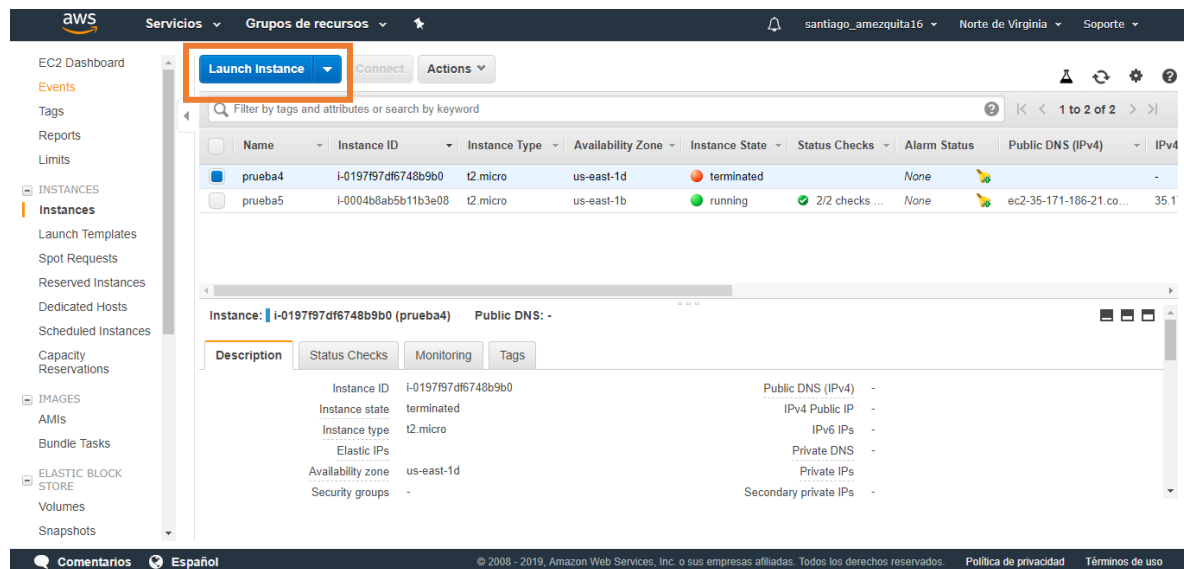


Uploading a shiny-server in AWS

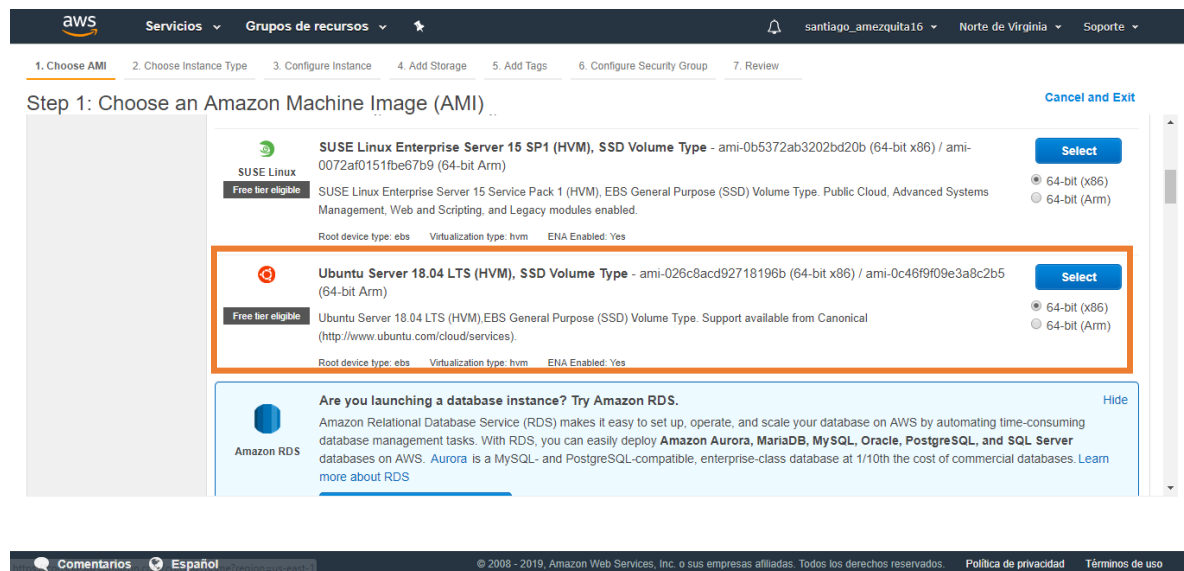
The following documentation is a step-by-step guide to upload a shiny server on AWS.

Creating the AWS server

1. Log into Amazon Web Services (AWS from now on).
2. In the “Services” section, look for EC2 to create a server
3. Click on launch instance:



4. Look for a free tier eligible server, we will use in this case an Ubuntu Server 18.04 LTS of 64-bits.



EC2 allows you to create elastic servers, which can be used for simple tasks of the shiny apps or “enlarge” themselves to do more complex tasks.

5. Once you choose the Amazon Machine Image (AMI from now on), click on “Review and Launch”,

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 types Current generation Show/Hide Columns

Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
<input type="checkbox"/>	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
<input checked="" type="checkbox"/>	General purpose	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.small	1	2	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.large	2	8	EBS only	-	Low to Moderate	Yes

Cancel Previous **Review and Launch** Next: Configure Instance Details

6. Click on “Launch”.

Step 7: Review Instance Launch
Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

AMI Details [Edit AMI](#)

Ubuntu Server 18.04 LTS (HVM), SSD Volume Type - ami-026c8acd92718196b
Free tier eligible
Ubuntu Server 18.04 LTS (HVM), EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).
Root Device Type: ebs Virtualization type: hvm

Instance Type [Edit instance type](#)

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	Variable	1	1	EBS only	-	Low to Moderate

Security Groups [Edit security groups](#)

Security group name: launch-wizard-6
Description: launch-wizard-6 created 2019-07-25T10:18:29.986-05:00

Cancel Previous **Launch**

Comentarios Español © 2008 - 2019, Amazon Web Services, Inc. o sus empresas afiliadas. Todos los derechos reservados. Política de privacidad Términos de uso

7. Once you click on launch, you should get something like this:

Select an existing key pair or create a new key pair

X

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about [removing existing key pairs from a public AMI](#).

Choose an existing key pair

Select a key pair

prueba-shiny-apps

☐ I acknowledge that I have access to the selected private key file (prueba-shiny-apps.pem), and that without this file, I won't be able to log into my instance.

Cancel

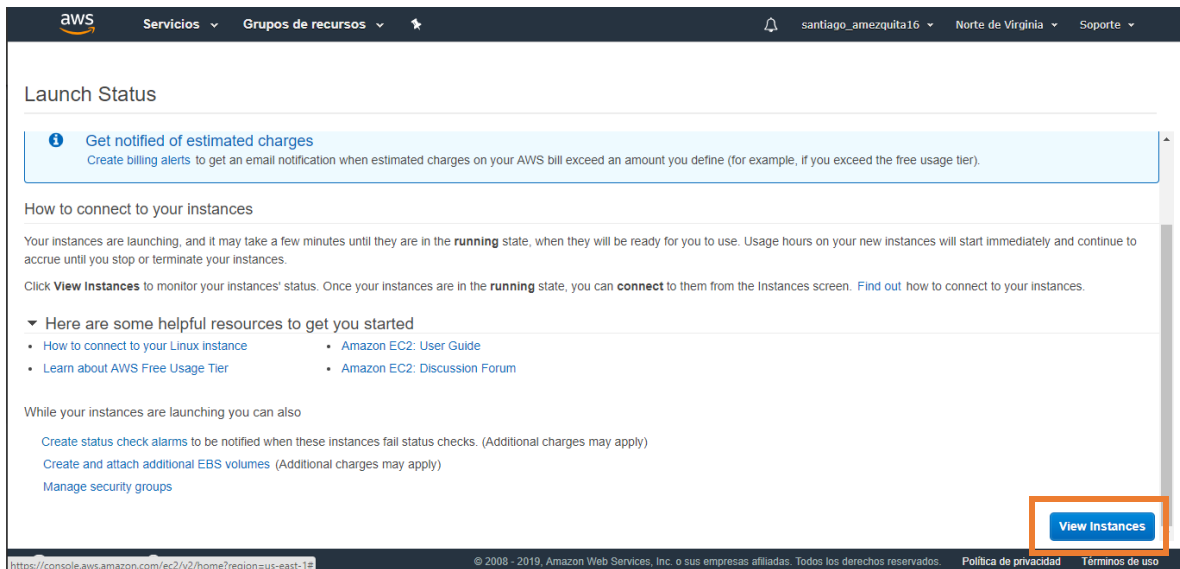
Launch Instances

This is to create a key for your AMI and access to it remotely. It will save a .pem file. **DO NOT LOOSE THIS FILE**, without it you cannot access your AMI in the future. Since you probably don't have a key pair, you should create one:

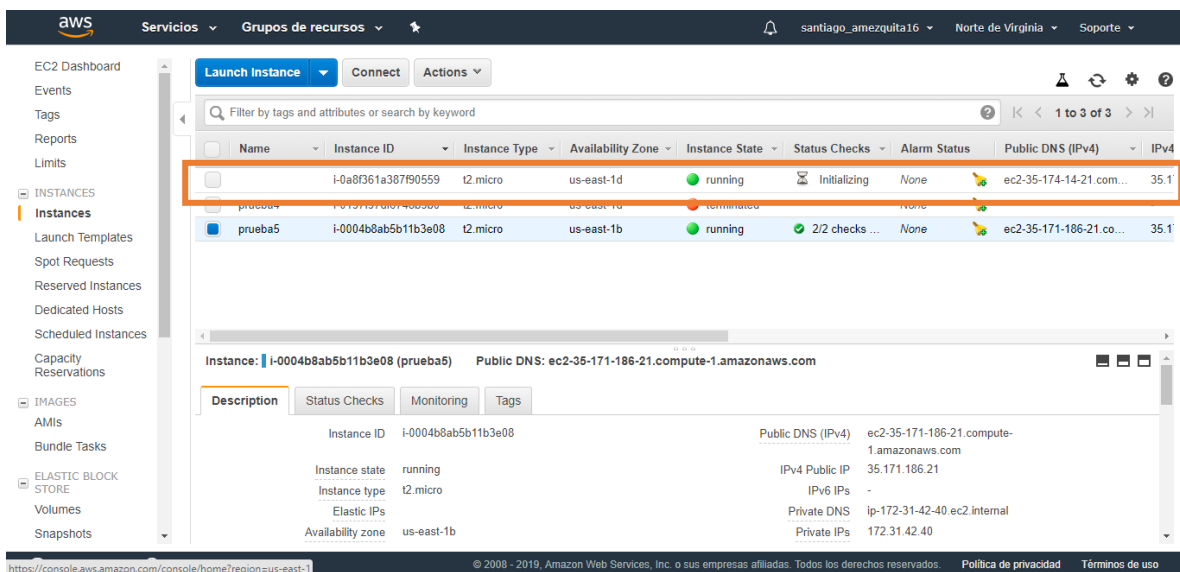
- 7.1. In the first list select "Create a new key pair"
- 7.2. Create a simple name separating words with "-".
- 7.3. Download the key pair and **SAVE IT IN A SAFE AND EASY TO ACCESS PLACE**.

In this case, we will use an existing key pair called "prueba-shiny-apps". If you're going to use an existing key pair, check the box on the bottom of this screen and click on "Launch Instances".

8. In the bottom of this new page, click on "View Instances" to see all of the instances that you may have.



Now, we can see our new instance with no name. If you want to, you can change the name of the instance.



We will change the name in this tutorial to “instance1”.

If we want to access from a windows machine, we will have to download PuTTY to access this machine. AWS contains the whole documentation with this process.

To connect with PuTTY, select the instance in AWS and click on “Connect”, you’ll get something like this:

Connect To Your Instance

I would like to connect with

☒ A standalone SSH client ⓘ

☐ EC2 Instance Connect (browser-based SSH connection) ⓘ

☐ A Java SSH Client directly from my browser (Java required) ⓘ

To access your instance:

1. Open an SSH client. (find out how to [connect using PuTTY](#))

2. Locate your private key file (prueba-shiny-apps.pem). The wizard automatically detects the key you used to launch the instance.

3. Your key must not be publicly viewable for SSH to work. Use this command if needed:

```
chmod 400 prueba-shiny-apps.pem
```

4. Connect to your instance using its Public DNS:

```
ec2-35-174-14-21.compute-1.amazonaws.com
```

Example:

```
ssh -i "prueba-shiny-apps.pem" ubuntu@ec2-35-174-14-21.compute-1.amazonaws.com
```

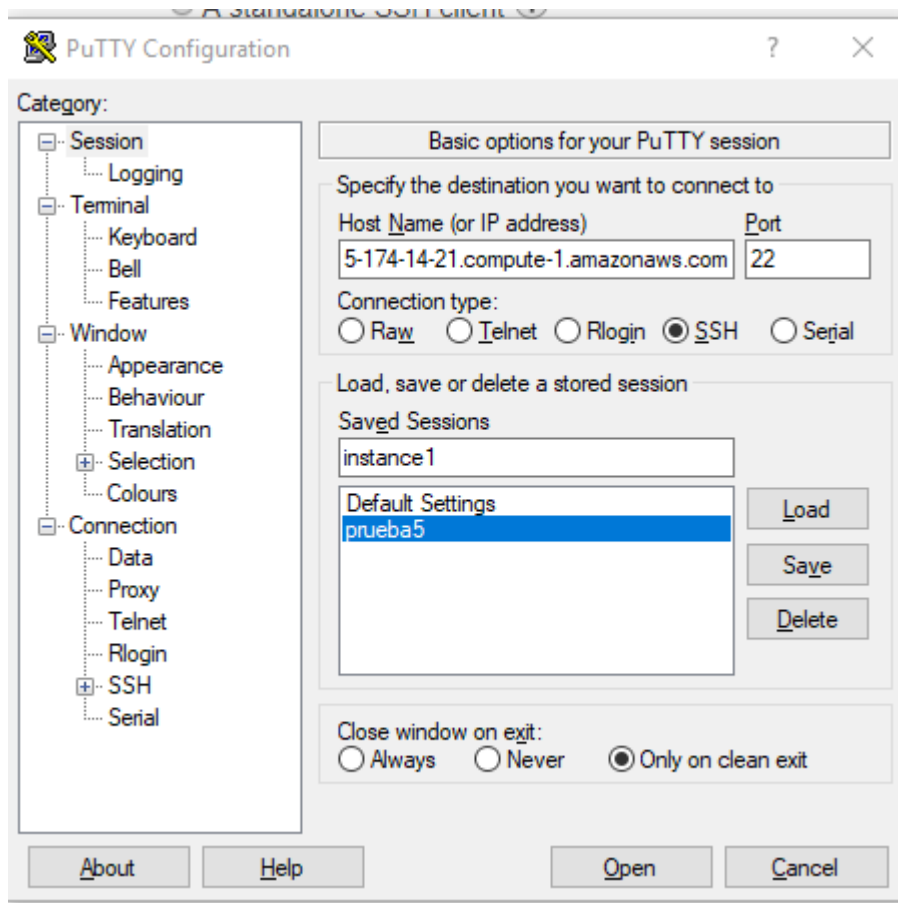
Please note that in most cases the username above will be correct, however please ensure that you read your AMI usage instructions to ensure that the AMI owner has not changed the default AMI username.

If you need any assistance connecting to your instance, please see our [connection documentation](#).

Close

The selected text is what we will need to access to this machine in PuTTY.

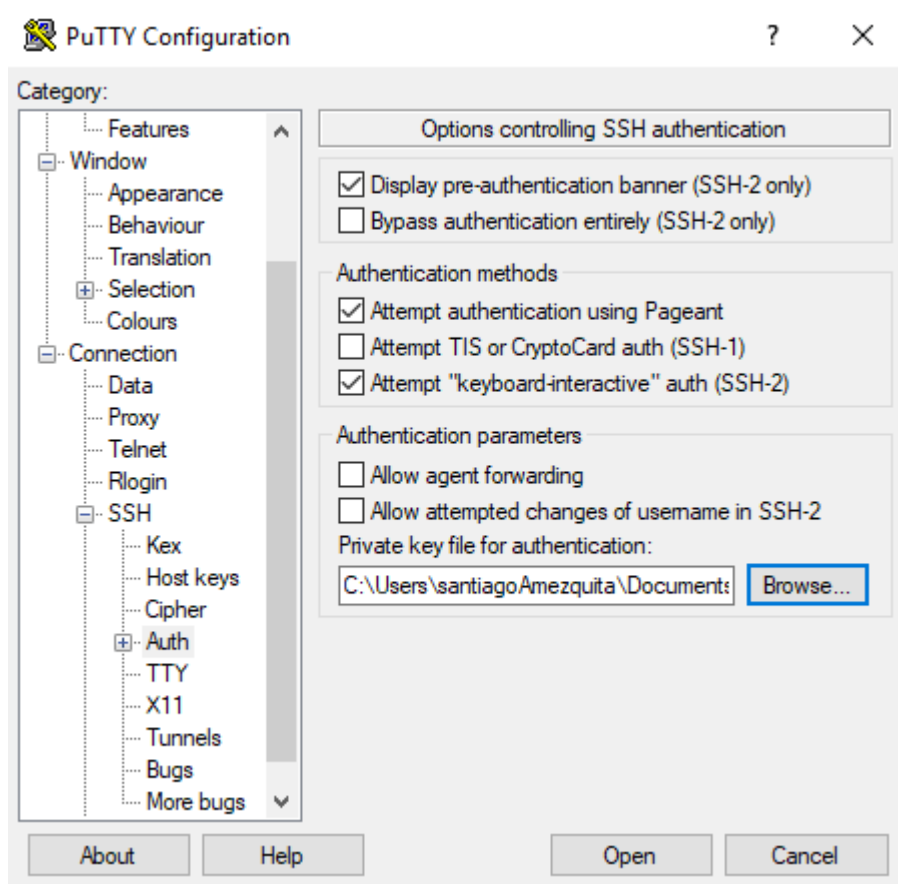
9. Open PuTTY and create the server:
 - 9.1. In the "Session" page, copy and paste your Instance's public DNS, in my case, it is "ec2-35-174-14-21.compute-1.amazonaws.com".
 - 9.2. In order to have an organized environment in PuTTY, save the instance name with the same name you have on AWS, in the case of this tutorial, "instance1".
 - 9.3. Click on Save. You'll have a PuTTY screen like this:



Check that all the sections in the “Session” screen are almost equal to the ones showed here (Select SSH, select “Only on clean exit”).

9.4. Now we have to connect to the machine with the key pair we downloaded previously, with a .ppk extension. This will be done in the section Connection -> SSH -> Auth

9.5. Browse for your key pair and upload it, you’ll get something like this:



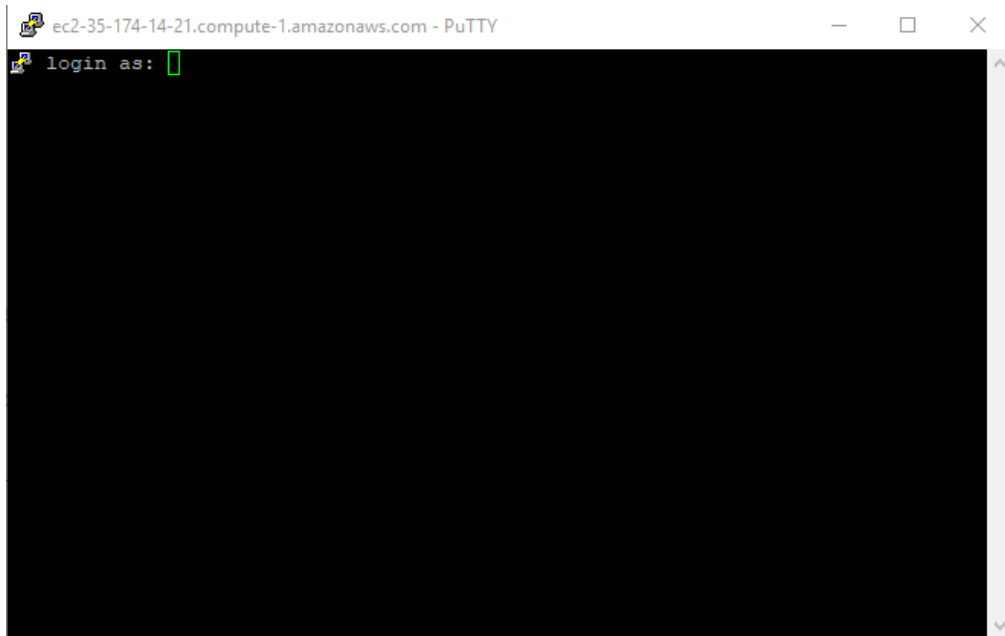
Again, check that everything has the same checkboxes checked.

9.6. Go the Session screen again and click on save again.

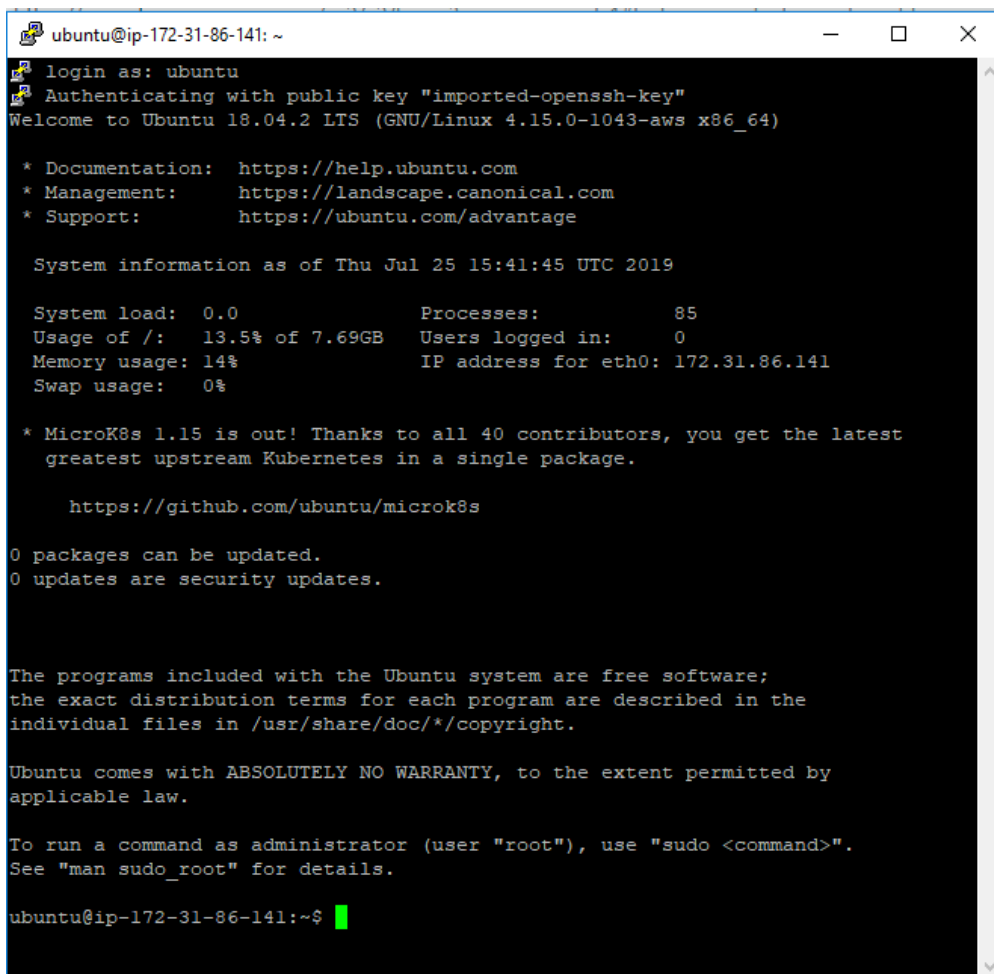
9.7. Click on "Open".

9.8. Click on "Yes".

10. When you successfully accessed to your AMI, you'll get a screen like this one:



We will login as “ubuntu”, so, type ubuntu and press enter, you’ll get something like this:



Install R and Shiny in your AMI

Now we must install R and shiny.

11. The whole documentation to install R is on the link: <https://cran.r-project.org/>
- 11.1. Click on “Download R for Linux”.
- 11.2. Click on “ubuntu”.
- 11.3. You’ll get a page like this:

UBUNTU PACKAGES FOR R

- [Installation](#)
- [Supported Packages](#)
- [Secure APT](#)
- [Administration and Maintanances of R Packages](#)
- [Pathways to R Packages](#)
- [Reporting Problems](#)
- [Acknowledgements](#)

R 3.6 packages for Ubuntu on i386 and amd64 are available for most stable Desktop releases of Ubuntu until their official end of life date. However, only the latest Long Term Support (LTS) release is fully supported. As of November 18, 2018 the supported releases are Xenial Xerus (16.04; LTS), Trusty Tahr (14.04; LTS), Bionic Beaver (18.04; LTS), Cosmic Cuttlefish (18.10), and Disco Dingo (19.04). Note, to install R 3.6 packages, a different sources.list entry is needed. See below for details. Even though R has moved to version 3.6, for compatibility the sources.list entry still uses the `cran3.5` designation.

R 3.4 packages for Ubuntu on i386 and amd64 are available for all stable Desktop releases of Ubuntu prior to Bionic Beaver (18.04) until their official end of life date. However, only the latest Long Term Support (LTS) release is fully supported. As of November 18, 2018 the supported releases are Xenial Xerus (16.04; LTS), and Trusty Tahr (14.04; LTS).

See <https://wiki.ubuntu.com/Releases> for details.

For additional binary packages for R (currently 4,000+), check out the CRAN2deb4ubuntu PPA:

<https://launchpad.net/~marutter/+archive/ubuntu/c2d4u> or <https://launchpad.net/~marutter/+archive/ubuntu/c2d4u3.5> depending on which version of R you are using.

Installation

To obtain the latest R 3.6 packages, add an entry like

```
deb https://cloud.r-project.org/bin/linux/ubuntu disco-cran35/
```

or

```
deb https://cloud.r-project.org/bin/linux/ubuntu cosmic-cran35/
```

or

```
deb https://cloud.r-project.org/bin/linux/ubuntu bionic-cran35/
```

or

```
deb https://cloud.r-project.org/bin/linux/ubuntu xenial-cran35/
```

or

```
deb https://cloud.r-project.org/bin/linux/ubuntu trusty-cran35/
```

The whole documentation to install the latest version of R is on that page, however, we will do a step-by-step guide in this tutorial to install R successfully on your AMI.

12. Open the file sources.list located on the route `/etc/apt/`. To access this file, write the following lines and press enter in each one of them:

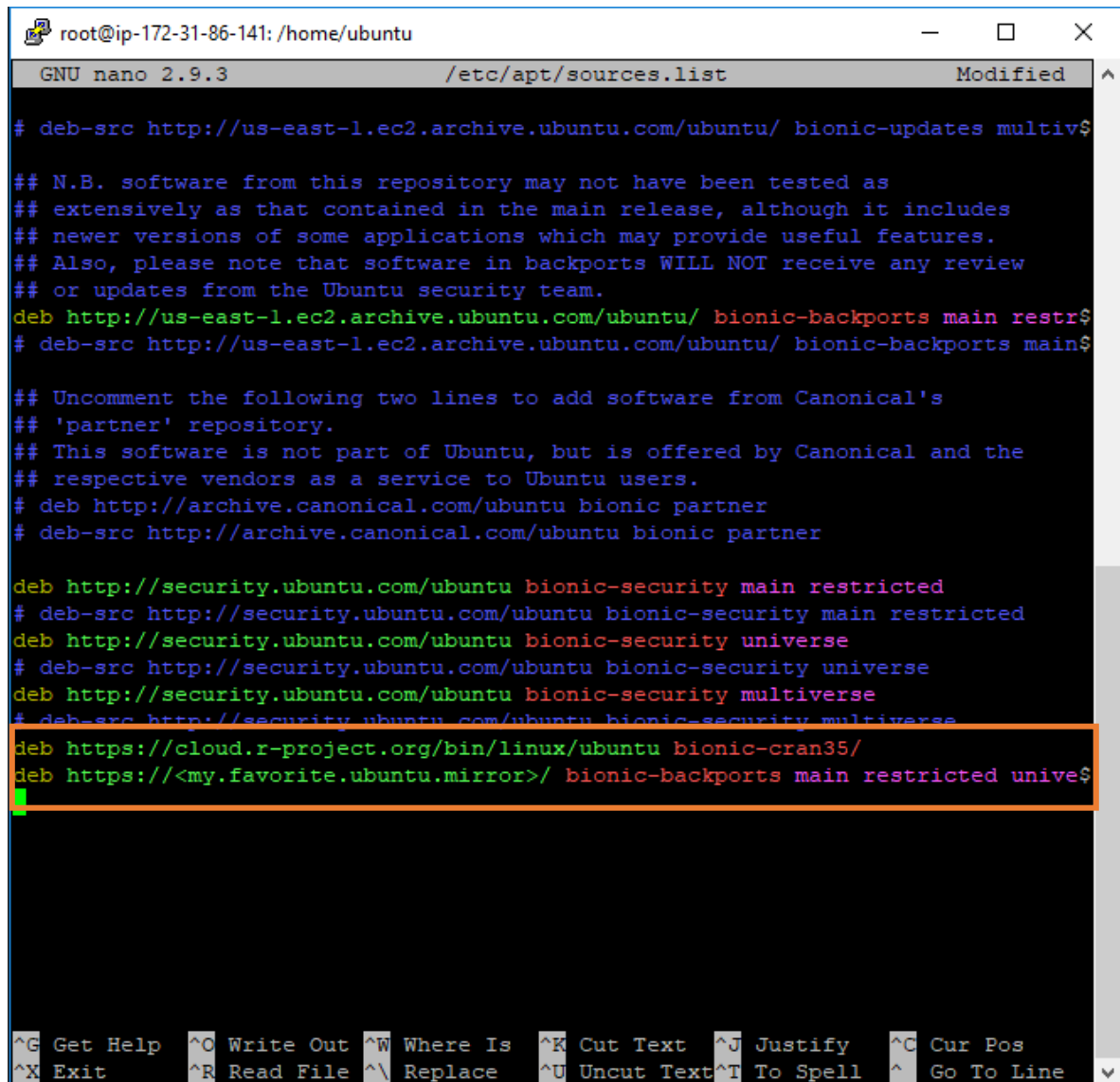
```
sudo su
```

```
nano /etc/apt/sources.list
```

Now that you're on the file, write the following lines in the bottom of the file:

```
deb https://cloud.r-project.org/bin/linux/ubuntu bionic-cran35/  
deb https://<my.favorite.ubuntu.mirror>/ bionic-backports main restricted universe
```

And end up with a file like this one:



```
root@ip-172-31-86-141: /home/ubuntu
GNU nano 2.9.3 /etc/apt/sources.list Modified
# deb-src http://us-east-1.ec2.archive.ubuntu.com/ubuntu/ bionic-updates multiv$
## N.B. software from this repository may not have been tested as
## extensively as that contained in the main release, although it includes
## newer versions of some applications which may provide useful features.
## Also, please note that software in backports WILL NOT receive any review
## or updates from the Ubuntu security team.
deb http://us-east-1.ec2.archive.ubuntu.com/ubuntu/ bionic-backports main restr$
# deb-src http://us-east-1.ec2.archive.ubuntu.com/ubuntu/ bionic-backports main$

## Uncomment the following two lines to add software from Canonical's
## 'partner' repository.
## This software is not part of Ubuntu, but is offered by Canonical and the
## respective vendors as a service to Ubuntu users.
# deb http://archive.canonical.com/ubuntu bionic partner
# deb-src http://archive.canonical.com/ubuntu bionic partner

deb http://security.ubuntu.com/ubuntu bionic-security main restricted
# deb-src http://security.ubuntu.com/ubuntu bionic-security main restricted
deb http://security.ubuntu.com/ubuntu bionic-security universe
# deb-src http://security.ubuntu.com/ubuntu bionic-security universe
deb http://security.ubuntu.com/ubuntu bionic-security multiverse
# deb-src http://security.ubuntu.com/ubuntu bionic-security multiverse
deb https://cloud.r-project.org/bin/linux/ubuntu bionic-cran35/
deb https://<my.favorite.ubuntu.mirror>/ bionic-backports main restricted unive$

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line
```

Press Ctrl+O

Press Enter

Press Ctrl+X

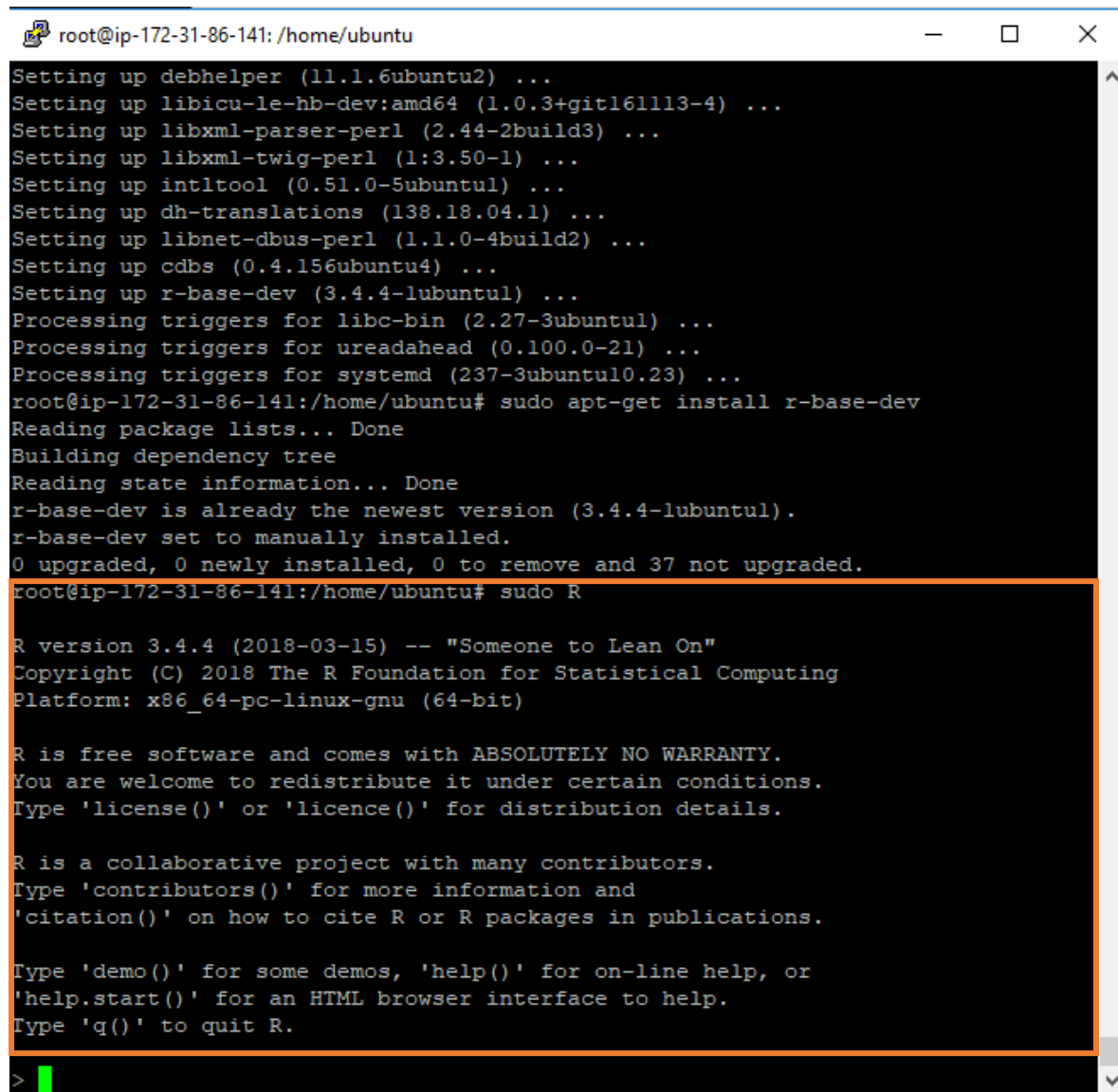
Now you must put these three commands in your AMI's console:

```
sudo apt-get update
sudo apt-get install r-base
sudo apt-get install r-base-dev
```

To access R from your AMI you can write:

```
sudo R
```

And you'll get something like this:

A terminal window titled 'root@ip-172-31-86-141: /home/ubuntu' showing the installation of R. The terminal output includes the installation of various dependencies like debhelper, libicu, libxml, and r-base-dev. It then shows the command 'sudo apt-get install r-base-dev' being executed, which reports that r-base-dev is already the newest version. Finally, the command 'sudo R' is executed, leading to the R startup screen. The startup screen displays the R version (3.4.4), copyright information, platform (x86_64-pc-linux-gnu), and a list of commands for help and quitting. The terminal window has a black background with white text and a green cursor at the bottom.

```
root@ip-172-31-86-141: /home/ubuntu
Setting up debhelper (11.1.6ubuntu2) ...
Setting up libicu-le-hb-dev:amd64 (1.0.3+git161113-4) ...
Setting up libxml-parser-perl (2.44-2build3) ...
Setting up libxml-twig-perl (1:3.50-1) ...
Setting up intltool (0.51.0-5ubuntu1) ...
Setting up dh-translations (138.18.04.1) ...
Setting up libnet-dbus-perl (1.1.0-4build2) ...
Setting up cdb (0.4.156ubuntu4) ...
Setting up r-base-dev (3.4.4-lubuntul) ...
Processing triggers for libc-bin (2.27-3ubuntu1) ...
Processing triggers for ureadahead (0.100.0-21) ...
Processing triggers for systemd (237-3ubuntu10.23) ...
root@ip-172-31-86-141:/home/ubuntu# sudo apt-get install r-base-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
r-base-dev is already the newest version (3.4.4-lubuntul).
r-base-dev set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 37 not upgraded.
root@ip-172-31-86-141:/home/ubuntu# sudo R

R version 3.4.4 (2018-03-15) -- "Someone to Lean On"
Copyright (C) 2018 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

>
```

At this point we have to install shiny directly from R with the following line:

```
install.packages("shiny")
```

If for some reason this doesn't work or takes too long, abort the task with Ctrl+C, quit R and use the command:

```
sudo apt-get install r-cran-shiny
```

This last command will be very important to install the R packages in case it doesn't work directly in R.

To exit from R console, you can type q() or press Ctrl+Z.

Configure your Shiny-server

13. Download the latest shiny-server version in the bottom of this page:
<https://www.rstudio.com/products/shiny/download-server/>

For me it is:

```
sudo apt-get install gdebi-core
wget https://download3.rstudio.org/ubuntu-14.04/x86_64/shiny-server-1.5.9.923-amd64.deb
sudo gdebi shiny-server-1.5.9.923-amd64.deb
```

14. Access the shiny app from the server

In order to access to the default shiny app that comes with the server, you have to locate the IPv4 from your AMI. This information is here:

Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP	IPv6 IPs	Key Name	Monitoring	Launch Time
running	2/2 checks ...	None	ec2-35-174-14-21.compute-1.amazonaws.com	35.174.14.21	-	prueba-shiny-a...	disabled	July 2
terminated	2/2 checks ...	None	-	-	-	prueba-shiny-a...	disabled	July 2
running	2/2 checks ...	None	ec2-35-171-186-21.compute-1.amazonaws.com	35.171.186.21	-	prueba-shiny-a...	disabled	July 2

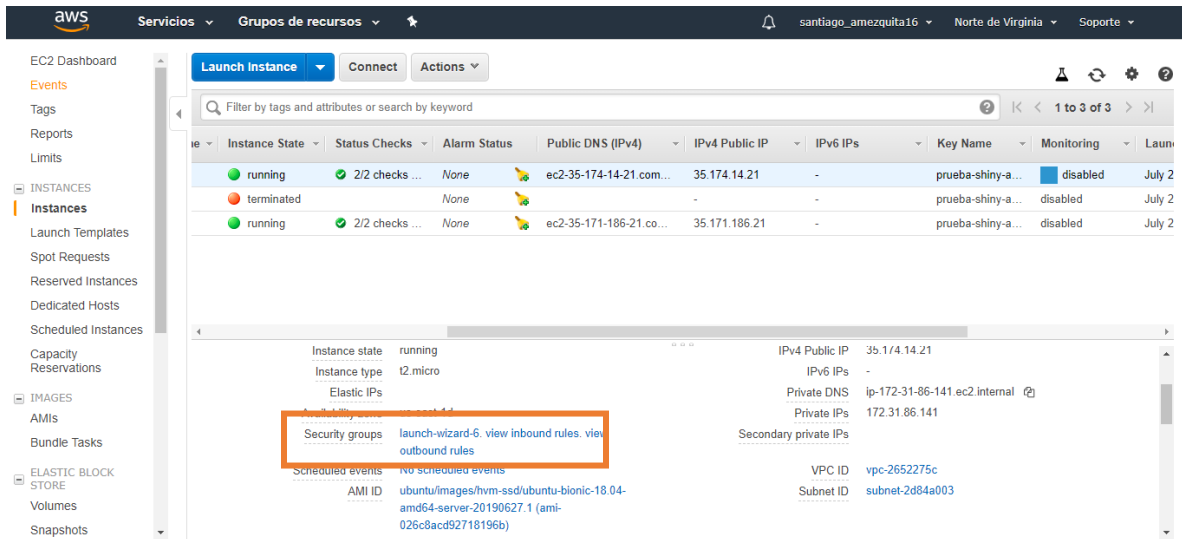
Instance: i-0a8f361a387f90559 (instance1)		Public DNS: ec2-35-174-14-21.compute-1.amazonaws.com	
Description	Status Checks	Monitoring	Tags
Instance ID	i-0a8f361a387f90559	Public DNS (IPv4)	ec2-35-174-14-21.compute-1.amazonaws.com
Instance state	running	IPv4 Public IP	35.174.14.21
Instance type	t2.micro	IPv6 IPs	-
Elastic IPs	-	Private DNS	ip-172-31-86-141.ec2.internal
Availability zone	us-east-1d	Private IPs	172.31.86.141

In my case it is 35.174.14.21 (This won't work because I shutted down the server after this tutorial).

If you type 35.174.14.21:3838 in your browser to access, it wouldn't work because the 3838 port isn't allowed for access.

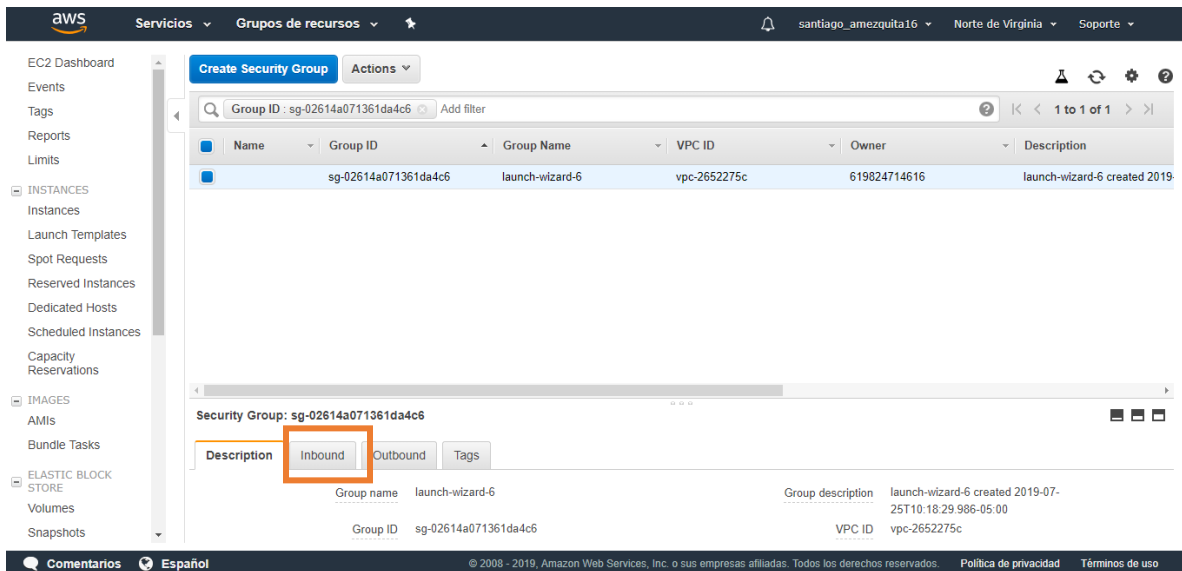
You must create a rule in the AWS server to allow the 3838 port to be accessible.

15. To allow access to the 3838 port you must select the instance and in the bottom part, there's a section called "Security groups":



Click in launch-wizard-6

In this new page, click on "Inbound"



Click on edit

Click on "Add Rule"

16. The information corresponding to shinyApps in the AMI is located on /srv/shiny-server. To access to this information on your AMI, you should type:

```
cd /srv/shiny-server
```

To see which elements are on that folder, type “ls”

Something like this will show up:

```
root@ip-172-31-86-141: /srv/shiny-server
2019-07-25 16:42:49 (22.6 MB/s) - 'shiny-server-1.5.9.923-amd64.deb' saved [61328916/61328916]

root@ip-172-31-86-141:/home/ubuntu# sudo gdebi shiny-server-1.5.9.923-amd64.deb
Reading package lists... Done
Building dependency tree
Reading state information... Done
Reading state information... Done

Shiny Server
Shiny Server is a server program from RStudio, Inc. that makes Shiny applications available over the web. Shiny is a web application framework for the R statistical computation language.
Do you want to install the software package? [y/N]:y
Selecting previously unselected package shiny-server.
(Reading database ... 69017 files and directories currently installed.)
Preparing to unpack shiny-server-1.5.9.923-amd64.deb ...
Unpacking shiny-server (1.5.9.923) ...
^[[ASetting up shiny-server (1.5.9.923) ...
Creating user shiny
Adding LANG to /etc/systemd/system/shiny-server.service, setting to C.UTF-8
Created symlink /etc/systemd/system/multi-user.target.wants/shiny-server.service -> /etc/systemd/system/shiny-server.service.
● shiny-server.service - ShinyServer
   Loaded: loaded (/etc/systemd/system/shiny-server.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2019-07-25 16:43:28 UTC; 80ms ago
 Main PID: 7079 (shiny-server)
    Tasks: 1 (limit: 1152)
   CGroup: /system.slice/shiny-server.service
           └─7079 /opt/shiny-server/ext/node/bin/shiny-server /opt/shiny-server...js

Jul 25 16:43:28 ip-172-31-86-141 systemd[1]: Started ShinyServer.
root@ip-172-31-86-141:/home/ubuntu# cd
root@ip-172-31-86-141:~# cd /srv/shiny-server/
root@ip-172-31-86-141:/srv/shiny-server# ls
index.html  sample-apps
root@ip-172-31-86-141:/srv/shiny-server#
```

That means that you have two files on that folder: index.html and a folder called sample-apps. If you look deep inside the folder “sample-apps” you can see two more folders (this is done using the command line “cd sample-apps” and then “ls”):

```
root@ip-172-31-86-141:/srv/shiny-server# cd sample-apps
root@ip-172-31-86-141:/srv/shiny-server/sample-apps# ls
hello  rmd
root@ip-172-31-86-141:/srv/shiny-server/sample-apps#
```

This shows you that you have two folders, one called “hello” and other called “rmd”. Lets see what “hello” folder has with the command line “cd hello” and “ls”:

```
root@ip-172-31-86-141:/srv/shiny-server/sample-apps# cd hello/
root@ip-172-31-86-141:/srv/shiny-server/sample-apps/hello# ls
server.R  ui.R
root@ip-172-31-86-141:/srv/shiny-server/sample-apps/hello#
```

In the folder “hello” are the files server.R and ui.R, the basic structures for any shinyApp.

If you want to go back to the previous folder, you must use the command “cd ..” and if you want to go back to the root just type “cd”.

To remove a file from the AMI you can use the command:

```
sudo rm fileName
```

```
root@ip-172-31-86-141:/srv/shiny-server/sample-apps# cd ..
root@ip-172-31-86-141:/srv/shiny-server# ls
index.html  sample-apps
root@ip-172-31-86-141:/srv/shiny-server# sudo rm index.html
root@ip-172-31-86-141:/srv/shiny-server# ls
sample-apps
root@ip-172-31-86-141:/srv/shiny-server#
```

In this case, we removed index.html, now our shiny server looks like this:



Where you can access all the apps within the sample-apps folder.

You can add files from your local machine to the AMI by using WinSCP, which can connect directly with PuTTY.

Installing R packages into your AMI

17. Installing packages on AWS machine is a fundamental part of the performance of any application. Sometimes, you must install them via R, and sometimes, via console commands.

In this example we're going to install a very important package for data science called "dplyr". In my case, when I tried to install it via R, it crashed my AMI, therefore, I used the console command to install it. My recommendation will be to install it with the console command and wait:

```
sudo apt-get install r-cran-dplyr
```

If you fail to do so via console, install the package via R with the command "install.packages("packageName")". If by some reason it fails again or takes too long, it may be due to the dependencies of the packages with other packages. You can stop the installation of the package in R by pressing Ctrl+C and it will show you which libraries are the dependencies to install the packages and try to install them using the command used above (sudo apt...).

```
root@ip-172-31-86-141: /srv/shiny-server
Selecting previously unselected package r-cran-rlang.
Preparing to unpack .../12-r-cran-rlang_0.2.0-1_amd64.deb ...
Unpacking r-cran-rlang (0.2.0-1) ...
Selecting previously unselected package r-cran-r6.
Preparing to unpack .../13-r-cran-r6_2.2.2-2_all.deb ...
Unpacking r-cran-r6 (2.2.2-2) ...
Selecting previously unselected package r-cran-utf8.
Preparing to unpack .../14-r-cran-utf8_1.1.3-1_amd64.deb ...
Unpacking r-cran-utf8 (1.1.3-1) ...
Selecting previously unselected package r-cran-pillar.
Preparing to unpack .../15-r-cran-pillar_1.0.1-1_all.deb ...
Unpacking r-cran-pillar (1.0.1-1) ...
Selecting previously unselected package r-cran-tibble.
Preparing to unpack .../16-r-cran-tibble_1.4.1-lubuntul_amd64.deb ...
Unpacking r-cran-tibble (1.4.1-lubuntul) ...
Selecting previously unselected package r-cran-dplyr.
Preparing to unpack .../17-r-cran-dplyr_0.7.4-3_amd64.deb ...
Unpacking r-cran-dplyr (0.7.4-3) ...
Setting up r-cran-bindr (0.1-1) ...
Setting up r-cran-littler (0.3.3-1) ...
Setting up littler (0.3.3-1) ...
Setting up r-cran-crayon (1.3.4-2) ...
Setting up r-cran-assertthat (0.2.0-lubuntul) ...
Setting up r-cran-pkgkitten (0.1.4-1.1) ...
Setting up r-cran-rcpp (0.12.15-1) ...
Setting up r-cran-r6 (2.2.2-2) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Setting up r-cran-utf8 (1.1.3-1) ...
Setting up r-cran-rlang (0.2.0-1) ...
Setting up r-cran-cli (1.0.0-1) ...
Setting up r-cran-magrittr (1.5-4) ...
Setting up r-cran-pkgconfig (2.0.1-2) ...
Setting up r-cran-glue (1.2.0-1) ...
Setting up r-cran-bindrcpp (0.2-1) ...
Setting up r-cran-pillar (1.0.1-1) ...
Setting up r-cran-tibble (1.4.1-lubuntul) ...
Setting up r-cran-dplyr (0.7.4-3) ...
root@ip-172-31-86-141: /srv/shiny-server#
```

Saving the logs of your apps in AMI

18. When an app fails, shiny has the bad habit to don't show the whole log of errors sometimes. To change this execute the following command:

```
sudo nano /etc/shiny-server/shiny-server.conf
```

In this file add the following line **AT THE TOP OF IT**:

```
preserve_logs true;
```

Then, press Ctrl+O and then Enter. Quit the file with Ctrl+X.

After this is done, execute the following command:

```
sudo systemctl reload shiny-server
```

The logs are stored on the path “/var/log/shiny-server”. This feature allows you to see where your app crashes (in case it does).

Important ubuntu Console commands

- Get the root user permissions: `sudo su`
- Access R in ubuntu console: `sudo R`

- Access to a path in ubuntu console: `cd /path/`
- See which files are in a folder: `ls`
- Edit a file in the ubuntu console: `nano filename.extension`
- Install a package in ubuntu console: `sudo apt-get install r-cran-packageName`
- Reset the AWS server through ubuntu console (this should be done each time you upload a new app or apply a change on any of the R scripts): `systemctl restart shiny-server`