Get unlimited access to the best of Medium for less than $1/week. Become a member

A (very) brief overview of Google Cloud Platform

Introduction and terminology for new users to start digging into GCP

Alexander Weston, PhD · Follow
Published in Towards Data Science
7 min read · May 2, 2022

Listen Share More

Ebene bei Auvers mit Regenwolken by Van Gogh. From Wikimedia Commons.

Recently, our statistics department started transitioning away from local servers to Google Cloud Platform (GCP). Probably many of you find yourselves in a similar situation.

What I've found over two years of using GCP is that there's surprisingly little help available online for new users learning Cloud for the first time. I thought it might be helpful to write a brief article underlining some of the core concepts of Cloud.

One disclaimer — I've adapted this from documentation written for our summer interns because I thought it might be useful for a wider audience. We use a Cloud environment that was built by our own IT department with additional security appropriate for a large academic health institution. Therefore, several features of the platform are modified or disabled (for example, the ability to request your own DLVMs).

If you also happen to work for the Three Shields, then you've come to the right article.

What is Cloud?

Cloud refers to a platform where another company managers your server, versus you managing the server yourself. The reason it's called cloud (a confusing term, in my opinion) is because you're accessing a distributed network of hundreds or thousands of servers located elsewhere, like a "cloud" of servers.

If interested, I really enjoy this video tour of a Google Cloud Center, which for us is located in South Carolina.

It's worth mentioning that Google Cloud Platform is the same tool that all Google employees, including data scientists, use internally for their own research.

What are the advantages of Cloud?

The biggest advantage, which is driving adoption at many large research centers, is the cost advantage. But even as an "end-user" we've found several advantages to Google Cloud Platform.

Unlimited data storage is what originally drew us to Cloud. We were constantly struggling with server storage issues. I now have over thirty terabytes of data dedicated just to my personal projects.

No need to share resources. Another struggle is too many people (or one greedy person…) running large computing jobs that slow everybody down. With GCP, this is no longer an issue.

Newer tools. Many of the tools we use, such as Tensorflow, were developed (or at least better managed) by Google. Having access to the latest pre-built packages is much easier than attempting to set up packages yourself.

Self-managed environment. An individual Cloud DLVM is typically more customizable than a shared server space would be, and having the freedom to configure our own individual environments has been a welcome change.

If none of these factors impact your daily work, I will add that Google has given us access to many of the tools that we need to deploy our own models. So if you're writing code that you hope others may be able to use, there will be a long-term advantage to using Cloud.

A brief word on security

Data security, especially at an academic health center, is rightfully a major concern. Google Cloud Architects have worked alongside our own IT department to develop a secure platform that cannot be accessed even by Google employees.

I will add that Google has much more experience with data security than myself, and I would much rather leverage their own platform than try to build one myself.

Cloud Resources and Terminology

This is intended to be a very brief, very basic overview of many of the concepts and terms you will encounter using Cloud. Many of these topics I intend to cover more fully in future articles.

Projects are a group of users who share storage and computing together, it is the basic level of organization in Cloud, similar to a shared server. Our Project consists of the fifteen members of our research lab.

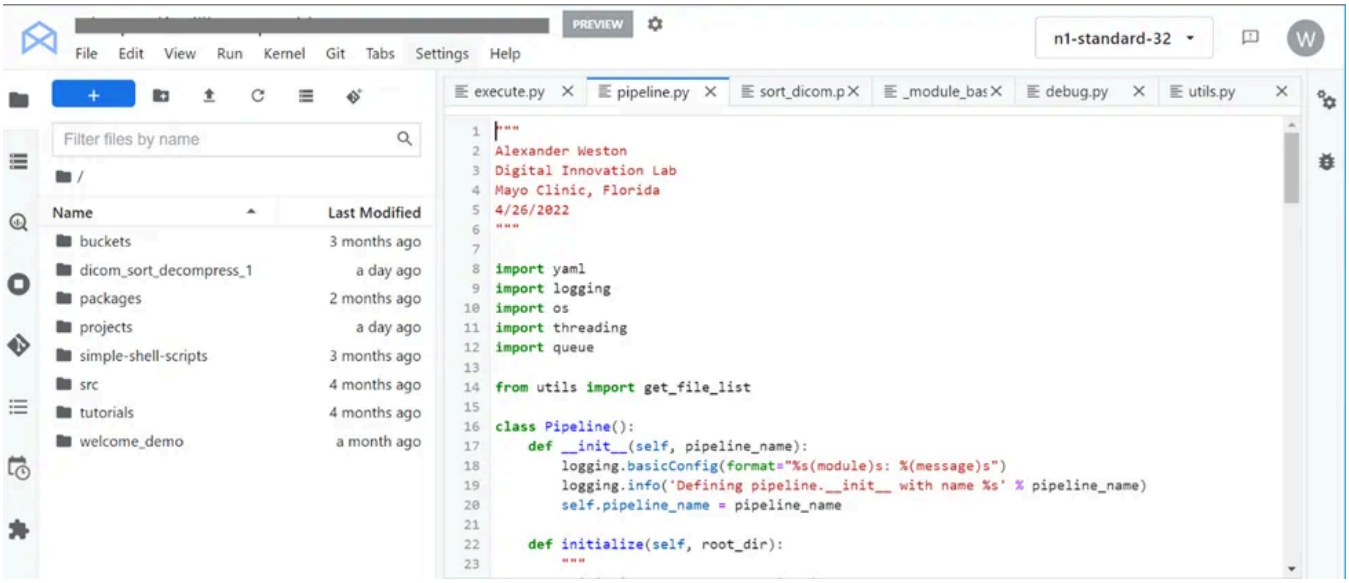
Each project has a unique project-ID, which appears at the top of the web interface. You will need to know this to log into the Platform.

Deep-Learning Virtual Machines, or *DLVMs*, are the basic unit of Cloud. For those who have used Azure or AWS, a DLVM is exactly the same as a *Virtual Machine*.

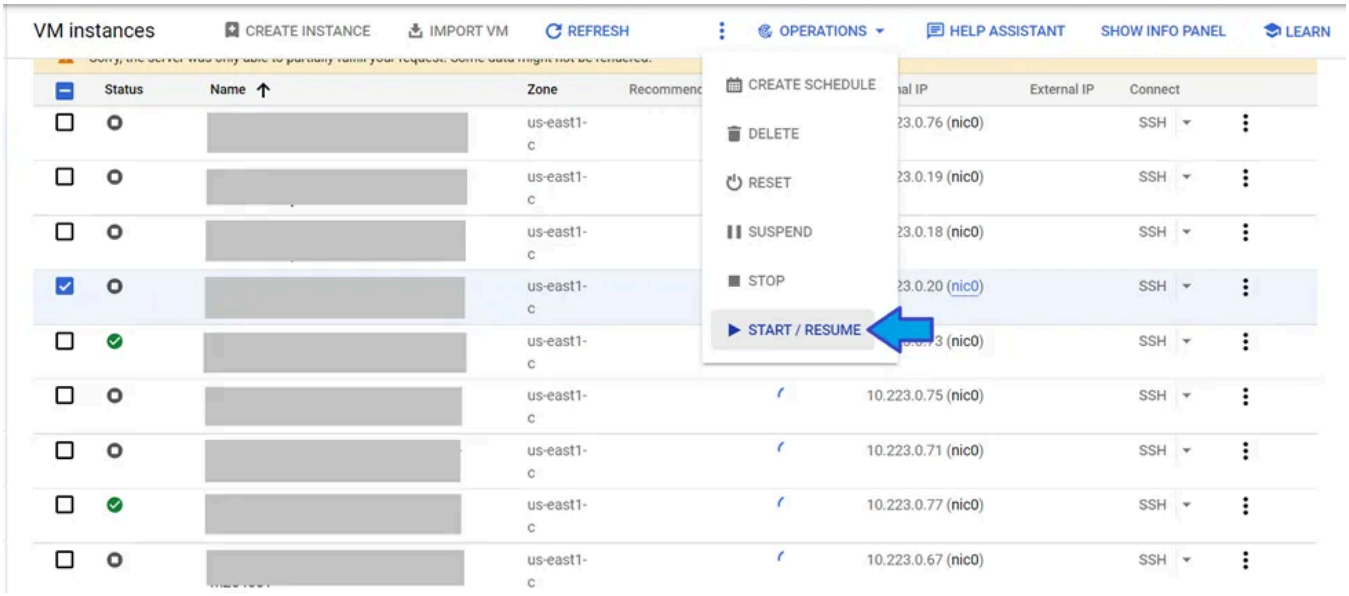
A DLVM is analogous to your personal computer. DLVMs are not intended to be shared; each DLVM is assigned to one user (although each user may have as many DLVMs as they wish). Each DLVM also has a unique ID which is used to log in.

DLVMs are almost infinitely customizable in terms of computing options. I have found that the *n1-standard-16* configuration, which has 16 CPU cores and 60GB of memory, is perfect for most all my work. I also add 2x T4 GPUs for training basic deep-learning models. This has a cost of approximately $4/hr.

You only pay for the DLVM when it's turned on. The cost is the same whether the DLVM is being used or sitting idle, so we do ask everyone to please be mindful and turn off their DLVMs at the end of the day.



An example of the JupyterLab interface of Google Cloud Platform. Image by author.



An example of the “Compute Engine” menu which shows the DLVMs assigned to our Project, and the options to Start/Stop the DLVM. I have unfortunately had to gray out the name of the DLVMs, a typically naming scheme consists of **projectName-zone-userID**. Image by author.

The Storage Bucket is the main place to store and share files. Unlike a Linux server where storage and compute are inseparable, the Storage Bucket is in a different location from the DLVMs, which is an aspect of the platform that can be confusing at first.

What this means, when you log into your DLVM, the files in the Storage Bucket will not be immediately visible to you. There is an easy way to solve this by fuse-mounting the bucket to the DLVM, which makes all the files visible.

Once the bucket has been fuse-mounted, it is indistinguishable from DLVM storage and can be accessed with the same commands. We like to include this command in a startup script so it runs automatically when the DLVM is turned on.

The Bucket is shared between all users of a project; anybody assigned to a project can read and write to any folder in the Bucket. This is changing, within the last few weeks, our institution has implemented an “Enhanced Teams” feature that allows for attaching multiple Buckets to a single project, each with different permissions. But, within that bucket, all members still have access to all files.

BigQuery is Google’s version of a SQL database tool. As a data scientist focused on imaging, I don’t use BigQuery often but my coworkers tell me it is one of the best features of the Platform. It is lightning fast and benefits from much of Google’s expertise as a world-leader in keyword searching.

Vertex AI is the service for submitting code to a compute engine. Submitting a job to Vertex AI is often much faster and cheaper than running it on the DLVM. For those who are familiar with Slurm or Sun Grid Engine, Vertex AI is analogous.

To use Vertex AI, you must wrap your code in a Docker container and submit it with a YAML file that tells the engine what computing resources to apply.

As a brief note on the subject of Docker, it has quickly become our favorite tool for leveraging GCP. For our interns, one of my colleagues built a custom Docker container with RStudio Pro which allows us to do the hard work of setting up their environment in advance.

It also makes working with Vertex AI easier, because you've already debugged your code in a docker container which can easily be pushed to the engine.

The Google Cloud Software Developer Kit, or Cloud SDK is a separate, optional command-line interface (CLI) for running Cloud commands from your local Desktop or laptop. I highly recommend it.

SDK allows you to manage GCP from your laptop, including copying files to/from the bucket, accessing your DLVM via SSH (including port forwarding for an interactive environment), and turning on/off your DLVM.

On Linux and Mac, SDK commands can be run from the shell terminal. On Windows, Google provides a separate terminal interface, they are also available through PowerShell.

Once you've gotten comfortable with SDK, there's almost no need to use the web interface, which frankly can be clunky. I wrap up many common commands such as turning on/off the DLVMs into shortcuts in my .bashrc file.

Cloud App. Finally, Google has also written a Cloud Console smartphone app. It's got limited uses, but it can be used to turn on/off your DLVM in case you've forgotten.

Conclusion

Thank you for reviewing and I hope the information has been helpful to you! Please leave a comment and I'll be happy to answer any questions.

I'm sure readers would love to hear your experience with the Platform, and any tricks or tips you can offer.

~AW

Cloud Data Science

tds

Follow

Published in Towards Data Science

802K Followers · Last published 4 days ago

Your home for data science and AI. The world's leading publication for data science, data analytics, data engineering, machine learning, and artificial intelligence professionals.

Follow

Written by Alexander Weston, PhD

149 Followers · 22 Following

Principal data scientist at Mayo Clinic. My views are entirely my own.

Responses (1)

What are your thoughts?

Respond

Mamady Konate
Nov 19, 2023

Between AWS and Google Cloud, which is better for Healthcare Data Management in your opinion?

Reply

More from Alexander Weston, PhD and Towards Data Science