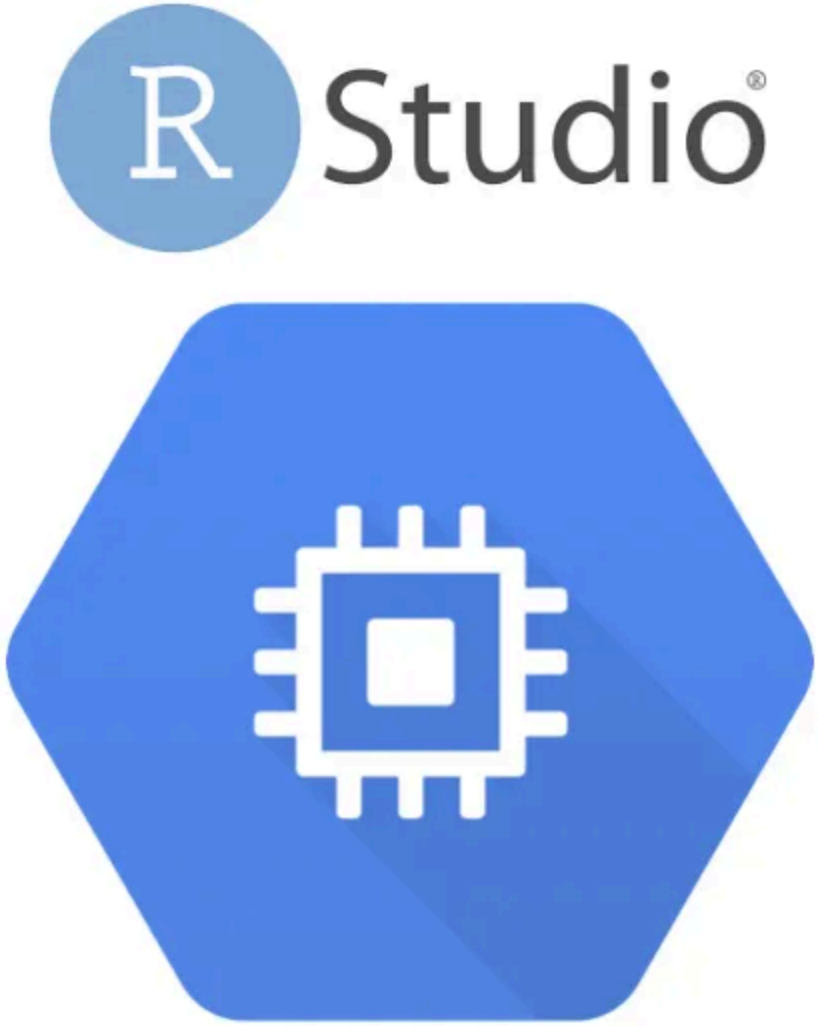
Running R/RStudio in a GCP VM

Stefan Gouyet · Follow

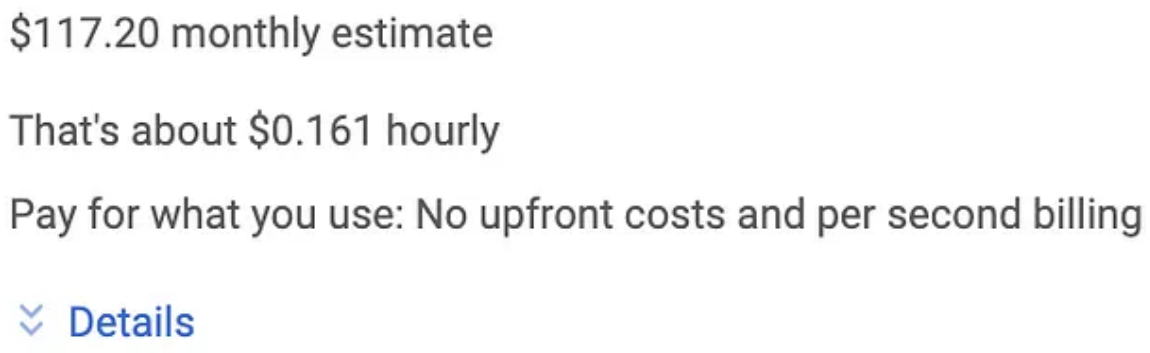
Published in Analytics Vidhya · 6 min read · May 8, 2020



Running large, computationally intensive R scripts is often a very slow process; how can we upgrade to a more powerful server, while keeping costs low?

The Google Cloud Platform (GCP) allows us to use Virtual Machines (VM) of various configurations and only pay for what we use. This pricing structure makes it feasible to run a large and intensive script for relatively cheap. And if your workflow is fault-tolerant, preemptible machines can reduce your total expenditure significantly more.

For the purpose of this article, I will use an n1-standard-16 machine (16 vCPUs and 60 GB of memory), with a Debian GNU/Linux 9 (stretch) disk image and preemptibility configured. The cost of this machine came out to be $0.161 hourly — my specific task took an average of 3 hours to run once a week, totaling around $2 dollars a month. *Make sure you turn the VM off after using it, as keeping it on would increase that number.*



Note: Of course, with a preemptible machine, there is always a risk that the instance will terminate while you are using it. That has personally never happened

to me, but it is always a risk. As always, if your workflow is not fault tolerant, preemptible machines are not recommended.

Step 1: Configure VM and Firewall Rules

Below is the gcloud command for creating my specific configuration (note that I have added a tag of http-server).

```
gcloud compute instances create task2r-vm --zone=us-west1-b --
machine-type=n1-standard-16 --image=debian-9-stretch-v20200420 --
image-project=debian-cloud --preemptible --
scopes=https://www.googleapis.com/auth/cloud-platform --tags=http-
server,https-server
```

Next, let's configure our firewall rules (under VPC Network> Firewall Rules) to allow us to access port 8787, which we will need to use our RStudio GUI. The firewall rule should use IP-ranges: 0.0.0.0/0 and port tcp:8787, and be applied to tag: http-server (so it connects to our VM).

Once these two steps have been finalized, we can SSH into our VM via the console or command line.

```
gcloud compute ssh task2r-vm --zone=us-west1-b
```

Step 2:

After successfully SSHing into our sever, let's install our relevant packages:

As always, begin with:

```
sudo apt-get update
```

Next, run these two commands (the first will allow us to install deb files while the second installs R).

```
sudo apt -y install gdebi-core
sudo apt -y install r-base r-base-dev
```

Additionally, if you will require the use of tidyverse, you will need to run this command as well:

```
sudo apt-get install libcurl4-openssl-dev libssl-dev libxml2-dev
```

Once these dependencies have been installed, make sure R can be accessed in your VM:

```
sudo R
```

Step 3: Install RStudio Server

With R now working, let's install RStudio as well (this is technically not a required step, as we are able to run R scripts without RStudio).

RStudio's documentation makes this part very simple.

We can run the second and third of the above commands (we've already ran the first):

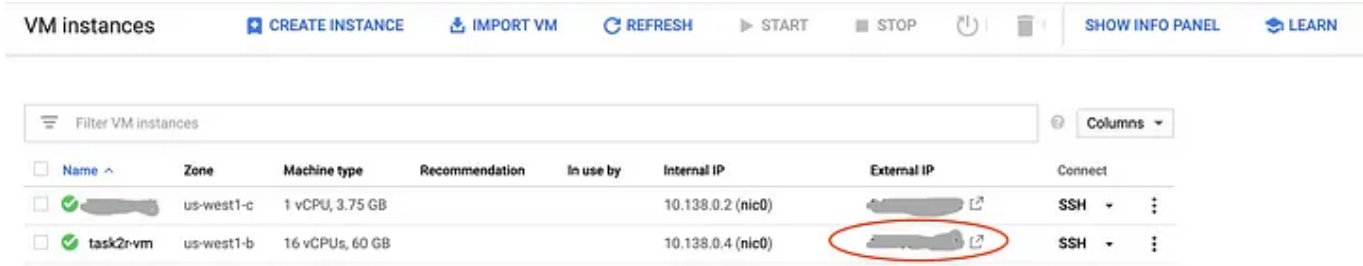
```
wget https://download2.rstudio.org/server/debian9/x86_64/rstudio-server-1.2.5042-amd64.deb

sudo gdebi rstudio-server-1.2.5042-amd64.deb
```

After the two commands have been successfully ran, we are returned with an acknowledgment that RStudio has been started.

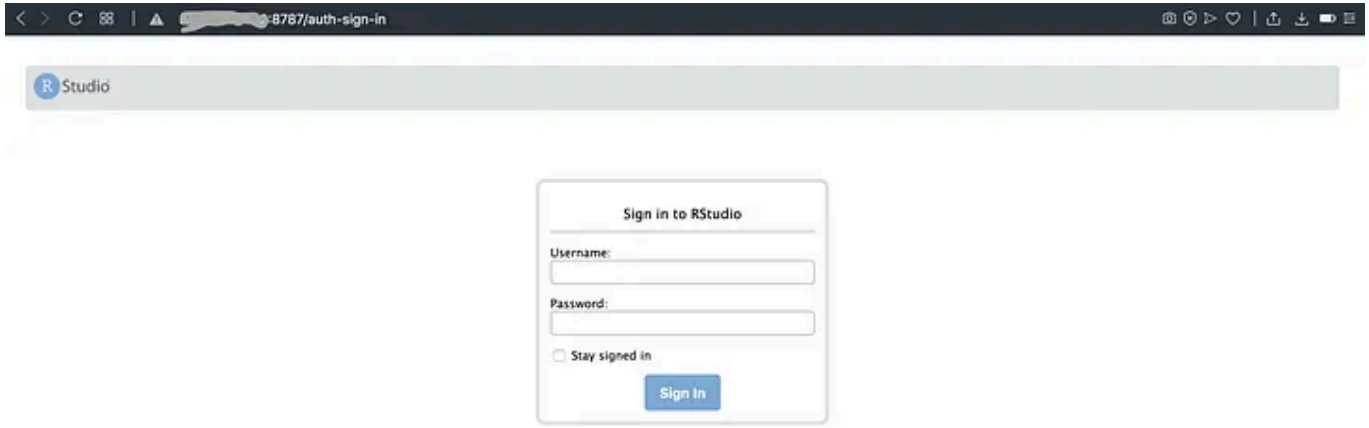
```
task2r-vm systemd[1]: Starting RStudio Server...
task2r-vm systemd[1]: Started RStudio Server.
```

To visually confirm that this is the case, navigate to your VM's **External IP**, specifying port number 8787. This should look like <External IP>:8787. Recall that tcp:8787 was the port that we specified in the firewall rule (step 1).



Find your instance's External IP

When you navigate to the URL, you will see a "/auth-sign-in" string attach itself after the 8787 port number.



Visit <External IP>:8787

Back in the shell, we need to add ourselves as a new user. To do so, we use the following command (replace <stefang> with your preferred username):

```
sudo adduser stefang
```

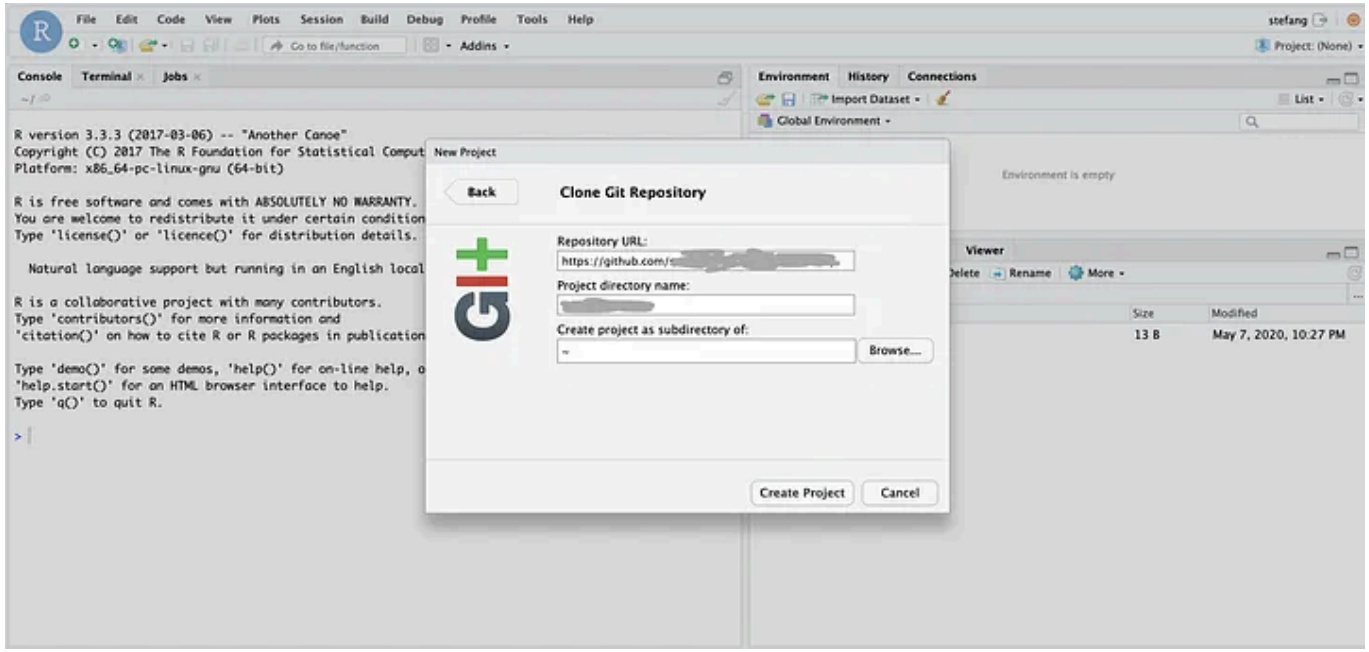
After running this, return to the RStudio sign-in page and use your new credentials:

Great! We have RStudio up and running. Let's now bring in our code via Github (I personally like to write all my code on my local computer — while my VM is turned off — and then push the code to the VM to run it if there are any changes).

Before doing so, we need to install git in the shell (the second line below is not necessary but allows for a longer “timeout” between sign-ins).

```
sudo apt-get install git-core -y
git config --global credential.helper "cache --timeout 28800"
```

After restarting RStudio, we can now create a new project by cloning our git repository.



Regarding directories: if you want to access your repository in the VM shell, you need to change users, from your default user to the user you create with the “sudo adduser <username>” command above.

```
cd /home
cd stefang/
```

From this new directory, we can access all of our files (including any outputs from the script) via the shell.

One final note:

I wanted to fully automate this process so that, unless I had new code to push, the script would run without me having to SSH into the instance.

There were a few steps required for this task, which I will briefly sum up:

First, I set up Cloud Scheduler to start/stop my instance at specific times each week. Cloud Scheduler allows us to limit our usage of our VM, keeping

costs low and reducing any wasted energy. The full process of setting this up also involves Cloud Functions and Pub/Sub.

Second, I created a bash script that would change current directories, mount a GCS bucket, and upload my script's output files (several CSVs and an HTML map file) to the bucket.

```
#!/bin/bash

cd /home
cd stefang/

gcsfuse --implicit-dirs <project-name-bucket> gcs-bucket/

sudo mv <project-name>/output/ gcs-bucket/

#unmount GCS bucket
fusermount -u gcs-bucket
```

Third, I configured Crontab to run the script every Thursday at 1:50 pm, preceded by the bash script at 6:00 pm (this gave me enough time in case the report took longer).

```
50 13 * * 4 cd /home/stefang/<project-name> && /usr/bin/Rscript -e
"rmarkdown::render('<script-name>')"
```

```
00 18 * * 4 cd /home/stefang/<project-name> && /usr/bin/bash
write_to_bucket.sh
```

— — —

So that's it! Thanks for reading and feel free to leave any comments/questions below.

Google Cloud Platform | Rstudio | R

Responses (4)

What are your thoughts?

Respond

Jpierrat
over 1 year ago

Hi Stephan,

I installed RStudio on a Google VM using a Nginx server so that I could use an SSL certificate. Everything works fine, but I can't upload files larger than 100Mb. I'm sure this isn't a problem with the RStudio configuration, as I have.....

Read More

Reply

Sailesh J
over 3 years ago

Hi Stefan,

Very nice article. How would you run the script on the VM for an on demand job ?

This means I want to run an r-script on my VM when users clicks a button on my UI. I have tried using cloud functions but not much luck SSH into VM

Any thoughts ?

Reply

Marcelo Alid
over 3 years ago

Great article, thanks!
Could you please fix the link to the full process of automation (the one using Cloud Scheduler, Cloud Function and Pub/Sub?

Reply

See all responses

More from Stefan Gouyet and Analytics Vidhya

In Analytics Vidhya by Stefan Gouyet

An Introduction to the TD Ameritrade API in Python

With the TD Ameritrade API, analyzing stock market data has never been so easy. With ju...

May 27, 2020 74 6

In Analytics Vidhya by Harikrishnan N B

Confusion Matrix, Accuracy, Precision, Recall, F1 Score

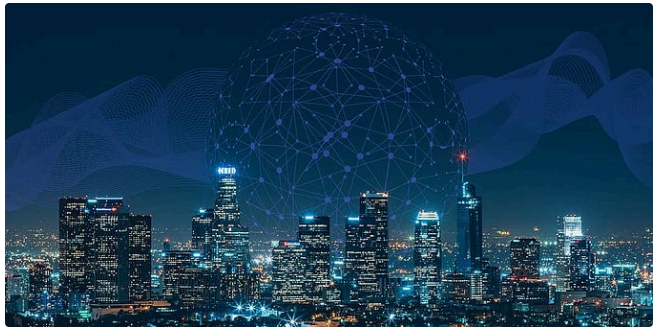
Binary Classification Metric

Dec 10, 2019 1.2K 6

In Analytics Vidhya by Kia Eisinga

How to create a Python library

Ever wanted to create a Python library, albeit for your team at work or for some open...



In Analytics Vidhya by Stefan Gouyet

Getting Started with Google's Data Loss Prevention API in Python

Jan 26, 2020 2.8K 31

Cloud Data Loss Prevention (DLP) is one of the many cloud security products that Goog...

Jul 13, 2020 57

See all from Stefan Gouyet

See all from Analytics Vidhya

Recommended from Medium

Supriti

A Beginner's Guide to Google Cloud Functions (GCF) with Python

Emad Dehnavi

Protect Your Sensitive Data with Piiranha-v1

As developers, one of the biggest headaches we face is managing infrastructure — settin...

Nov 1, 2024 80

Are you worried about protecting your users' sensitive data from unauthorized access?...

Sep 14, 2024 3

Lists

Staff picks

796 stories · 1559 saves

Stories to Help You Level-Up at Work

19 stories · 912 saves

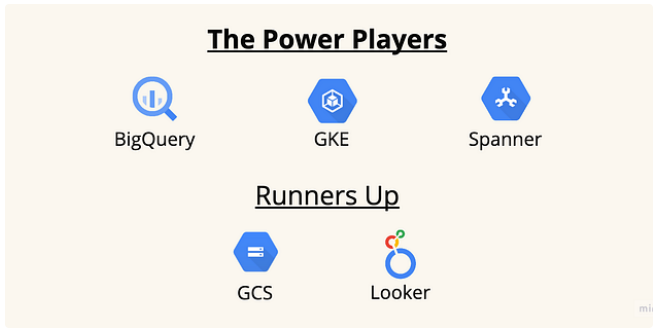
Self-Improvement 101

20 stories · 3191 saves

Productivity 101

20 stories · 2705 saves

In The Data Ledger by AndH



Serhat Soydan

Data Analysis with ydata-profiling: Practical Insights and Examples

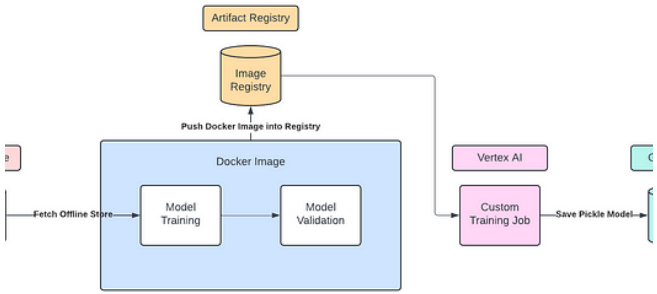
In today's data-driven world, efficient data profiling is essential for gaining insights and...

Jul 12, 2024

4 Years with Google Cloud Platform (GCP): The Good, the No...

My honest take on GCP services—what shines, what lags, and what needs a serious...

Dec 10, 2024 208 3



In Google Cloud - Community by Maxell Milay

Custom ML Training with Docker and Google Cloud Vertex AI

Google Cloud Vertex AI provides high levels of abstraction in handling simple Machine...

Sep 15, 2024 53

In CodeX by Christianlauer

Google brings BigQuery to VS Code

Using the Google Cloud Code extension for VS Code to work with BigQuery datasets an...

Dec 17, 2024 14

See more recommendations