

Universidad Francisco Marroquín
Análisis de Oportunidad de Negocios
Catedrático: Carlos Zelada
Auxiliar: Obed Espinoza

Laboratorio Shiny Docker: Tutorial AWS

:

Manuel Palencia 20160391
Juan Barillas 20170490
Guatemala 03 de Noviembre, 2020

Introducción

En el siguiente tutorial podremos utilizar Amazon Web Services (AWS) para poder hacer un deploy de un contenedor de Docker en el cual correremos una imagen que contiene nuestra Shiny App que desarrollamos previamente sobre el dataset de Streamlit. Esperamos les sea de su interés el tutorial y los pueda ayudar en futuros trabajos.

Pueden ver nuestro demo funcionando en el siguiente link <http://3.19.60.38:3838/>

Así también como pueden ver nuestra imagen dentro de Docker hub
<https://hub.docker.com/repository/docker/mapg23/shiny-steam-analysis>

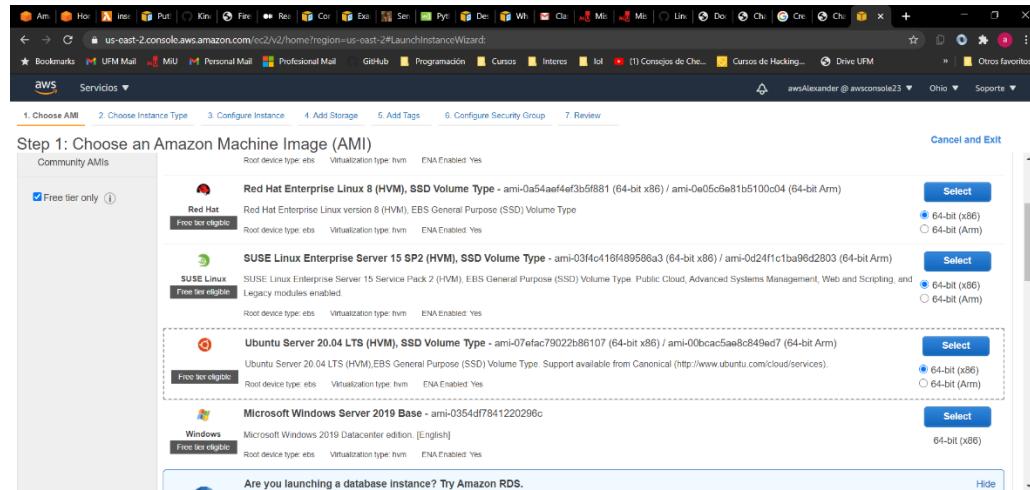
AWS Account

Ingresar a <https://aws.amazon.com/> cree su cuenta siguiendo el paso a paso del sitio web o ingrese a su cuenta si tiene una.

AWS EC2 Server Setup

En esta primera parte del tutorial montaremos una instancia de un EC2 en AWS.

Seleccionaremos un Amazon Machine Image (AMI) que nos funcione para el tutorial. En este tutorial seleccionaremos un Ubuntu Server con una t2.micro que tenga 25GB de almacenamiento.



Step 1: Choose an Amazon Machine Image (AMI)

Community AMIs

- Red Hat Enterprise Linux 8 (HVM), SSD Volume Type** - ami-0a54ae4ef3b5f881 (64-bit x86) / ami-0e05c6e81b5100c04 (64-bit Arm)
 64-bit (x86)
 64-bit (Arm)
 Select
- SUSE Linux Enterprise Server 15 SP2 (HVM), SSD Volume Type** - ami-03fc4c16f48956a3 (64-bit x86) / ami-0d24ff1c1ba96d2803 (64-bit Arm)
 64-bit (x86)
 64-bit (Arm)
 Select
- Ubuntu Server 20.04 LTS (HVM), SSD Volume Type** - ami-07efac79022bb86107 (64-bit x86) / ami-00bcac5ae8c849ed7 (64-bit Arm)
 64-bit (x86)
 64-bit (Arm)
 Select
- Microsoft Windows Server 2019 Base** - ami-0354df7841220296c
 64-bit (x86)
 Select

Are you launching a database instance? Try Amazon RDS. [Hide](#)



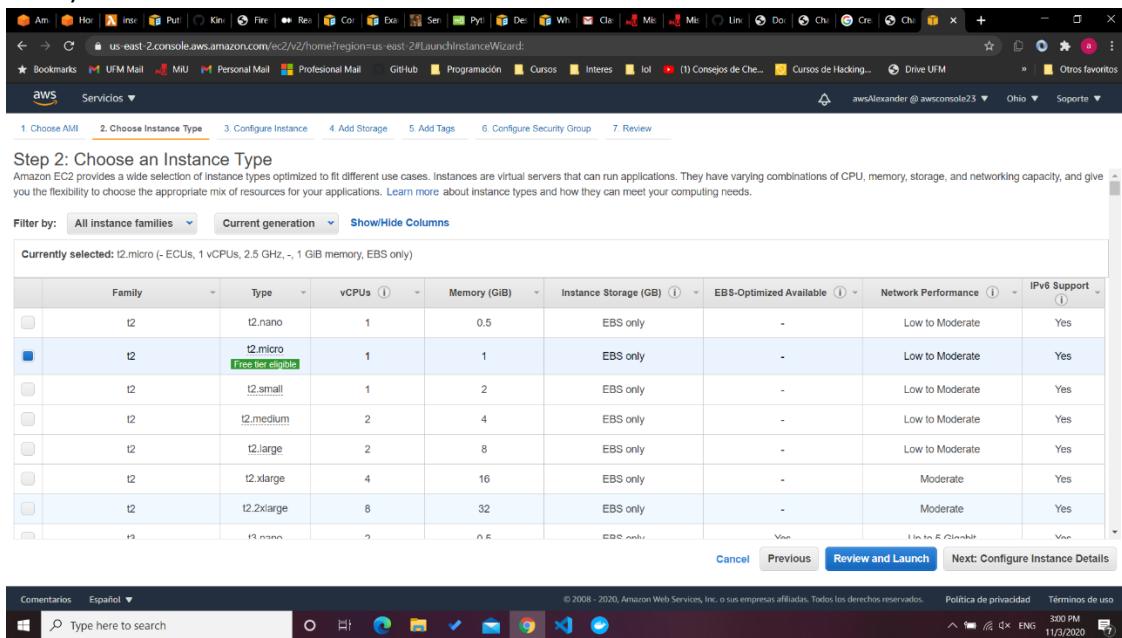
Comments Español ▾

Type here to search

© 2008 - 2020, Amazon Web Services, Inc. o sus empresas afiliadas. Todos los derechos reservados. [Política de privacidad](#) [Términos de uso](#)

3:00 PM ENG 11/3/2020

Seleccione el tipo de instancia entre las opciones (Seleccionaremos CPU y RAM)



Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. Learn more about instance types and how they can meet your computing needs.

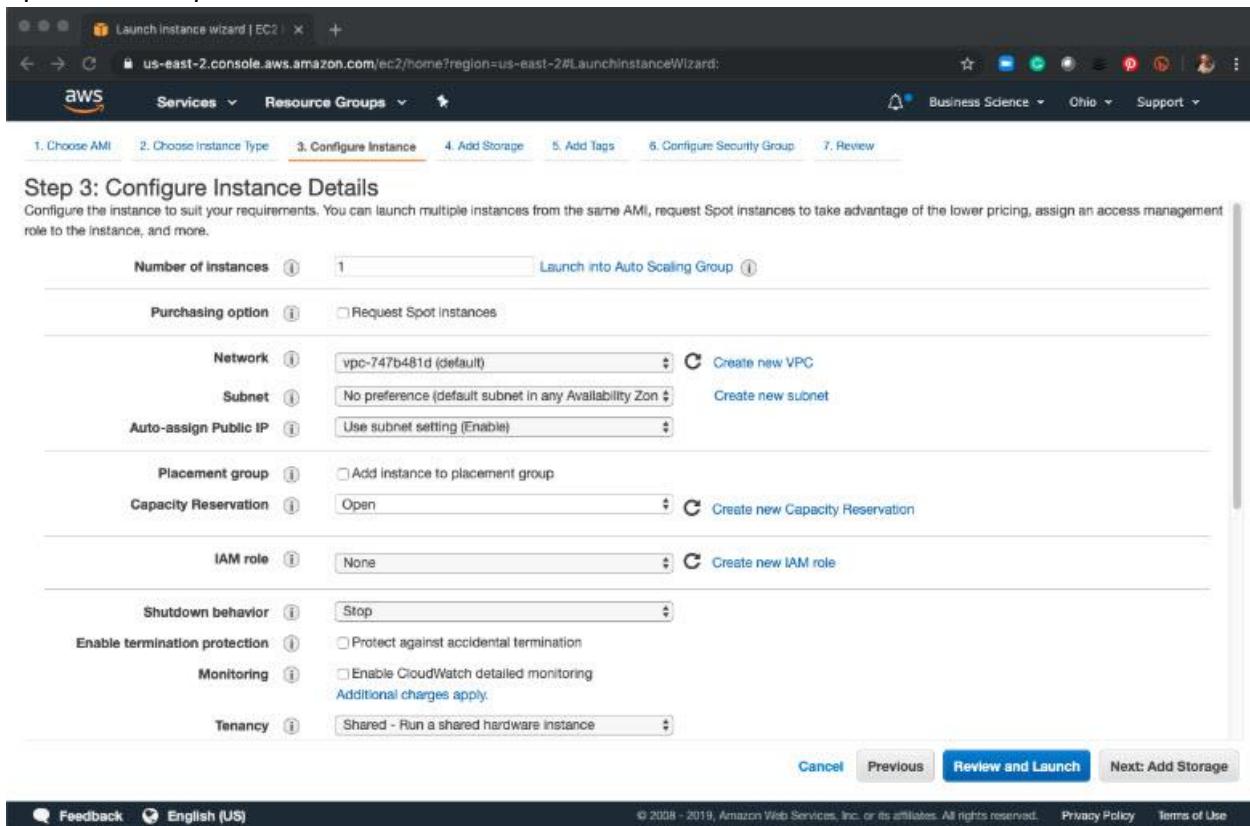
Filter by: All instance families Current generation Show/Hide Columns

Currently selected: t2.micro (- ECUs, 1 vCPUs, 2.5 GHz, -, 1 GiB memory, EBS only)

Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
t2	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
t2	t2.micro <small>Free tier eligible</small>	1	1	EBS only	-	Low to Moderate	Yes
t2	t2.small	1	2	EBS only	-	Low to Moderate	Yes
t2	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
t2	t2.large	2	8	EBS only	-	Low to Moderate	Yes
t2	t2.xlarge	4	16	EBS only	-	Moderate	Yes
t2	t2.2xlarge	8	32	EBS only	-	Moderate	Yes
t3	t3.nano	0.5	0.5	EBS Only	Via	Low to Moderate	Via

Cancel Previous Review and Launch Next: Configure Instance Details

A continuación configuraremos nuestra instancia con las configuraciones que creamos sean optimas estas pueden variar.



Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot Instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of Instances: 1 Launch into Auto Scaling Group

Purchasing option: Request Spot Instances

Network: vpc-747b481d (default) Create new VPC

Subnet: No preference (default subnet in any Availability Zone) Create new subnet

Auto-assign Public IP: Use subnet setting (Enable)

Placement group: Add instance to placement group

Capacity Reservation: Open Create new Capacity Reservation

IAM role: None Create new IAM role

Shutdown behavior: Stop

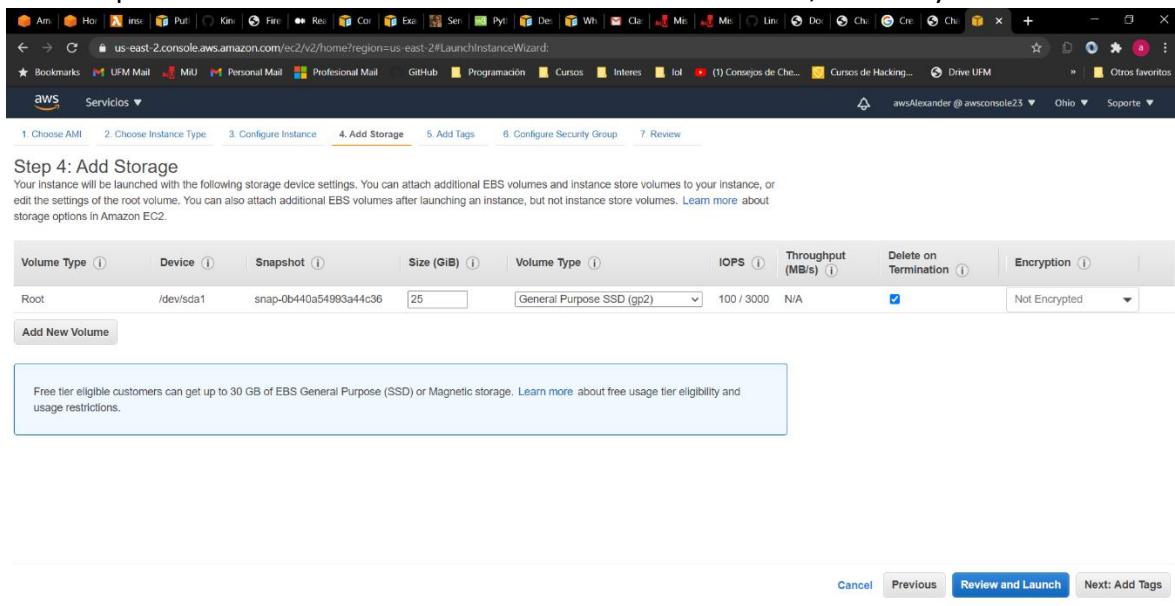
Enable termination protection: Protect against accidental termination

Monitoring: Enable CloudWatch detailed monitoring Additional charges apply.

Tenancy: Shared - Run a shared hardware instance

Cancel Previous Review and Launch Next: Add Storage

Seleccionaremos la cantidad de almacenamiento que permitiremos dentro de nuestra instancia, es importante tener suficiente espacio para almacenar todo lo que necesitamos en este caso la instalación de Docker, RStudio y RServer



Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more](#) about storage options in Amazon EC2.

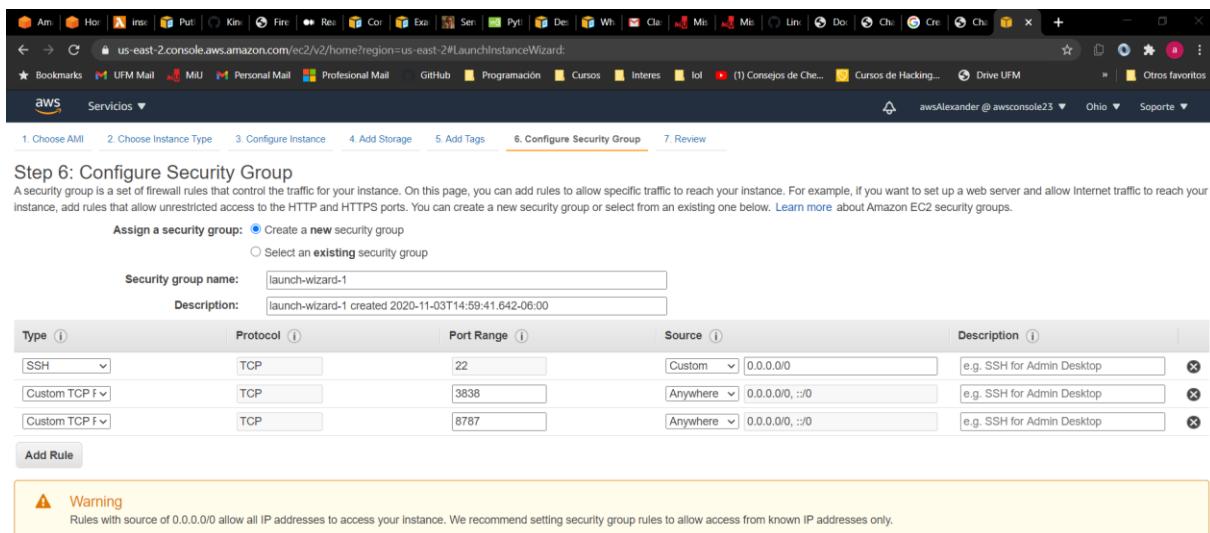
Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encryption
Root	/dev/sda1	snap-0b440a54993a44c36	25	General Purpose SSD (gp2)	100 / 3000	N/A	<input checked="" type="checkbox"/>	Not Encrypted

Add New Volume

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. [Learn more](#) about free usage tier eligibility and usage restrictions.

Cancel Previous Review and Launch Next: Add Tags

Configuraremos el grupo de seguridad para nuestra instancia. En esta debemos asegurarnos que como parte de las reglas se encuentren un Custom TCP con los puertos 3838 y 8787 Estos corresponden a los puertos que utilizaremos para Shiny Server y RStudio Server respectivamente.



Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group: Create a new security group Select an existing security group

Security group name:

Description:

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom <input type="text" value="0.0.0.0/0"/>	e.g. SSH for Admin Desktop
Custom TCP	TCP	3838	Anywhere <input type="text" value="0.0.0.0/0"/>	e.g. SSH for Admin Desktop
Custom TCP	TCP	8787	Anywhere <input type="text" value="0.0.0.0/0"/>	e.g. SSH for Admin Desktop

Add Rule

⚠ Warning
Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Cancel Previous Review and Launch



Comentarios Español ▾

© 2008 - 2020, Amazon Web Services, Inc. o sus empresas afiliadas. Todos los derechos reservados. Política de privacidad Términos de uso

Type here to search

Cancel Previous Review and Launch

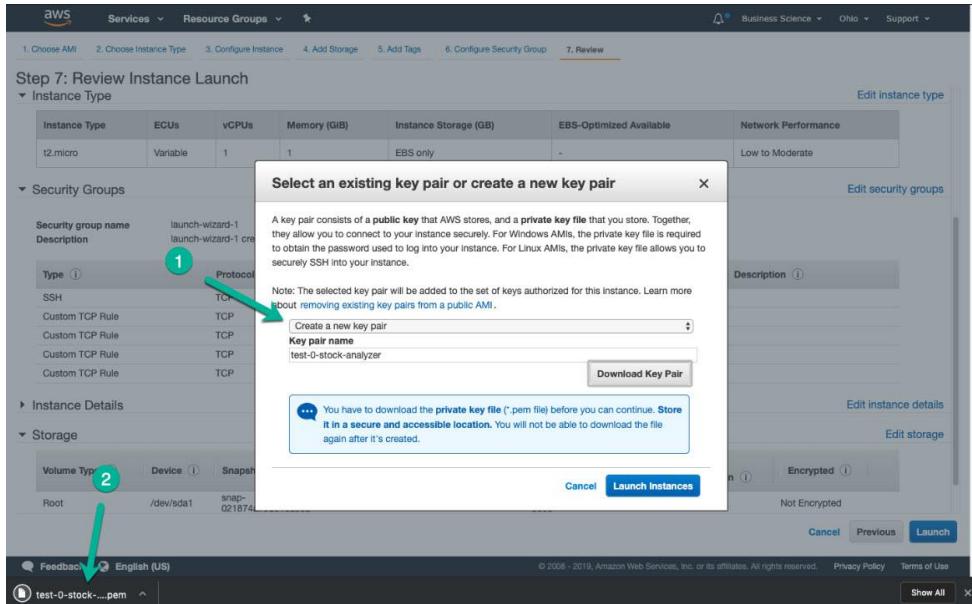
Comments Spanish ▾

© 2008 - 2020, Amazon Web Services, Inc. o sus empresas afiliadas. Todos los derechos reservados. Política de privacidad Términos de uso

Type here to search

Cancel Previous Review and Launch

Luego como paso final necesitamos descargar la llave para poder realizar una conexión hacia la instancia.

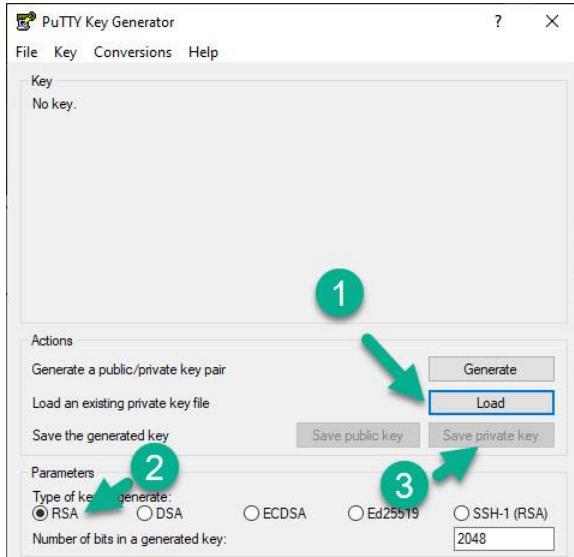


EC2 Connection with SSH

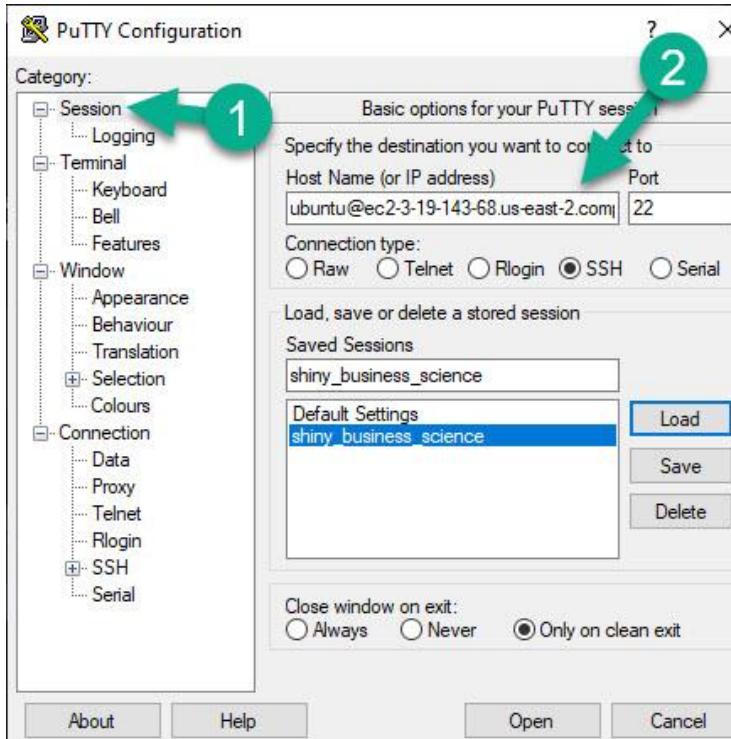
Las instrucciones a continuación nos servirán para poder conectarnos hacia la instancia de EC2 de AWS mediante una consola para poder operar y montar sobre la instancia. Para ello necesitaremos la herramienta de Windows – Putty.

Como primer paso necesitamos descargar la herramienta. La puedes encontrar aquí <https://www.putty.org/>

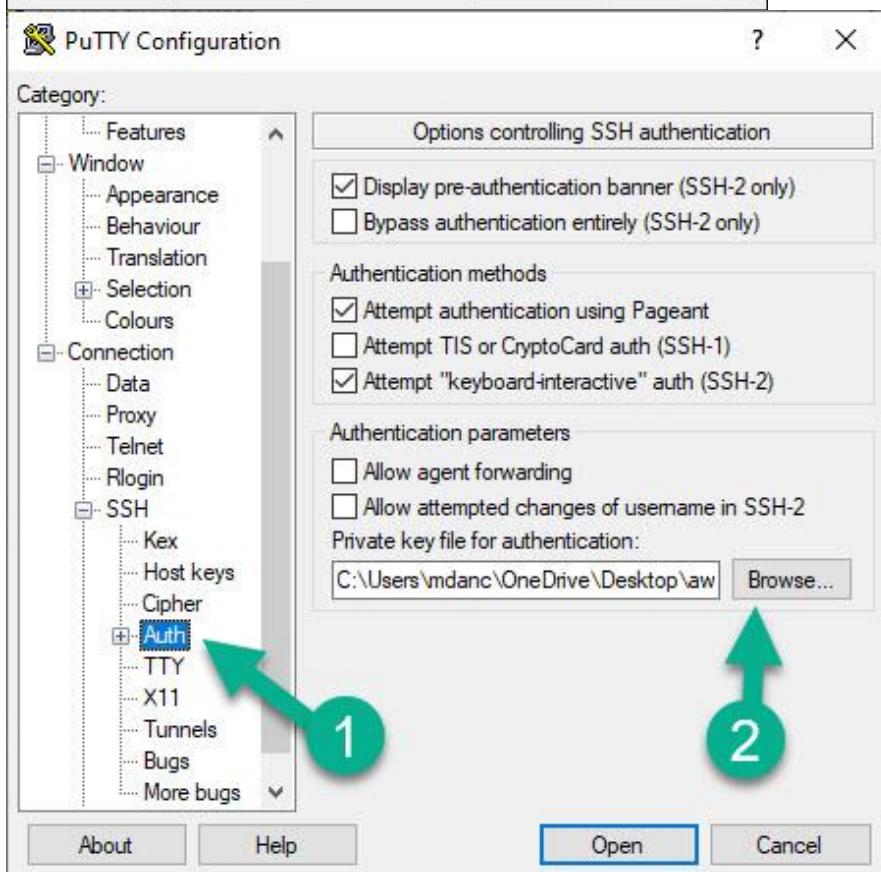
El segundo paso a seguir es convertir nuestro .pem que obtuvimos al final de crear nuestra instancia y debemos abrir el PuttyGen ingresando este archivo para crear un archivo .ppk que nos ayudara a realizar la conexión.



Ahora nos conectaremos a la instancia. En Putty necesitamos agregar el hostname con username y el puerto previamente agregado al security group.



Como podemos observar tenemos Ubuntu como username seguido de la dirección publica de AWS todo esto mapeado al puerto 22 que debe encontrarse en el security group de la instancia.



En la parte de SSH o Auth debemos ingresar nuestro archivo .ppk es importante agregarlo pues sin este no podremos tener una conexión exitosa.



Upgrade and Update Ubuntu

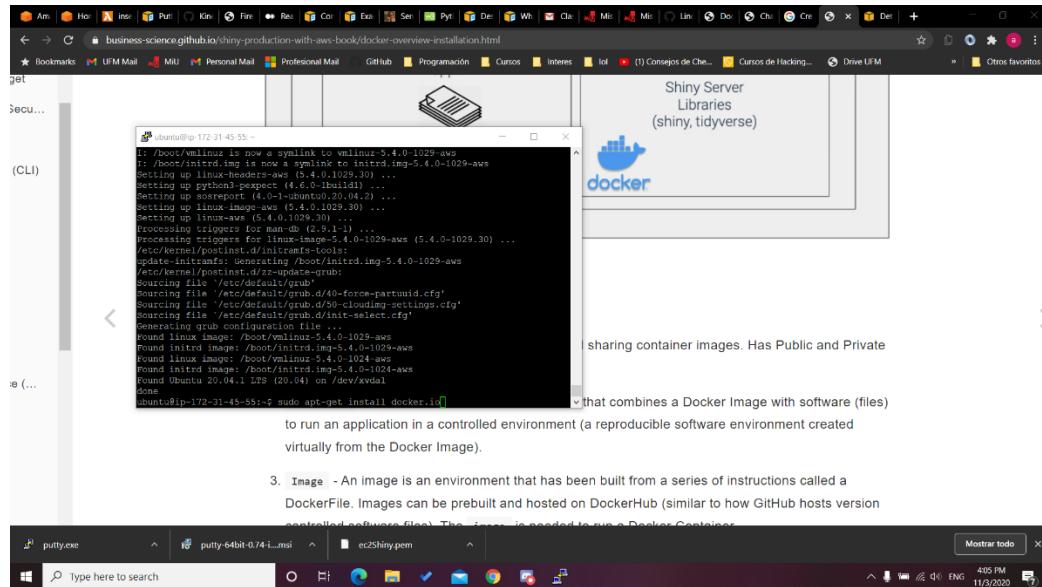
En esta parte estaremos actualizando nuestro sistema de Ubuntu en caso de que no tengamos los últimos paquetes y nos aseguremos de tener las versiones más actuales de las librerías.

- sudo apt update
 - sudo apt-get upgrade -y
 - sudo apt-get dist-upgrade -y

Docker Installation

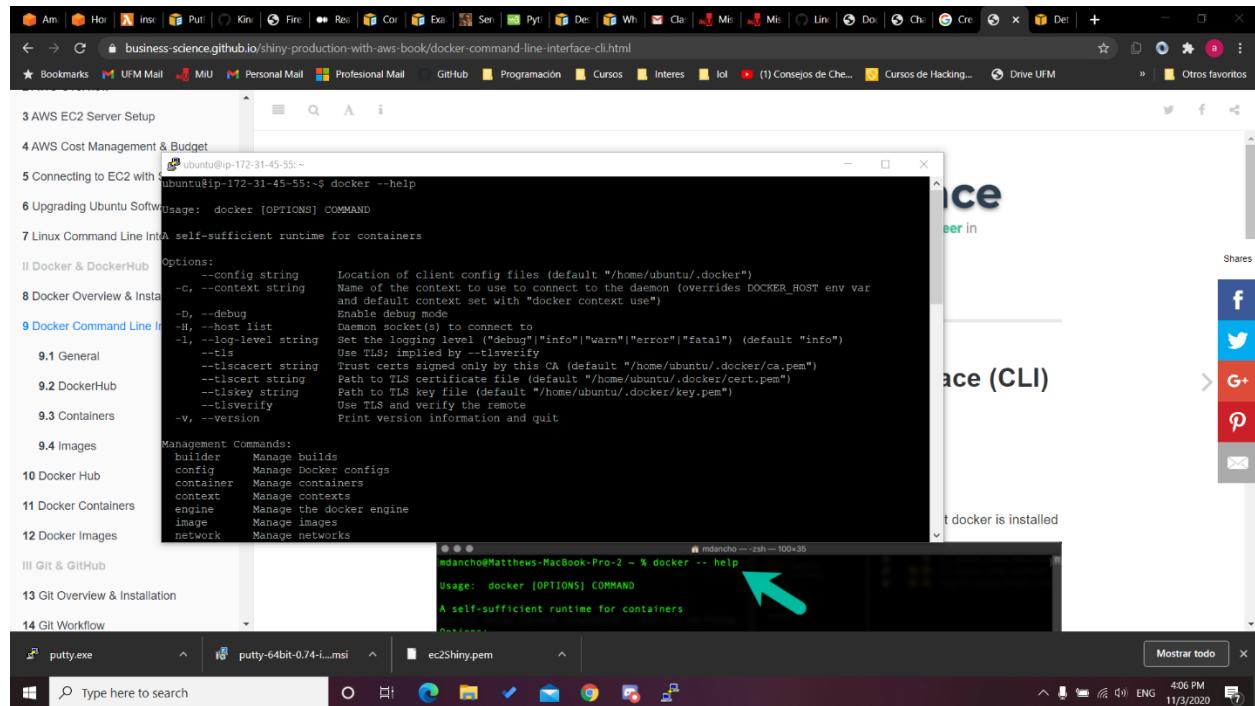
Instalar Docker mediante la siguiente línea de comando

➔ sudo apt-get install docker.io



Asegurarse que Docker corre

➔ docker --help

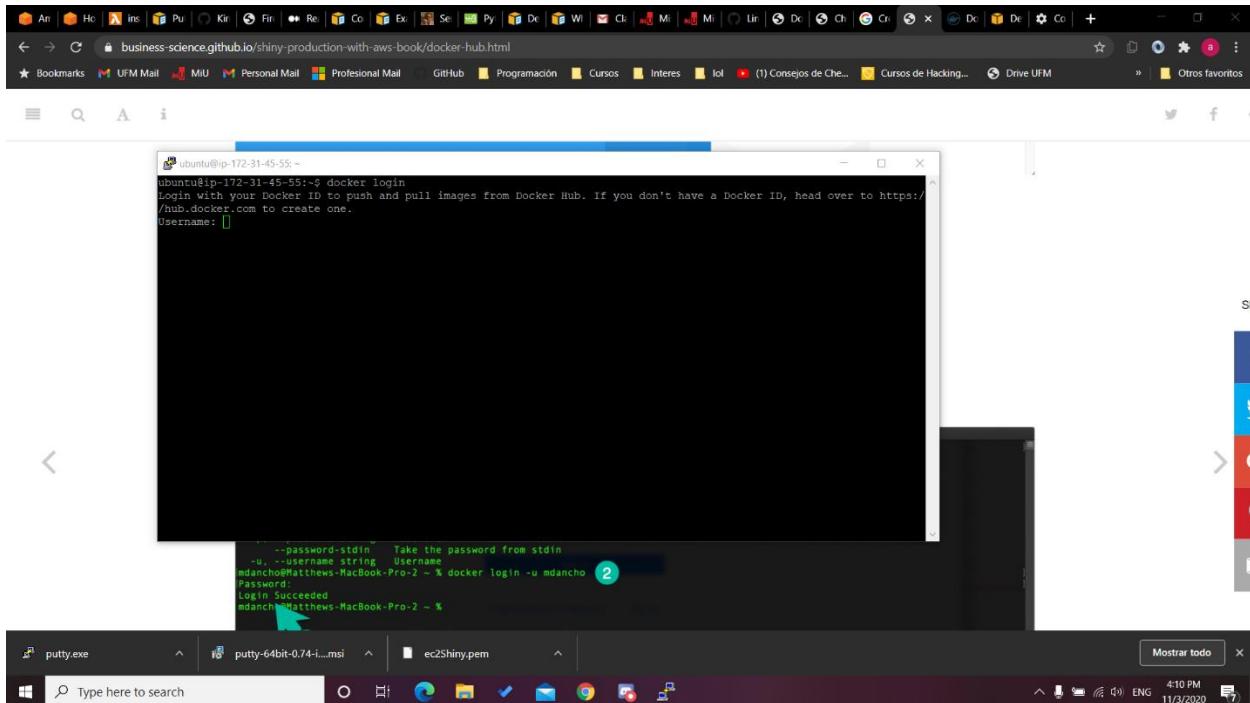


Docker Hub Connection

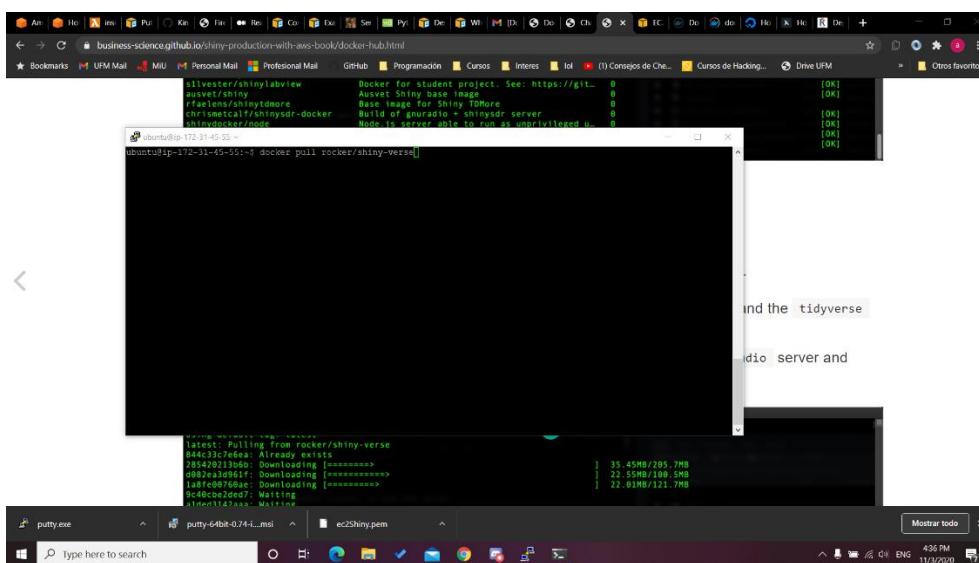
Para continuar con nuestro tutorial ingresa a <https://hub.docker.com/> y crea una cuenta para poder hacer push y pull a imágenes en Docker hub.

Con tu cuenta ingresa mediante la conexión de Putty con el comando Docker login [username]

- ➔ docker login [username]
- ➔ Password: [password]



- ➔ Docker pull rocker/shiny-verse
- ➔ Docker pull rocker/tidyverse





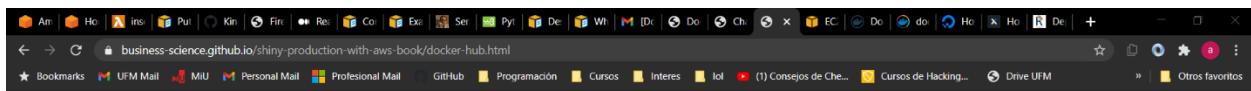
10.5 List the Docker Images

```
Ubuntu@ip-172-31-45-55: ~
ubuntu@ip-172-31-45-55:~$ sudo !!
sudo docker pull rocker/tidyverse
Using default tag: latest
latest: Pulling from rocker/tidyverse
4a42e09f9ba4: Already exists
4039240d2e0b: Already exists
4106205e95: Already exists
4039240d2e0b: Already exists
4b491e1a2dc: Already exists
4efc7a29f9ba4: Already exists
4bd47ae7cd00: Extracting [=====] 144.3MB/236.3MB
4039240d2e0b: Download complete
4b491e1a2dc: Download complete
4bd47ae7cd00: Extracting [=====]
```

10.6 Wrapup



→ Docker image ls



10.5 List the Docker Images

1. docker image ls - List the local docker images installed on the EC2 Server.

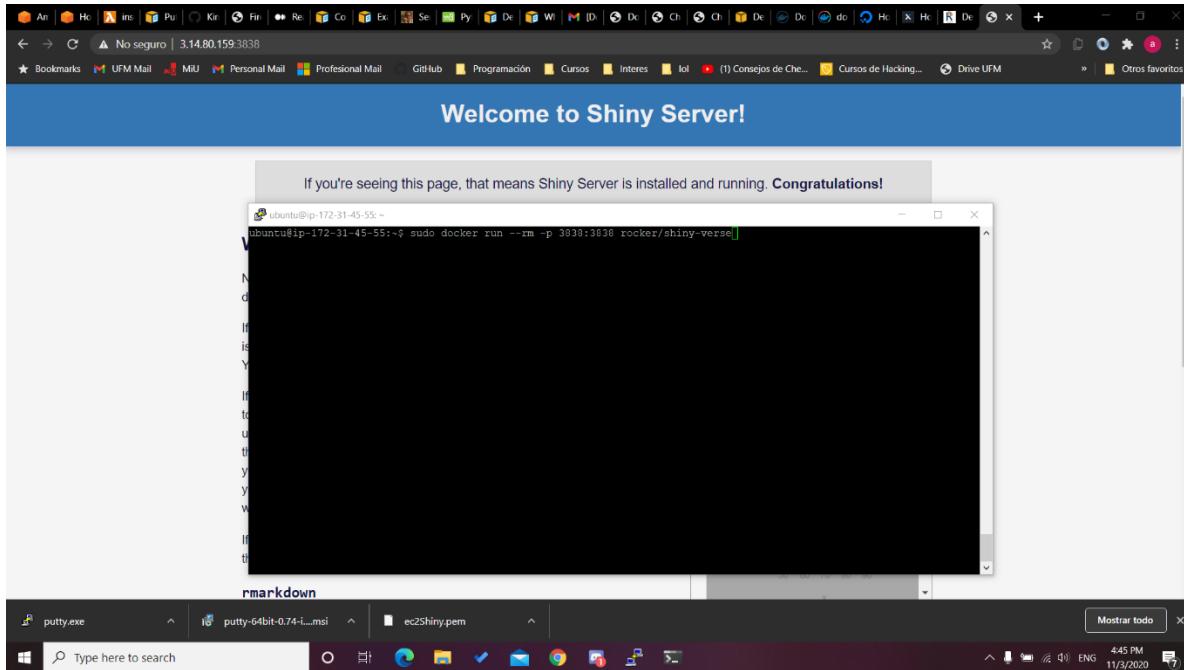
```
Ubuntu@ip-172-31-45-55: ~
ubuntu@ip-172-31-45-55:~$ sudo !!
sudo docker pull rocker/tidyverse
Using default tag: latest
latest: Pulling from rocker/tidyverse
4a42a2a29f9ba4: Already exists
mdan127c9761dcba: Already exists
REPO113b203e905: Already exists
rockerb341d2e0b: Already exists
shiny-3d7aa7c3: Already exists
mdan1feca4dd95c3: Already exists
rock3d3d7aa7c3a0: Pull complete
rock3d3d7aa7c3a0: Pull complete
mdan10c6e28d8e7d: Pull complete
mdan10c6e28d8e7d: Pull complete
mdan10c6e28d8e7d: Pull complete
mdan10c6e28d8e7d: Pull complete
sha256:e60ef005f0c790360c447a4ad29e67a71d96d900e3527adeeb07d5656ad97a11
mdanStatus: Downloaded newer image for rocker/tidyverse:latest
mdanDocker.io/rocker/tidyverse:latest
rockUbuntu@ip-172-31-45-55:~$ sudo docker image ls
rockREPOSITORY TAG IMAGE ID CREATED SIZE
mdan:rocker/shiny-verse latest 19975715e26d 9 days ago 1.91GB
rock:rocker/tidyverse latest 037f93874719 9 days ago 2.48GB
mdan:ubuntu@ip-172-31-45-55:~$
```

10



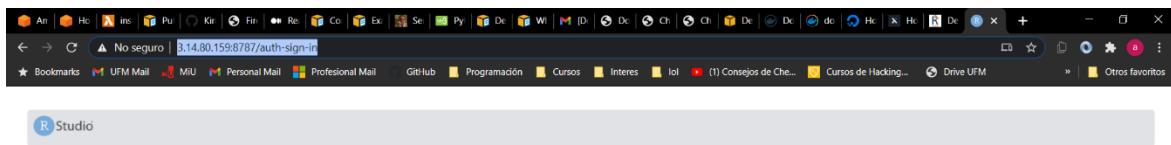
Docker Container

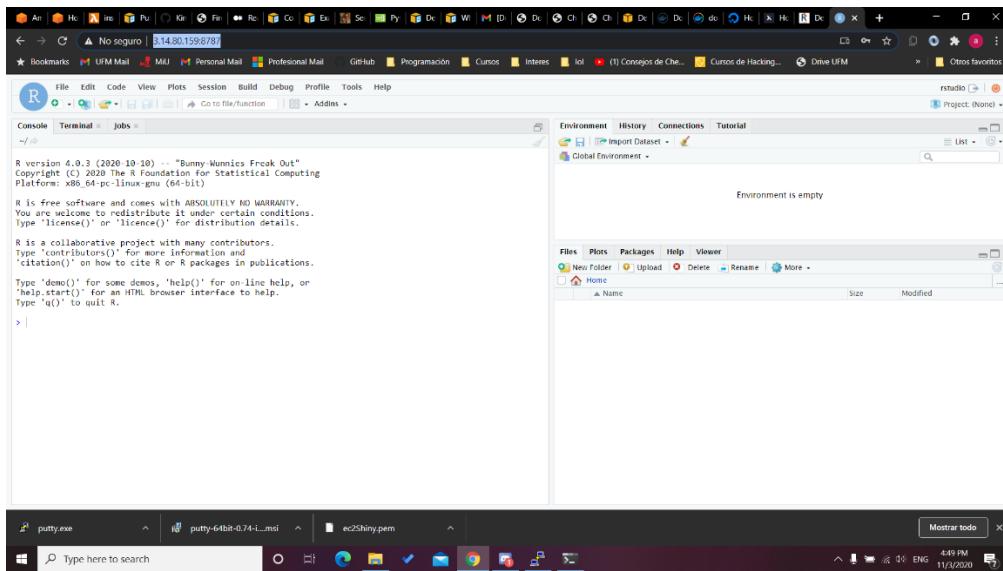
1. sudo docker run --rm -p 3838:3838 rocker/shinyverse
2. Abre localhost:3838 o Tambien el ippublico de AWS:3838



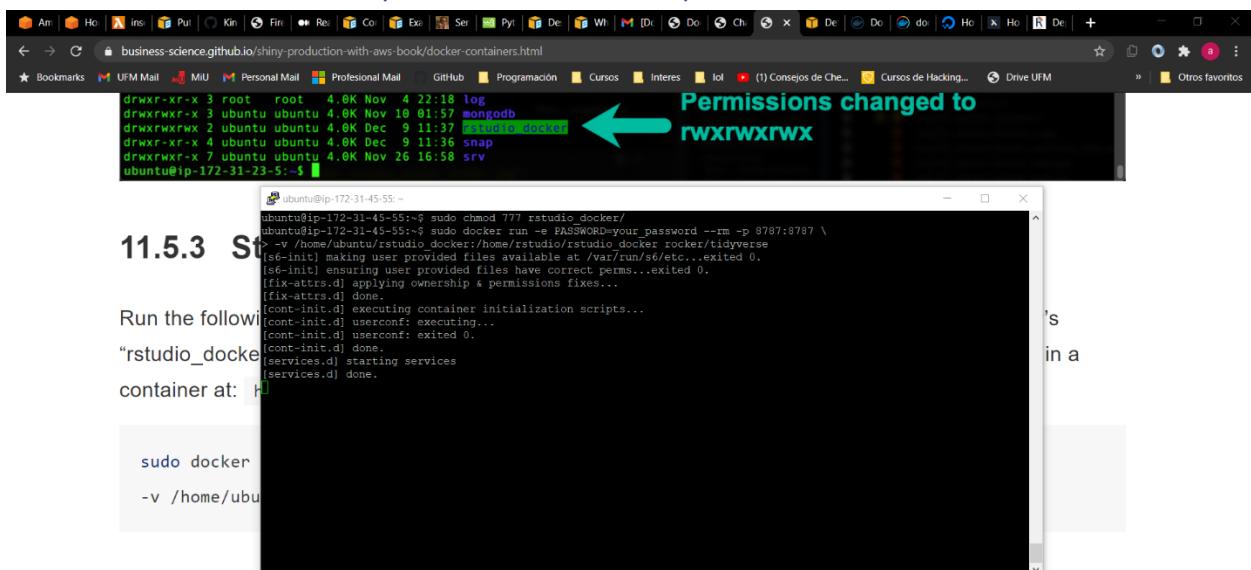
sudo docker run -e PASSWORD=your_password --rm -p 8787:8787 rocker/tidyverse

Abre localhost:8787 o ip publico de AWS:8787





Creación de volumen y autorización de permisos

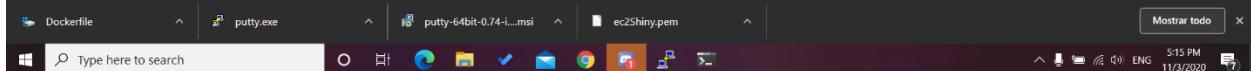


11.5.3 Step 3 - Create a Docker Container

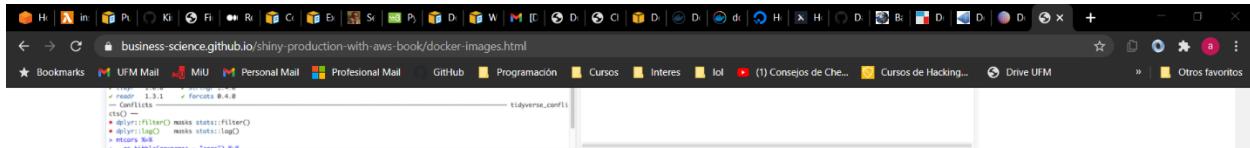
Run the following command to create a Docker container named 'rstudio_docker' with a volume mounted at '/home/ubuntu':

```
sudo docker
-v /home/ubuntu
```

11.5.4 Step 4 - Navigate to Your Server's RStudio Session



Docker Image Build



The screenshot shows a browser window with the URL business-science.github.io/shiny-production-with-aws-book/docker-images.html. The page contains text and code snippets related to Docker image builds for Shiny applications.

12.3.3 Docker

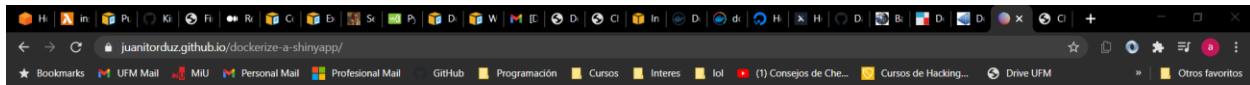
The docker build command is used to build a Docker image from a Dockerfile. It takes the path to the Dockerfile as an argument. If no path is specified, it looks for a file named Dockerfile in the current directory.

Navigate into the directory where your Dockerfile is located:

```
sudo docker build .
```

Here's what is happening:

- `docker build` - The command to build a Docker image.
- `..` - The directory path to the Dockerfile.
- somewhere else or named differently than Dockerfile, use tab completion to locate it (i.e. `docker build path/to/your/file/dockerfile_name`)
- `-t` - Adds an image name and tag.

The screenshot shows a browser window with the URL juantorduz.github.io/dockerize-a-shinyapp/. The page contains a Dockerfile example and instructions for building a Docker image for a shiny application.

select port

EXPOSE 3838

all DONE (shinymaterial)

RUN The downloaded source packages are in

'/tmp/RtmpfmBqPD/downloaded_packages'

rm Removing intermediate container d67f483a93d4

--> 0ea80a6ff0b

CMD Step 4/8 : COPY app.R /srv/shiny-server/

--> eaaaaf81fe4

Step 5/8 : COPY data /srv/shiny-server/data

--> d5e746e9e1aa

To build Step 6/8 : EXPOSE 3838

--> RUN sudo chmod -R shiny:shiny /srv/shiny-server

--> Removing intermediate container 7cled0d82326

--> 418c4d9aff00

dock Step 7/8 : RUN sudo chown -R shiny:shiny /srv/shiny-server

--> Removing intermediate container 36bd61b5243e

--> 5cc03aef09f

To create Step 8/8 : CMD ["usr/bin/shiny-server.sh"]

--> Running in ec8f15292657

Removing intermediate container ec8f15292657

--> 1b8c0e1f50fe

dock Successfully built 1b8c0e1f50fe

Successfully tagged shiny:steam:latest

ubuntu@ip-172-31-45-55:~/rstudio_docker/lab\$ sudo docker image ls

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
shiny:steam	latest	1b0c0e1f50fe	24 seconds ago	2.49GB
rocker/shiny-verse	latest	19975715e264	9 days ago	1.91GB
rocker/tidyverse	latest	037f93874719	9 days ago	2.49GB

ubuntu@ip-172-31-45-55:~/rstudio_docker/lab\$



Docker Push

hub.docker.com/u/mapg23

Pull rate limits for certain users are being introduced to Docker Hub starting November 2nd. [Learn more](#)

[docker hub](#) Search for great content (e.g., mysql)

Explore Repositories Organizations Get Help mapg23

```
ubuntu@ip-172-31-45-55:~$ sudo docker image ls
REPOSITORY TAG IMAGE ID CREATED SIZE
shiny-verse latest 19975715e26d 9 days ago 1.91GB
rocker/tidyverse latest 037f93874719 9 days ago 2.48GB
ubuntu@ip-172-31-45-55:~$ sudo docker tag lb8c0elf50fe mapg23/shiny-widgets:latest
ubuntu@ip-172-31-45-55:~$ sudo docker image ls
REPOSITORY TAG IMAGE ID CREATED SIZE
shiny-steam latest 1b8c0elf50fe 8 minutes ago 2.48GB
mapg23/shiny-widgets latest 1b8c0elf50fe 8 minutes ago 2.48GB
rocker/shiny-verse latest 19975715e26d 9 days ago 1.91GB
rocker/tidyverse latest 037f93874719 9 days ago 2.48GB
ubuntu@ip-172-31-45-55:~$ sudo docker tag lb8c0elf50fe mapg23/shiny-steam:latest
ubuntu@ip-172-31-45-55:~$ sudo docker image ls
REPOSITORY TAG IMAGE ID CREATED SIZE
shiny-steam latest 1b8c0elf50fe 8 minutes ago 2.48GB
mapg23/shiny-steam latest 1b8c0elf50fe 8 minutes ago 2.48GB
mapg23/shiny-widgets latest 1b8c0elf50fe 8 minutes ago 2.48GB
rocker/shiny-verse latest 19975715e26d 9 days ago 1.91GB
rocker/tidyverse latest 037f93874719 9 days ago 2.48GB
ubuntu@ip-172-31-45-55:~$ sudo docker push mapg23/shiny-steam:latest
The push refers to repository [docker.io/mapg23/shiny-steam]
699d22a4e505: Pushed
3269a5100635: Pushed
e799f0fb7a14: Pushed
dbee19f6fe42: Pushed
a51e650d81ee: Pushed
19927af949: Mounted from rocker/shiny-verse
e34bdc9ce4b: Mounted from rocker/shiny-verse
ca46f3a2a2: Mounted from rocker/tidyverse
cc31da247282: Mounted from rocker/tidyverse
0513be7ed530: Mounted from rocker/tidyverse
ec1017c93e7c: Mounted from rocker/tidyverse
9e97312b63ff: Mounted from rocker/tidyverse
e1c75a5e0bfa: Mounted from rocker/tidyverse
latest: digest: sha256:4990a3a3fd504627af88bd01689396b0edd54b68fee52914699e830adSacb43 size: 3056
ubuntu@ip-172-31-45-55:~$
```

Docker HUB with image pushed

hub.docker.com/u/mapg23

Pull rate limits for certain users are being introduced to Docker Hub starting November 2nd. [Learn more](#)

[docker hub](#) Search for great content (e.g., mysql)

Explore Repositories Organizations Get Help mapg23

 **mapg23** [Edit profile](#)

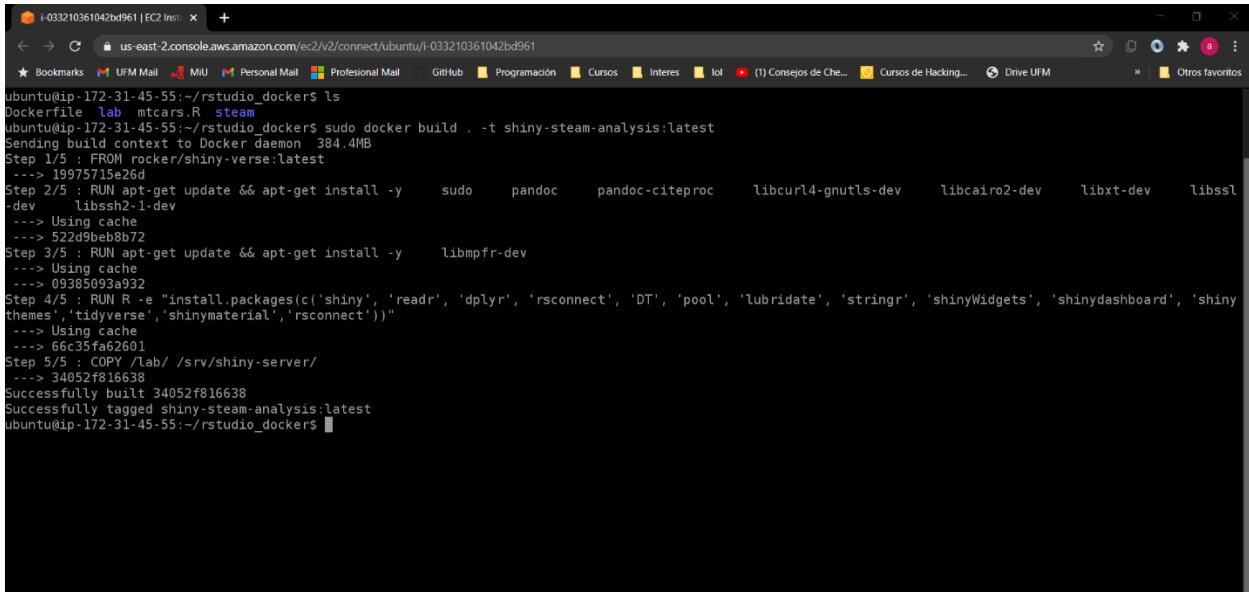
Community User Joined August 3, 2018

[Repositories](#) [Starred](#) [Contributed](#)

Displaying 1 of 1 repository

	mapg23/shiny-steam	By mapg23 • Updated 2 minutes ago	1 Download	0 Stars
	Container			

Dockerfile working



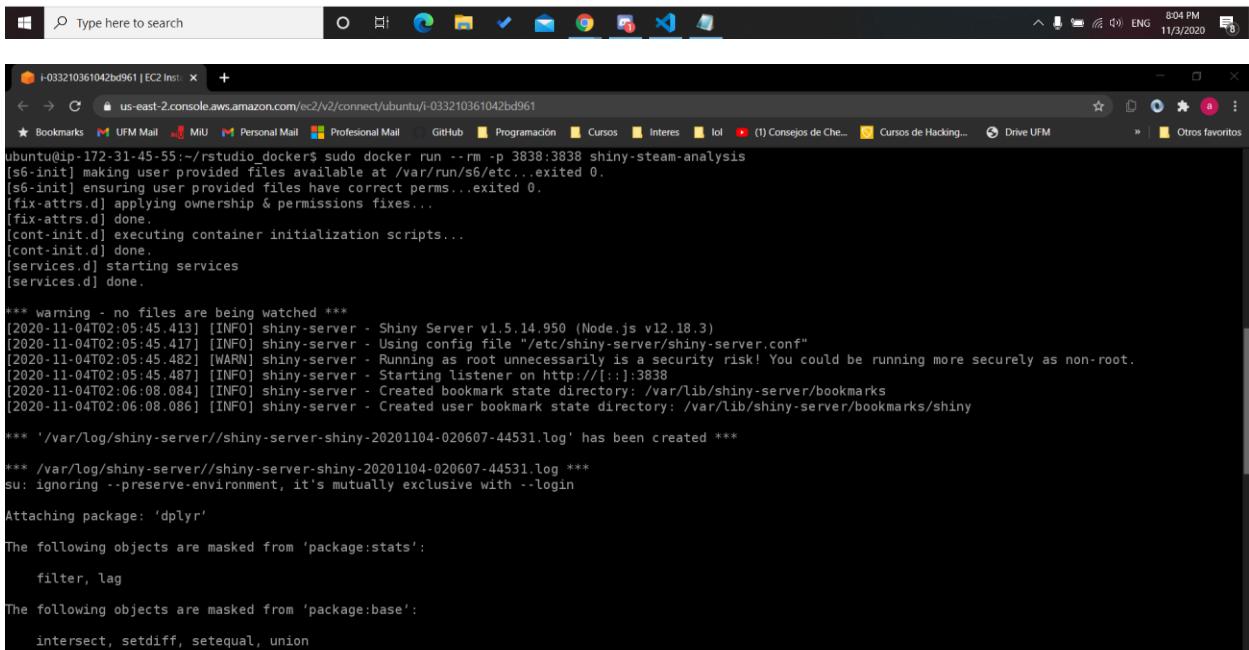
```

ubuntu@ip-172-31-45-55:~/rstudio_dockers$ ls
Dockerfile  lab  mtcars.R  steam
ubuntu@ip-172-31-45-55:~/rstudio_dockers$ sudo docker build . -t shiny-steam-analysis:latest
Sending build context to Docker daemon 384.4MB
Step 1/5 : FROM rocker/shiny-verse:latest
--> 19975715e26d
Step 2/5 : RUN apt-get update && apt-get install -y      sudo      pandoc      pandoc-citeproc      libcurl4-gnutls-dev      libcairo2-dev      libxt-dev      libssl-dev
--> Using cache
--> 522d9be8b872
Step 3/5 : RUN apt-get update && apt-get install -y      libmpfr-dev
--> Using cache
--> 09385093a932
Step 4/5 : RUN R -e "install.packages(c('shiny', 'readr', 'dplyr', 'rsconnect', 'DT', 'pool', 'lubridate', 'stringr', 'shinyWidgets', 'shinydashboard', 'shinythemes', 'tidyverse', 'shinymaterial', 'rsconnect'))"
--> Using cache
--> 66c35fa62601
Step 5/5 : COPY /lab/ /srv/shiny-server/
--> 34052f816638
Successfully built 34052f816638
Successfully tagged shiny-steam-analysis:latest
ubuntu@ip-172-31-45-55:~/rstudio_dockers$ 

```

i-033210361042bd961

Public IPs: 3.19.60.38 Private IPs: 172.31.45.55



```

ubuntu@ip-172-31-45-55:~/rstudio_dockers$ sudo docker run --rm -p 3838:3838 shiny-steam-analysis
[s6-init] making user provided files available at /var/run/s6/etc..exited 0.
[s6-init] ensuring user provided files have correct perms...exited 0.
[fix-attrs.d] applying ownership & permissions fixes...
[fix-attrs.d] done.
[cont-init.d] executing container initialization scripts...
[cont-init.d] done.
[services.d] starting services
[services.d] done.

*** warning - no files are being watched ***
[2020-11-04T02:05:41Z] [INFO] shiny-server - Shiny Server v1.5.14.950 (Node.js v12.18.3)
[2020-11-04T02:05:41Z] [INFO] shiny-server - Using config file "/etc/shiny-server/shiny-server.conf"
[2020-11-04T02:05:45.48Z] [WARN] shiny-server - Running as root unnecessarily is a security risk! You could be running more securely as non-root.
[2020-11-04T02:05:45.48Z] [INFO] shiny-server - Starting listener on http://[::]:3838
[2020-11-04T02:06:08.08Z] [INFO] shiny-server - Created bookmark state directory: /var/lib/shiny-server/bookmarks
[2020-11-04T02:06:08.08Z] [INFO] shiny-server - Created user bookmark state directory: /var/lib/shiny-server/bookmarks/shiny

*** '/var/log/shiny-server/shiny-server-shiny-20201104-020607-44531.log' has been created ***
*** '/var/log/shiny-server/shiny-server-shiny-20201104-020607-44531.log' ***
su: Ignoring --preserve-environment, it's mutually exclusive with --login

Attaching package: 'dplyr'

The following objects are masked from 'package:stats':
  filter, lag

The following objects are masked from 'package:base':
  intersect, setdiff, setequal, union

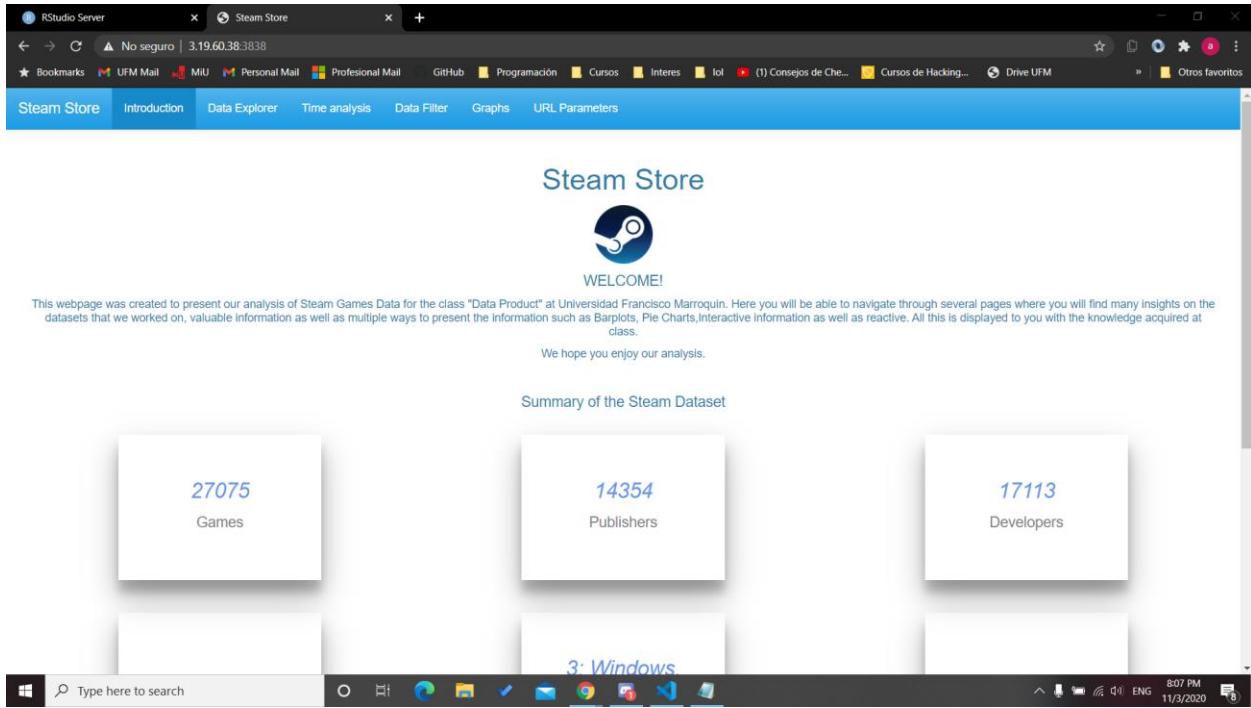
```

i-033210361042bd961

Public IPs: 3.19.60.38 Private IPs: 172.31.45.55



Project working



Steam Store

WELCOME!

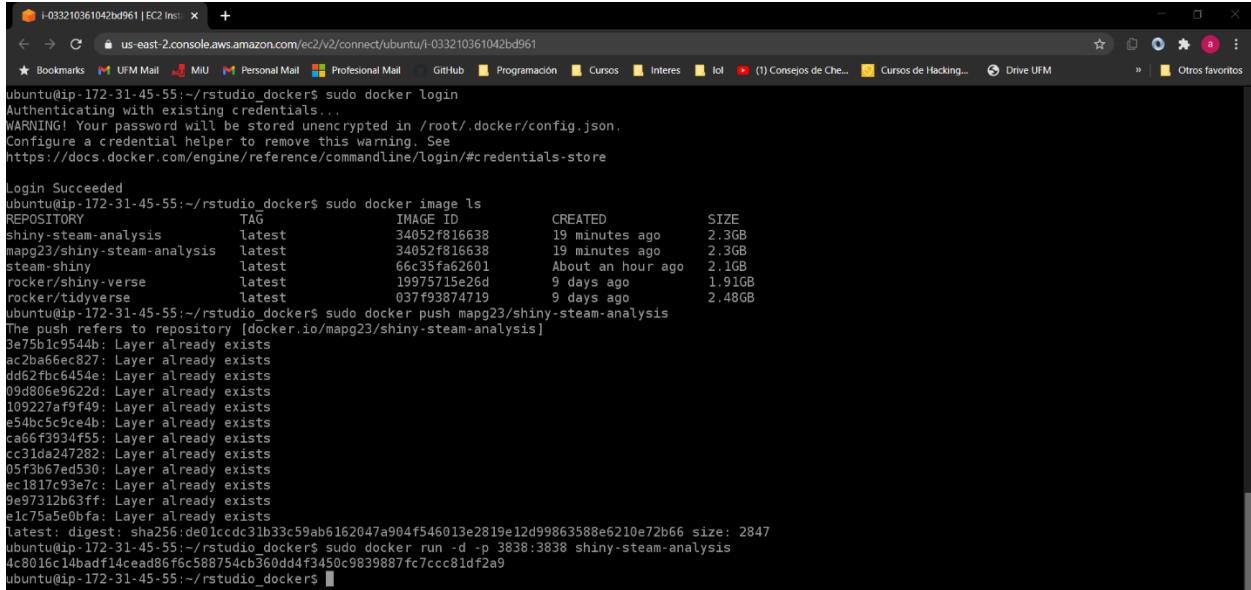
This webpage was created to present our analysis of Steam Games Data for the class "Data Product" at Universidad Francisco Marroquin. Here you will be able to navigate through several pages where you will find many insights on the datasets that we worked on, valuable information as well as multiple ways to present the information such as Barplots, Pie Charts, Interactive information as well as reactive. All this is displayed to you with the knowledge acquired at class.

We hope you enjoy our analysis.

Summary of the Steam Dataset

Category	Count
Games	27075
Publishers	14354
Developers	17113

Instance running detach until container is killed



```

ubuntu@ip-172-31-45-55:~/rstudio_dockers$ sudo docker login
Authenticating with existing credentials...
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
ubuntu@ip-172-31-45-55:~/rstudio_dockers$ sudo docker image ls
REPOSITORY          TAG      IMAGE ID      CREATED       SIZE
shiny-steamp-analysis latest   34052f816638  19 minutes ago  2.3GB
mapg23/shiny-steamp-analysis latest   34052f816638  19 minutes ago  2.3GB
steam-shiny          latest   66c35fa62601  About an hour ago  2.1GB
rocker/shiny-verse   latest   19975715e26d  9 days ago    1.91GB
rocker/tidyverse    latest   037f93874719  9 days ago    2.48GB
ubuntu@ip-172-31-45-55:~/rstudio_dockers$ sudo docker push mapg23/shiny-steamp-analysis
The push refers to repository [docker.io/mapg23/shiny-steamp-analysis]
3e75b1c9544b: Layer already exists
ac2ba66ec827: Layer already exists
dd62fbcc6454e: Layer already exists
09d806e9622d: Layer already exists
109227af9749: Layer already exists
e54bc5c9ce4b: Layer already exists
ca66f3934f55: Layer already exists
cc31da247282: Layer already exists
05f3067ed530: Layer already exists
ec1817c93e7c: Layer already exists
9e97312b63ff: Layer already exists
e1c73a5e0bfa: Layer already exists
latest: digest: sha256:de01cccd31b33c59ab6162047a904f546013e2819e12d99863588e6210e72b66 size: 2847
ubuntu@ip-172-31-45-55:~/rstudio_dockers$ sudo docker run -d -p 3838:3838 shiny-steamp-analysis
4c8016c14badf14cead86f6c588754cb360dd4f3450c9839887fc7ccc81df2a9
ubuntu@ip-172-31-45-55:~/rstudio_dockers$ 

```

i-033210361042bd961

Public IPs: 3.19.60.38 Private IPs: 172.31.45.55

