

Email: vicsanjinez@gmail.com
Twitter: @victorsanjinez
Github: <https://github.com/vicsanjinez>
Linkedin: <https://www.linkedin.com/in/victor-sanjinez-78484170/>

Parte 3 – Amazon EC2 como servidor Web

Ahora que tenemos una instancia de Amazon Web Services corriendo, es tiempo de administrarla.

En mi caso al usar una AMI Ubuntu me conecto a mi instancia por **SSH**.

(Documentación Accediendo a EC2 por SSH

<http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AccessingInstancesLinux.html>)

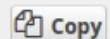
Utilizaremos nuestro Key Pair para realizar la conexión (archivo .pem) (fue descargado al momento de inicializar la instancia con un nombre que nosotros mismos le dimos al archivo)

El primer paso es modificar los permisos del archivo .pem para que no sea accesible por otros usuarios.

(Permisos en linux <https://www.linux.com/learn/understanding-linux-file-permissions>)

2. In a command-line shell, change directories to the location of the private key file that you created when you launched the instance.
3. Use the **chmod** command to make sure that your private key file isn't publicly viewable. For example, if the name of your private key file is `my-key-pair.pem`, use the following command:

```
chmod 400 /path/my-key-pair.pem
```



4. Use the **ssh** command to connect to the instance. You specify the private key (.pem) file and `user_name@public_dns_name`. For Amazon Linux, the user name is `ec2-user`. For RHEL, the user name is `ec2-user` or `root`. For Ubuntu, the user name is `ubuntu` or `root`.

Teniendo el archivo .pem con los permisos correctos podemos conectarnos a nuestra instancia por SSH.

El formato del comando utilizado es el siguiente:

#ssh -i archivo.pem usuario@ip

Ya que nosotros técnicamente no instalamos el sistema operativo de nuestra instancia, con que usuario debemos conectarnos?

Email: vicsanjinez@gmail.com
Twitter: [@victorsanjinez](https://twitter.com/victorsanjinez)
Github: <https://github.com/vicsanjinez>
Linkedin: <https://www.linkedin.com/in/victor-sanjinez-78484170/>

El usuario depende de la AMI (Amazon Machine Image) que seleccionamos, por ejemplo:

Si la AMI es Amazon Linux, el usuario es: **ec2-user**

Si la AMI es Ubuntu, el usuario es: **ubuntu**

```
testaws |⇒ ssh -i testaws.pem ubuntu@34.211.255.42
```

Ejecutando el siguiente comando el proceso de autenticación se inicia.

```
testaws |⇒ ssh -i testaws.pem ubuntu@34.211.255.42
The authenticity of host '34.211.255.42 (34.211.255.42)' can't be established.
ECDSA key fingerprint is SHA256:RZRzL1TZqLl9CLdcAC82Ij+PsmKvkb0SbI8dtQ14MC8.
Are you sure you want to continue connecting (yes/no)?
```

Cuando finaliza la autenticación la conexión por SSH queda establecida.

```
testaws |⇒ ssh -i testaws.pem ubuntu@34.211.255.42
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.4.0-1038-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

Get cloud support with Ubuntu Advantage Cloud Guest:
http://www.ubuntu.com/business/services/cloud

0 packages can be updated.
0 updates are security updates.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-25-3:~$
```

Teniendo acceso a nuestra instancia, podemos instalar un servidor web. Si bien es cierto que podemos instalar los servicios por separado con el comando **apt**, en este caso utilizaremos un instalador completo proporcionado por **Bitnami**.

(Instalador de Bitnami <https://bitnami.com/stack/lamp/installer>)

Email: vicsanjinez@gmail.com
Twitter: @victorsanjinez
Github: <https://github.com/vicsanjinez>
Linkedin: <https://www.linkedin.com/in/victor-sanjinez-78484170/>

Para lo cual debemos descargar el instalador con el comando **wget**:

```
ubuntu@ip-172-31-25-3:~$ wget https://bitnami.com/redirect/to/163849/bitnami-lampstack-5.6.32-0-linux-x64-installer.run
```

También tenemos que modificar los permisos del instalador para que pueda ser ejecutable.

```
ubuntu@ip-172-31-25-3:~$ chmod 755 bitnami-lampstack-5.6.32-0-linux-x64-installer.run
```

Ahora ejecutamos el instalador con el siguiente comando:

#!/nombre_del_instalador

```
ubuntu@ip-172-31-25-3:~$ ./bitnami-lampstack-5.6.32-0-linux-x64-installer.run
Bitnami LAMP Stack requires at least 1000MB of memory and the installer has detected 990MB of memory. This may prevent the application from installing, working properly or cause it to stop functioning due to lack of memory. Visit the following link to learn how to increase the swap space.
https://bitnami.com/lowmemory
Continue with installation? [Y/n]: Y
```

Recordemos que nuestra instancia es **t2.micro**, por lo que el instalador detecta que existe poca memoria, sin embargo, este no es un problema al momento de la instalación.

El instalador de Bitnami incluye los frameworks web mas utilizados, si se desea se pueden seleccionar para que sean instalados junto a los servidores.

Email: vicsanjinez@gmail.com
Twitter: [@victorsanjinez](https://twitter.com/victorsanjinez)
Github: <https://github.com/vicsanjinez>
Linkedin: <https://www.linkedin.com/in/victor-sanjinez-78484170/>

```
to install. Click Next when you are ready to continue.

Varnish [Y/n] :n

Zend Framework [Y/n] :n

Symfony [Y/n] :n

CodeIgniter [Y/n] :n

CakePHP [Y/n] :n

Smarty [Y/n] :n

Laravel [Y/n] :n

PhpMyAdmin : Y (Cannot be edited)

Is the selection above correct? [Y/n]: Y

-----
Installation folder

Please, choose a folder to install Bitnami LAMP Stack

Select a folder [/home/ubuntu/lampstack-5.6.32-0]:

-----
Create MySQL 'root' Account

Bitnami LAMP Stack database root user creation

Password :
Re-enter :

-----
Setup is now ready to begin installing Bitnami LAMP Stack on your computer.

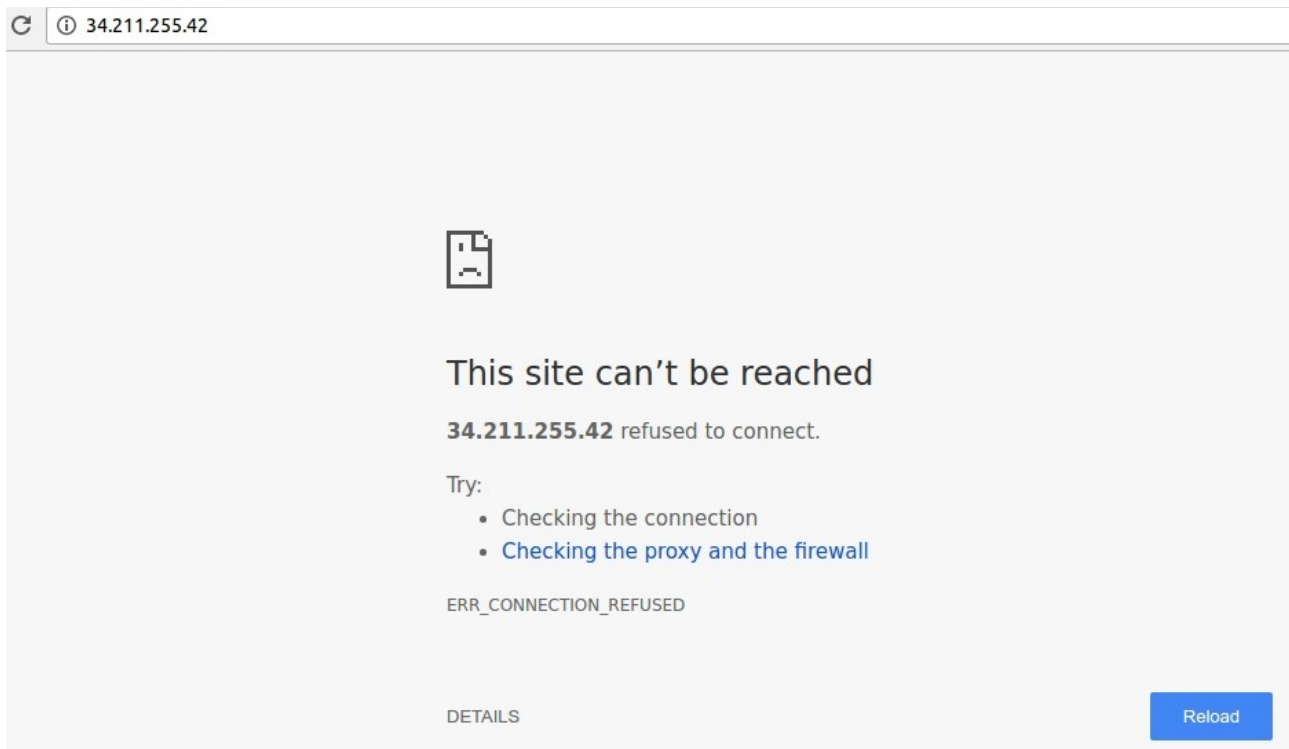
Do you want to continue? [Y/n]: Y

-----
Please wait while Setup installs Bitnami LAMP Stack on your computer.

Installing
0% _____ 50% _____ 100%
#####
```

Con el proceso de instalación terminado, realizamos un test para ver si el servidor web esta funcionando, para ello **ingresamos a la IP Publica de nuestra instancia desde un navegador.**

Email: vicsanjinez@gmail.com
 Twitter: @victorsanjinez
 Github: https://github.com/vicsanjinez
 LinkedIn: https://www.linkedin.com/in/victor-sanjinez-78484170/



Al ver que el servidor web no esta disponible, debemos verificar su funcionamiento. Para ello debemos ingresar hasta la carpeta de instalación haciendo uso del comando **cd**.

#cd /path_instalacion/lampstack-version

Con el comando **ls** podemos observar todos los archivos.

```
ubuntu@ip-172-31-25-3:~$ cd /home/ubuntu/lampstack-5.6.32-0/
ubuntu@ip-172-31-25-3:~/lampstack-5.6.32-0$ ls
README.txt      config          licenses        scripts
apache2         ctlscrip.sh    manager-linux-x64.run  sqlite
apps           docs           mysql           uninstall
changelog.txt  git           php            uninstall.dat
common         img           properties.ini  use_lampstack
ubuntu@ip-172-31-25-3:~/lampstack-5.6.32-0$
```

El archivo **ctlscrip.sh** nos sirve para administrar los servidores, es decir, para iniciar, detener, reiniciar los mismos.

El formato del comando a utilizar es:

#!/ctlscrip.sh operación servicio

En nuestro caso utilizamos el siguiente comando:

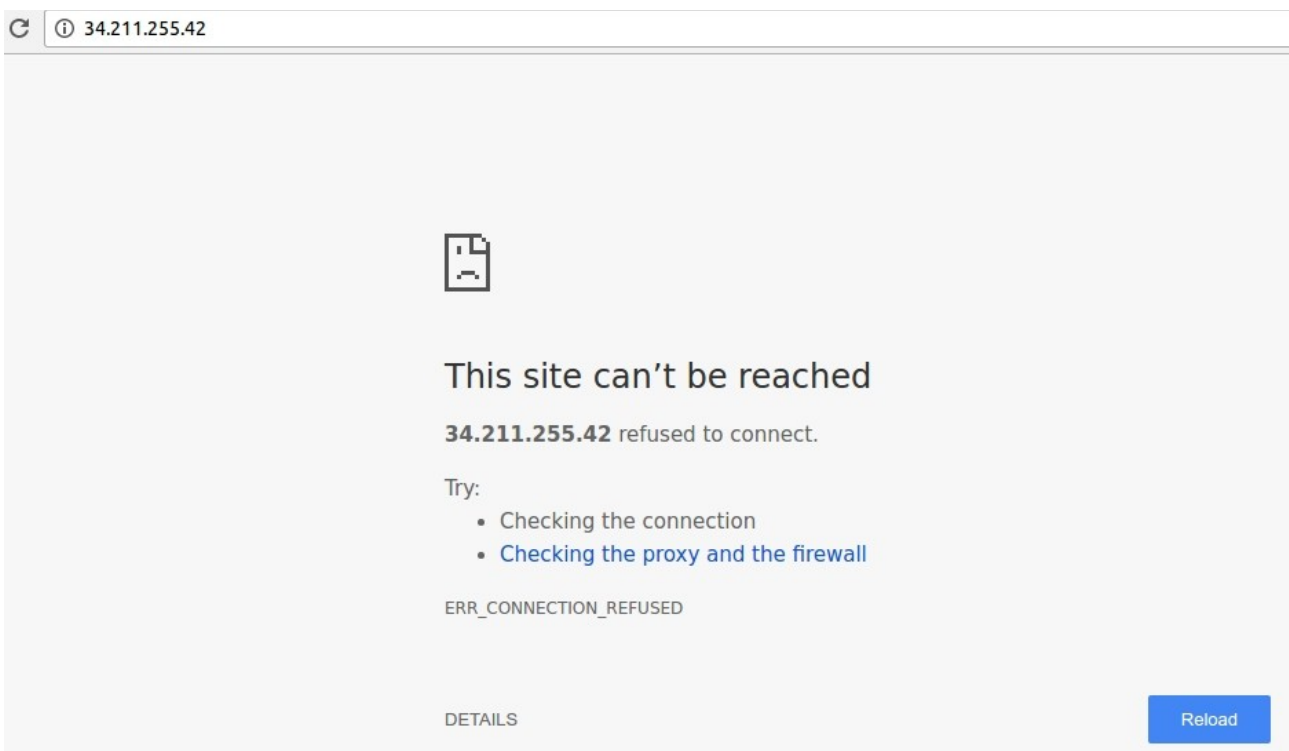
#!/ctlscrip.sh start apache

Email: vicsanjinez@gmail.com
Twitter: @victorsanjinez
Github: <https://github.com/vicsanjinez>
Linkedin: <https://www.linkedin.com/in/victor-sanjinez-78484170/>

#./ctlscript.sh start mysql

```
ubuntu@ip-172-31-25-3:~/lampstack-5.6.32-0$ ./ctlscript.sh start apache
Syntax OK
/home/ubuntu/lampstack-5.6.32-0/apache2/scripts/ctl.sh : httpd (pid 2410) already running
ubuntu@ip-172-31-25-3:~/lampstack-5.6.32-0$ ./ctlscript.sh start mysql
/home/ubuntu/lampstack-5.6.32-0/mysql/scripts/ctl.sh : mysql started at port 3306
```

Y volvemos a realizar el test desde el navegador.



Como vemos aun el servidor no esta disponible, pero porque aun no funciona?

La respuesta son los **Security Groups** de Amazon Web Services

(Documentación de Security Groups

<http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-network-security.html>)

Quando se crea una instancia **todos los puertos están bloqueados a excepción del puerto TCP 22 SSH.**

Email: vicsanjinez@gmail.com

Twitter: @victorsanjinez

Github: https://github.com/vicsanjinez

Linkedin: https://www.linkedin.com/in/victor-sanjinez-78484170/

Group ID : sg-f1a2498d Add filter

Name	Group ID	Group Name	VPC ID	Description
sg-f1a2498d	launch-wizard-4	vpc-7bd9f41c	launch-wizard-4 created 2017-11-23T11:49:29.157-04:00	

Security Group: sg-f1a2498d

Description Inbound Outbound Tags

Edit

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	0.0.0.0/0	

Ya que nosotros intentamos utilizar el **puerto 80 para el servidor web** debemos habilitar el puerto en el Security Group. Para ello en el tab **“Inbound”** hacemos clic en el boton **Edit**.

Edit inbound rules

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom 0.0.0.0/0	ssh
HTTP	TCP	80	Anywhere 0.0.0.0, ::/0	http
HTTPS	TCP	443	Anywhere 0.0.0.0, ::/0	https
Custom TCP F	TCP	20	Anywhere 0.0.0.0, ::/0	ftp 20
Custom TCP F	TCP	21	Anywhere 0.0.0.0, ::/0	ftp 21

Add Rule

NOTE: Any edits made on existing rules will result in the edited rule being deleted and a new rule created with the new details. This will cause traffic that depends on that rule to be dropped for a very brief period of time until the new rule can be created.

Save

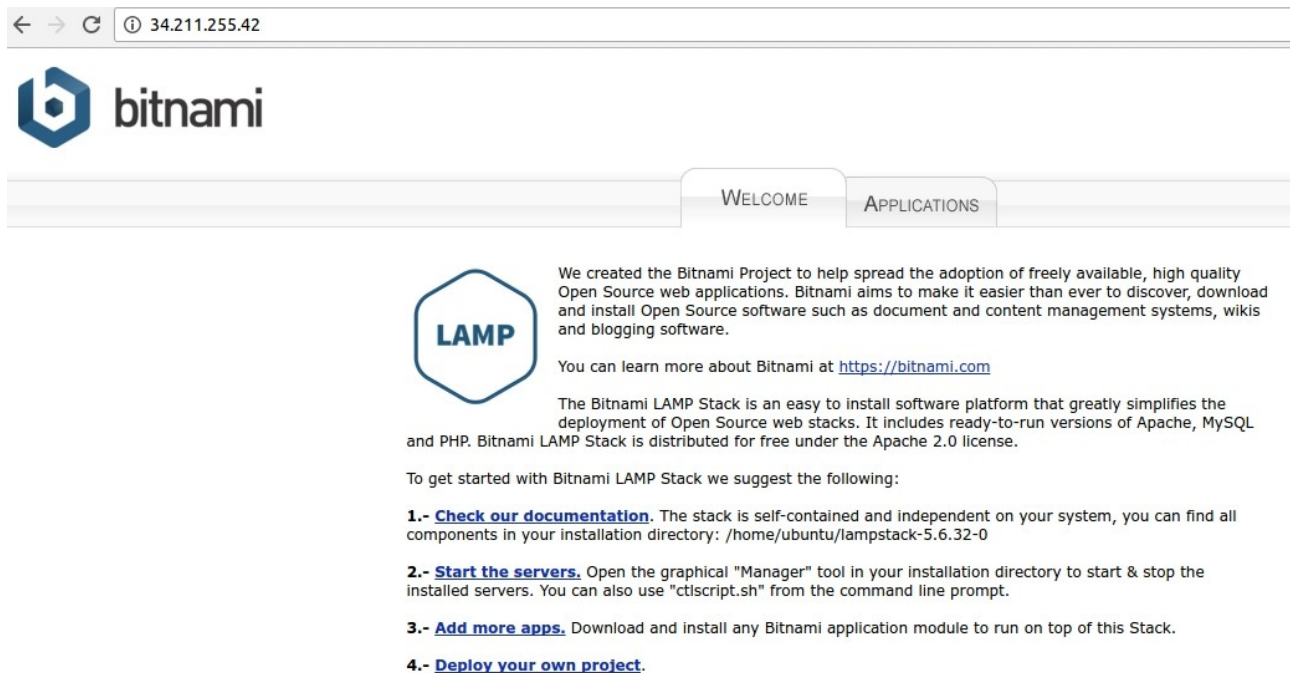
Hacemos clic en el botón **“Add Rule”** y seleccionamos el tipo **HTTP**, el cual **implica el puerto 80 TCP**, también podemos seleccionar desde que IP recibiremos el trafico y por ultimo una descripción de la regla.

Luego hacemos clic en el botón **“Save”** para guardar los cambios.

Nuevamente realizamos el test y finalmente nuestro servidor web se encuentra funcionando correctamente.

Email: vicsanjinez@gmail.com

Twitter: @victorsanjinez

Github: <https://github.com/vicsanjinez>Linkedin: <https://www.linkedin.com/in/victor-sanjinez-78484170/>


The screenshot shows the Bitnami website with a navigation bar containing 'WELCOME' and 'APPLICATIONS'. The main content area features a 'LAMP' logo and text explaining the Bitnami Project's goal to help spread the adoption of freely available, high quality Open Source web applications. It mentions that Bitnami aims to make it easier to discover, download, and install Open Source software such as document and content management systems, wikis, and blogging software. A link to <https://bitnami.com> is provided for more information. The text also states that the Bitnami LAMP Stack is an easy to install software platform that greatly simplifies the deployment of Open Source web stacks, including ready-to-run versions of Apache, MySQL, and PHP. It is distributed for free under the Apache 2.0 license. A section titled 'To get started with Bitnami LAMP Stack we suggest the following:' lists four steps: 1. Check our documentation (The stack is self-contained and independent on your system, you can find all components in your installation directory: /home/ubuntu/lampstack-5.6.32-0), 2. Start the servers (Open the graphical "Manager" tool in your installation directory to start & stop the installed servers. You can also use "ctscript.sh" from the command line prompt), 3. Add more apps (Download and install any Bitnami application module to run on top of this Stack), and 4. Deploy your own project.

Una cosa mas...

La ubicación donde estarán nuestras webs, es:

#cd /path_instalacion/lampstack-version/apache2/htdocs

Podemos crear una web de prueba para hacer un ultimo test.

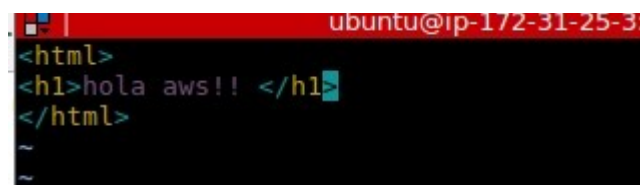
```
ubuntu@ip-172-31-25-3:~/lampstack-5.6.32-0/apache2/htdocs$ mkdir web
ubuntu@ip-172-31-25-3:~/lampstack-5.6.32-0/apache2/htdocs$ cd web/
ubuntu@ip-172-31-25-3:~/lampstack-5.6.32-0/apache2/htdocs/web$ echo '' > index.html
ubuntu@ip-172-31-25-3:~/lampstack-5.6.32-0/apache2/htdocs/web$ vim index.html
```

Para ello creamos una carpeta nueva con el comando **mkdir**.

Accedemos a la carpeta con el comando **cd**.

Creamos un nuevo archivo

Modificamos el archivo con el editor de texto **vim**.



The screenshot shows a terminal window with the command prompt 'ubuntu@ip-172-31-25-3:'. The content of the file 'index.html' is displayed, showing the following HTML code:

```
<html>
<h1>hola aws!! </h1>
</html>
```

Para modificar un archivo con **vim**, una vez abierto con el comando:

Email: vicsanjinez@gmail.com
Twitter: @victorsanjinez
Github: <https://github.com/vicsanjinez>
Linkedin: <https://www.linkedin.com/in/victor-sanjinez-78484170/>

#vim nombre_archivo

- Presionamos la tecla “i”
- Introducimos el texto que queramos
- Para guardar y salir presionamos la tecla **Esc**
- Luego tecleamos **:wq**

(Editar con vim http://www.radford.edu/~mhtay/CPSC120/VIM_Editor_Commands.htm)

Desde el navegador podemos observar que nuestra web funciona correctamente y esta disponible e internet.



hola aws!!

Un saludo y hasta la próxima!