

TALLER 8. Servidor Remoto

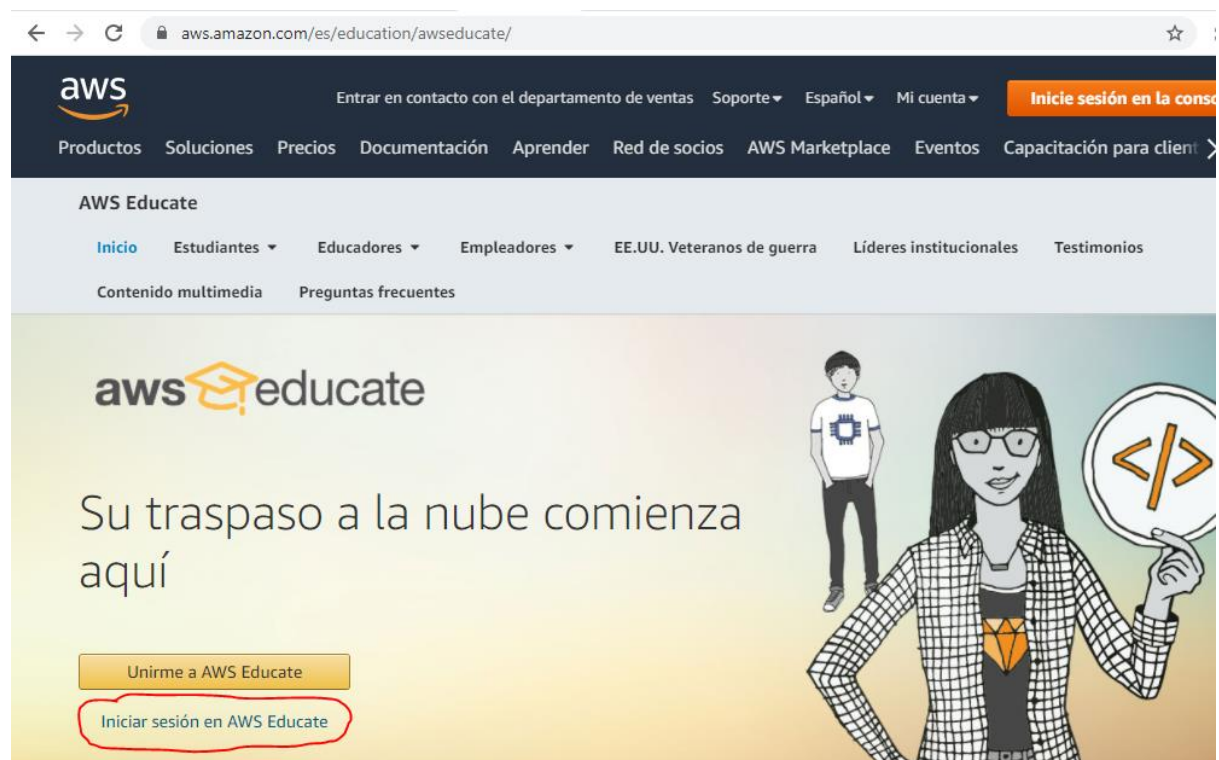
Inicio

El objetivo de este taller es entender como desplegar un servidor en un ambiente remoto.

Se desplegará en un servidor remoto todo el código del servidor desarrollado en los talleres anteriores.

1. Creación de una máquina remota

Activar la cuenta de AWS e ingresar.



awseducate.com/educator/s/

awseducate Contenido Aulas y Créditos Recursos profesionales **AWS Account** Perfil

Zeida Solarte Idioma preferido: Spanish

Hi, welcome back to Educate.

Featured Content

Getting the Most out of AWS Educate
Looking to get the most from your AWS Educate membership? Seeking answers to frequently

awseducate.com/educator/s/awssite

awseducate Contenido Aulas y Créditos Recursos profesionales **AWS Account** Perfil

AWS Educate Starter Account

Your cloud journey has only just begun. Use your AWS Educate Starter Account to access the AWS Console and resources, and start building in the cloud!

AWS Educate Starter Account

Your account has an estimated **134** credits remaining and access will end on **oct. 8, 2021**.

labs.vocareum.com/main/main.php?m=editor&nav=1&asnid=14334&stepid=14335

Vocareum My Classes Help zsol

Welcome to your AWS Educate Account

AWS Educate provides you with access to a wide variety of AWS Services for you to get your hands on and build on AWS! To get started, click on the AWS Console button to log in to your AWS console.

Please read the FAQ below to help you get started on your Starter Account.

- What are the list of services supported?
- What regions are supported with Starter Accounts or Classroom Accounts?
- I can't start any resources. What happened?

Your AWS Account Status

Active
full access ()

\$134.02
remaining credits (estimated)

2:60
session time

Account Details **AWS Console**

Please use AWS Educate Account responsibly. Re shut down your instances when not in use to mal use of your credits. And, don't forget to logout on done with your work!

En la consola de administración de AWS, seleccionar EC2 en la opción de todos los servicios.

us-west-2.console.aws.amazon.com/console/home?nc2=h_ct®ion=us-west-2&src=header-signin#

aws Servicios Grupos de recursos

Consola de administración de AWS

Servicios de AWS

Buscar servicios
Puede escribir nombres, palabras clave o acrónimos.
Ejemplo: Relational Database Service, base de datos, RDS

Servicios visitados recientemente
EC2

Todos los servicios
Informática
EC2
Lightsail
Lambda
Cadena de bloques
Amazon Managed Blockchain
Seguridad, identidad y conformidad
IAM
Resource Access Manager
Coquito

Acceda a los recursos d
Acceda a la consola de aplicación móvil de la
Más información

Explorar AWS
Amazon Redshift
Almacén de datos rápido, senc
ampliar las consultas a su lago
Más información
Ejecute contenedores sin se
AWS Fargate ejecuta y escala s

En el menú a la izquierda seleccionar Instancias

us-west-2.console.aws.amazon.com/ec2/v2/home?region=us-west-2#Home:

aws Servicios Grupos de recursos

New EC2 Experience
Tell us what you think

EC2 Dashboard
Events
Tags
Reports
Limits

INSTANCES
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Scheduled Instances
Capacity Reservations

Welcome to the new EC2 console!
We're redesigning the EC2 console to make it easier to use and improve performance. We'll release new screens periodically. We encourage you to let us know where we can make improvements. To switch between the old console and the new console, use the New EC2 Experience toggle in the top right corner.

EC2

Resources
You are using the following Amazon EC2 resources in the EE.UU. Oeste (Oregón) Region:

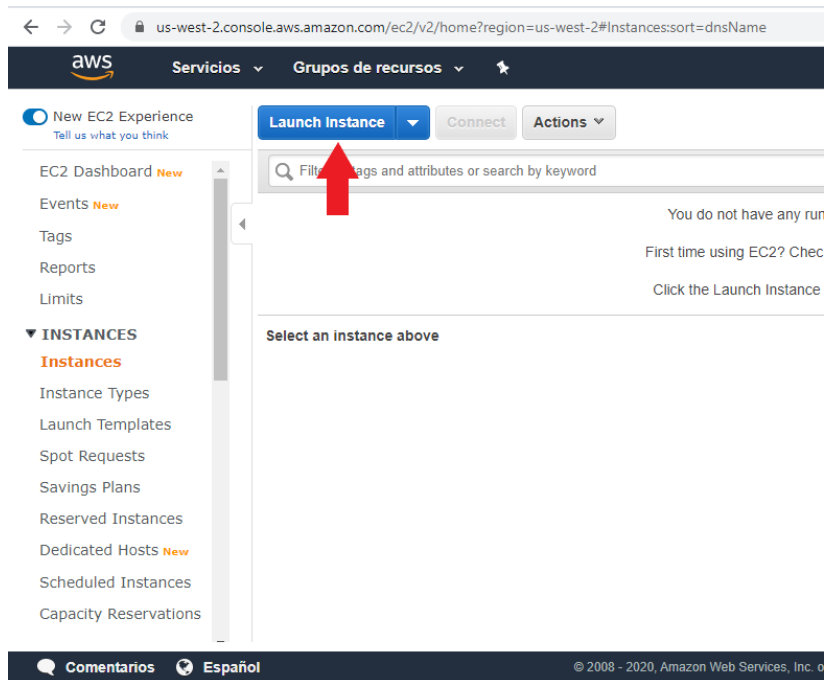
Running instances	0	Elastic IPs	0
Dedicated Hosts	0	Snapshots	0
Volumes	0	Load balancers	0
Key pairs	1	Security groups	3
Placement groups	0		

Facile sine confusione et rianro Microsoft 678: Server Amazon On availability remove on AWS

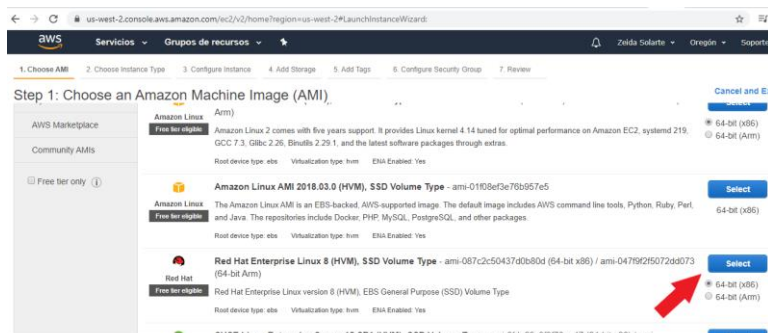
Comentarios Español

© 2008 - 2020 Amazon Web Services, Inc. o sus empresas afiliadas. Todos los derechos reservados.

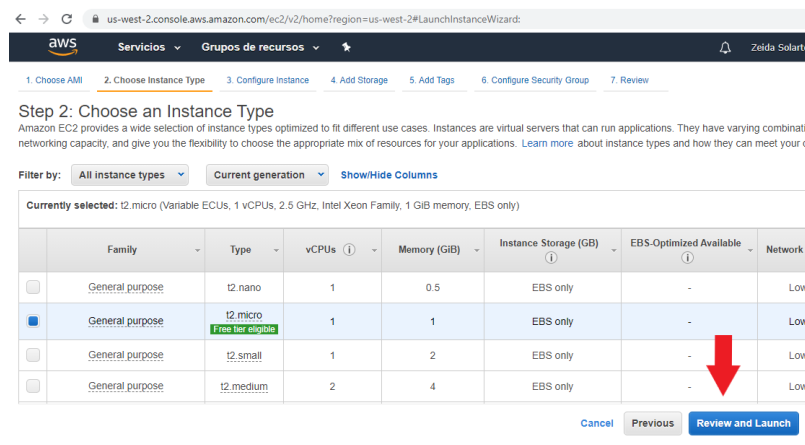
Dar clic en el botón Launch Instance



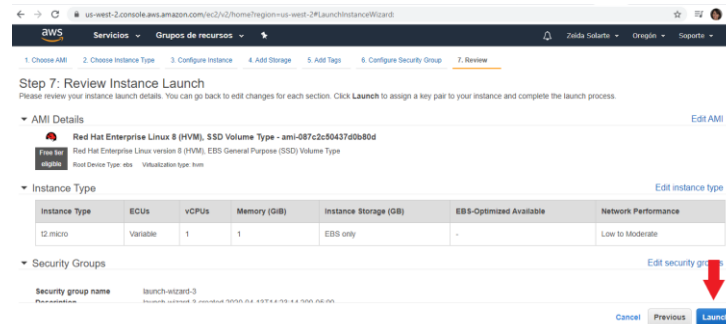
Seleccionar una de las opciones gratis. Recomiendo Red Hat Enterprise Linux 8 (HVM)



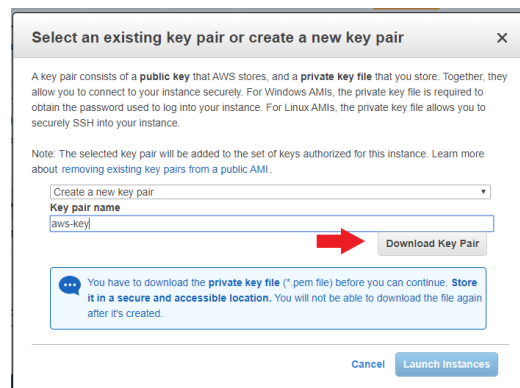
En el tipo de instancia, seleccionar la opción gratis de propósito general y dar clic en el botón review and launch



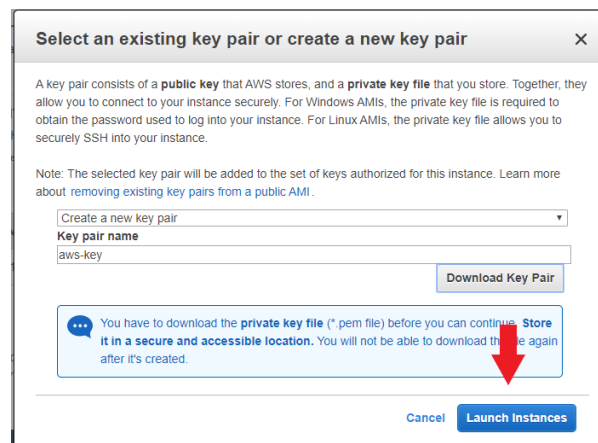
Dar clic en el botón launch



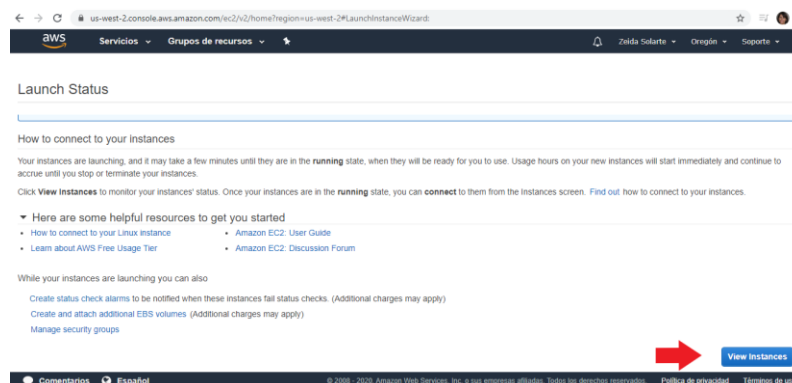
Crear una nueva key-par, darle un nombre y descargarla.



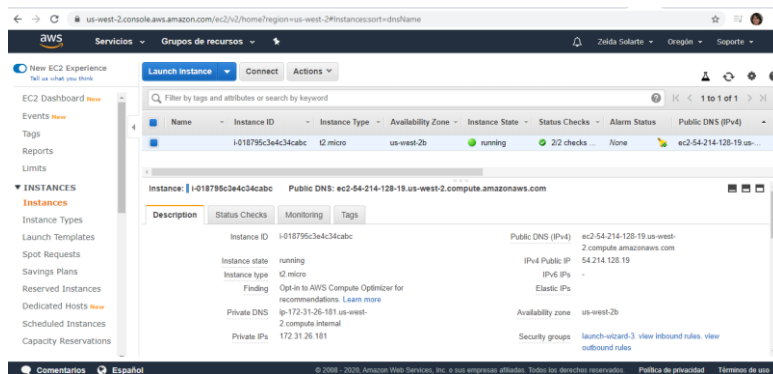
Se debe lanzar la instancia



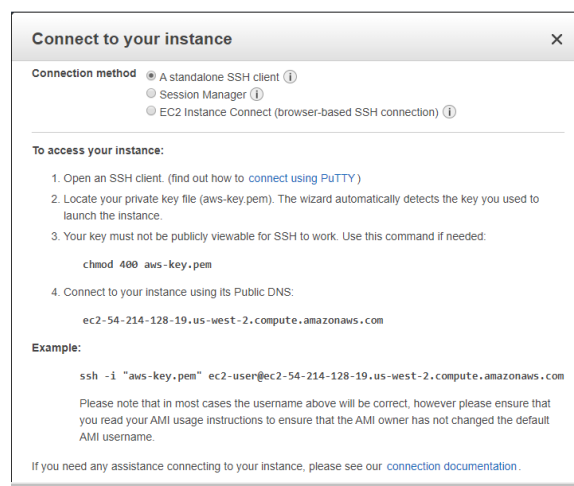
En esta pantalla aparece el resultado de haber lanzado la instancia y nos permite visualizarla.



Aparece la instancia creada, el estado (running) y las características de la misma. Dirección publica, nombre DNS, dirección privada, etc.

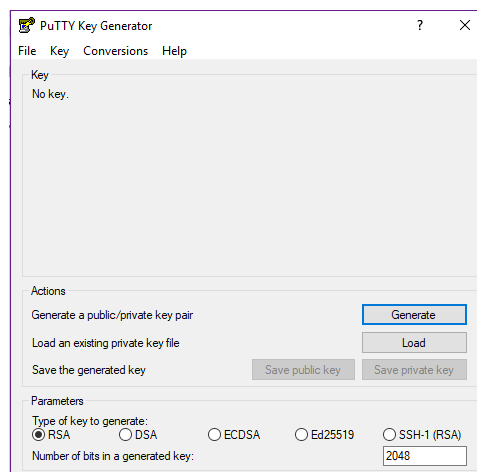


Al darle clic en el botón Connect mostrará las opciones para conectarse a la instancia creada.

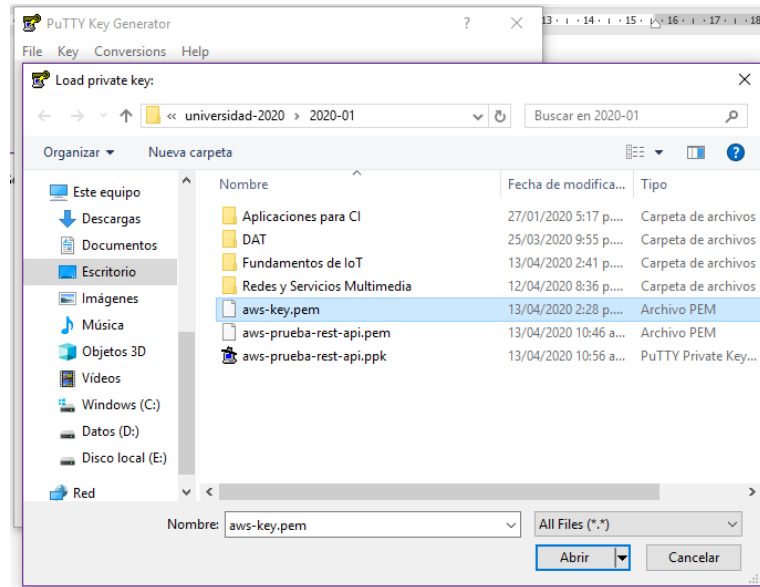


La opción recomendada es usando Putty. Que es un cliente SSH. Se debe descargar también Puttygen para poder convertir la clave a un formato entendible por Putty.

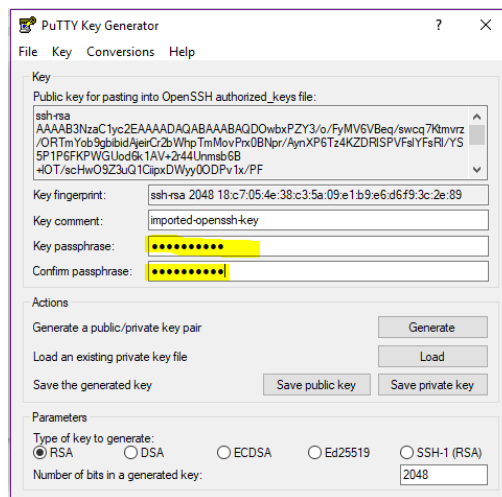
Con Putty.gen se debe cambiar el formato de la clave. Para ello se abre Puttygen con las opciones que se muestran en la gráfica.



Se da clic en el botón Load y se abre el archivo que contiene la key par, generada anteriormente.

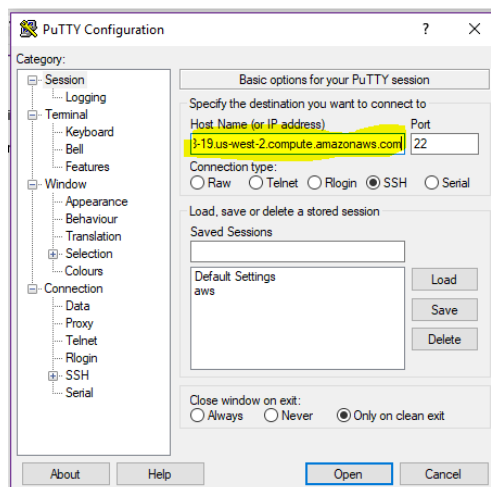


Le damos aceptar en el mensaje de advertencia que sale y añadimos una key passphrase.

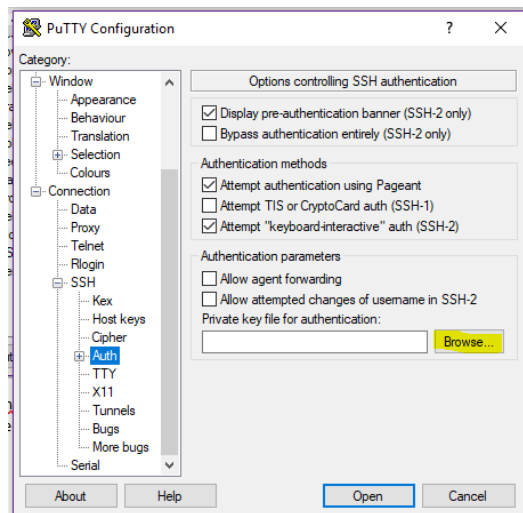


Y damos clic sobre la opción Save private key, la guardamos con el nombre original.

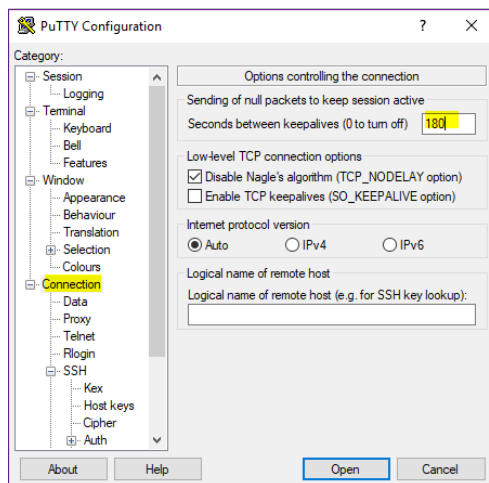
Para establecer la conexión con la instancia creada usamos Putty.



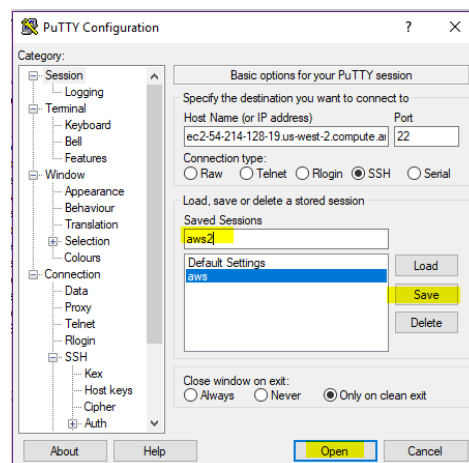
En el hostname colocamos el Public DNS de nuestra instancia. Después desplegamos la opción SSH que se encuentra en Connection y ubicamos la clave privada que creamos con Puttygen.



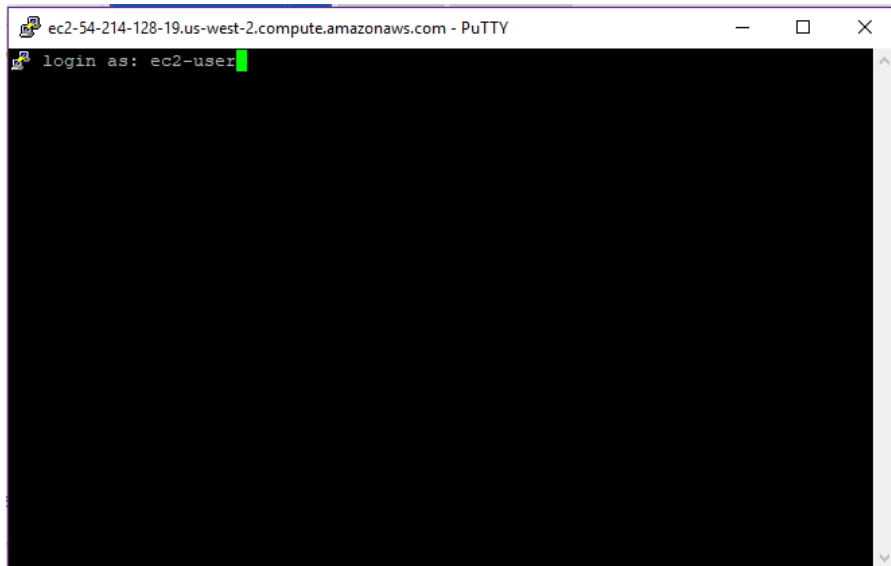
Vamos a connections y lo configuramos para que no se desconecte del servidor tan seguido y se mantengan las sesiones por más tiempo.



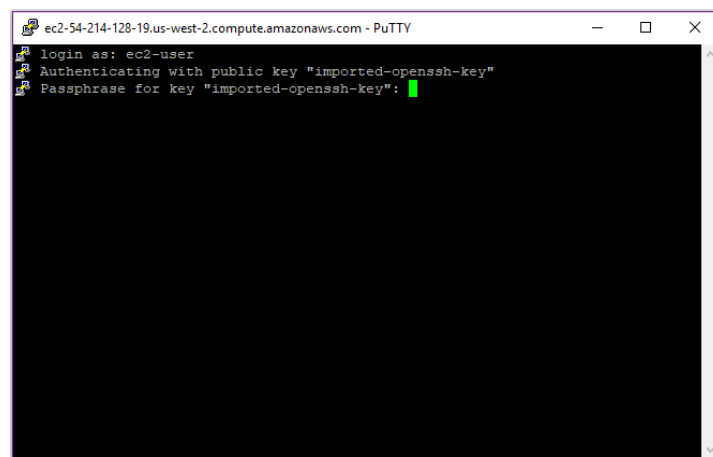
Volvemos a Session y le damos un nombre a la configuración y le damos salvar, para que se almacene la configuración para la siguiente vez que la necesitemos. Por último le damos open para conectarnos a la instancia.



El usuario para conectarnos a la máquina es ec2-user



Y luego pide la passphrase que creamos anteriormente.



Y ya quedamos conectados.

2. Instalación del software requerido para el servidor

- Instalación de paquetes de nodejs

Instalar node.js y los paquetes asociados. Para ello utilizamos la herramienta yum para la instalación de los paquetes.

El comando a usar es:

sudo yum module install nodejs/development

```
ec2-user@ip-172-31-26-181:~$ login as: ec2-user
Authenticating with public key "imported-openssh-key"
Passphrase for key "imported-openssh-key":
[ec2-user@ip-172-31-26-181 ~]$ pwd
/home/ec2-user
[ec2-user@ip-172-31-26-181 ~]$ pwd
/home/ec2-user
[ec2-user@ip-172-31-26-181 ~]$ sudo yum module install nodejs/development
```

- **Instalación del bróker (mosquitto)**

Primero se debe instalar un repositorio de software adicional llamado Paquetes adicionales para Enterprise Linux o EPEL. El comando es:

sudo yum install epel-release

Si este comando falla, ejecutar el siguiente proceso:

sudo yum install wget

wget <https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm>

sudo yum install epel-release-latest-7.noarch.rpm

Esto agrega la información del repositorio de EPEL a nuestro sistema. Ahora podemos instalar mosquitto.

sudo yum install mosquitto

Para poder usar mosquitto se debe iniciar, esto se hace con el comando:

sudo service mosquitto start

- **Instalación de mysql**

```
[ec2-user@ip-172-31-12-176 ~]$ pwd
/home/ec2-user
[ec2-user@ip-172-31-12-176 ~]$ sudo yum install mysql mysql-server
```

...

```
perl-podlators-4.11-1.el8.noarch
perl-threads-1:2.21-2.el8.x86_64
perl-threads-shared-1.58-2.el8.x86_64
policycoreutils-python-utils-2.9-9.el8.noarch
protobuf-lite-3.5.0-7.el8.x86_64

Complete!
[ec2-user@ip-172-31-12-176 ~]$
```

Iniciamos el servicio de mysql

```
[ec2-user@ip-172-31-12-176 ~]$ sudo service mysqld start
Redirecting to /bin/systemctl start mysqld.service
[ec2-user@ip-172-31-12-176 ~]$
```

Se recomienda ejecutar `mysql_secure_installation` para hacer frente a varios problemas de seguridad en una instalación predeterminada de MySQL.

```
[ec2-user@ip-172-31-12-176 ~]$ sudo mysql_secure_installation

Securing the MySQL server deployment.

Connecting to MySQL using a blank password.

VALIDATE PASSWORD COMPONENT can be used to test passwords
and improve security. It checks the strength of password
and allows the users to set only those passwords which are
secure enough. Would you like to setup VALIDATE PASSWORD component?

Press y|Y for Yes, any other key for No:
```

Responda cualquier otra tecla para no validar el nivel de seguridad de la contraseña. Después coloque una contraseña para el ingreso a MySQL y a todas las preguntas siguientes se recomienda contestar yes.

- Creación de bases de datos y tablas.

Se ingresa a mysql.

```
[ec2-user@ip-172-31-12-176 ~]$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 8.0.21 Source distribution

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Aquí se ejecutan los comandos para la creación de la base de datos y tablas de la misma manera como lo hicimos en el servidor local.

```
mysql> create database CaliDatos;
Query OK, 1 row affected (0.01 sec)

mysql> use caliDatos
ERROR 1049 (42000): Unknown database 'caliDatos'
mysql> use CaliDatos
Database changed
mysql> create table datosNodo(
    -> id int auto_increment,
    -> idnodo int,
    -> temperatura double,
    -> humedad double,
    -> horafecha varchar(30),
    -> primary key(id));
Query OK, 0 rows affected (0.05 sec)

mysql>
```

- **Instalación de mongoDB**

Lo primero que se debe hacer es crear un repositorio para la versión de mongodb que se desea instalar. En este caso se hará para la versión 4.2 de mongo. Para esto se crea un archivo con el nombre `mongodb-org-4.2.repo` en la ubicación `/etc/yum.repos.d`

```
[ec2-user@ip-172-31-12-176 proyectoREST]$ cd /etc/yum.repos.d/
[ec2-user@ip-172-31-12-176 yum.repos.d]$ sudo vi mongodb-org-4.2.repo
```

Dentro del archivo copiamos lo siguiente

```
[mongodb-org-4.2]
name=MongoDB Repository
baseurl=https://repo.mongodb.org/yum/redhat/$releasever/mongodb-org/4.2/x86_64/
gpgcheck=1
enabled=1
gpgkey=https://www.mongodb.org/static/pgp/server-4.2.asc
```

```
ec2-user@ip-172-31-12-176:/etc/yum.repos.d
[mongodb-org-4.2]
name=MongoDB Repository
baseurl=https://repo.mongodb.org/yum/redhat/$releasever/mongodb-org/4.2/x86_64/
gpgcheck=1
enabled=1
gpgkey=https://www.mongodb.org/static/pgp/server-4.2.asc
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
-- INSERT --
```

Salvamos y salimos.

Nota: para salvar y salir presiona la tecla esc, luego la tecla dos puntos(:) y por último la tecla x.

Luego instalamos mongo db, con el siguiente comando : `sudo yum install mongodb-org` (a todas las preguntas responder yes)

```
[ec2-user@ip-172-31-12-176 yum.repos.d]$ sudo yum install mongodb-org
```

...

```
Installed:
mongodb-org-4.2.10-1.el8.x86_64
mongodb-org-mongos-4.2.10-1.el8.x86_64
mongodb-org-server-4.2.10-1.el8.x86_64
mongodb-org-shell-4.2.10-1.el8.x86_64
mongodb-org-tools-4.2.10-1.el8.x86_64
python2-2.7.17-1.module+el8.2.0+4561+f4e0d66a.x86_64
python2-libs-2.7.17-1.module+el8.2.0+4561+f4e0d66a.x86_64
python2-pip-9.0.3-16.module+el8.2.0+5478+b505947e.noarch
python2-pip-wheel-9.0.3-16.module+el8.2.0+5478+b505947e.noarch
python2-setuptools-39.0.1-11.module+el8.1.0+3446+c3d52da3.noarch
python2-setuptools-wheel-39.0.1-11.module+el8.1.0+3446+c3d52da3.noarch

Complete!
[ec2-user@ip-172-31-12-176 yum.repos.d]$
```

y después se inicia mongo con el comando: `sudo service mongod start`

```
[ec2-user@ip-172-31-12-176 yum.repos.d]$ sudo service mongod start
Redirecting to /bin/systemctl start mongod.service
[ec2-user@ip-172-31-12-176 yum.repos.d]$
```

Ya estaría mongo instalado y ejecutándose. Para usarlo solo habría que escribir el comando `mongo` en el terminal:

```
2020-10-21T21:46:36.588+0000 I  CONTROL  [initandlisten]
2020-10-21T21:46:36.588+0000 I  CONTROL  [initandlisten] ** WARNING: /sys/kernel
/mm/transparent_hugepage/enabled is 'always'.
2020-10-21T21:46:36.588+0000 I  CONTROL  [initandlisten] **          We suggest se
tting it to 'never'
2020-10-21T21:46:36.588+0000 I  CONTROL  [initandlisten]
---
Enable MongoDB's free cloud-based monitoring service, which will then receive an
d display
metrics about your deployment (disk utilization, CPU, operation statistics, etc)
.

The monitoring data will be available on a MongoDB website with a unique URL acc
essible to you
and anyone you share the URL with. MongoDB may use this information to make prod
uct
improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeM
onitoring()
---
> █
```

Con el comando `exit` salimos del ambiente de mongo.

```
To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeM
onitoring()
---
> exit
bye
[ec2-user@ip-172-31-12-176 yum.repos.d]$ █
```

3. Creación y configuración del proyecto

- Creación del espacio del proyecto

Creamos una carpeta para almacenar nuestro proyecto, en este caso la he llamado `servidorIoT` y la he creado en la carpeta del usuario `ec2-user`.

```
[ec2-user@ip-172-31-21-23 ~]$ mkdir servidorIoT
[ec2-user@ip-172-31-21-23 ~]$ cd servidorIoT/
[ec2-user@ip-172-31-21-23 servidorIoT]$ █
```

Ejecutamos el comando: `npm init --yes`

```
[ec2-user@ip-172-31-21-23 servidorIoT]$ npm init --yes
Wrote to /home/ec2-user/servidorIoT/package.json:

{
  "name": "servidorIoT",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}
```

- **Instalación de los paquetes de soporte a MQTT y REST**

Se debe instalar la librería de MQTT que permitirá conectarnos al bróker, para esto usamos el comando

npm install mqtt --save

Se deben instalar los paquetes express y morgan como soporte a REST, esto se hace con el siguiente comando:

npm i express morgan

- **Instalación de los paquetes de soporte a MySQL y MongoDB**

Se debe añadir la librería de mysql al proyecto. para esto usamos el comando

npm i mysql

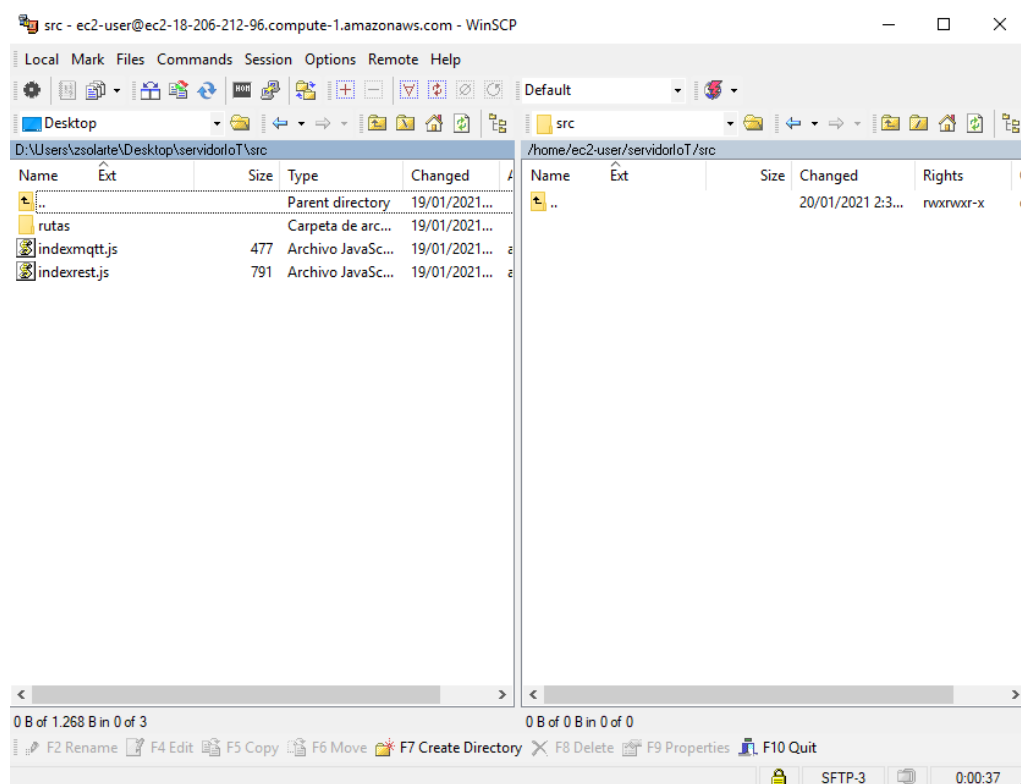
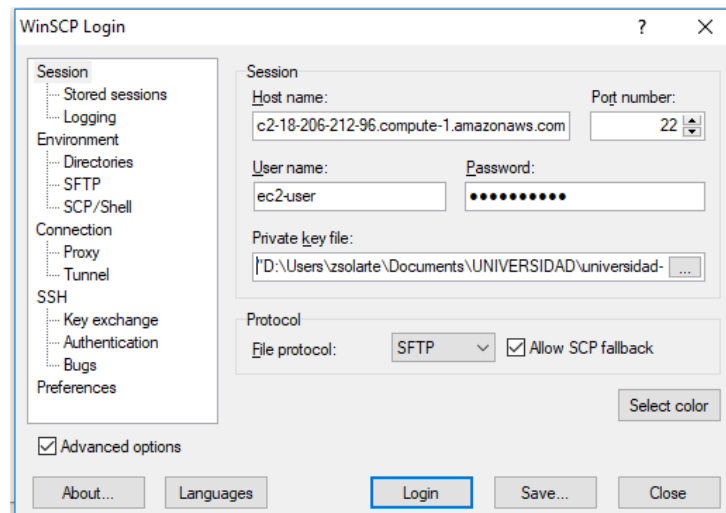
y para instalar las librerías de mongodb, lo hacemos con el siguiente comando:

npm i mongodb

4. Transferencia de archivos

El código a desarrollar es exactamente el mismo que usamos en la máquina local, por tanto lo que haremos es subir los archivos ya desarrollados.

Para pasar archivos hacia la maquina creada, se puede usar winscp. El cual se configura con los datos de la instancia y con la clave creada. Generalmente él toma los datos de la conexión realizada con Putty y se conecta.



Aquí simplemente se arrastran los archivos a las carpetas correspondientes.

5. Habilitar puertos

Para poder acceder al servidor desde un cliente externo se deben habilitar los puertos 1883 (mosquitto) y 3000 (servidor REST) en la instancia. Para ello seleccionamos el primer security group asociado a la instancia.

console.aws.amazon.com/ec2/v2/home?region=us-east-1#InstanceDetails:instanceId=i-044a240ce67f590b1

Resumen de instancia de i-044a240ce67f590b1 Información

Se ha actualizado hace less than a minute

ID de la instancia: i-044a240ce67f590b1

Estado de la instancia: **En ejecución**

Tipo de instancia: t2.micro

Hallazgo de AWS Compute Optimizer: **User: arn:aws:sts::367311873900:assumed-role/vocstartsoft/user783988-zsolate@uao.edu.co is not authorized to perform: compute-optimizer:GetEnrollmentStatus on resource: * with an explicit deny**

Dirección IPv4 pública: 18.206.212.96 | [dirección abierta](#)

DNS de IPv4 pública: ec2-18-206-212-96.compute-1.amazonaws.com | [dirección abierta](#)

Direcciones IP elásticas: -

Rol de IAM: -

Seguridad

Detalles de seguridad

Rol de IAM: -

ID del propietario: 367311873900

Grupos de seguridad: sg-021d3a2978e1083f3 (launch-wizard-7)

Damos clic en el security group asociado a la instancia

sg-021d3a2978e1083f3 - launch-wizard-7 Acciones

Detalles

Nombre del grupo de seguridad: launch-wizard-7

ID del grupo de seguridad: sg-021d3a2978e1083f3

Descripción: launch-wizard-7 created 2021-01-20T13:53:45-730-05:00

ID de la VPC: vpc-22a9605f

Propietario: 367311873900

Número de reglas de entrada: 1 Entrada de permiso

Número de reglas de salida: 1 Entrada de permiso

Reglas de entrada Reglas de salida Etiquetas

Reglas de entrada Editar reglas de entrada

Tipo	Protocolo	Intervalo de puertos	Origen	Descripción: opcional
SSH	TCP	22	0.0.0.0/0	-

Y editamos las reglas de entrada y las guardamos:

Editar reglas de entrada Información

Las reglas de entrada controlan el tráfico entrante que puede llegar a la instancia.

Reglas de entrada Información

Tipo	Protocolo	Intervalo de puertos	Origen	Descripción: opcional	
SSH	TCP	22	Personalizada	Q	Eliminar
TCP personalizado	TCP	3000	Personalizada	Q	Eliminar
TCP personalizado	TCP	1883	Personalizada	Q	Eliminar

Agregar regla

NOTA: al modificar las reglas existentes, se eliminará la regla editada y se creará otra nueva con los nuevos detalles. Esto provocará que se interrumpa el tráfico que depende de dicha regla durante un breve periodo de tiempo, hasta que se pueda crear la regla nueva.

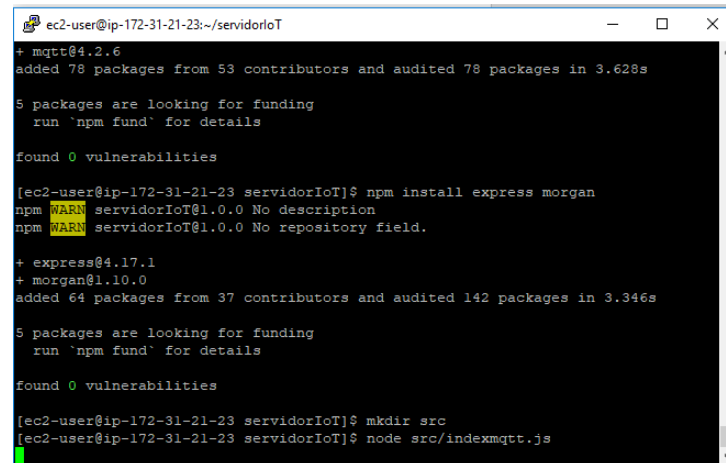
Cancelar Previsualizar los cambios **Guardar reglas**

6. Levantar los servicios

Para hacer la prueba lanzamos el servidor, de la misma manera como lo hacíamos en el servidor local.

node src/indexmqtt.js

node src/indexrest.js



```
ec2-user@ip-172-31-21-23:~/servidorIoT
+ mqtt@4.2.6
added 78 packages from 53 contributors and audited 78 packages in 3.628s

5 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

[ec2-user@ip-172-31-21-23 servidorIoT]$ npm install express morgan
npm WARN servidorIoT@1.0.0 No description
npm WARN servidorIoT@1.0.0 No repository field.

+ express@4.17.1
+ morgan@1.10.0
added 64 packages from 37 contributors and audited 142 packages in 3.346s

5 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

[ec2-user@ip-172-31-21-23 servidorIoT]$ mkdir src
[ec2-user@ip-172-31-21-23 servidorIoT]$ node src/indexmqtt.js
```

Tener en cuenta que hay que tener activos todos los servicios necesarios para la ejecución del proyecto.

Para que el servidor no quede bloqueado en espera de las peticiones de los clientes, se puede usar PM2 que es un administrador de procesos para aplicaciones de Node.js. PM2 permite implementar demonios en aplicaciones para que puedan funcionar en segundo plano como servicios.

Use npm para instalar la última versión de PM2 en su servidor:

sudo npm install pm2@latest -g

La opción -g indica a npm que instale el módulo de forma global, de modo que esté disponible en todo el sistema.

Se usa el comando pm2 start para ejecutar la aplicación en segundo plano:

pm2 start src/indexmqtt.js

Al ejecutar este comando se le asigna un nombre y un identificador a la aplicación.

Para detener una aplicación se usa este comando (especifique App name o id de PM2):

pm2 stop app_name_or_id

Para reiniciar una aplicación:

pm2 restart app_name_or_id

Para obtener una lista de las aplicaciones actualmente administradas por PM2:

pm2 list

7. Envío de datos al servidor

El envío de datos al servidor se hace de la misma manera que lo hacíamos al servidor local teniendo en cuenta que debemos cambiar “localhost” por el nombre DNS de la maquina remota, lo cual lo encontramos en la información de la instancia en la consola de AWS.

Realizar la configuración de la aplicación en java para que envíe los datos al servidor remoto, en los diferentes escenarios (MQTT y REST, MySQL y MongoDB) y verifique que los datos se almacenen de manera correcta.