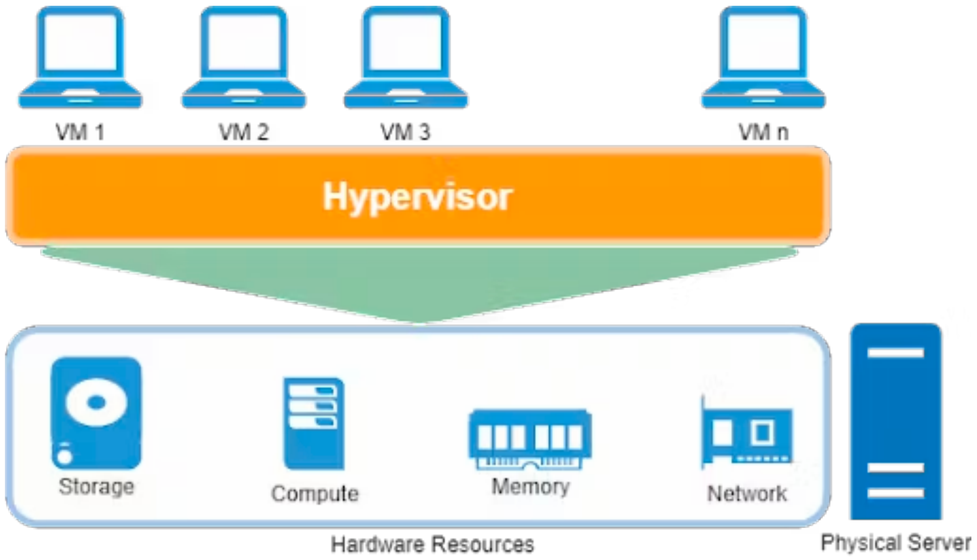
RShiny Deployment on GCP

Tutorial

Claire

Intro: why use a computing platform?



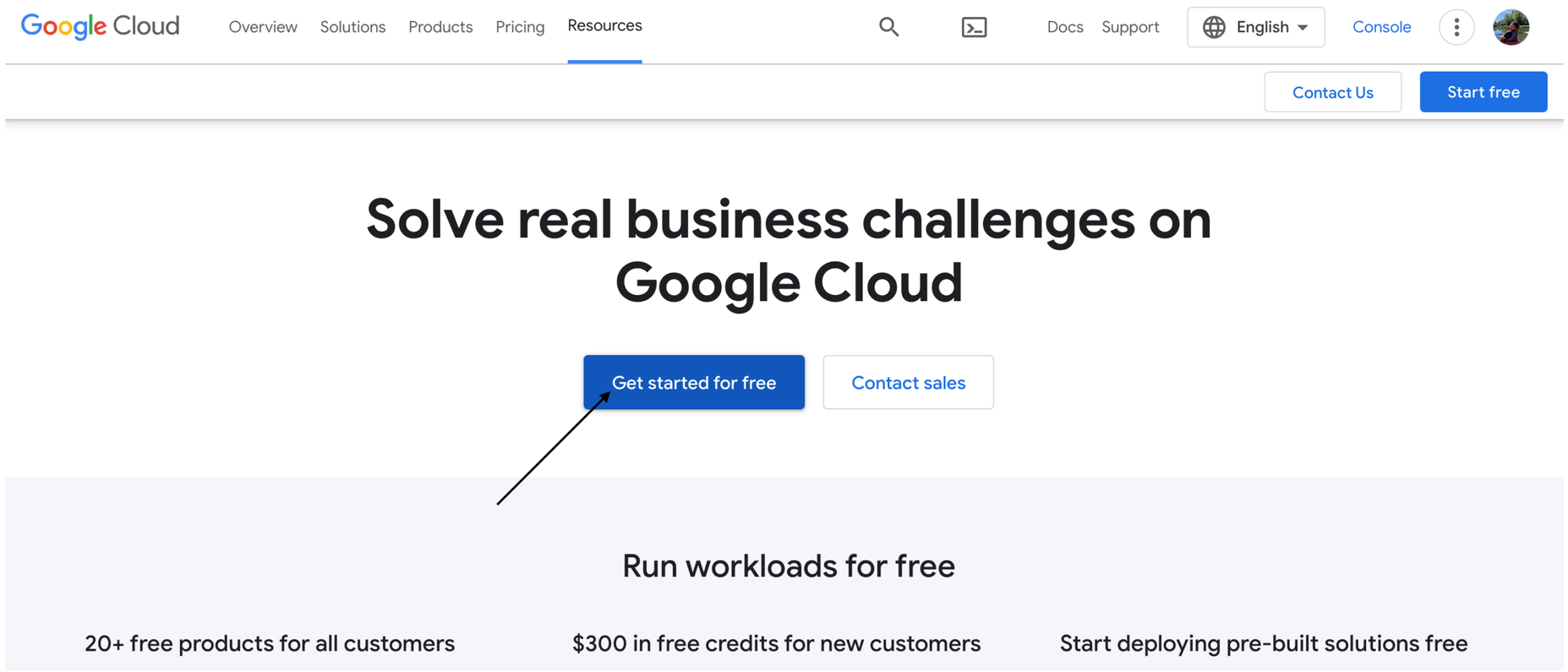
Remote access to VM: virtual machines

Host machine: can be bigger, simpler, more powerful, and far away.

Nowadays, many platforms lend computing resources: AWS, Google, Microsoft Azure, etc. and most of them work similarly.

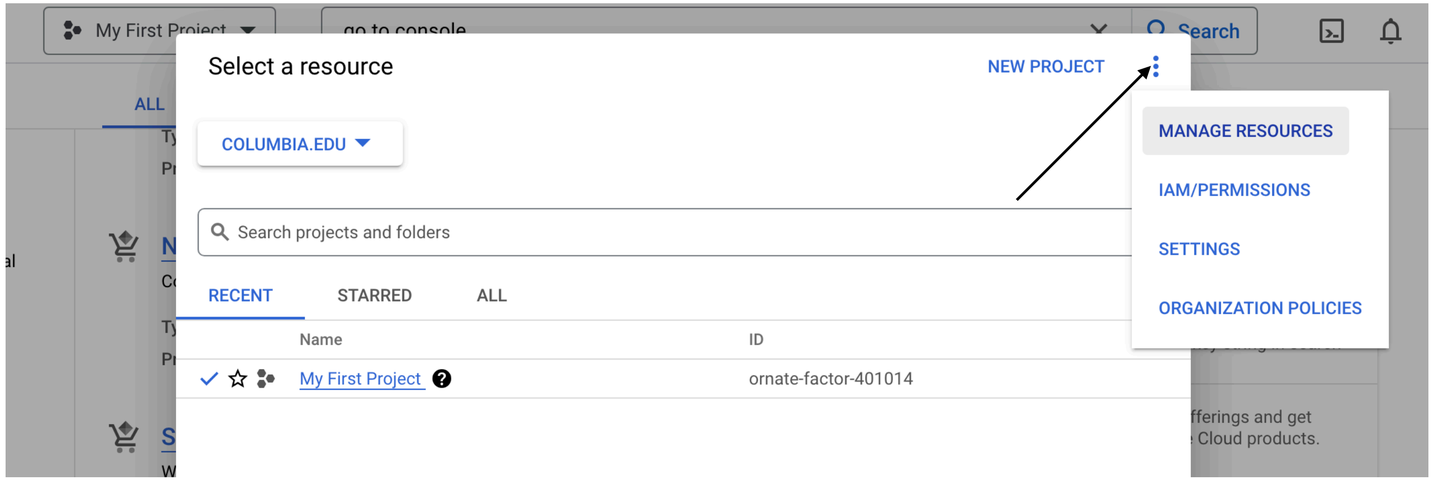
Step 0: Register on GCP and create project

You will have to enter your details after registering for a 12-month free trail.

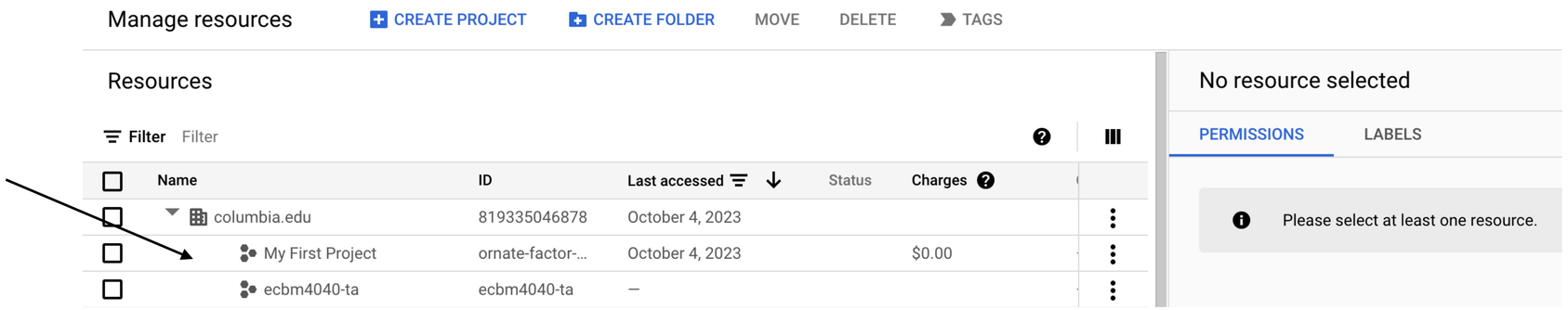


Step 0: Register on GCP and create project

Click on ‘My First Project’ and select the drop down menu on the side to go to ‘Manage resources’



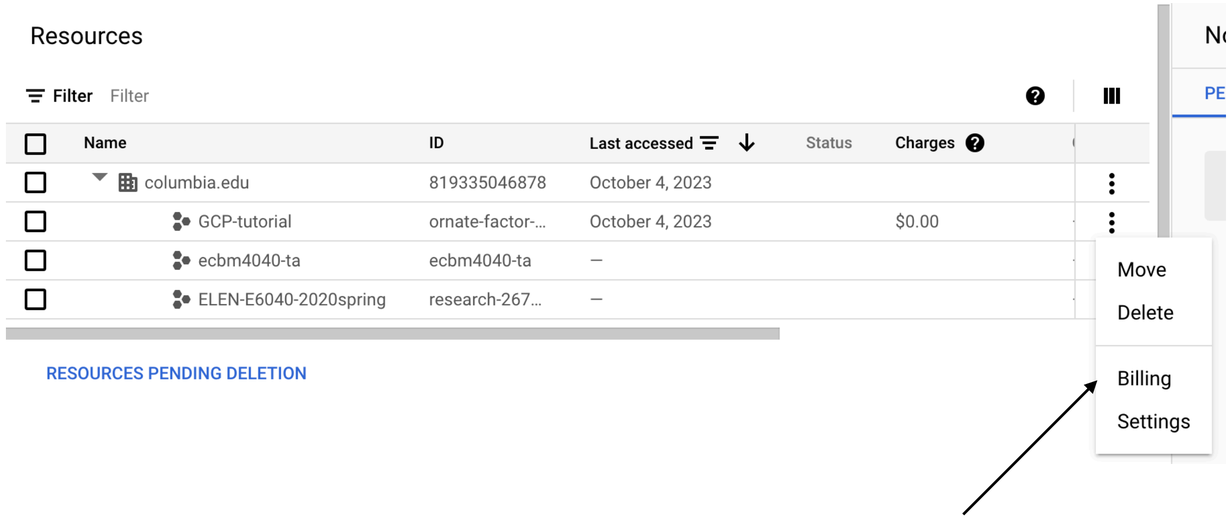
This will open the console view of the GCP and you can create your project.



Step 0: Register on GCP and create project

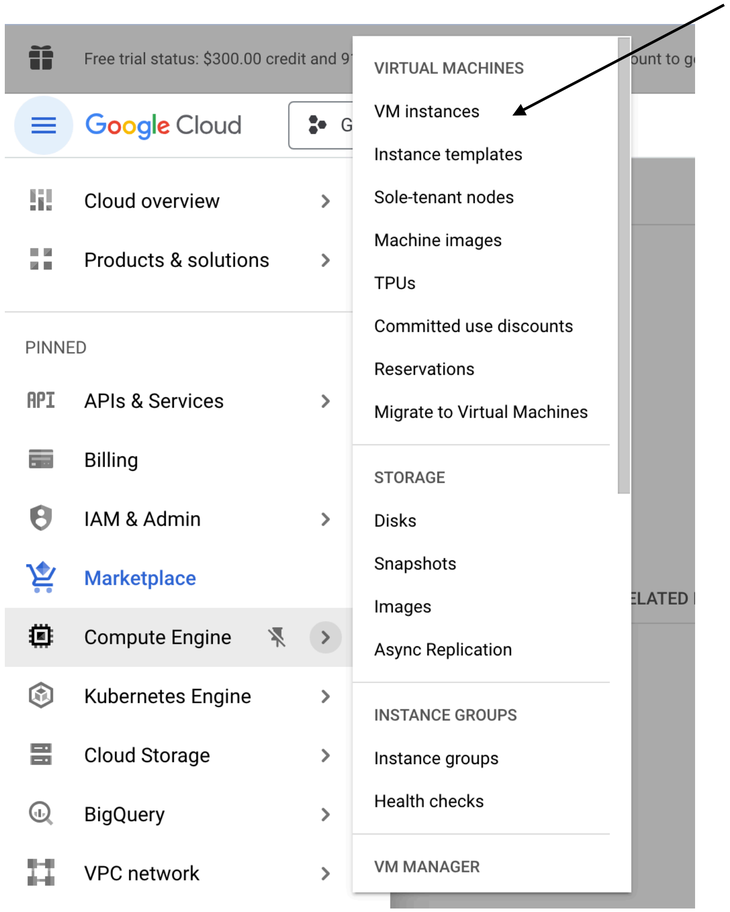
Check that billing has been enabled on your project.

If not you can access billing details in the drop down menu.

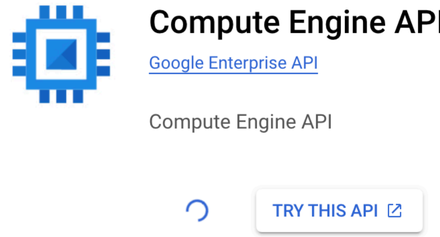


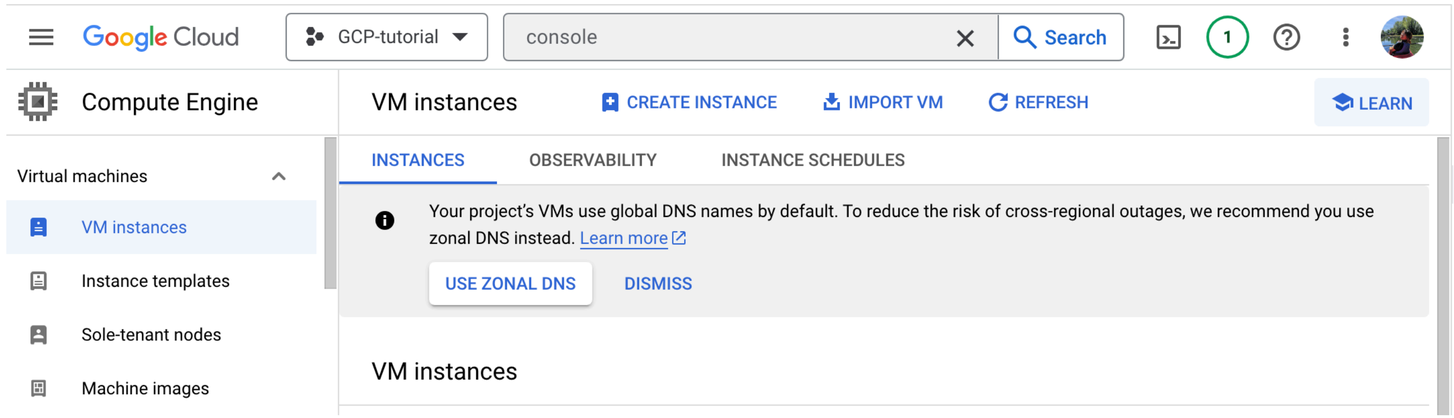
Step 1: Create a virtual machine instance

In Compute Engine menu from the sidebar, go to VM instances.



Create a VM instance by enabling the Compute Engine API

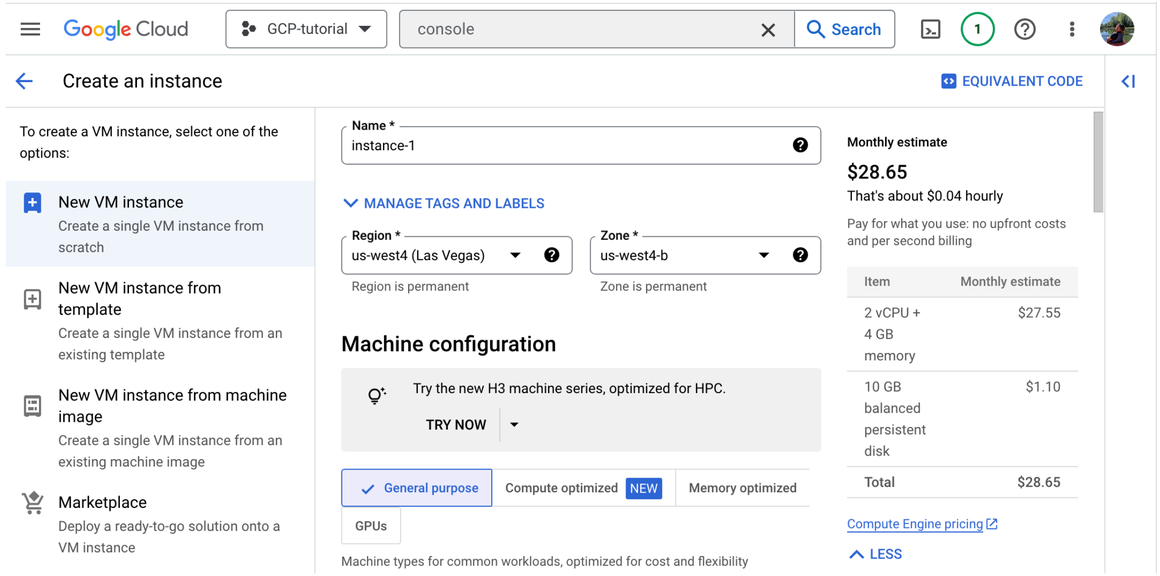




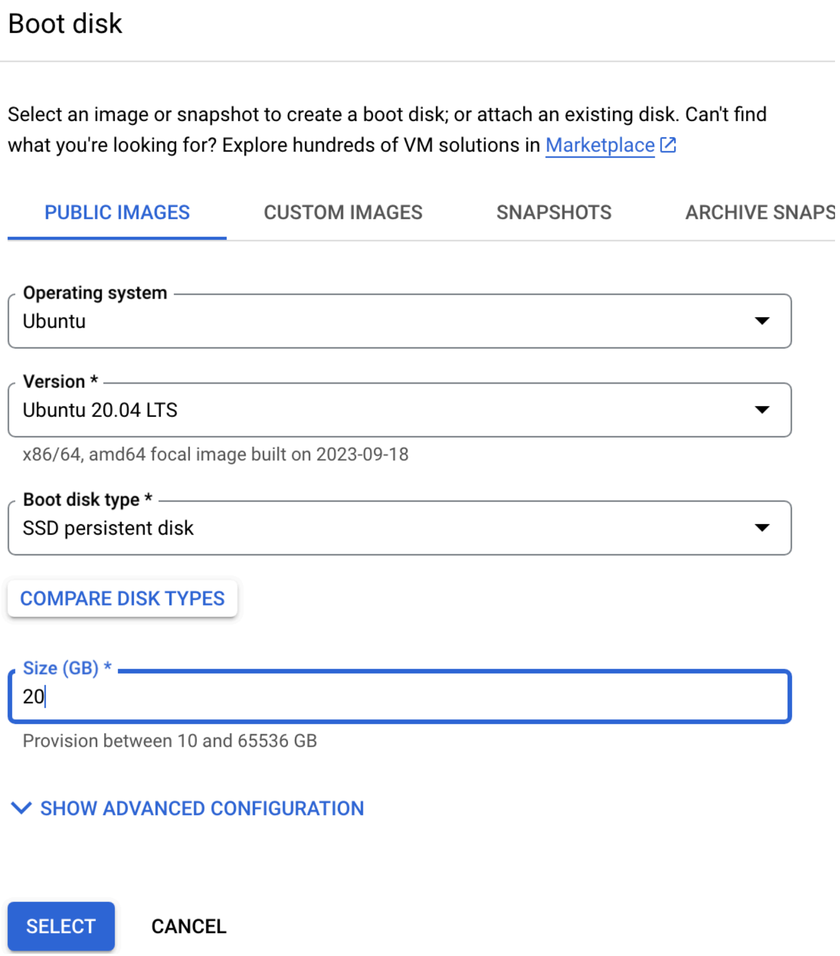
The VM instances page should look like the above.

Step 1: Create a virtual machine instance

Create your instance by completing the following



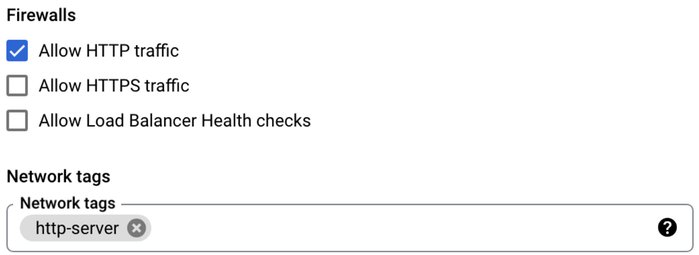
You can choose the boot disk you want to install.

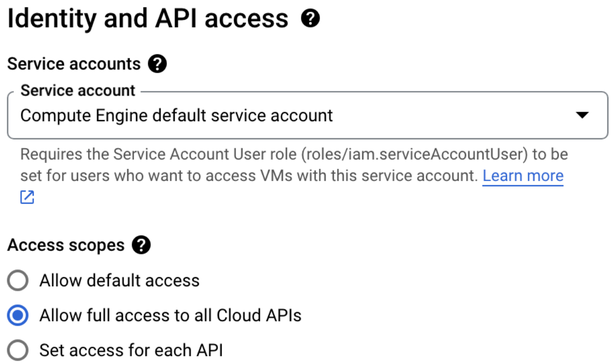


Step 1: Create a virtual machine instance

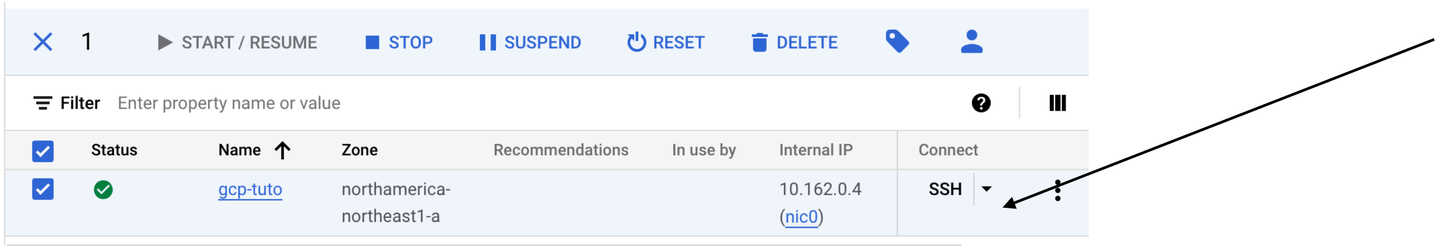
In Firewall, allow HTTP traffic.

In Access scopes allow full access to all Cloud APIs.



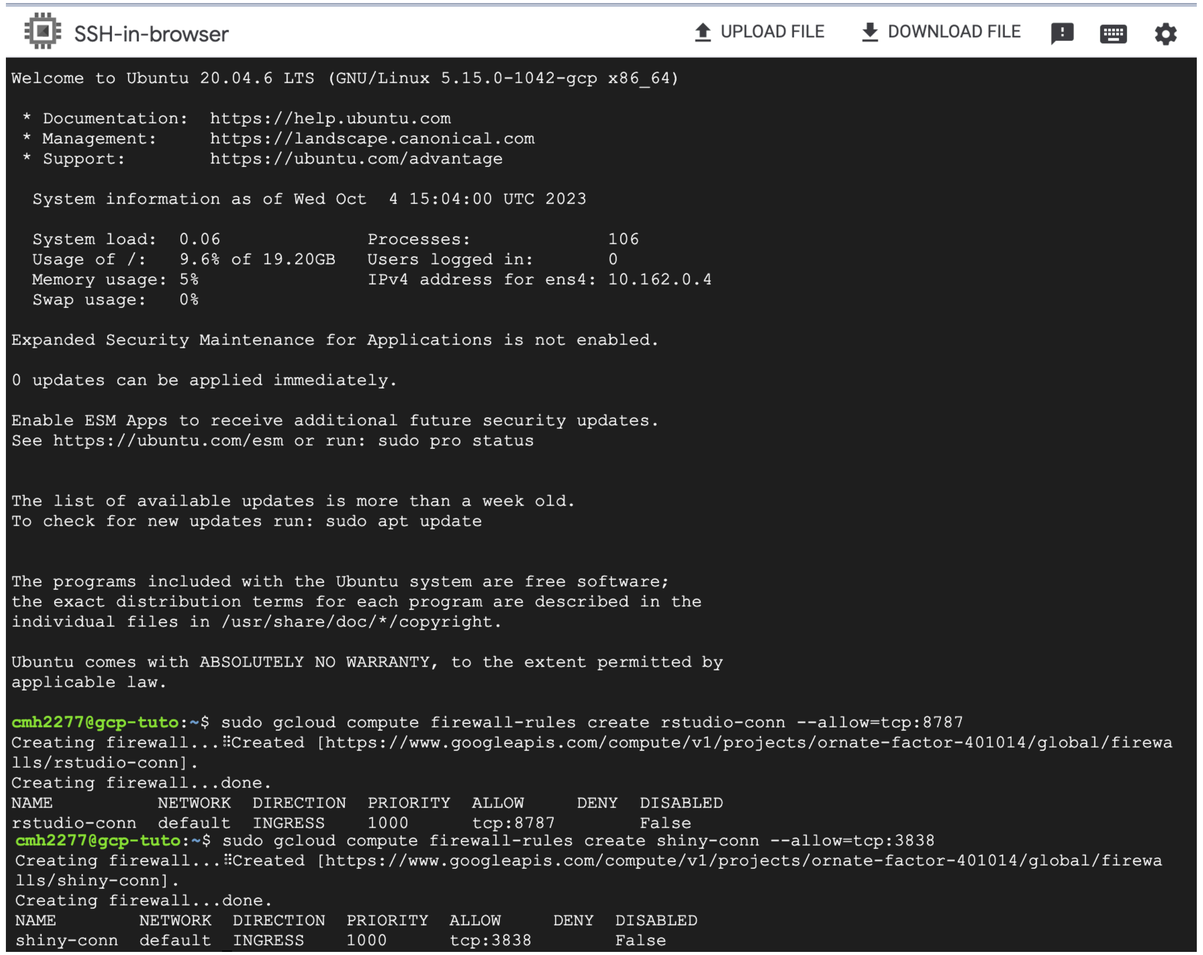


Run your VM instance by clicking start.



Step 2: Create a virtual machine instance

Now we want to set up the virtual machine so that it supports R Shiny apps



In these steps we allow R Shiny app and R Shiny server connections to our Virtual Machine.

We can now move on to installing R and R Studio on the VM.

Using a VM is like giving you access to a remote but powerful computer on which to run the usual operations you would on your local machine.

Step 3/4/5: Install R/RStudio on the VM

We will install R and RStudio on the VM using the pop-up window which is the terminal of your VM.

Remember you clicked on 'SSH' to open the window? This guarantees you can remote login safely.

For most Unix systems you can google the bash script to do your tasks… or ask your CompSci friends to help out.

We do all the necessary upgrading. And use the code to download R and the needed packages.

- The code chunk below adds a line to the repository list, then passes a key for the Ubuntu server to download R, updates the existing packages, and installs r-base and r-base dev.

```
sudo sh -c 'echo "deb https://cloud.r-project.org/bin/linux/ubuntu bionic-cran35/" >> /etc/apt/sources.list'
sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys E084DAB9
sudo apt-get update
sudo apt-get install r-base r-base-dev
```

- A few features won't work using only the r-base, since the packages are based on other programs as well, so to cover a few of those, below are the codes used to install the software needed.
  - Spatial Libraries:

```
sudo apt-get install libgeos-dev libproj-dev libgdal-dev
```

  - Tidyverse Universe:

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

  - other:

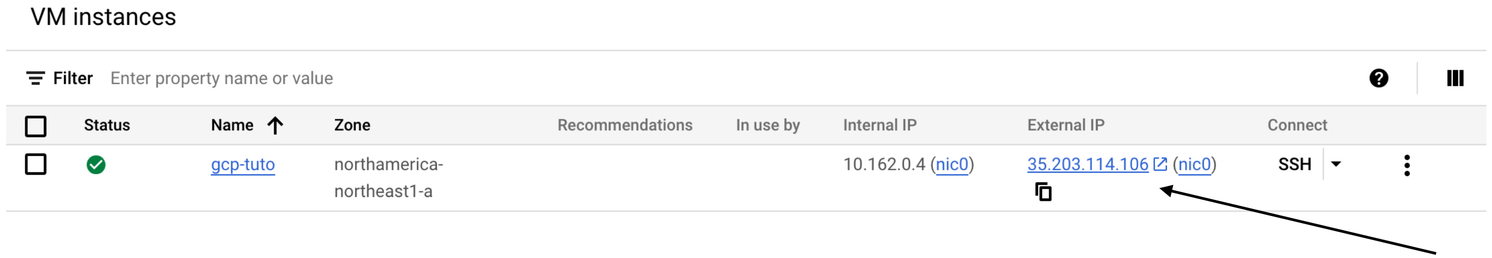
```
sudo add-apt-repository -y ppa:opencpu/jq
sudo apt-get update
sudo apt-get install libjq-dev

sudo apt-get install libudunits2-dev
sudo apt-get install libprotobuf-dev
sudo apt-get install libv8-dev
sudo apt-get install protobuf-compiler
```

These are the lines that are automatically fed to/ executed by your computer when you install R by downloading the packages.

Step 3/4/5: Install R/RStudio on the VM

- Follow the steps that allow you to install RStudio likewise.
- Note that we can run R without RStudio from the command line as soon as we have installed R.
- The command 'sudo -i R' tells the machine you are now writing lines in R. You can install packages that way like *dplyr, rmarkdown, shiny*.
- Remember that you need to quit the R app that is running in order to use the command line of you VM by typing 'quit()'.
- Note that this works similarly with Python and other languages.



For Ubuntu 20 + users, the wget url might change to

wget https://download2.rstudio.org/server/jammy/amd64/rstudio-server-2023.09.0-463-amd64.deb

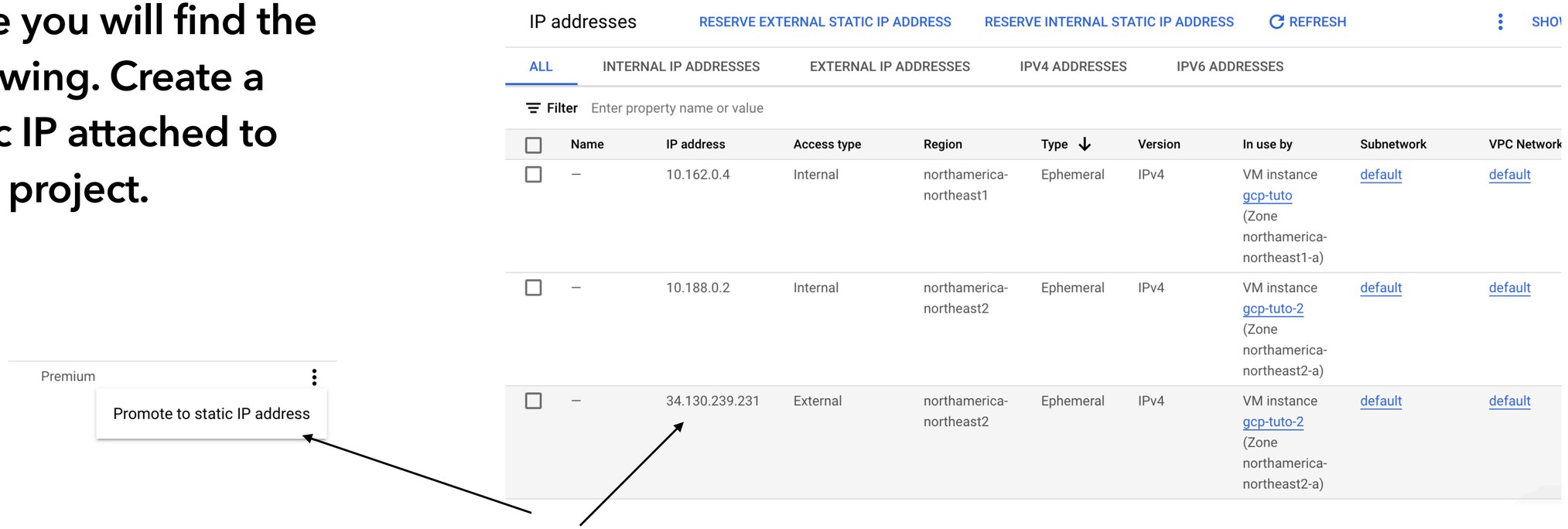
sudo gdebi rstudio-server-2023.09.0-463-amd64.deb

Other debugs here (https://posit.co/download/rstudio-server/)

To get your external IP, look up the VM instance you have created. Mine looks like http://35.203.114.1065:8787

Step 6/7: Reserve a static external IP/Develop your app

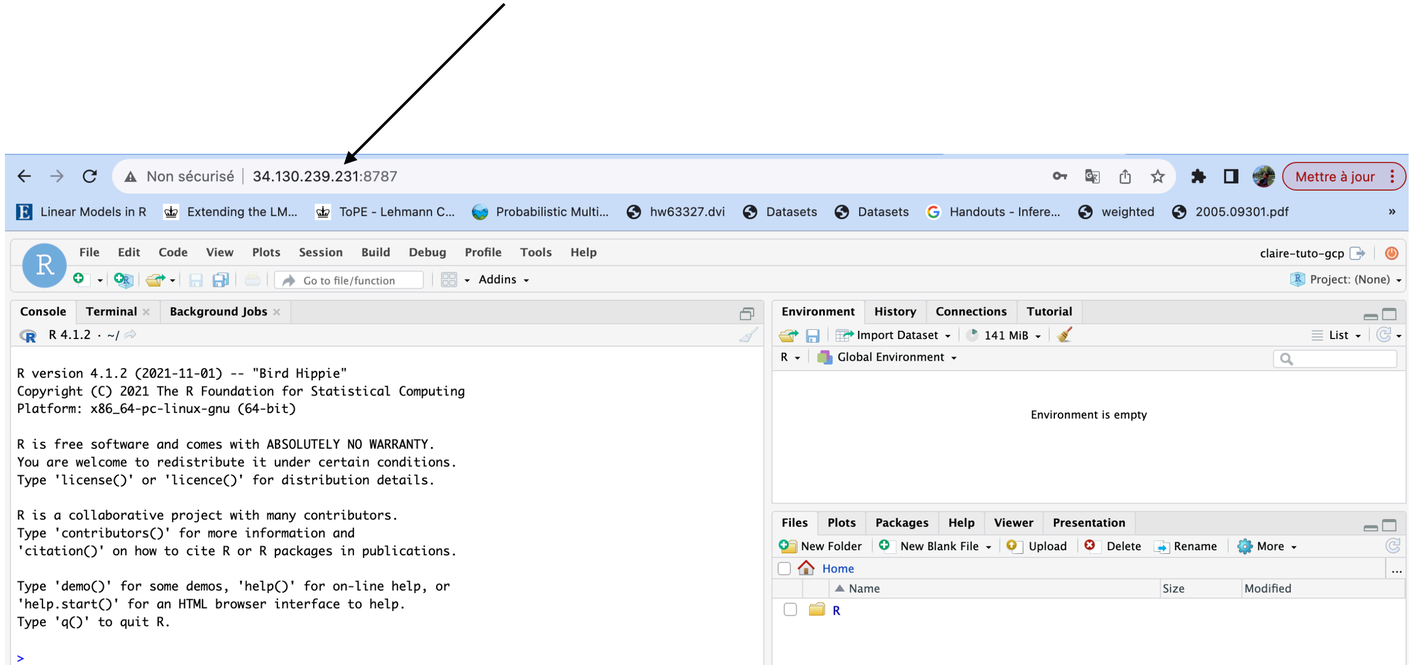
In the linked networking page you will find the following. Create a static IP attached to your project.

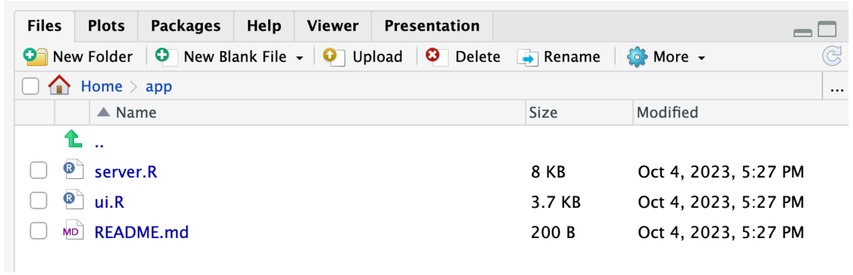


Step 6/7: Reserve a static external IP/Develop your app

In the linked networking page you will find the following. Create a static IP attached to your project.

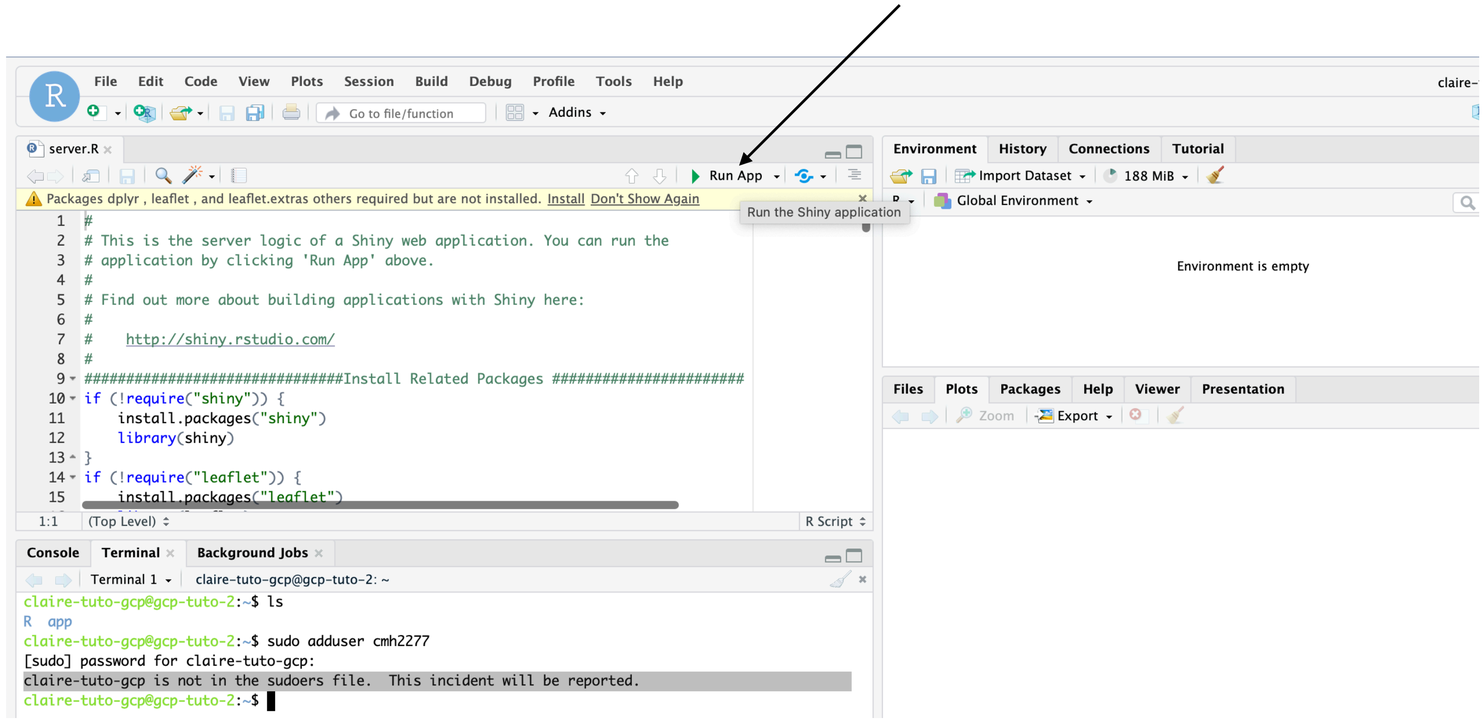
You can now upload your files for your Shiny App.





Step 8: Publish your Rshiny App

Last week, the tutorial showed you could use shinyapp.io.com to deploy your app using the package *rsconnect*. With GCP you can use a command line to copy all the files to your shiny-server *(Step 5)* which publishes your RShiny App *as long as the vm is running*.



Take away

You can log into the R studio server using your username and password and work remotely on a more powerful computational engine than you own laptop.