

Developing, Deploying, and Monitoring in the Cloud

GCP Fundamentals: Core Infrastructure



Getting Started with Deployment Manager and Google Stackdriver



Google Cloud

Last modified 2018-08-13

© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

Agenda

Development in the cloud

Deployment: Infrastructure as code

Monitoring: Proactive instrumentation

Lab

Cloud Source Repositories

- Fully featured Git repositories hosted on Google Cloud Platform
- Supports collaborative development of cloud apps
- Includes integration with Stackdriver Debugger



 Google Cloud

Cloud Source Repositories provides Git version control to support collaborative development of any application or service, including those that run on App Engine and Compute Engine. If you are using the Stackdriver Debugger, you can use Cloud Source Repositories and related tools to view debugging information alongside your code during application runtime. Cloud Source Repositories also provides a source viewer that you can use to browse and view repository files from within the Google Cloud Platform Console.

With Cloud Source Repositories, you can have any number of private Git repositories, which allows you to organize the code associated with your cloud project in whatever way works best for you. Google Cloud diagnostics tools like the Debugger and Error Reporting can use the code from your Git repositories to let you track down issues to specific errors in your deployed code without slowing down your users. If you've already got your code in GitHub or BitBucket repositories, you can bring that into your cloud project and use it just like any other repository, including browsing and diagnostics.

Cloud Functions

- Create single-purpose functions that respond to events without a server or runtime
 - Event examples: New instance created, file added to Cloud Storage
- Written in Javascript; execute in managed Node.js environment on Google Cloud Platform



 Google Cloud

Cloud Functions is a lightweight, event-based, asynchronous compute solution that allows you to create small, single-purpose functions that respond to cloud events without the need to manage a server or a runtime environment. You can use these functions to construct applications from bite-sized business logic. You can also use Cloud Functions to connect and extend cloud services.

You are billed, to the nearest 100 milliseconds, only while your code is running.

Cloud Functions are written in Javascript and execute in a managed Node.js environment on Google Cloud Platform. Events from Cloud Storage and Cloud Pub/Sub can trigger Cloud Functions asynchronously, or you can use HTTP invocation for synchronous execution.

Cloud Events are things that happen in your cloud environment. These might be things like changes to data in a database, files added to a storage system, or a new virtual machine instance being created.

Events occur whether or not you choose to respond to them. Creating a response to an event is done with a trigger. A trigger is a declaration that you are interested in a certain event or set of events. You create triggers to capture events and act on them.

Agenda

Development in the cloud

Deployment: Infrastructure as code

Monitoring: Proactive instrumentation

Lab

Deployment Manager

- Infrastructure management service
- Create a `.yaml` template describing your environment and use Deployment Manager to create resources
- Provides repeatable deployments



Deployment Manager is an infrastructure management service that automates the creation and management of your Google Cloud Platform resources for you.

Setting up your environment in GCP can entail many steps: setting up compute, network, and storage resources and keeping track of their configurations. You can do it all by hand if you want to, taking an imperative approach. But it is more efficient to use a template. That means a specification of what the environment should look like, declarative rather than imperative.

GCP provides Deployment Manager to let do just that. It's an infrastructure management service that automates the creation and management of your Google Cloud Platform resources for you.

To use Deployment Manager, you create a template file, using either the YAML markup language or Python, that describes what you want the components of your environment to look like. Then you give the template to Deployment Manager, which figures out and does the actions needed to create the environment your template describes. If you need to change your environment, edit your template, and then tell Deployment Manager to update the environment to match the change.

Here's a tip: you can store and version-control your Deployment Manager templates in Cloud Source Repositories.

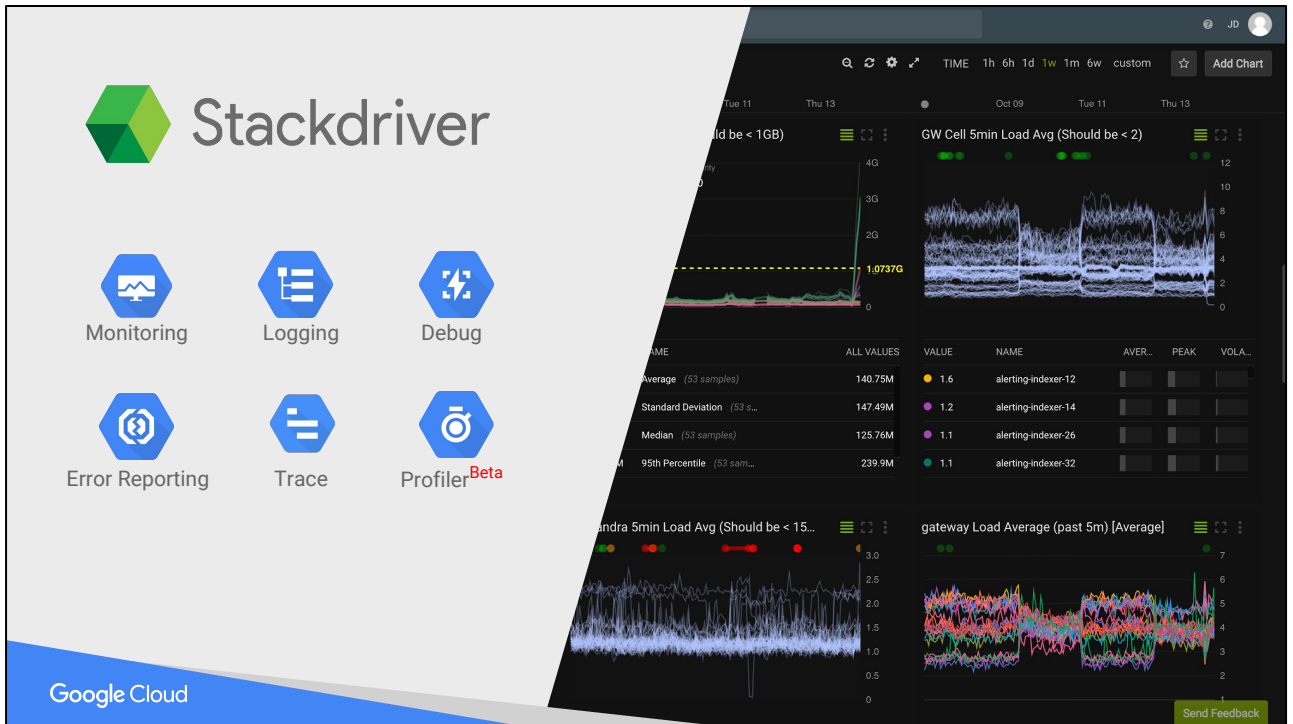
Agenda

Development in the cloud

Deployment: Infrastructure as code

Monitoring: Proactive instrumentation

Lab



You can't run an application stably without monitoring. Monitoring lets you figure out whether the changes you made were good or bad. It lets you respond with information rather than with panic when one of your end users complains that your application is down.

Stackdriver is GCP's tool for monitoring, logging, and diagnostics. Stackdriver gives you access to many different kinds of signals from your infrastructure platforms, virtual machines, containers, middleware, and application tier: logs, metrics, traces. It gives you insight into your application's health, performance, and availability, so if issues occur you can fix them faster.

Stackdriver offers capabilities in six areas

Monitoring



Platform, system, and application metrics

Uptime/health checks

Dashboards and alerts

Error Reporting



Error notifications

Error dashboard

Logging



Platform, system, and application logs

Log search, view, filter, and export

Log-based metrics

Debugger



Debug applications

Trace



Latency reporting and sampling

Per-URL latency and statistics

Profiler ^{Beta}



Continuous profiling of CPU and memory consumption

Here are the core components of Google Stackdriver: monitoring, logging, trace, error reporting, and debugging.

Stackdriver Monitoring checks the endpoints of 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. You can set up alerts on interesting criteria, like when health check results or uptimes fall into levels that need action. You can use Monitoring with a lot of popular notification tools. And you can create dashboards to help you visualize the state of your application.

Stackdriver Logging lets you view logs from your applications, and filter and search on them. Logging also lets you define metrics based on log contents that are incorporated into dashboards and alerts. You can also export logs to BigQuery, Cloud Storage, and Cloud Pub/Sub.

Stackdriver Error Reporting tracks and groups the errors in your cloud applications, and it notifies you when new errors are detected.

With Stackdriver Trace, you can sample the latency of App Engine applications and report per-URL statistics.

A painful way to debug an existing application is to go back into it and add lots of logging statements. Stackdriver Debugger offers a different way. It connects your application's production data to your source code, so you can inspect its state of your application at any code location in production. That means you can view the application state without adding logging statements. Stackdriver Debugger works best when your application's source code is available, such as in Cloud Source Repositories, although it can be in other repositories too.

To profile an application is to look at its execution of a program and observe the call patterns between functions, how much CPU time each function consumes, and how much memory is allocated for each. Stackdriver Profiler is a statistical, low-overhead profiler that continuously gathers CPU usage and memory-allocation information from your production applications. It attributes that information to the application's source code, helping you identify the parts of the application consuming the most resources, and otherwise illuminating the performance characteristics of the code. At the time of this writing, Stackdriver Profiler is in beta. This means that it is not covered by any SLA or deprecation policy and may be subject to backward-incompatible changes.

Agenda

Development in the cloud

Deployment: Infrastructure as code

Monitoring: Proactive instrumentation

Lab

Lab Instructions

In this lab you will create a deployment using Deployment Manager and use it to maintain a consistent state of your deployment. You will also view resource usage in a VM instance using Stackdriver.

- Create a Deployment Manager deployment.
- Update a Deployment Manager deployment.
- View the load on a VM instance using Google Stackdriver.

More resources

Cloud Source Repositories

<https://cloud.google.com/source-repositories/docs/>

Deployment Manager

<https://cloud.google.com/deployment-manager/docs/>

Google Stackdriver

<https://cloud.google.com/stackdriver/docs/>



© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.