# Development in Cloud

People create great applications in the Google Cloud.

Popular tools for development, deployment and monitoring just work in GCP. You also have options for tools that are tightly integrated with GCP and in this module, I'll explain them. Let's start with talking about development.

Lots of GCP customers use Git to store and manage their source code trees. That means, running their own Git instances or using a hosted Git provider. Running your own is great because you have total control. Using a hosted Git provider is great because it's less work. What if there were a third way?

Maybe a way to keep code private to a GCP project and use IAM permissions to protect it, but not have to maintain the Git instance yourself.

That's what **Cloud Source Repositories** is. It provides Git version control to support your team's development of any application or service, including those that run on App Engine, Compute Engine, and Kubernetes Engine.

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.

Cloud Source Repositories also contains a source viewer so that you can browse and view repository files from within the GCP console.

Many applications contain event-driven parts. For example, maybe you have an application that lets users upload images. Whenever that happens, you need to process that image in various ways: convert it to a standard image format, thumbnail into various sizes, and store each in a repository. You could always integrate this function into your application, but then you have to worry about providing compute resources for it, no matter whether it happens once a day or once a millisecond. What if you could just make that provisioning problem go away? It would be great if you could write a single purpose function that did the necessary image manipulations and then arrange for it to automatically run whenever a new image gets uploaded. That's exactly what Cloud Functions lets you do. You don't have to worry about servers or runtime binaries.

You just write your code in JavaScript for a Node.js environment that GCP provides and then configure when it should fire. There's no need for you to pay for servers either. You just pay whenever your functions run, in 100 millisecond intervals.

**Cloud Functions** can trigger on events in Cloud Storage, Cloud Pub/Sub, or in HTTP call.

Here's how setting up a Cloud Function works. You choose which events you care about. For each event type, you tell Cloud Functions you're interested in it. These declarations are called **triggers**. Then you attach JavaScript functions to your triggers. From now on, your functions will respond whenever the events happen. Some applications, especially those that have microservices architecture, can be implemented entirely in Cloud Functions. People also use Cloud Functions to enhance existing applications without having to worry about scaling.

# Deployment: Infrastructure as code

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. In other words, you figure out the commands you need to set up your environment the way you want. If you want to change your environment, you figure out the commands to change it from the old state to the new. If you want to clone your environment, you do all those commands again**. This is a lot of work. It's more efficient to use a template**. That means a specification of what the environment should look like**. It's declarative rather than imperative**. GCP provides **Deployment Manager** to let you 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 it,

1, 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.

2, Then, you give the template to Deployment Manager, which figures out and does the actions needed to create the environment your template describes.

3, 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.

Quiz:

What is the advantage of putting event-driven components of your application into Cloud Functions?

Your code executes whenever an event triggers it, no matter whether it happens rarely or many times per second. That means you don't have to provision compute resources to handle these operations

# Monitoring: Proactive instrumentation

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 and traces. It gives you insight into your application's health, performance and availability. So if issues occur, you can fix them faster.

Here are the core components of 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 PubSub.

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

Stackdriver Trace, you can sample the latency of app engine applications and report Per-URL statistics.

How about debugging? 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 applications production data to your source code. So you can inspect the state of your application at any code location in production. That means you can view the application stage without adding logging statements. Stackdriver Debugger works best when your application source code is available, such as in Cloud Source repositories. Although it can be in other repositories too.

# Lab

$export MY\_ZONE=us-center1-a

$echo $DEVSHELL\_PROJECT\_ID

$nano mydeploy.yaml

resources:

-name: my-vm

$sed –i –e ‘s/PROJECT\_ID/’$DEVSHELL\_PROJECT\_ID/ mydeploy.yaml

$sed –i –e ‘s/ZONE/’$MY\_ZONE/ mydeploy.yaml

$nano mydeploy.yaml

$gcloud deployment-manager deployments create my-first-depl \

--config mydeploy.yaml

$gcloud deployment-manager deployment list

Go to VM install to verify the deployment

To change the deploy.yaml

Nano mydeploy.yaml

“;apt-get install nginx-light –y”

Rerun following:

$gcloud deployment-manager deployments create my-first-depl \

* --config mydeploy.yaml

Go to VM install to verify the deployment

To change the deploy.yaml

$ dd if=/dev/urandom | gzip -9 >> /dev/null&

Use stackdriver to monitor:

Create account/skip AWS setup /launch monitoring

Quiz1: Why might a GCP customer choose to use Deployment Manager?

Deployment Manager is an infrastructure management system for GCP resources.

Quiz: Which statements are true about Stackdriver Logging? Choose all that are true (2 statements)

* Stackdriver Logging lets you view logs from your applications, and filter and search on them.
* Stackdriver Logging lets you define metrics based on your logs.