Resource Monitoring

Essential Cloud Infrastructure: Core Services

STACKDRIVER, MONITORING, DEBUGGER, LOGGING, ERROR REPORTING, TRACE

RESOURCE MONITORING (STACKDRIVER), ERROR REPORTING AND DEBUGGING (STACKDRIVER)

Last modified 2018-1-29

Stackdriver overview

- Integrated monitoring, logging, diagnostics
- Manages across platforms
  - GCP and AWS
  - Dynamic discovery of GCP with smart defaults
  - Open source agents and integrations
- Access to powerful data and analytics tools
- Collaborations with third-party software

Stackdriver

Google Cloud

One key value of Stackdriver is that it integrates features and services that are otherwise multiple separate tools.

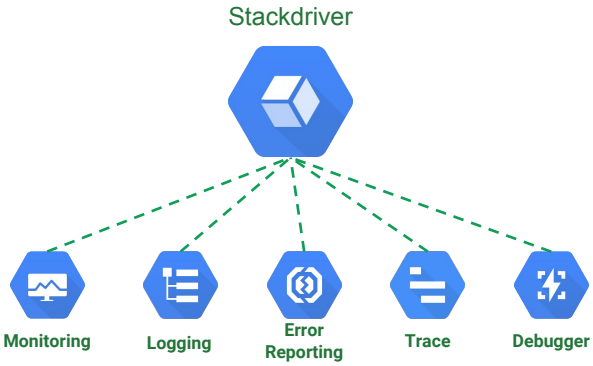
Discovers cloud resources and application services automatically based on deep integration with Cloud Platform and Amazon Web Services. Discovers and maintains relationships between key application components and identifies functional groups and clusters.

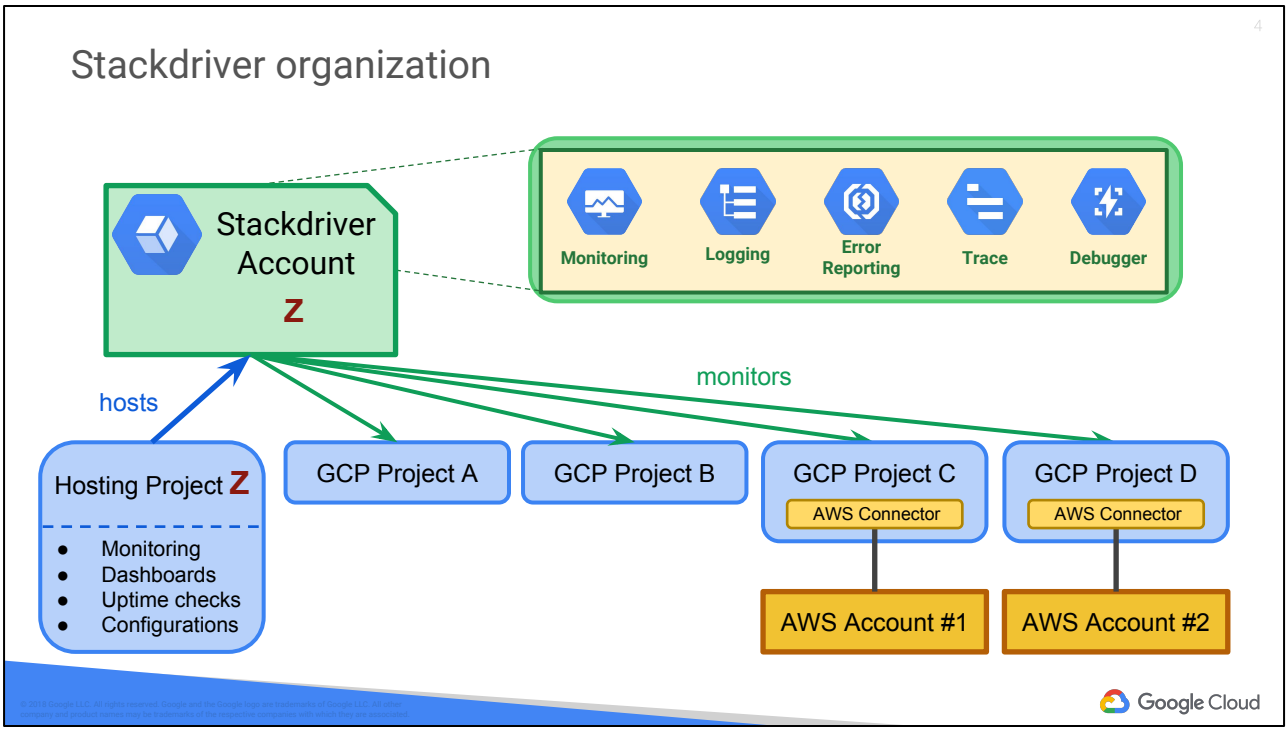
Thanks to Stackdriver's smart defaults, you can get up and running with core visibility into your cloud platform in less than two minutes. As you deploy Google's open-source agents, additional metrics and logs are automatically incorporated into the service.

Multiple integrated products

Benefits of Stackdriver:

- Monitors multi-cloud
  - GCP and AWS
- Identify trends and prevent issues
  - Charts
- Reduce monitoring overhead
- Improve signal-to-noise
  - Advanced alerting
- Fix problems faster
  - Alerts -> Dashboards -> Logs





A Stackdriver account is the root entity that holds monitoring and configuration information. You add projects into an account to monitor them.
You can have multiple accounts to organize separate systems and applications. However, all items that you want to monitor together should be in the same account.
To access an AWS account, you must configure a project in GCP to hold the AWS Connector. Each account or project can only be monitored by one Stackdriver account at a time.
A special project called the "hosting" project contains monitoring, configurations, dashboards, uptime checks, etc. The Stackdriver account acquires the name of the hosting account project (in the diagram, Project "Z" becomes Stackdriver Account "Z").
Stackdriver is available in two service tiers: "Basic" (free) and "Premium," which is charged by monitored resources in the account. AWS connections require the "Premium" service.

Partner integrations

BMC	Hybrid cloud security, compliance, and cloud lifecycle management suite
Splunk	Real-time security information and event management (SIEM) and operational intelligence platform
PagerDuty	Incident management solution provides view across GCP application development pipelines Reduces incident response times and improves reliability
Logentries (Rapid7)	SaaS service that provides centralized search, monitoring of log data, analytics, and security.
Tenable Network Security	Security continuous visibility and cross-platform context
Atlassian Hipchat	Team collaboration suite
Netskope	Cloud access security broker

Google Cloud

Google's growing Stackdriver partner ecosystem and tools make working with Stackdriver even easier.
https://cloud.google.com/stackdriver/partners

Agenda

- **Monitoring**
- Lab
- Logging
- Error Reporting
- Tracing
- Debugging
- Lab
- Quiz

Monitoring

- Dynamic config and intelligent defaults
- Platform, system, and application metrics
  - Ingests data: Metrics, events, metadata
  - Generates insights through dashboards, charts, alerts
- Uptime/health checks
- Dashboards
- Alerts

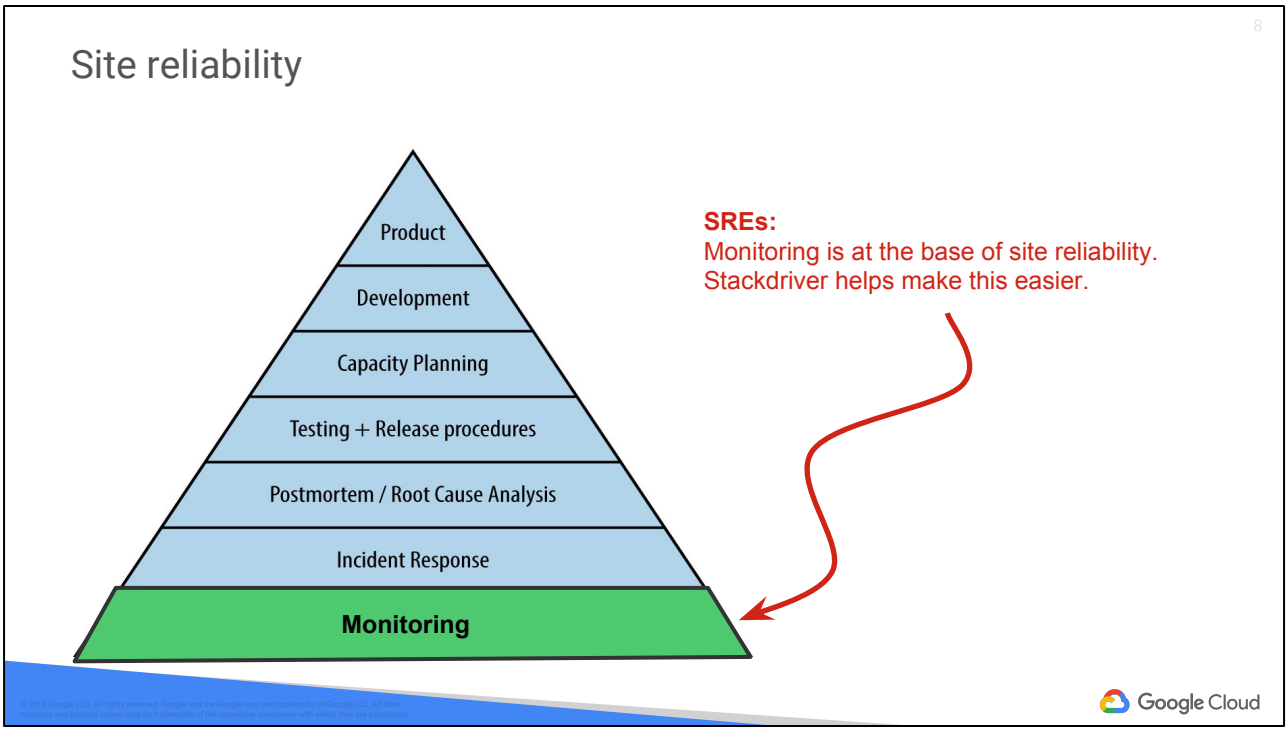
Monitoring

Google Cloud

Monitoring allows you to monitor your platform, system, and application metrics. You ingest data into cloud logging and you can create different metrics, custom events, monitor for specific metadata changes. You can also monitor and measure uptime and health checks.

Dynamic configuration: Configure monitoring tools dynamically (after resources deployed)

Intelligent defaults: easily create charts for basic monitoring activities



Site reliability engineering is a discipline that incorporates aspects of software engineering and applies that to operations whose goals are to create ultra-scalable and highly reliable software systems.

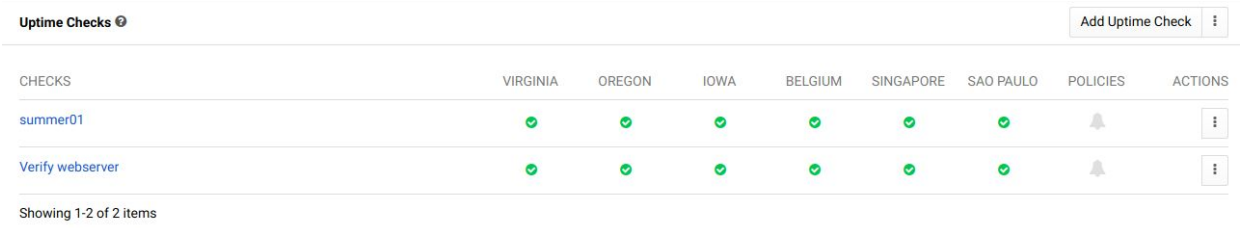
Site reliability starts at the core of any infrastructure. SREs are site reliability engineers: they are responsible for keeping the lights on at Google. SREs monitor, but they don't necessarily stare at a screen. There are certain reactive options that they take care of and they also try to automate responses. Their real core is to identify root cause analysis and identify how to test and re-release any types of fixes.

For more information, see: https://landing.google.com/sre/book.html

Platform, system, and application

- Platform
  - GCP and AWS
- System
  - Hosted probes
  - GCP Linux (native, minimal); Linux/Windows expanded (agent)
- Application instrumentation
- Common applications
  - Cassandra, Nginx, Apache Web Server, Elasticsearch ...

Uptime check overview



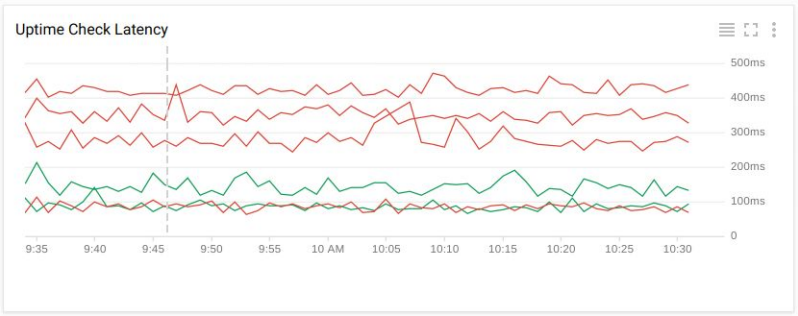
- Defaults to verifying from six global locations
- Global check of HTTP, HTTPS, TCP
- App Engine, Compute Engine, URLs ...

Google Cloud

When you make a change to the uptime checks, there can be a delay of up to 25 minutes before you see the new uptime check results.
During that time, the results of the former uptime check are displayed in the dashboard and is used in alerting policies.
In this image, summer01 is the name of a VM. A check was established to make sure that the VM was up. Another check was added to verify that a service was running, in this case a web server.

If "Verify webserver" interval is set to 1 minute, each location will check the site once a minute, so the server will see receive one request every 10 seconds.

Uptime check details



Uptime
100.000%

Outages
0 minutes

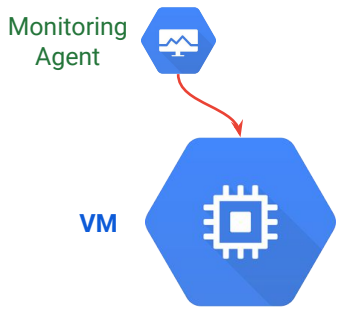
Location Results All locations passed

Check config

Check Type	HTTP
Resource	summer01
Path	/
Check Every	1 minute
Port	80
Locations	Global
Timeout	10 seconds

Google Cloud

Monitoring agent



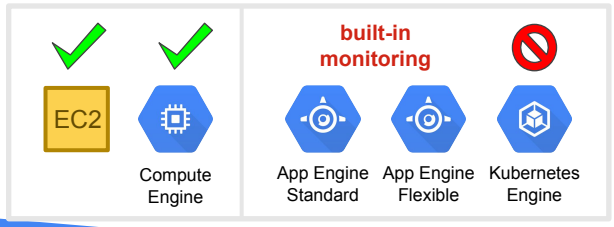
`collectd-daemon`

Default

- Disk, CPU, network, and process metrics

Application plugins

- *Curated* metrics



Apache web server
Cassandra
CouchDB
Elasticsearch
HBase
JVM Monitoring
Kafka
Memcached
MongoDB
MySQL
Nginx
PostgreSQL
RabbitMQ
Redis
Riak
Tomcat
Varnish
ZooKeeper

Google Cloud

The Monitoring agent is supported for EC2 VMs and GCP VMs. It adds the ability to connect application plugins and to collect more advanced metrics.

The Monitoring agent is not supported; however there is built-in monitoring support for App Engine in both Standard and Flexible environments.
The Monitoring agent is not currently supported for Kubernetes Engine.
Additionally, only specific OS's and versions are supported.
https://cloud.google.com/monitoring/agent/

Third-party applications
https://cloud.google.com/monitoring/agent/plugins/

Exhaustive metrics
https://cloud.google.com/monitoring/api/metrics#agent

Installing Monitoring agent

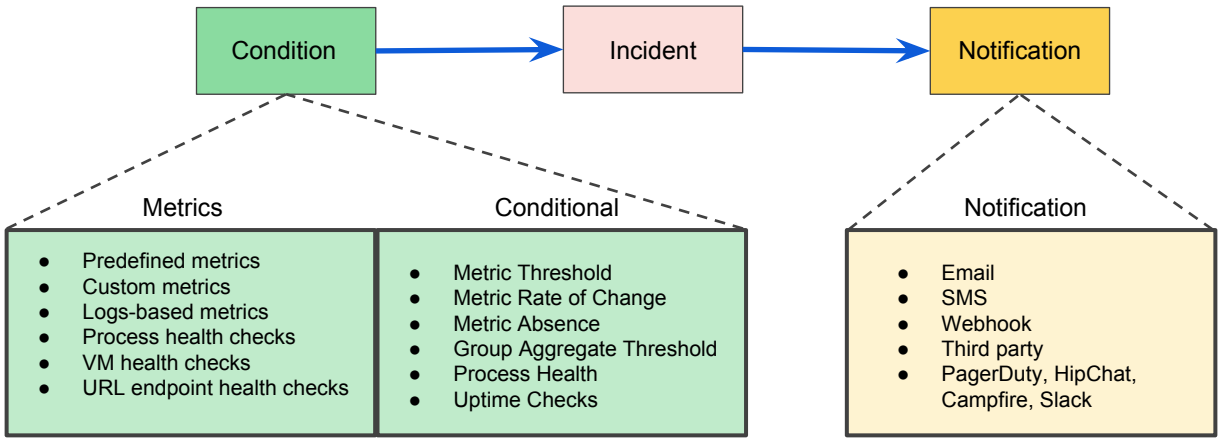
Install Monitoring agent

```
curl -O https://repo.stackdriver.com/stack-install.sh
sudo bash stack-install.sh --write-gcm
```


Google Cloud

Using the Monitoring agent is optional but recommended. Stackdriver Monitoring can access some metrics without the Monitoring agent, including CPU utilization, some disk traffic metrics, network traffic, and uptime information. Stackdriver Monitoring uses the Monitoring agent to access additional system resources and application services in virtual machine (VM) instances. If you want these additional capabilities, install the Monitoring agent.

Uptime monitoring



Google Cloud

Stackdriver Monitoring provides endpoint checks to web applications and other internet-accessible services running on your cloud environment. You can configure uptime checks associated with URLs, groups, or resources, such as instances and load balancers. Metrics are collected by Stackdriver periodically from a resource. In some cases the resource pushes its metric to Stackdriver, and in some cases Stackdriver pulls the metric from the resource. There is a delay before metrics become available in Stackdriver, called *metric latency*. For example, CPU utilization is gathered once a minute from a VM, and it can take 3 to 4 minutes for that metric to become available from a newly booted VM.

Examples of standard metrics

Compute Engine

```
firewall/dropped_bytes_count
firewall/dropped_packets_count
instance/cpu/reserved_cores
instance/cpu/usage_time
instance/cpu/utilization
instance/disk/read_bytes_count
instance/disk/read_ops_count
instance/disk/throttled_read_bytes_count
instance/disk/throttled_read_ops_count
instance/disk/throttled_write_bytes_count
instance/disk/throttled_write_ops_count
instance/disk/write_bytes_count
instance/disk/write_ops_count
instance/network/received_bytes_count
instance/network/received_packets_count
instance/network/sent_bytes_count
instance/network/sent_packets_count
instance/uptime
```

Cloud Storage

```
api/request_count
network/received_bytes_count
network/sent_bytes_count
```

Cloud Datastore

```
api/request_count
entity/read_sizes
entity/write_sizes
index/write_count
```

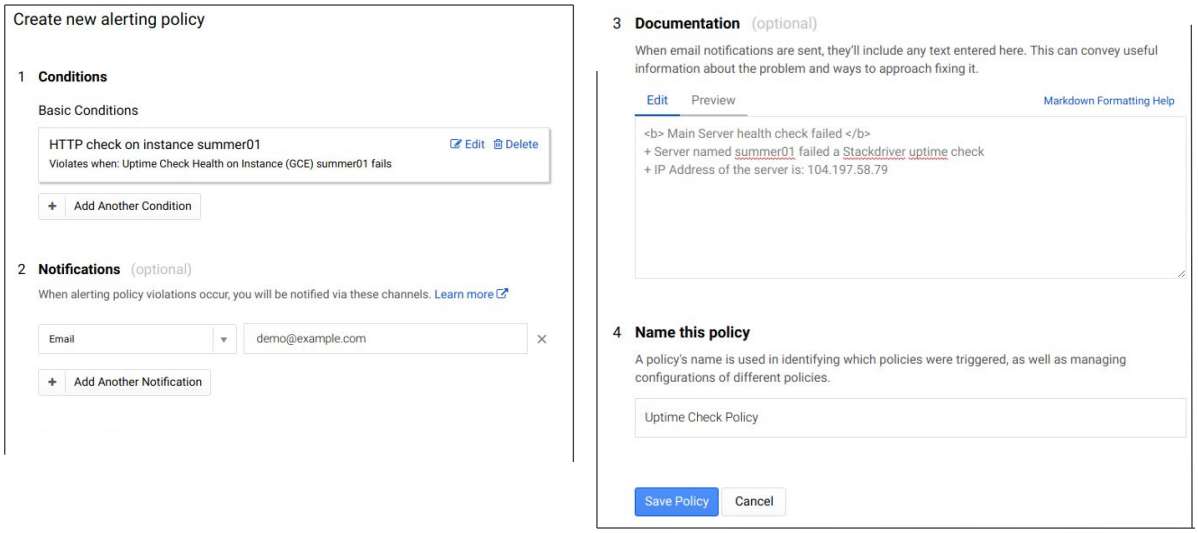
Cloud Pub/Sub

```
subscription/backlog_bytes
subscription/byte_cost
subscription/config_updates_count
subscription/mod_ack_deadline_message_operation_count
subscription/mod_ack_deadline_request_count
subscription/num_outstanding_messages
subscription/num_undelivered_messages
subscription/oldest_unacked_message_age
subscription/pull_ack_message_operation_count
subscription/pull_ack_request_count
subscription/pull_message_operation_count
subscription/pull_request_count
subscription/push_request_count
subscription/push_request_latencies
topic/byte_cost
topic/config_updates_count
topic/message_sizes
topic/send_message_operation_count
topic/send_request_count
```


Google Cloud

There are a lot of standard metrics, spanning Compute Engine, Cloud Pub/Sub, Cloud Storage and Cloud Datastore. For some examples, see https://cloud.google.com/monitoring/api/metrics#agent

Creating an alert



Google Cloud

Documentation associated with Alert makes it easier to scale and distribute monitoring responsibilities.

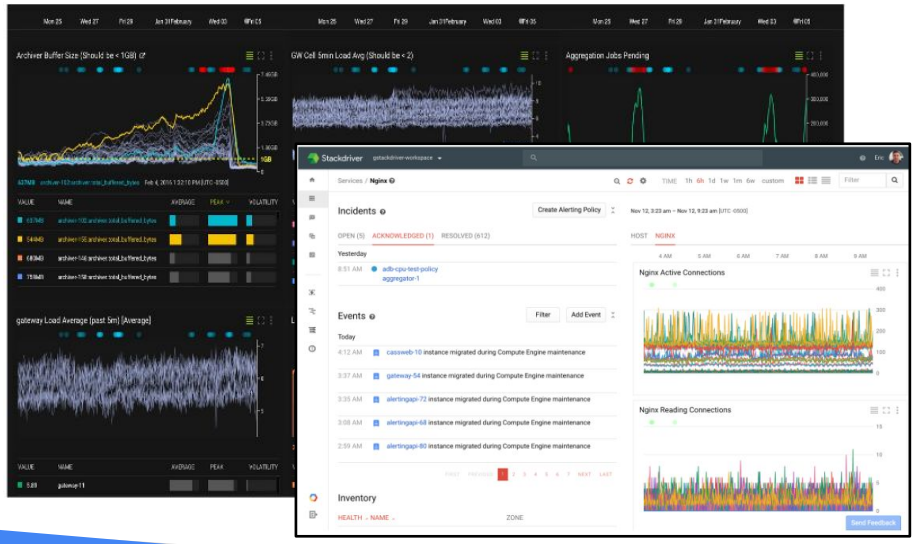
Groups

- Groups: aggregate metrics across a set of machines
  - Dynamically defined
  - Useful for high-change environments
- Separate production from development
- Filter Kubernetes Engine data by name and custom tags for cluster

Google Cloud

Groups can be created from the Stackdriver console: Groups -> Create Group
Dynamically defined:Because groups are filter-driven, any existing or future resources that match the criteria for the group will be automatically included. This means that you don’t have to update the group or related dashboards/alerts when you add or remove resources.

Dashboards



- No setup needed for GCP
- Quick visualization of core metrics for insights
- Customizable dashboards
- Auto-generated dashboards for common applications

Google Cloud

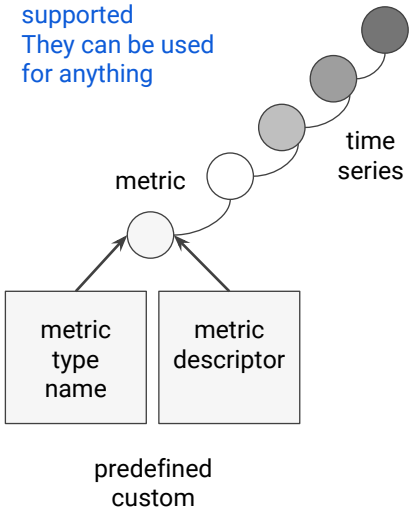
Custom metrics

This is a simple example of creating a custom metric in Python:

```
client = monitoring.Client()
descriptor = client.metric_descriptor(
    'custom.googleapis.com/my_metric',

metric_kind=monitoring.MetricKind.GAUGE,
    value_type=monitoring.ValueType.DOUBLE,
    description='This is a simple example
of a custom metric.')
descriptor.create()
```

- Complex metrics are supported
- They can be used for anything



Google Cloud

You can get metrics *into* Stackdriver by API calls or by using the agents.

Define custom metrics in the API

- Unique names
- Set time series
- Custom metrics be used for anything - reporting, charts …

https://cloud.google.com/monitoring/custom-metrics/creating-metrics#monitoring-create-metric-python

Stackdriver accounts

- Stackdriver accounts are a "single pane of glass," so monitor all your GCP projects in a single account.
- Determine your monitoring needs up front.
- Consider using separate Stackdriver accounts for data and control isolation.

Google Cloud

Why is it a best practice to use separate Stackdriver accounts?
Because the "single pane of glass" means that **all** Stackdriver users have access to **all** data by default.
It also means that a Stackdriver role assigned to one person on one project applies equally to all projects monitored by that account.

So to give people different roles per-project and to control visibility to data, consider placing the monitoring of those projects in separate Stackdriver accounts.

Alerts

- Alert on symptoms, not causes.
  - Example: monitor failing queries not database down
- Use multiple channels.
  - Avoid a single point of failure in your alert strategy.
  - Multiple notification channels, email, SMS …
- Customize alerts to audience needs.
  - Use descriptions to tell them what actions to take, what resources to examine.
- Avoid noise.
  - Adjust monitoring so alerts are actionable, not dismissable.

The slide describes “best practices”

Agenda

- Monitoring
- **Lab**
- Logging
- Error Reporting
- Tracing
- Debugging
- Lab
- Quiz

Lab: Resource Monitoring (Stackdriver)

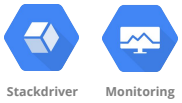
Objectives

In this lab, you learn how to perform the following tasks:

- Enable Stackdriver Monitoring
- Add charts to dashboards
- Create alerts with multiple conditions
- Create resource groups
- Create uptime checks

Completion: 60 minutes

Access: 120 minutes



Google Cloud

Lab Review

In this lab you learned how to:

- Monitor your projects
- Create a Stackdriver account
- Create alerts with multiple conditions
- Add charts to dashboards
- Create resource groups
- Create uptime checks for your services

Agenda

- Monitoring
- Lab
- **Logging**
- Error Reporting
- Tracing
- Debugging
- Lab
- Quiz

Logging

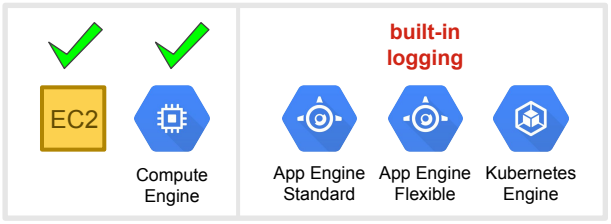
- Platform, system, and application logs
  - API to write to logs
  - 30-day retention + option to transfer to Cloud Storage
- Log search/view/filter
- Log-based metrics
- Monitoring alerts can be set on log events
- Data can be exported to BigQuery

Logging

Installing Logging agent

Install Logging agent

```
curl -sSO https://dl.google.com/cloudagents/install-logging-agent.sh
sudo bash install-logging-agent.sh
```



Google Cloud

Logging agent is based on `fluentd`

- **App Engine:** You do not need the agent.
- **Kubernetes Engine:** You do not need the agent.
- **Compute Engine:** Install the agent on your VM instances. The VM instances already have the required authorization. See Installing on Linux and Windows.
- **Amazon EC2:** Install authorization credentials on your VM instances, and then install the agent. See Authorizing the agent and then Installing on Linux and Windows.

For more information about the agent and the VM instances it supports, see Logging Agent.

Viewing and exporting

- Configure sinks for export
- Stackdriver logs must have access to the resource
  - Cloud Storage: storage and archiving
  - BigQuery: analysis
  - Cloud Pub/Sub: software integration
- Export process
  - As new log entries arrive, they are exported to sinks.
  - Existing log entries at the time the sink is created are not exported.
  - Cloud Storage entries are batched and sent out ~ hourly.

Google Cloud

As new log entries arrive in Stackdriver Logging, they are compared against the export sinks. If an entry matches a sink, it is exported to the destination. Log entries received before a sink is created will not be exported through that sink.

Log entries exported to Cloud Storage are batched and sent out approximately every hour. Log entries exported to BigQuery and Cloud Pub/Sub are streamed to those destinations immediately.

Exporting logs

- Retain data longer by exporting to different storage classes of Cloud Storage
- Search and analyze logs with BigQuery
- Advanced visualization with Cloud Datalab
- Stream logs to application or endpoint with Cloud Pub/Sub

Google Cloud

Logging

- Don't use substring matches on service names or resource types
- Faster search: search for specific values of indexed fields
  - Example: Log entry's name, resource type, resource label
- Advanced filter
  - More effective queries
  - Add to filter directly from log entries
- Learn the advanced viewing interface
  - Set up selections and filters in basic viewing interface
  - Then switch to advanced interface

Google Cloud

The slide describes best practices. As for not using substrings:

In the basic viewing interface, all searches are case-insensitive substring matches. That is, the searches text:abc or somefieldname:abc will match log entries containing abc, xyabcyx, and ABc. In advanced log filters, you must use the "has" search operator (:) for the same behavior.

For an exact match, use the equals operator (=). The comparison field=abc requires that field to contain exactly abc, in any letter case. That search cannot be expressed in the basic viewing interface.

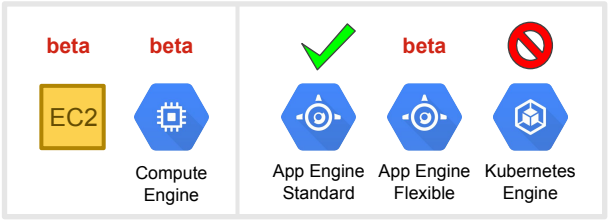
Agenda

- Monitoring
- Lab
- Logging
- **Error Reporting**
- Tracing
- Debugging
- Lab
- Quiz

Error Reporting

Aggregate and display errors for running cloud services

- Error notifications
- Error dashboard
- Java, Python, JavaScript, Ruby, C#, PHP, and Go



Error Reporting

Google Cloud

Stackdriver Error Reporting analyzes and aggregates the errors in your cloud applications. Notifies you when new errors are detected.
Error Reporting brings you the processed data directly to help you understand and fix the root causes faster.
Receive email or mobile alerts via mobile application.
Sign up for notifications on new errors

Agenda

- Monitoring
- Lab
- Logging
- Error Reporting
- **Tracing**
- Debugging
- Lab
- Quiz

Tracing

Tracing system
- Displays data in near real time
- Latency reporting
- Per-URL latency sampling

Collects latency data
- App Engine
- Google HTTPS load balancers
- Applications instrumented with the Stackdriver Trace SDKs

Tracing

Google Cloud

Stackdriver Trace provides latency sampling and reporting for App Engine, including per-URL statistics and latency distributions.
Tracing system that collects latency data from App Engine, Google HTTPS load balancers, and applications instrumented with the Stackdriver Trace SDKs, and displays it in near real time. Managing the amount of time it takes for your application to handle incoming requests and perform operations is an important part of managing overall application performance.

Agenda

- Monitoring
- Lab
- Logging
- Error Reporting
- Tracing
- **Debugging**
- Lab
- Quiz

Debugging

- Inspect an application without stopping it or slowing it down significantly
- App Engine Standard or Flexible and Compute Engine
- Java, Python, or Go
- Debug snapshots
  - Capture call stack and local variables of a running application
- Debug logpoints
  - Inject logging into a service without stopping it

Debugging

Google Cloud

Stackdriver Debugger connects your application's production data to your source code by inspecting the state of your application at any code location in production without stopping or slowing your requests. Use the production debugger to capture local variables and call stacks and link it back to a specific line location in the source code. Use that data to analyze the production state of your application and understand the behavior of your code in production or development.

Agenda

- Monitoring
- Lab
- Logging
- Error Reporting
- Tracing
- Debugging
- **Lab**
- Quiz

Google Cloud

Lab: Error Reporting and Debugging (Stackdriver)

Objectives

In this lab, you learn how to perform the following tasks:

- Launch a simple App Engine application
- Introduce an error into the application
- Explore Stackdriver Error Reporting
- Use Stackdriver Debugger to identify the error in the code
- Fix the bug and monitor in Stackdriver

Completion: 30 minutes

Access: 60 minutes

Lab Review

In this lab you:

- Deployed an application to App Engine
- Introduced a code bug and broke the application
- Used Stackdriver Error Reporting to identify the issue
- Analyzed the problem, finding the root cause using Stackdriver Debugger
- Modified the code to fix the problem, and then saw the results in Stackdriver

Agenda

- Monitoring
- Lab
- Logging
- Error Reporting
- Tracing
- Debugging
- Lab
- **Quiz**

Quiz

What is the foundational process at the base of Google's Site Reliability Engineering (SRE) ?

1. Capacity planning
2. Testing and release procedures
3. Monitering
4. Root cause analysis

Quiz

What is the purpose of the Stackdriver Trace service?

1. Reporting on latency as part of managing performance
2. Reporting on GCP system errors
3. Reporting on application errors
4. Reporting on GCP resource consumption as part of managing performance

Quiz

Stackdriver integrates several technologies, including monitoring, logging, error reporting, and debugging, that are commonly implemented in other environments as separate solutions using separate products. What are key benefits of integration of these services?

1. Reduces overhead, reduces noise, streamlines use, and fixes problems faster
2. Ability to replace one tool with another from a different vendor
3. Detailed control over the connections between the technologies
4. Better for GCP only so long as you don't need to monitor other applications or clouds

More resources

Monitoring
https://cloud.google.com/monitoring/docs/

Logging
https://cloud.google.com/logging/docs/

Error Reporting
https://cloud.google.com/error-reporting/docs/

Tracing
https://cloud.google.com/trace/docs/

Debugging
https://cloud.google.com/debugger/docs/

Google Cloud