Introduction to RStudio and Shiny servers

Nicolas Attalides

Workshop Setup:

- Wi-Fi

Network Name:

Password:

- Requirements

An active Gmail account

What is Google Cloud Platform?

Google Cloud Platform – known as GCP - is a collection of cloud computing services that use the resources available at Google. GCP offers services via the cloud that access Google's physical hardware infrastructure such as: computers, hard disk drives, solid state drives and networking. This is a fast and cost effective alternative to having to build and maintain your own physical infrastructure.

Other popular services:

- Microsoft Azure Cloud Computing Platform & Services
- Amazon Web Services (AWS)

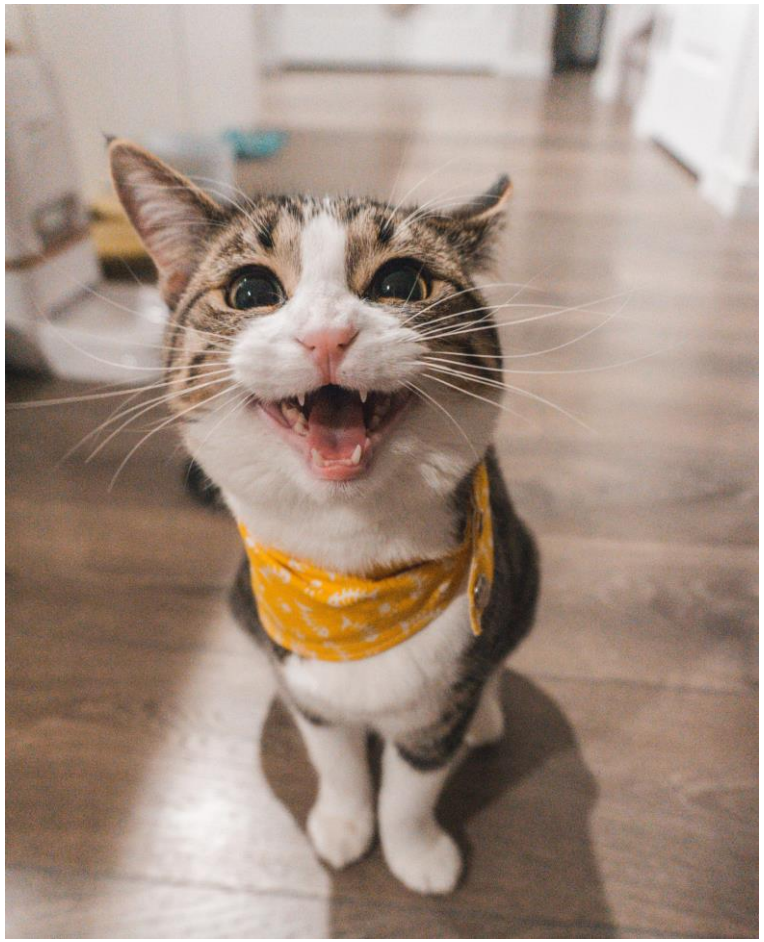
Topics

- Workshop aim:

Learn how to setup RStudio and Shiny servers on GCP and host a shiny app online.

- Topics:
  - Setup GCP Virtual Machine (VM) instance
  - Setup RStudio server
  - Setup Shiny server
  - Host a shiny app online

Setup GCP VM instance (for free)

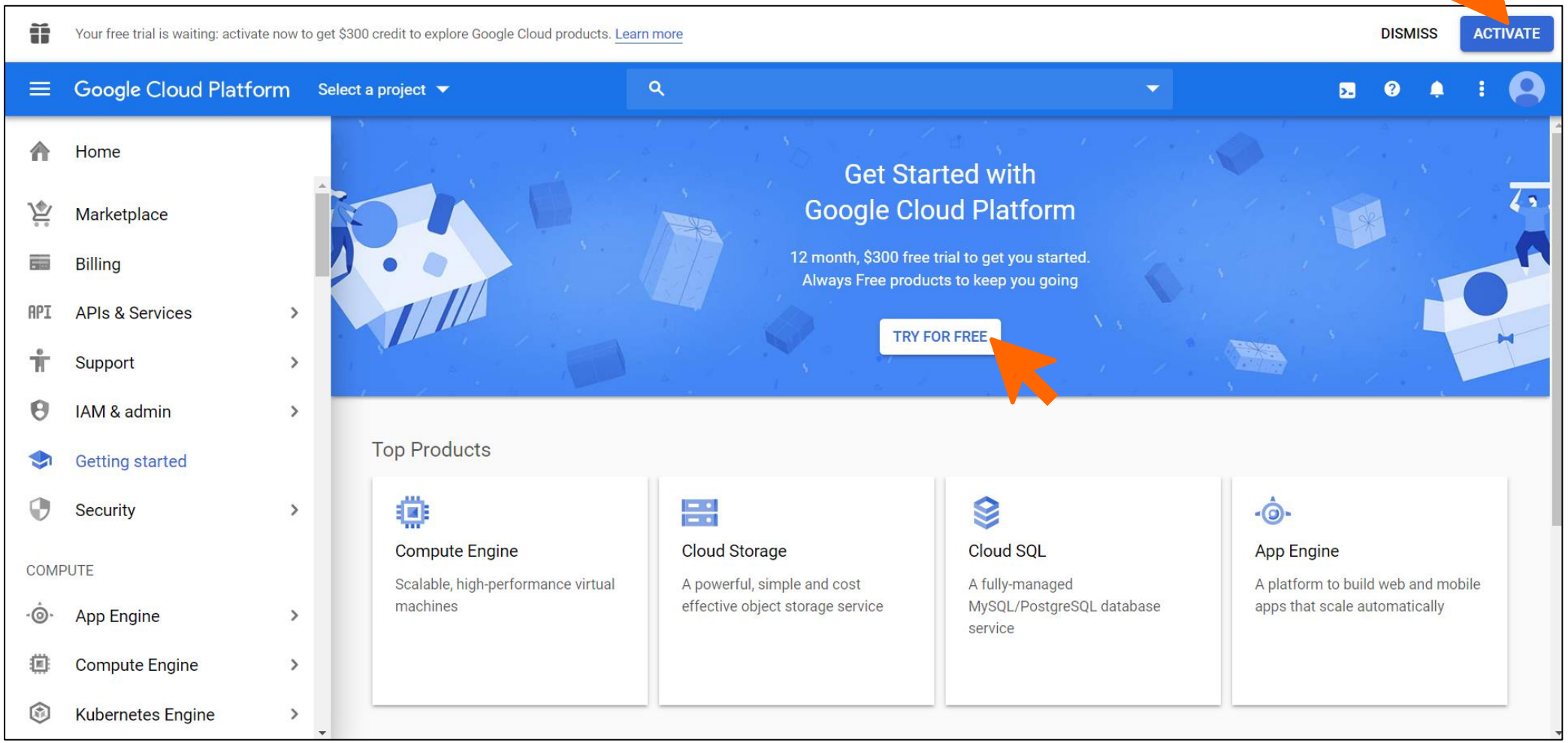


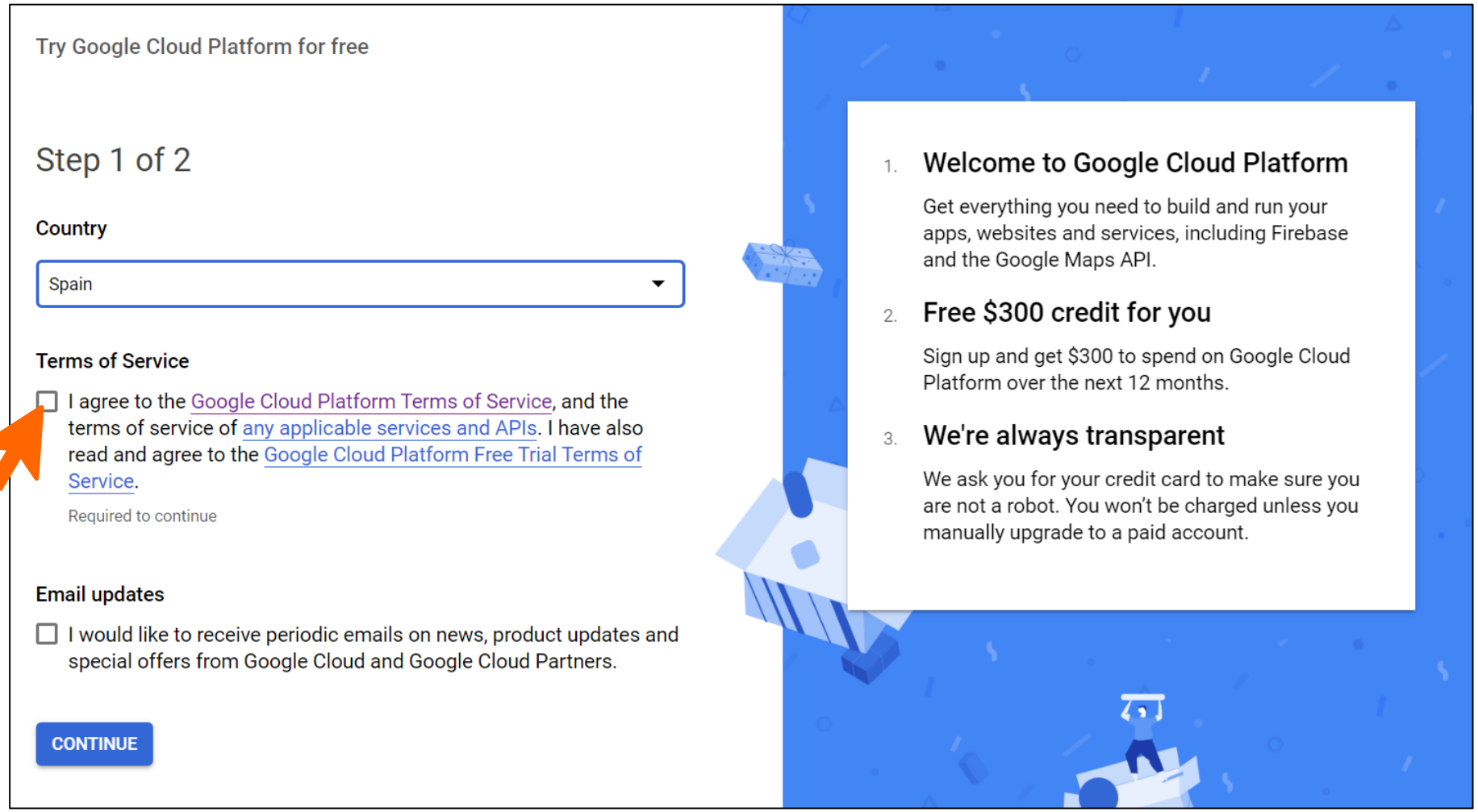
In order to access the Google Cloud Platform you will need to do the following:

1. Create a Gmail account
   (if you don't have one already 😱)
2. Visit https://console.cloud.google.com/
   (you might need to sign in)
3. Deal with the boring stuff (Terms of Service)
4. Get $300 free trial (for 12 months) 🥳
5. Get started!

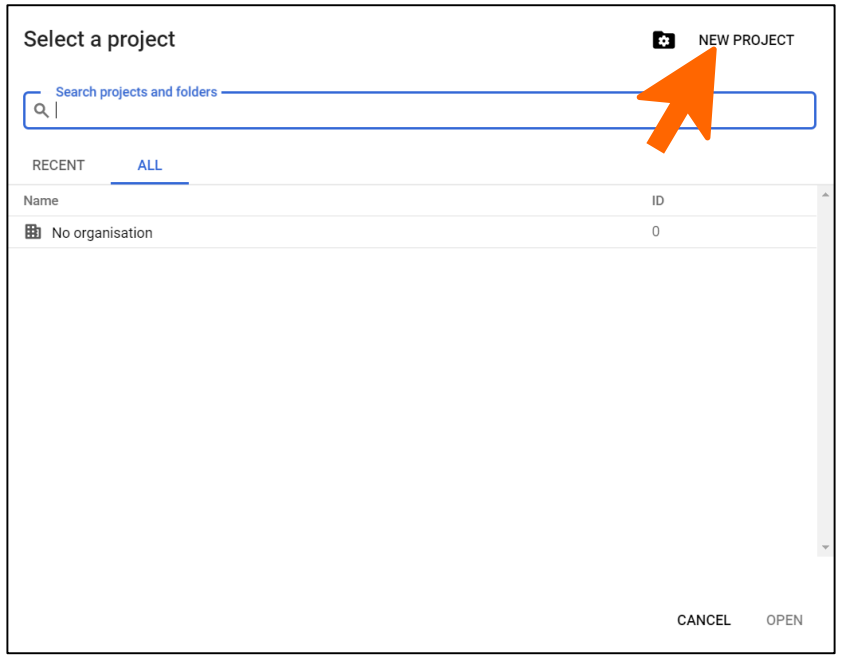
You might need to provide your card details – don't worry you are able to close your billing account if you want.

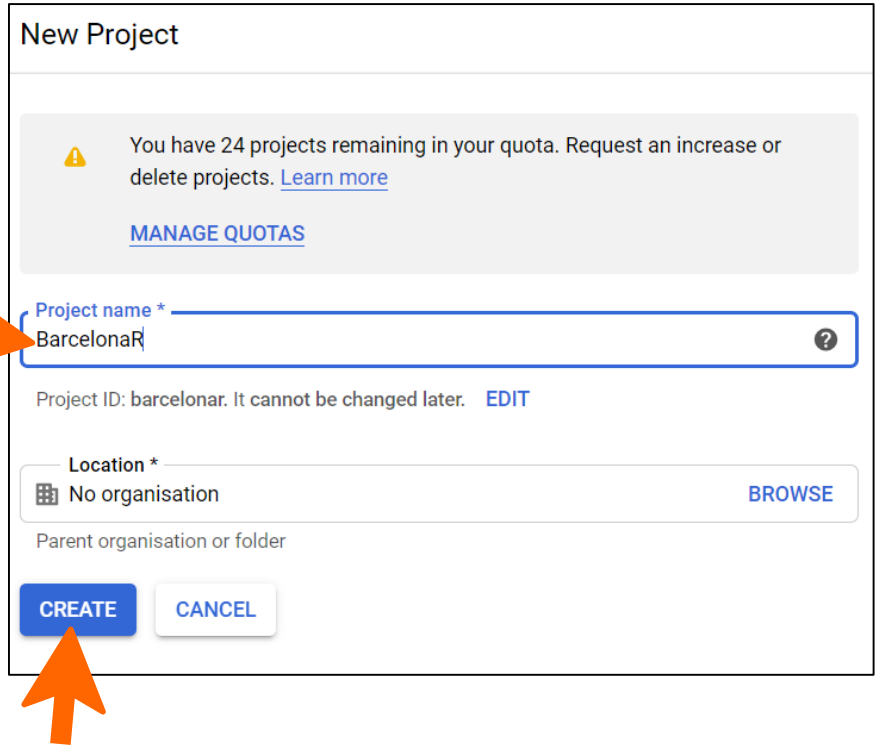
Live Demo Part 1



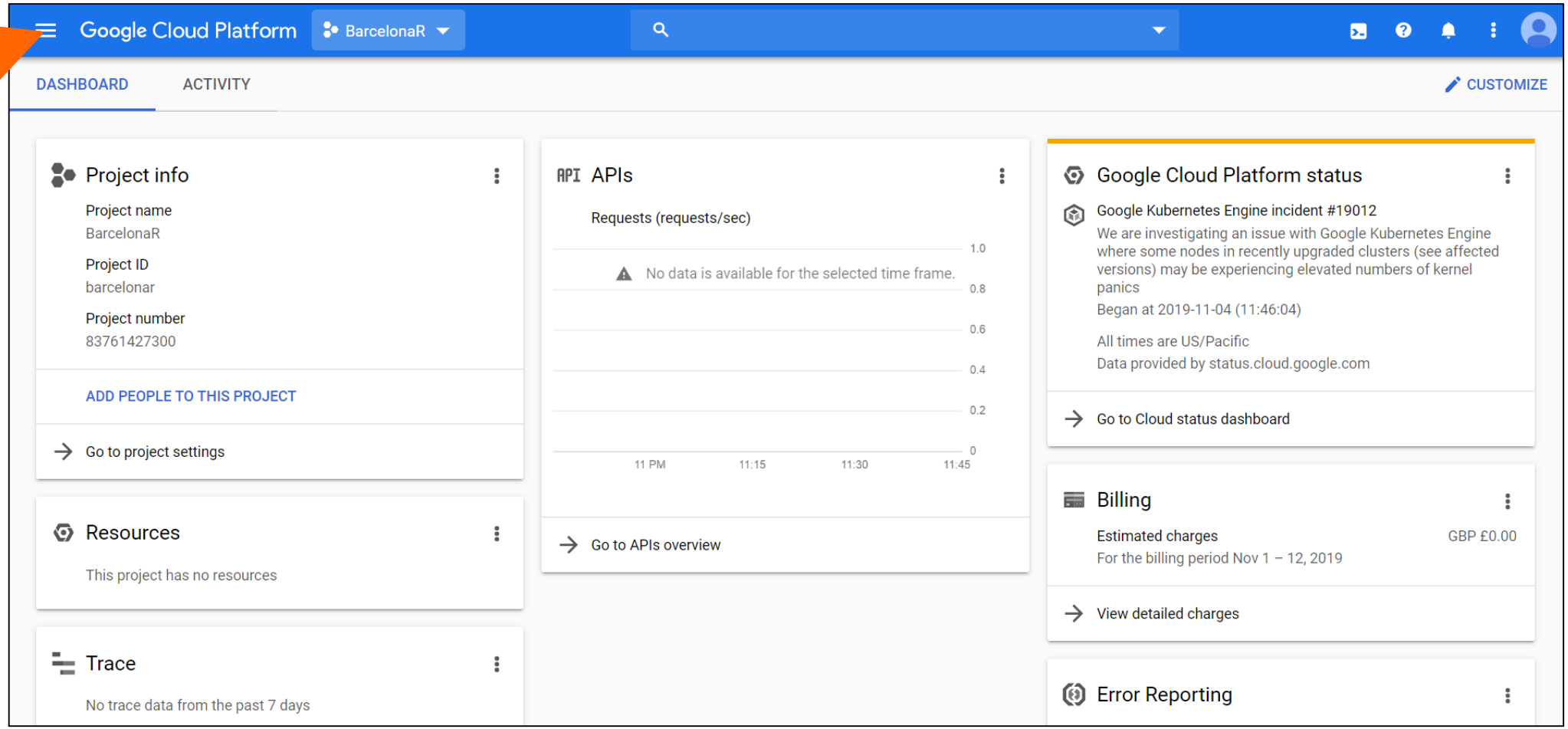

Try Google Cloud Platform for free
Step 1 of 2
Country
Spain
Terms of Service
I agree to the Google Cloud Platform Terms of Service, and the terms of service of any applicable services and APIs. I have also read and agree to the Google Cloud Platform Free Trial Terms of Service.
Required to continue
Email updates
I would like to receive periodic emails on news, product updates and special offers from Google Cloud and Google Cloud Partners.
CONTINUE
1. Welcome to Google Cloud Platform
Get everything you need to build and run your apps, websites and services, including Firebase and the Google Maps API.
2. Free $300 credit for you
Sign up and get $300 to spend on Google Cloud Platform over the next 12 months.
3. We're always transparent
We ask you for your credit card to make sure you are not a robot. You won't be charged unless you manually upgrade to a paid account.

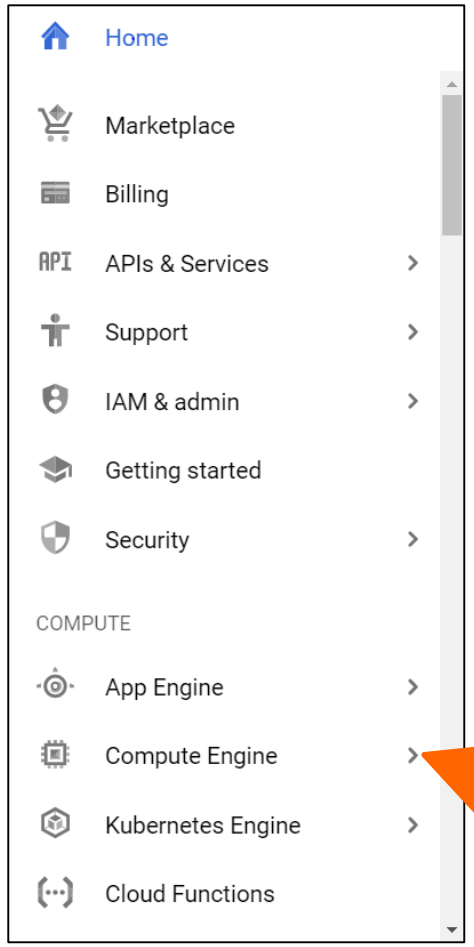
Create a project

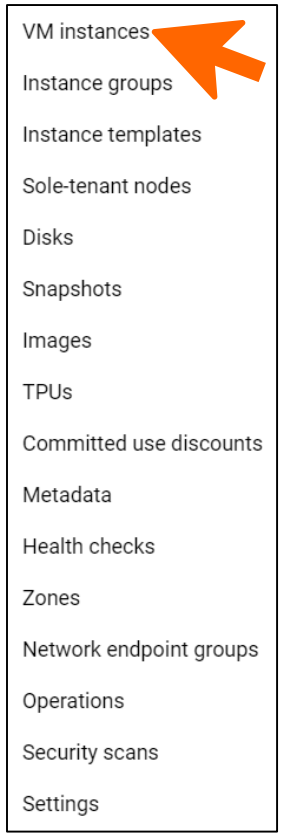


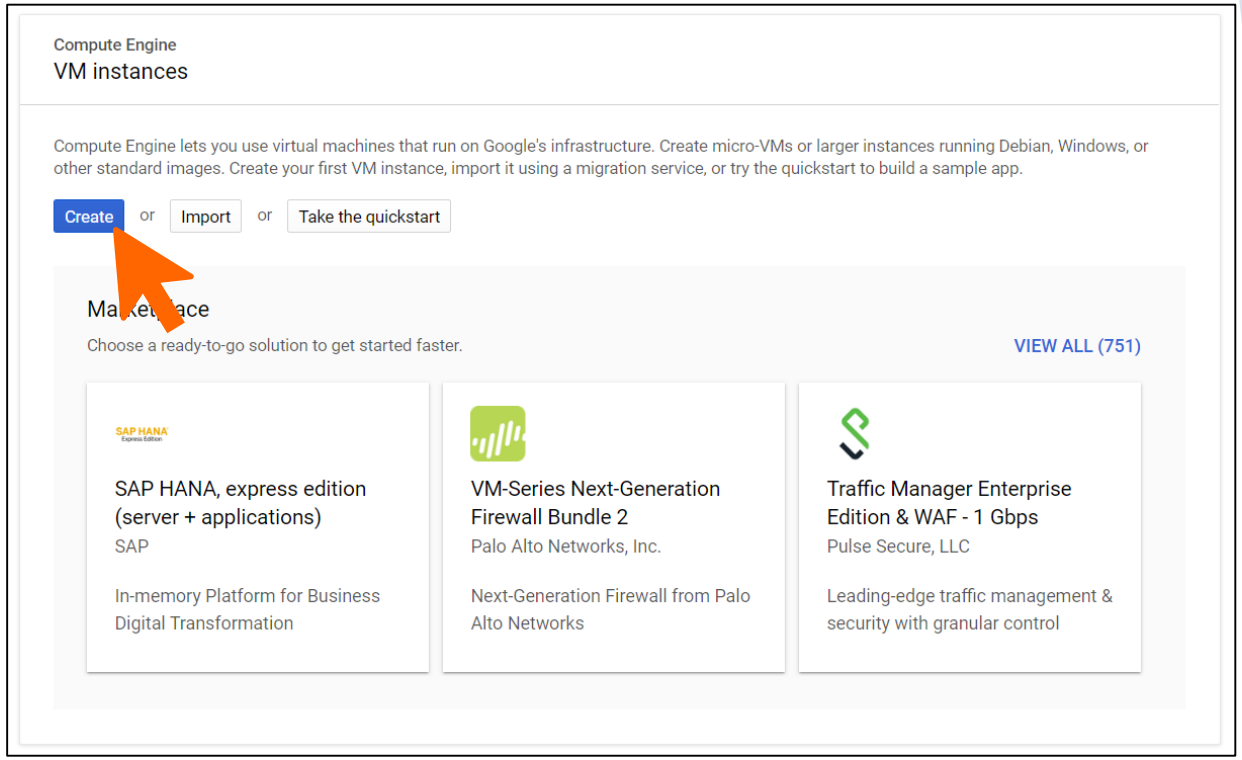


Create a VM instance



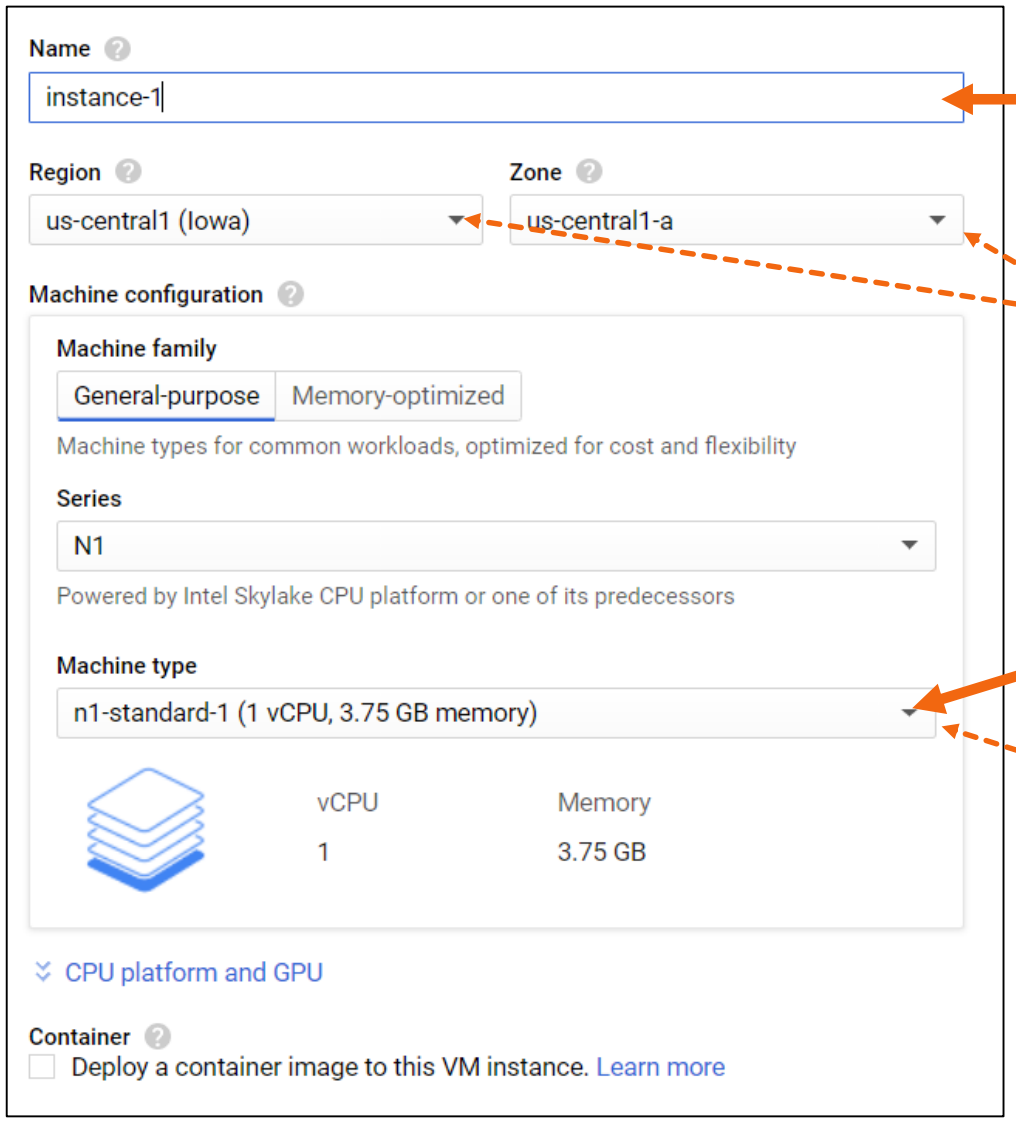






Use the pin 

functionality to pin the services you use most frequently!

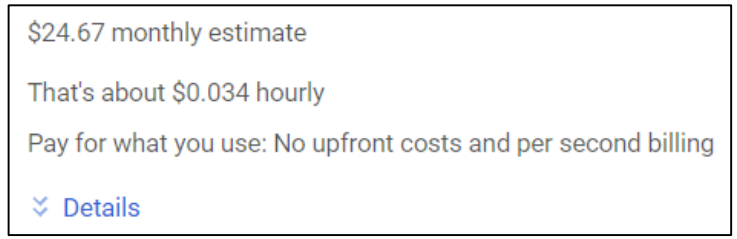


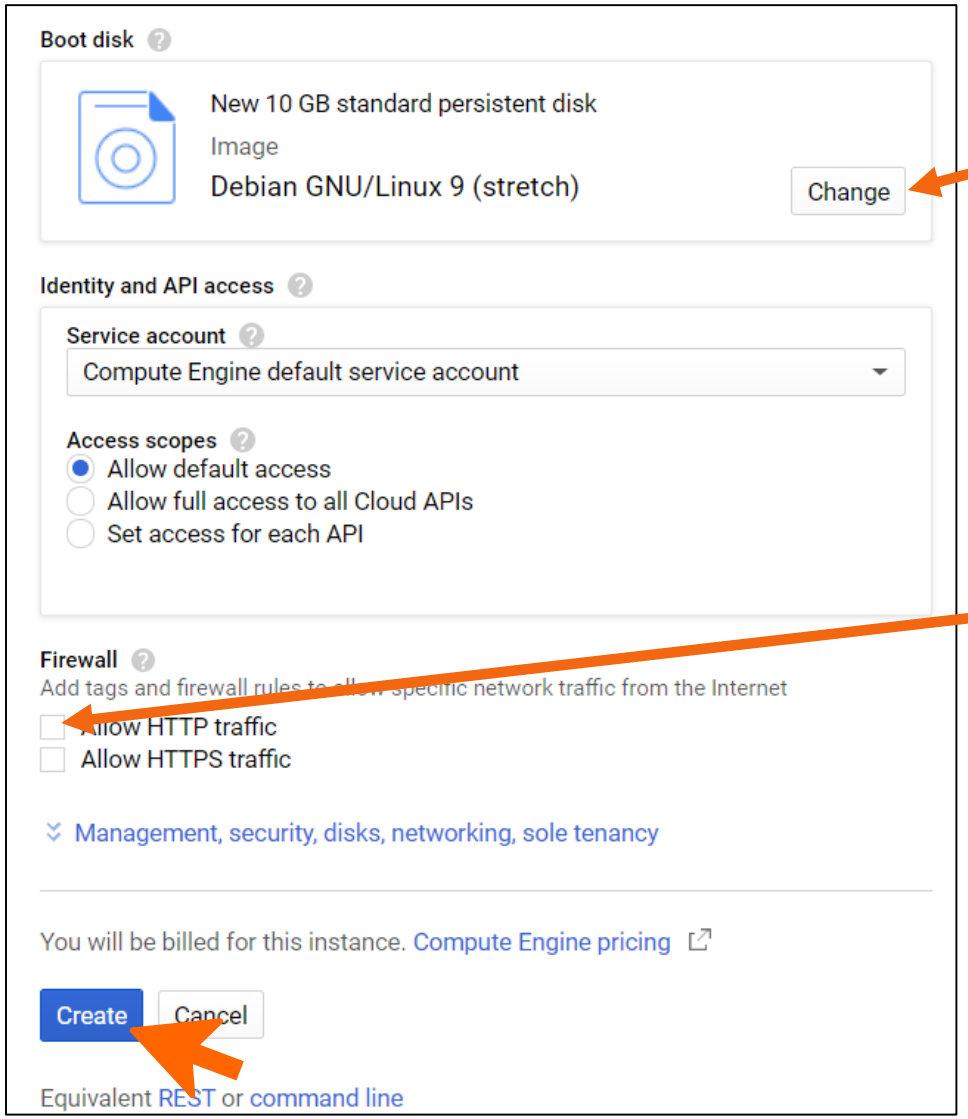
Name you VM instance
e.g. “rstudio-shiny-servers”

You can select a different region/zone to specify the location the resource is used and where the data is stored.

Select the VM machine type
e.g. “n1-standard-1” is OK

The “stronger” the machine type (more CPUs and/or more memory) the more expensive it is to run!





Select the OS image.
We will use “Ubuntu 16.04 LTS”

Other OS images may also work but you would need to adjust the installation procedure.

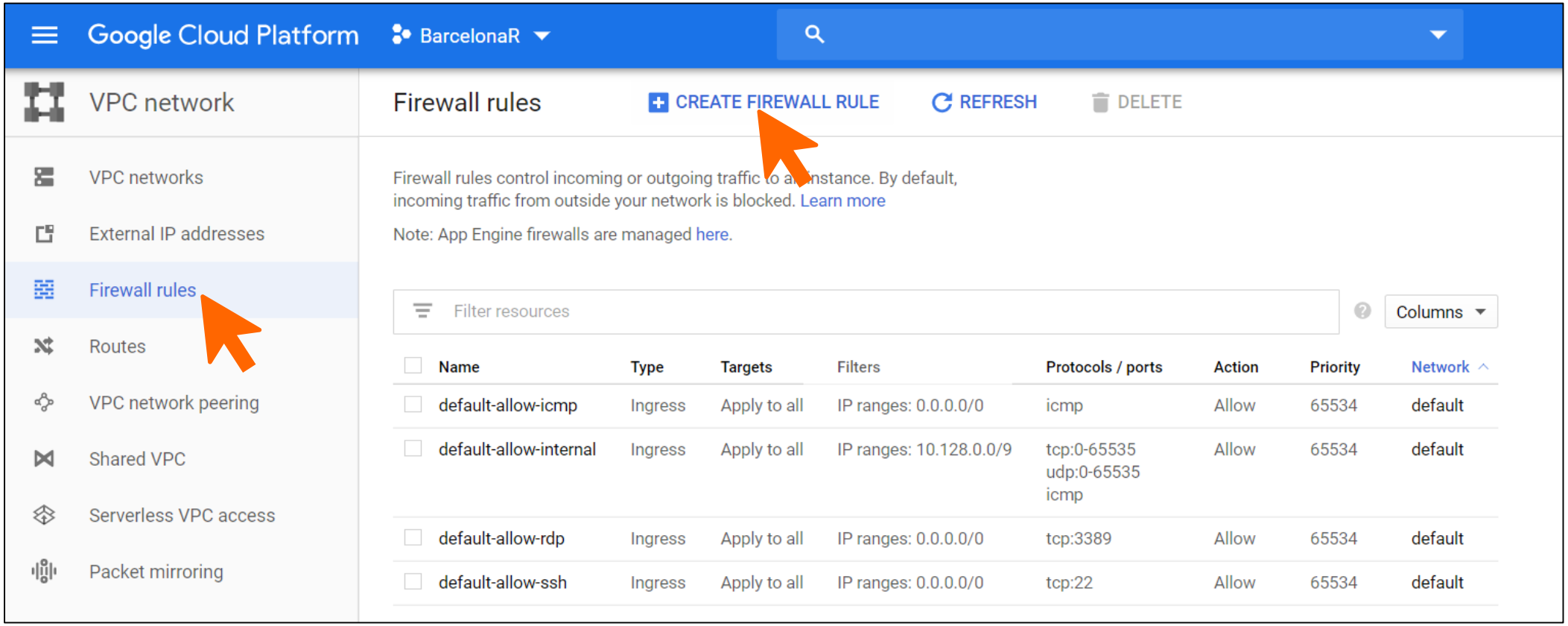
Check the “Allow HTTP traffic” box to allow incoming traffic.

We will later setup specific Firewall rules to allow incoming traffic to RStudio and Shiny servers

VM instance is running!

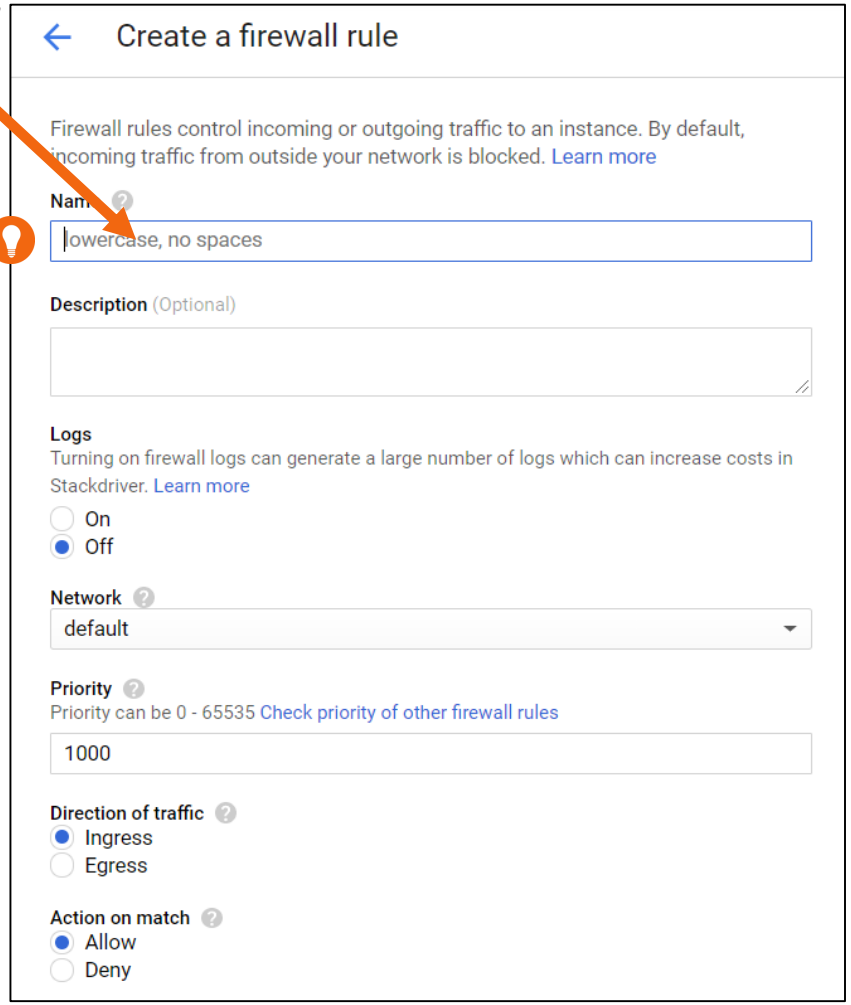
Make a note of this IP address...
it will be important!

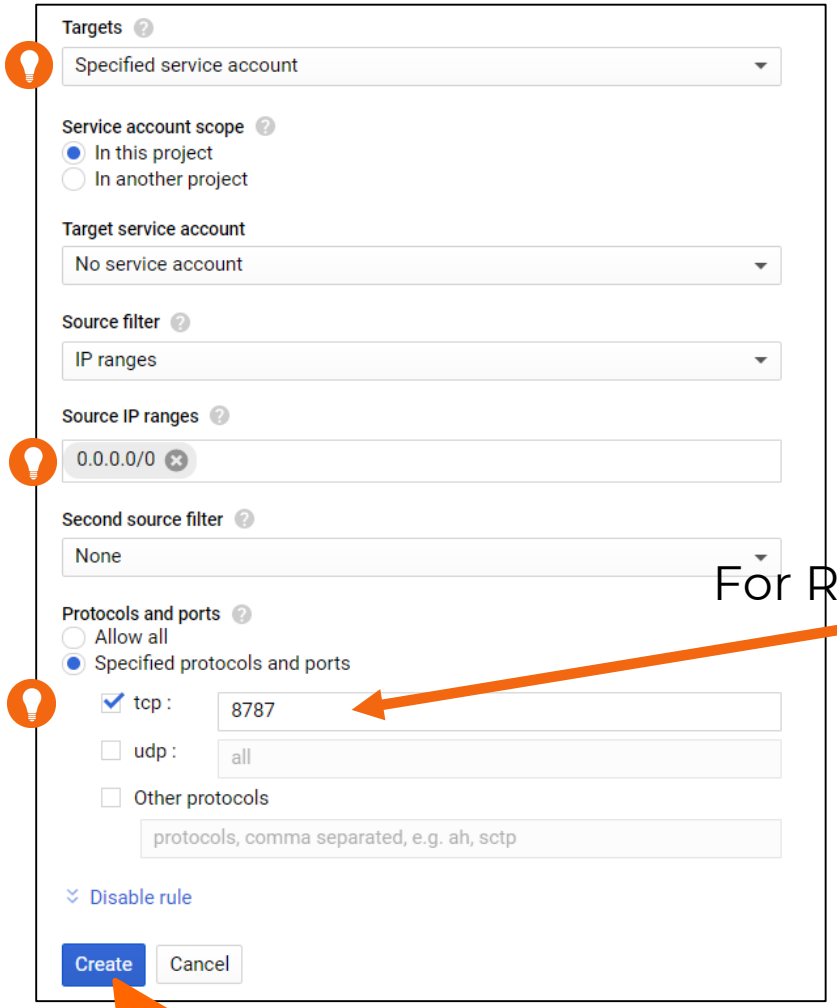
Create a firewall rule – Go to VPC network



Configuring firewall rules to allow access via ports 8787 and 3838 means that you and others can access RStudio and Shiny servers from a web browser such as Chrome

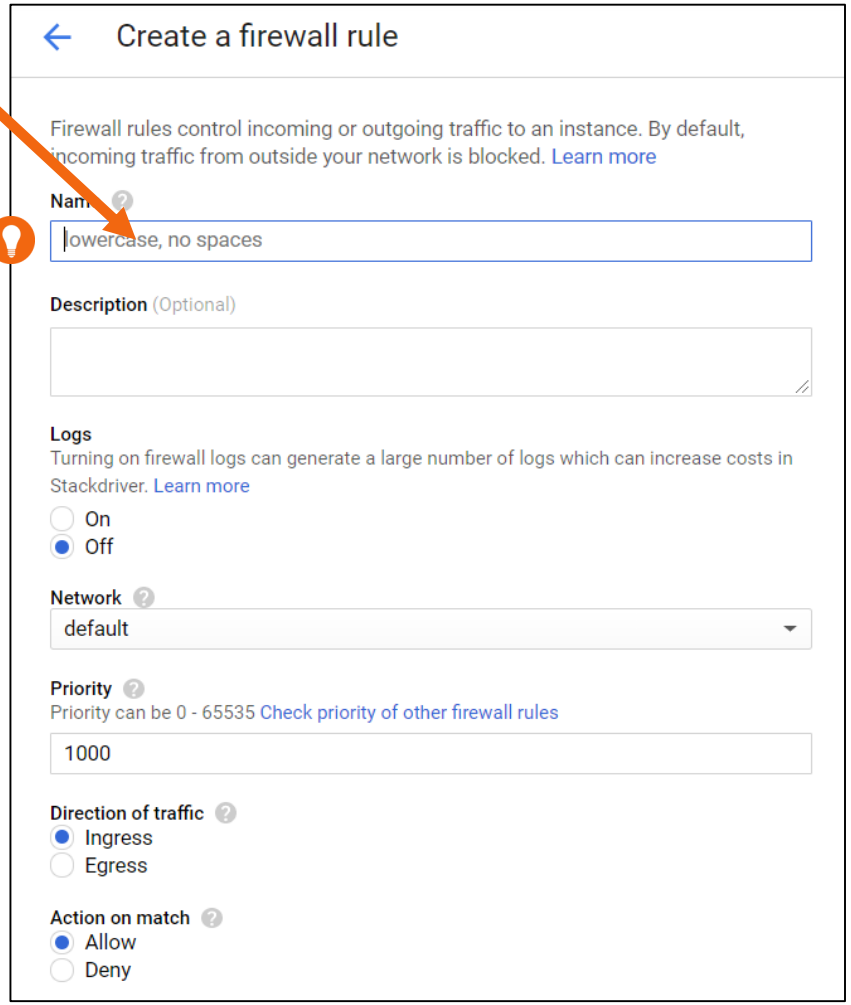
“rstudio”

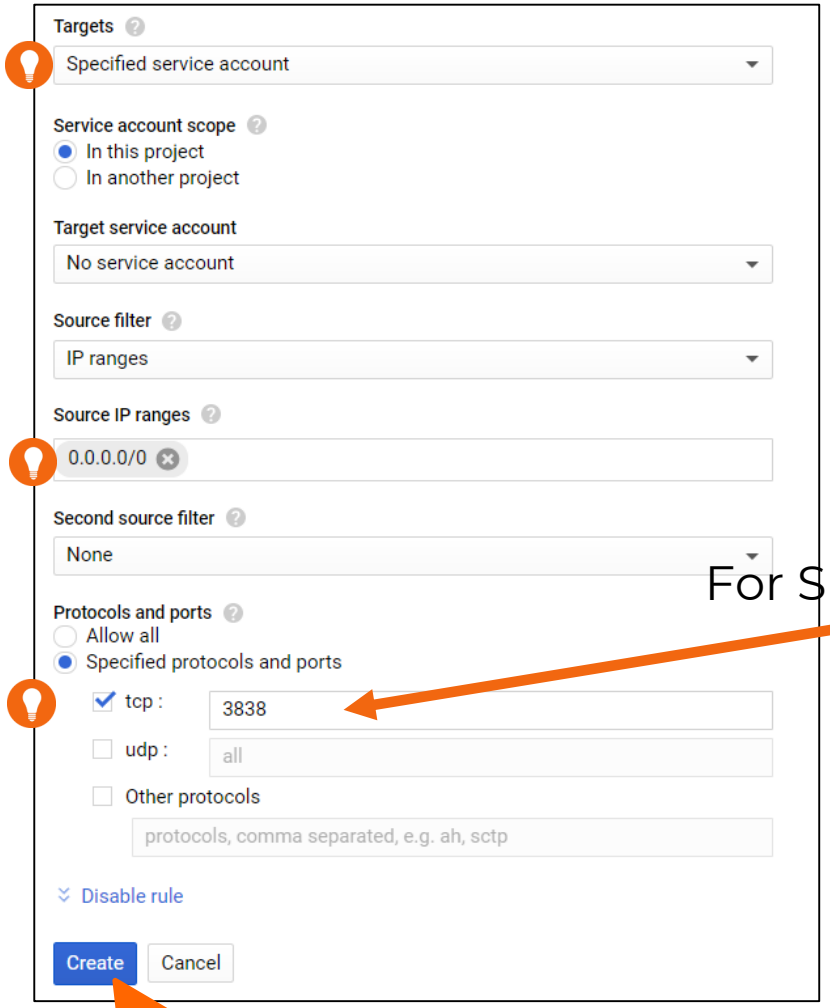




For RStudio server

“shiny”





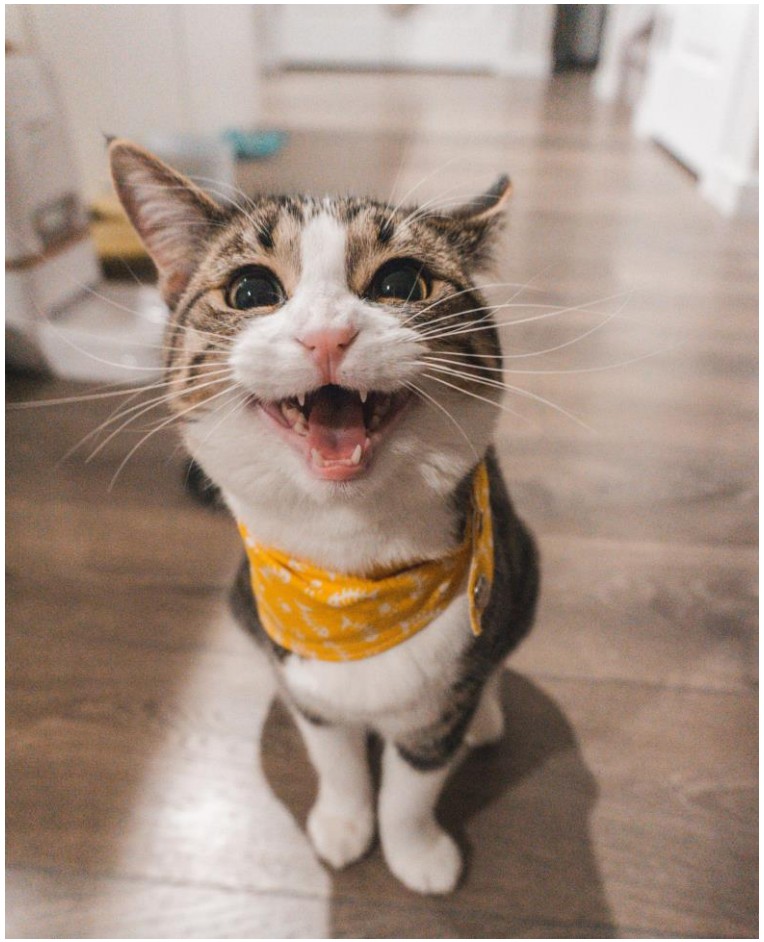
For Shiny server

Filter resources

Columns

Name	Type	Targets	Filters	Protocols / ports	Action	Priority	Network
default-allow-http	Ingress	http-server	IP ranges: 0.0.0.0/0	tcp:80	Allow	1000	default
rstudio	Ingress	Apply to all	IP ranges: 0.0.0.0/0	tcp:8787	Allow	1000	default
shiny	Ingress	Apply to all	IP ranges: 0.0.0.0/0	tcp:3838	Allow	1000	default
default-allow-icmp	Ingress	Apply to all	IP ranges: 0.0.0.0/0	icmp	Allow	65534	default
default-allow-internal	Ingress	Apply to all	IP ranges: 10.128.0.0/9	tcp:0-65535 udp:0-65535 icmp	Allow	65534	default
default-allow-rdp	Ingress	Apply to all	IP ranges: 0.0.0.0/0	tcp:3389	Allow	65534	default
default-allow-ssh	Ingress	Apply to all	IP ranges: 0.0.0.0/0	tcp:22	Allow	65534	default

Setup RStudio server

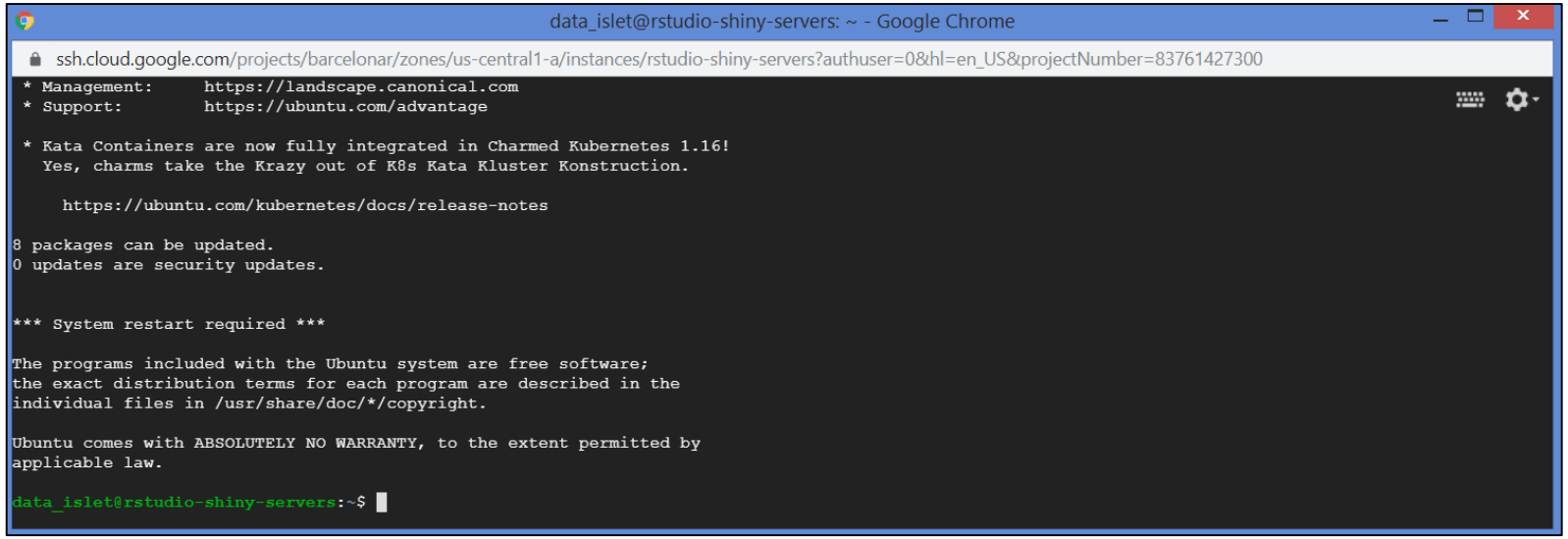


In order to setup RStudio server you will need to do the following:

1. Connect to your VM instance (via Secure Shell - SSH)
2. Update and Upgrade stuff
3. Install R
4. Install R packages
5. Install RStudio server
6. Create a user
7. Access RStudio server! 🥳

Remember you need to run commands as the "superuser" = `sudo`

Live Demo Part 2 – Connect to VM



Update / Upgrade

```
data_islet@rstudio-shiny-servers:~$ sudo apt-get update
...

data_islet@rstudio-shiny-servers:~$ sudo apt-get upgrade
...

data_islet@rstudio-shiny-servers:~$ sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys
E298A3A825C0D65DFD57CBB651716619E084DAB9
...

data_islet@rstudio-shiny-servers:~$ sudo echo "deb https://cloud.r-project.org/bin/linux/ubuntu
xenial-cran35/" | sudo tee -a /etc/apt/sources.list
...


data_islet@rstudio-shiny-servers:~$ sudo apt-get update
...
```

You might need: `sudo apt-get install dirmngr`

Install R / Install packages

```
data_islet@rstudio-shiny-servers:~$ sudo apt-get install r-base r-base-dev
...
Do you want to continue? [Y/n] Y

data_islet@rstudio-shiny-servers:~$ sudo apt-get install libcurl4-openssl-dev libssl-dev libxml2-
dev
...
Do you want to continue? [Y/n] Y
...

data_islet@rstudio-shiny-servers:~$ sudo R
...

data_islet@rstudio-shiny-servers:~$ install.packages(c('shiny', 'rmarkdown', 'dplyr'))
...

q("no")
```

Time for a break and let it run!

Install RStudio server & add a user

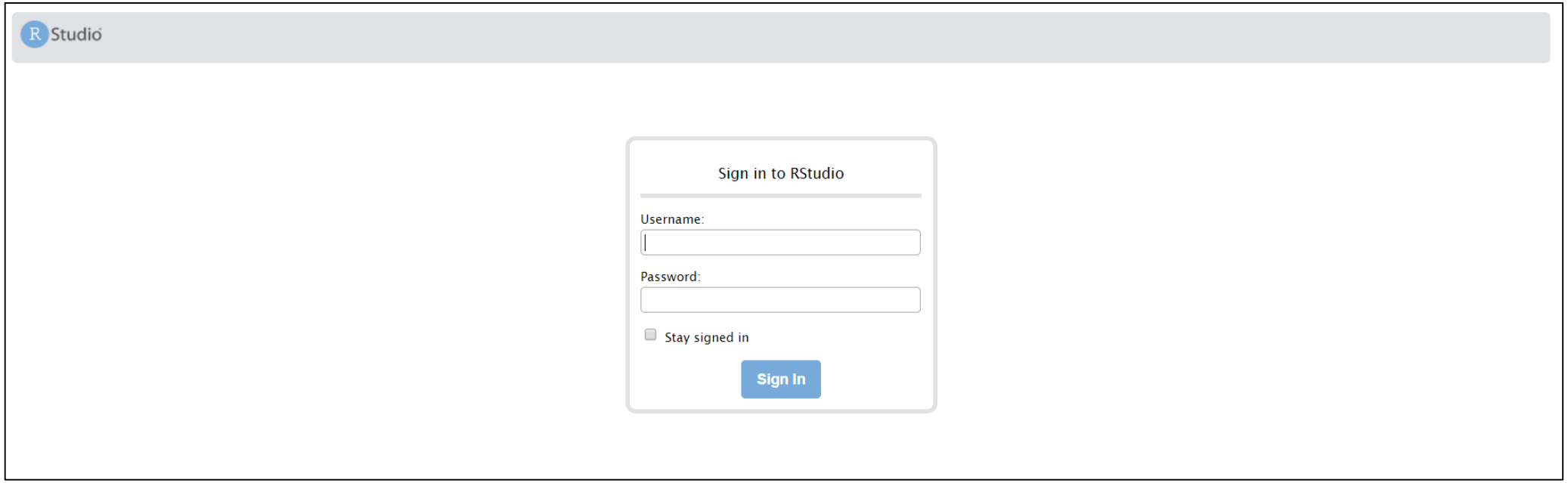
```
data_islet@rstudio-shiny-servers:~$ sudo gpg --keyserver keys.gnupg.net --recv-keys
3F32EE77E331692F
...

data_islet@rstudio-shiny-servers:~$ sudo apt-get install gdebi-core
...

data_islet@rstudio-shiny-servers:~$ wget
https://download2.rstudio.org/server/trusty/amd64/rstudio-server-1.2.5019-amd64.deb
...

data_islet@rstudio-shiny-servers:~$ sudo gdebi rstudio-server-1.2.5019-amd64.deb
...
Do you want to install the software package? [y/N]: Y
...

data_islet@rstudio-shiny-servers:~$ sudo adduser <username>
Enter new UNIX password:
Retype new UNIX password:
...
Is the information correct? [Y/n] Y
```

Choose your <username>

RStudio server is running!

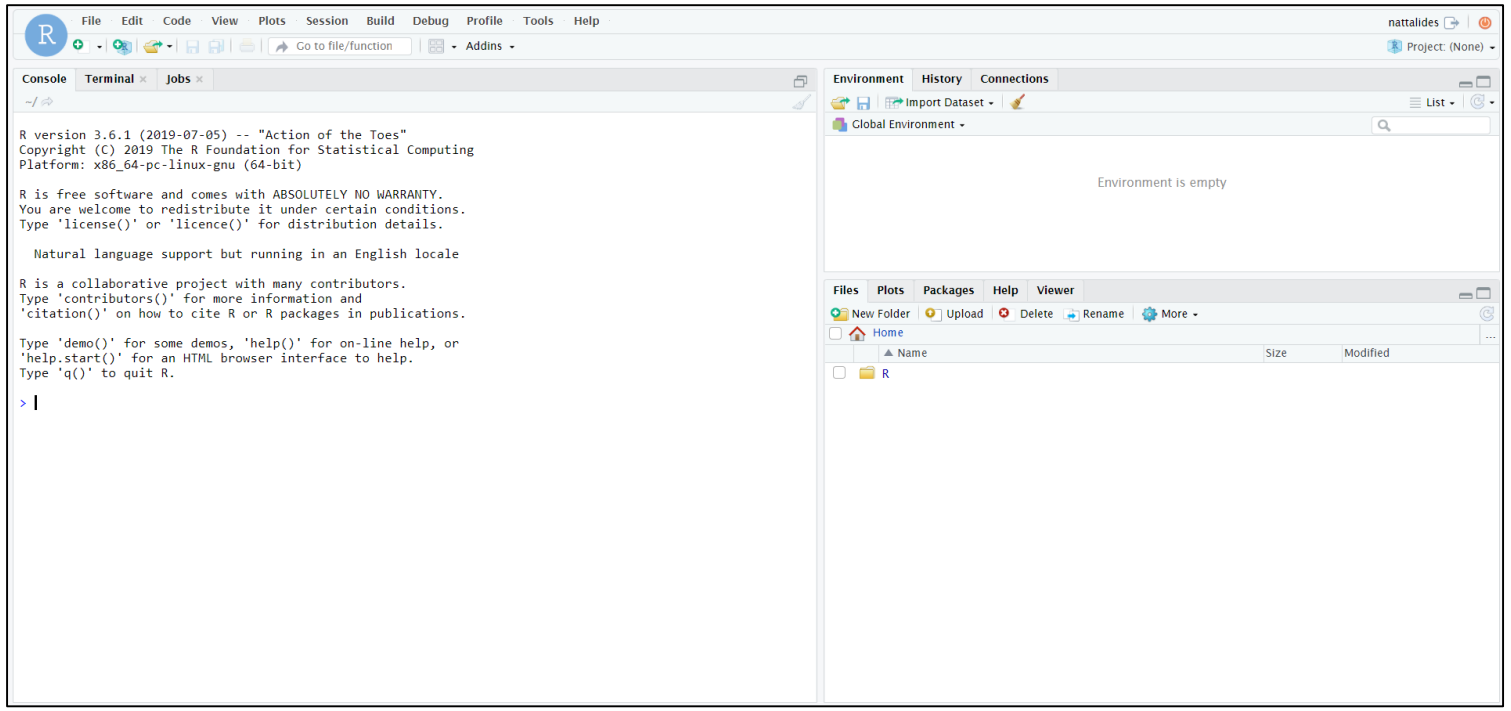
In a web browser navigate to the following address: http://<External IP>:8787

Where <External IP> is found ...

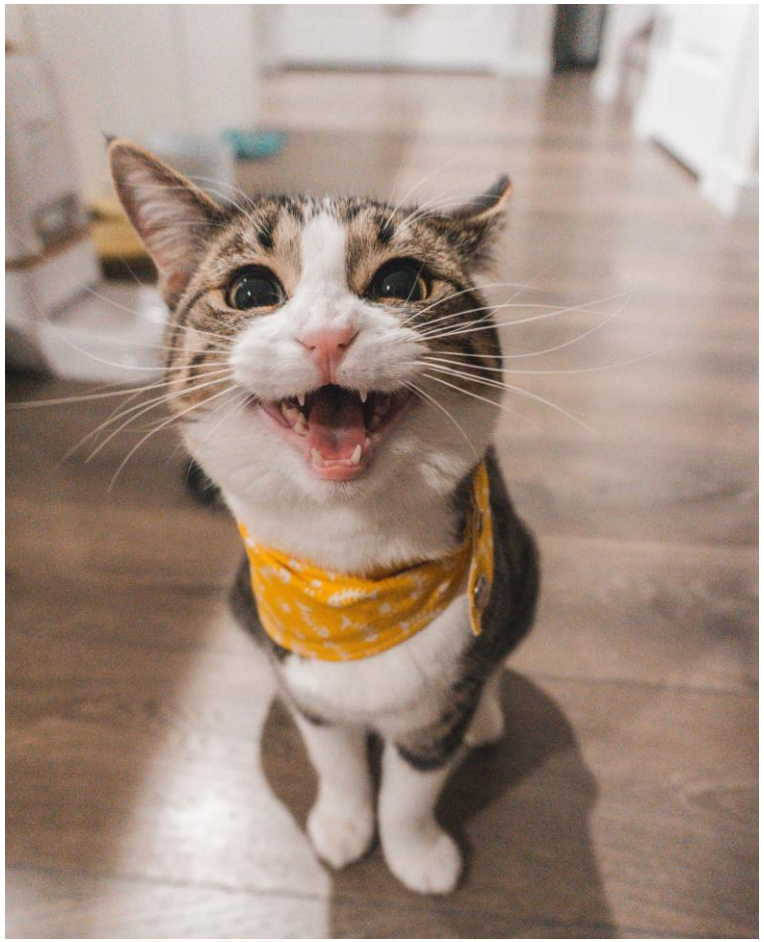


Your chosen <username> and <password>

RStudio server is running!



Setup Shiny server

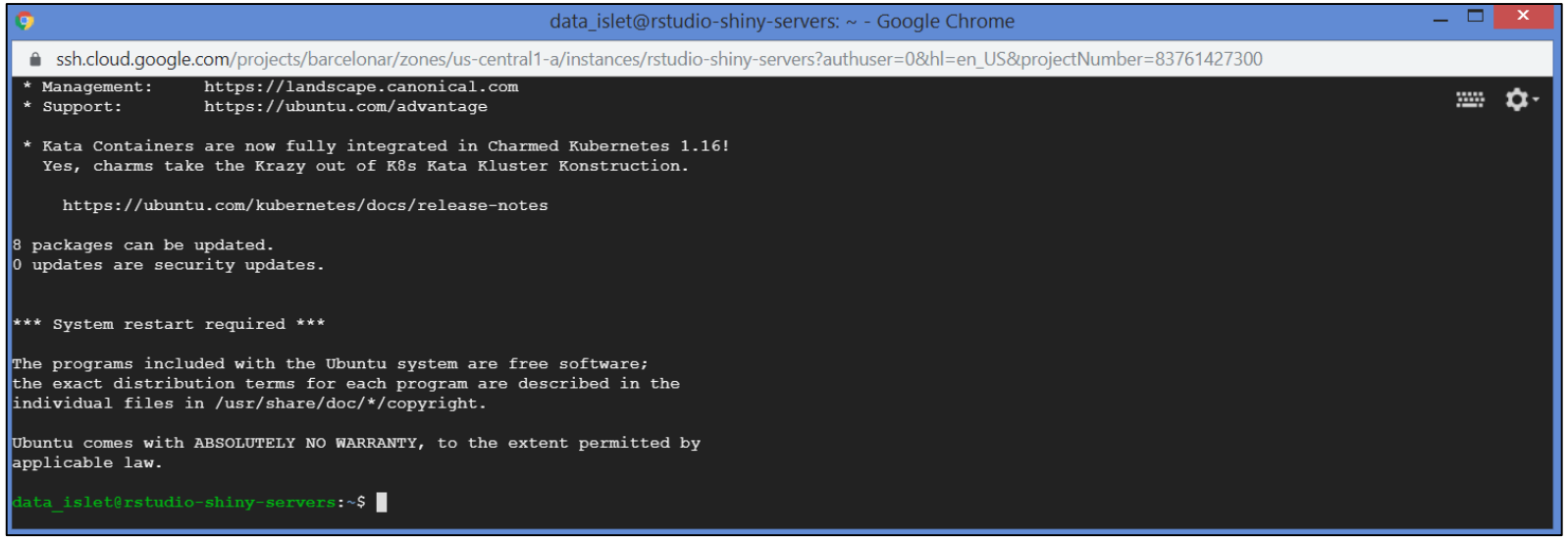


In order to setup Shiny server you will need to do the following:

1. Connect to your VM instance (via SSH)
2. Install Shiny server
3. Check shiny server status
4. Success! 🥳

Remember you need to run commands as the “superuser”

Live Demo Part 3 – Connect to VM



Install Shiny server & check status

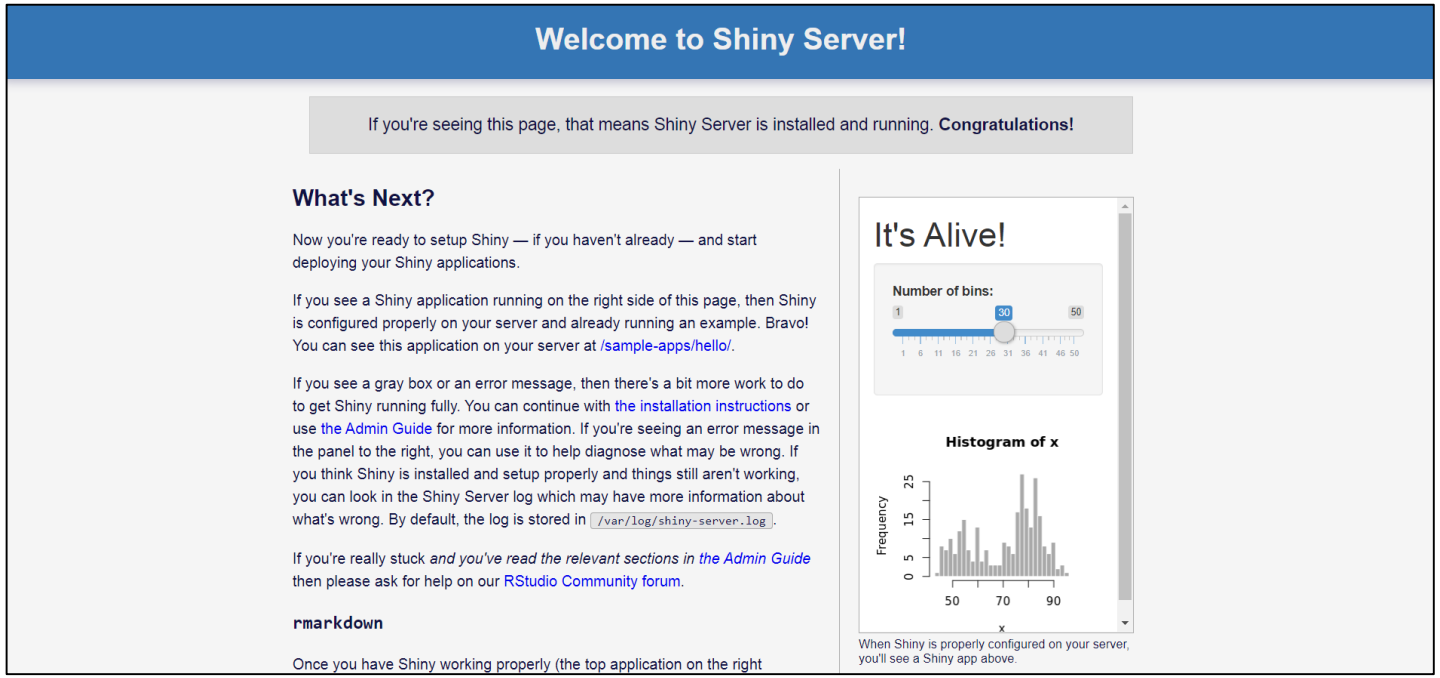
```
data_islet@rstudio-shiny-servers:~$ sudo apt-get install gdebi-core
...

data_islet@rstudio-shiny-servers:~$ wget https://download3.rstudio.org/ubuntu-14.04/x86_64/shiny-server-1.5.12.933-amd64.deb
...

data_islet@rstudio-shiny-servers:~$ sudo gdebi shiny-server-1.5.12.933-amd64.deb
...
Do you want to install the software package? [y/N]: Y
...

data_islet@rstudio-shiny-servers:~$ sudo systemctl status shiny-server
...
```

- To stop the shiny server: `sudo systemctl stop shiny-server`
- To start the shiny server: `sudo systemctl start shiny-server`

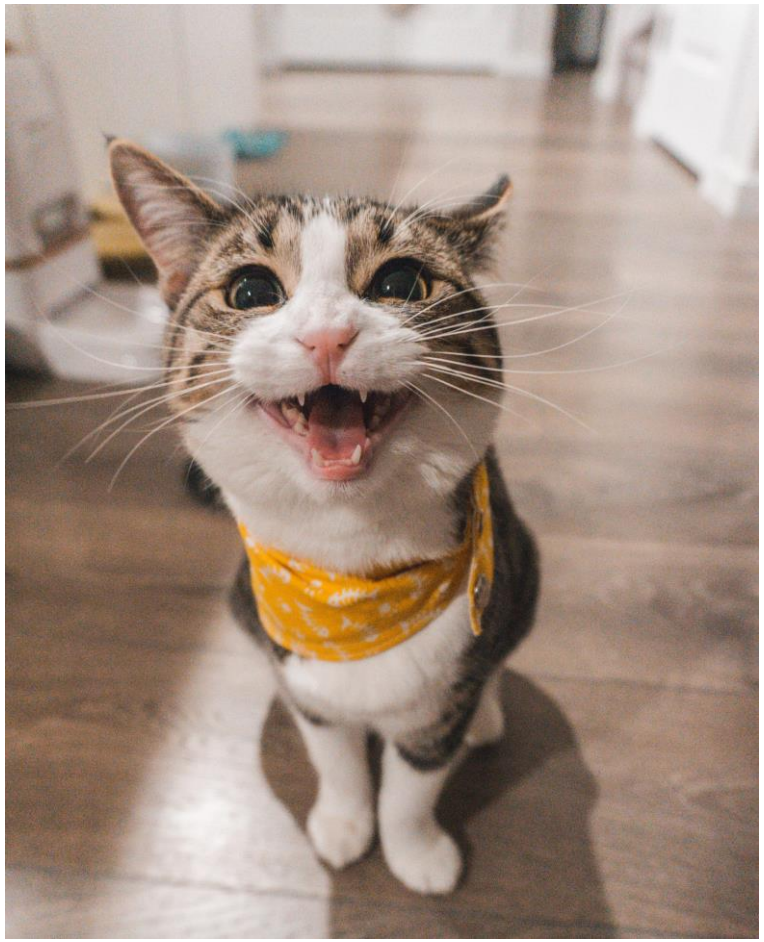
Shiny server is running!

In a web browser navigate to the following address: http://<External IP>:3838

Where <External IP> is found …



Live Demo Part 4 - Host a shiny app online



In order to host a shiny app online you will need to do the following:

1. Log in to your RStudio server user account
2. Create a folder that will contain the shiny app scripts
3. Write the ui.R script
4. Write the server.R script
5. Create a symbolic link to the folder * 😱
6. Test that it works
7. Share the URL! 🥳

*It's easier than it sounds

Example ui.R script

```
# Define UI for application
ui <- fluidPage(

  # Application title
  titlePanel("Hello BarcelonaR!"),

  # Sidebar with an input
  sidebarLayout(
    sidebarPanel(
      textInput("text_input", "Input text here:")
    ),

    # Main with output
    mainPanel(
      textOutput("text_output")
    )
  )
)
```

Example server.R script

```
library(shiny)

# Define server logic and R code
server <- function(input, output) {

  output$text_output <- renderText({
    # Display text input
    paste("You typed:", input$text_input)
  })
}
```

Create symbolic link

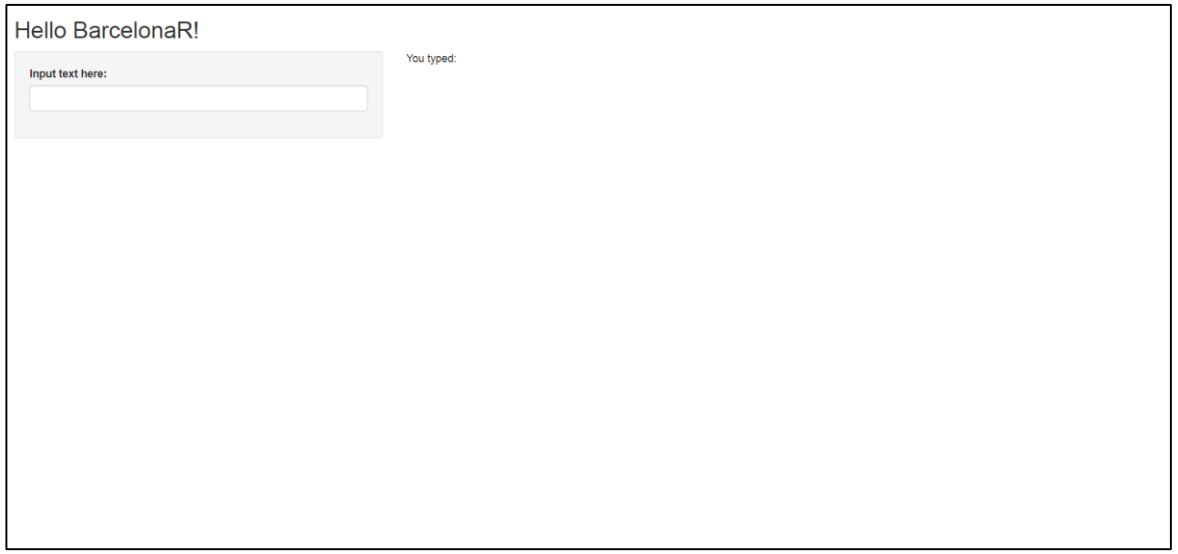
```
data_islet@rstudio-shiny-servers:~$ sudo ln -s /home/<username>/my-app /srv/shiny-server/my-app
```

Your chosen RStudio server <username>

Your shiny app is hosted online!

In a web browser navigate to the following address:
http://<External IP>:3838/<app-folder>

Where <External IP> is found just as before and <app-folder> is the name of the folder that contains the ui.R and server.R scripts



If you stop and start the GCP VM instance you will most likely get assigned a different <External IP> address!

Other improvements

- If your shiny app code is becoming larger and more complex then why not Build a Production Grade Shiny App with {golem}
- Write your shiny app in a project with code version control (such as GitHub)
- Make <External IP> static
- Add user authentication to Shiny Server with Nginx
- Create an SSL certificate for Shiny server (https)
- Control who can access your shiny apps (via GCP firewall settings)
- Create custom domains for RStudio server, Shiny server and for your shiny apps

Check out: https://docs.rstudio.com/shiny-server/ for a useful guide on how to customise other aspects of the Shiny server

Tips for troubleshooting

- If your shiny app crashes you can use the stored shiny server logs to view what happened
  1) navigate to `cd /var/log/shiny-server/`
  2) list of available logs `ls`
  3) view log `cat <file-name>.log`
- In most cases the issue might be due to file and/or folder permissions – you might need to give root permissions to read/write/execute
- You might need to install necessary libraries using:
  1) `sudo R`
  2) `install.packages('magrittr')`

  so that they are available at root level and to all users

Barcelona R

Thank you to our sponsors and partners!