-real time, and Trace automatically analyzes all of your application's traces to generate in-depth latency reports that can surface performance degradations.

Trace continuously gathers and analyzes trace data to automatically identify recent changes to your application's performance.



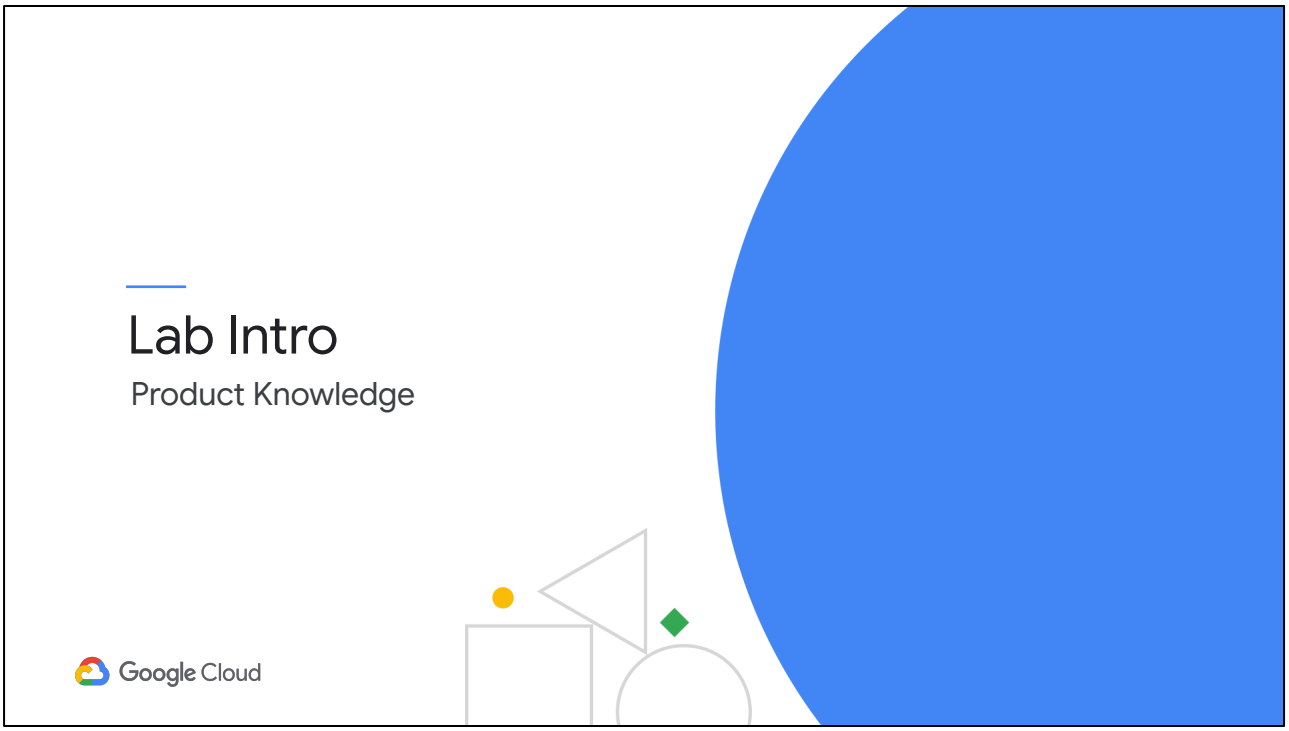
Poorly performing code increases the latency and cost of applications and web services every day, without anyone knowing or doing anything about it.

Cloud Profiler changes this by using statistical techniques and extremely low-impact instrumentation that can run across all production application instances to provide a complete CPU and heap picture of an application without slowing it down.

With broad platform support that includes Compute Engine VMs, App Engine, and Google Kubernetes Engine.

It allows developers to analyze applications running anywhere, including Google Cloud, other cloud platforms, or on-premises, with support for Java, Go, Python, and Node.js.

Cloud Profiler presents the call hierarchy and resource consumption of the relevant function in an interactive flame graph that helps developers understand which paths consume the most resources and the different ways in which their code is actually called.



In this lab, you will work to define technical and/or business benefits for various Google Cloud operations products. You'll then work through various scenarios and pick the best product for each.