

Universidad industrial de Santander

Instalación y configuración de la maquina virtual proporcionada por AWS.

Enero de 2024

Carlos Eduardo Silva Sepúlveda

Brayan Hernando Gonzales Mendoza

Tabla de contenido

Introducción	3
Conceptos previos	4
Tarea 1: Inscripción a AWS	5
Tarea 2: Crear una VM (instancia) de tipo EC2	5
Tarea 3. Configuración de la VM para soportar clientes externos iot.	7
Tarea 4. El día a día para encender la máquina y contar con una terminal de trabajo	9
Tarea 5. Instalación de Node.js	11
Tarea 6. Instalación de Node Red.....	12
Tarea 7. Pruebas de reconocimiento a Node Red	13
Tarea 8. Configuración de Node Red.	17
Tarea 9. Mantenimiento de Node RED.	19
REFERENCIAS	19

Introducción

En la era de la computación en la nube, la plataforma Amazon Web Services (AWS) se ha destacado como un líder indiscutible, ofreciendo una amplia gama de servicios para satisfacer las necesidades de empresas y desarrolladores. En este contexto, la implementación de una máquina virtual en AWS se convierte en un paso esencial para desplegar y ejecutar diversas aplicaciones y servicios de manera eficiente.

Una herramienta valiosa que puede potenciar la funcionalidad y la automatización en tu máquina AWS es Node-RED. Node-RED es un entorno de programación basado en nodos de código abierto que facilita la conexión de dispositivos, APIs y servicios en un flujo de trabajo visual. Este entorno intuitivo permite a los usuarios diseñar flujos de trabajo fácilmente, lo que resulta especialmente útil para la creación rápida de aplicaciones IoT (Internet de las cosas) y la automatización de tareas.

Conceptos previos

VM: Significa Virtual Machine o máquina virtual. En general, una máquina virtual se refiere a una réplica virtualizada de un entorno de hardware de computadora. Esto incluye recursos como CPU, memoria, almacenamiento y red. Una VM se utiliza para ejecutar sistemas operativos y aplicaciones, y puede ser equivalente a una computadora física en muchos aspectos. Las plataformas en la nube, como Google Cloud, AWS, Azure, Alibaba usan ese concepto para referirse a un servicio que ofrecen para que tu cuentes con algo que hace lo mismo que un computador, pero está en la nube de alguna de esas plataformas.

Instancia: Este término es conocido en la programación orientada a objetos, donde cada objeto es una instancia de una clase. Pero aquí tiene una connotación un poco diferente porque hablar de una instancia en AWS es hablar de una máquina virtual que se ha creado a imagen de otra que ofrece la plataforma. Imagina que el profesor te da la guía de un taller, tu le sacas una copia y luego trabajas sobre esa copia. Eso significa que tú has instanciado la guía del taller y ahora trabajas en una instancia del mismo. A diferencia de las VM tradicionales, las instancias son flexibles en términos de poder reconfigurar sus componentes. En adelante, en este trabajo, cuando usemos el término instancia o VM nos referiremos a lo mismo.

EC2: EC2 significa Elastic Cloud. Pero en realidad es un tipo de máquinas virtuales que ofrece AWS que permiten ser adaptadas aún después de creadas para contener mayores o menores recursos como CPU, memoria, almacenamiento y rendimiento de red, sistemas operativos, reglas de seguridad, el tipo de IP elástica o estática, para satisfacer tus necesidades específicas. Eso las hace flexibles y escalables. Es en el fondo una idea de negocios, donde la persona paga justo por lo que usa, ni más ni menos. EC2 es uno de los servicios más populares de AWS y es ampliamente utilizado para alojar aplicaciones web, ejecutar cargas de trabajo de análisis de datos, ejecutar servidores de aplicaciones, alojar sitios web y más. AWS también cuenta con una amplia gama de EC2 preconfiguradas para ser instanciadas para diversos casos de uso, como cómputo general, cómputo de alto rendimiento, cómputo optimizado para memoria, y más.

EBS: Significa Elastic Block Store. Es un tipo de almacenamiento que se le puede dar a una EC2 que es flexible.

AMI: "Amazon Machine Image" (Imagen de Máquina Amazon). Una AMI es una plantilla preconfigurada que contiene la información necesaria para lanzar una instancia de EC2 (Elastic Compute Cloud), que es una máquina virtual en la nube de AWS. Las amis son fundamentales en AWS, ya que te permiten crear instancias de EC2 de manera eficiente y reproducible.

IP estática: Una máquina tiene una IP estática si cuando se reinicia mantiene siempre la misma IP.

IP dinámica: Una máquina tiene una IP dinámica cuando se reinicia cambia la misma IP.

IP elástica: AWS es fanático del término “elastic”, quieren proyectar que todo es elástico. Por eso, cuando hablan de IP elástico se refieren en realidad a una IP estática, pero que el usuario puede usar a su antojo, es decir, la puede usar en una VM o trasladarla a otras VM o a recursos en la nube, como instancias EC2 o balanceadores de carga. La principal ventaja de una IP elástica en AWS es que se mantiene constante incluso cuando detienes y luego reinicias la instancia a la que está asociada. Esto permite una mayor flexibilidad y facilita la alta disponibilidad y la recuperación ante desastres. En conclusión, AWS no ofrece un servicio que se llame “IP estática”, eso lo suple con un servicio de IP estática. Desde ese punto de vista “IP estática” no es un tipo de IP, sino un servicio AWS.

Tarea 1: Inscripción a AWS

Proceso a seguir
https://aws.amazon.com/ > registra tus datos

Tarea 2: Crear una VM (instancia) de tipo EC2

VM es la sigla que usaremos para referirnos a una máquina virtual, El término que usa AWS para referirse a lo mismo es “instancia” que significa que estamos instanciando un modelo de máquina que ya tiene AWS.

Paso 1. Entra a la consola. Es decir, inicia sesión en AWS
https://aws.amazon.com/ > Sign in to the Console >
Paso 2. Creación de la VM
Lunch a virtual machine With EC2 > Name and tags (das un nombre que permita diferenciar esta VM de otras)> Entre las imágenes disponibles selecciona “Ubuntu”> OJO: No oprima “Launch Instance” hasta que no se le indique.

Paso 3. Creación de par de claves y lanzamiento (osea que la VM quede corriendo).

Algunos considerandos previos:

- Esta parte es muy necesaria en AWS. Está pensada en la seguridad, para que personas no autorizadas no puedan acceder a una VM

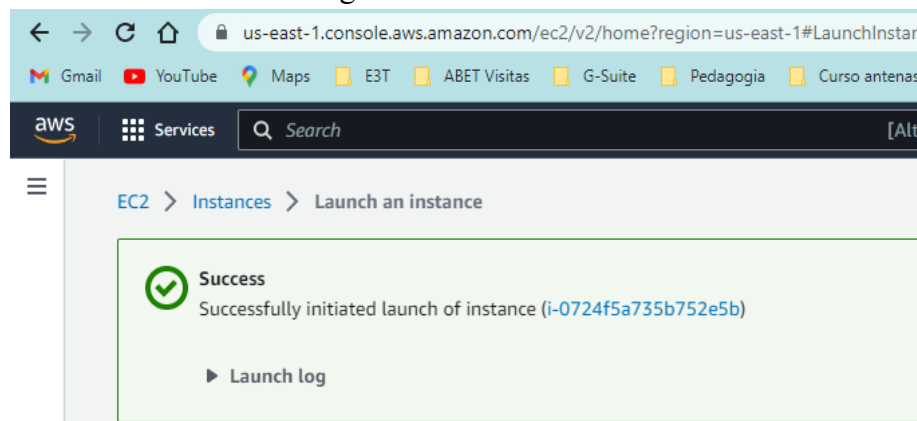
- Esas claves son necesarias cuando se intente acceder a la VM desde terminales remotos usando aplicaciones como powershell, putty o VNC. En nuestra experiencia no usaremos esas aplicaciones, sino que aprovecharemos una terminal en línea que ofrece AWS. De esta manera evitaremos usar estas claves, pero las vamos a cuidar por si se requieren a futuro.
- Son dos claves, la pública queda en la VM y se puede compartir con otros usuarios o sistemas, se usa para cifrar datos que se envían a la VM. La privada baja al computador del usuario en forma de un archivo .pem. También es posible bajar como un archivo .ppk. **Pero es más recomendable el archivo en formato .pem.** Sirve a la hora de configurar una terminal de acceso remoto a la VM, debe mantenerse segura y se utiliza para descifrar los datos que se reciben en la instancia EC2.
- Una recomendación es que le des a ese archivo el mismo nombre de la VM, ya que en el futuro puedes tener varias VM y por consiguiente varios de estos archivos

Paso 3. Crea el par de claves


Continuando desde donde estás creando la instancia > haz click en “Create new key pair” > cuando te pida un nombre para el par de claves dale el mismo dado a la máquina en el paso anterior> Oprime “Create key par” > Revisa el disco de tu computadora, debe haber bajado un archivo con nombre igual al tag de la VM y con extensión .pem > Guarda ese archivo en una carpeta selecta para que no se te pierda.

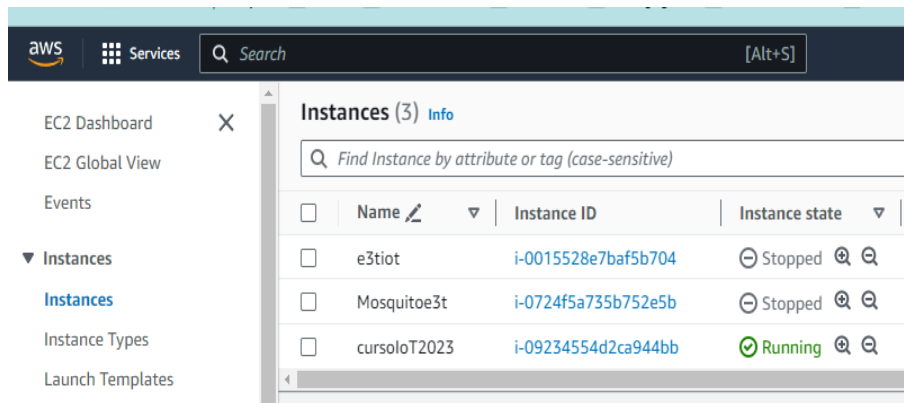
Paso 4. Finaliza la creación de la instancia (VM)

Oprime “Launch instance”. Verás algo así como resultado:



Revisa el estado de tu máquina

Menú lateral izquierdo > Instances > Instances. El siguiente es un ejemplo de lo que verás si la máquina creada se llamara cursoiot2023. Nota: si no vez tu VM refresca la vista del navegador con 



Vemos que la máquina queda encendida por defecto “Running”

Prueba apagar y encender la máquina. NOTA: es importante apagarla cuando no la uses si no deseas consumir recursos.

Chequeas la instancia de interés > Menu horizontal > Instance state > “Stop instance” para apagarla o “Start Instance” para encenderla

Tarea 3. Configuración de la VM para soportar clientes externos iot.

Las VM en la nube vienen usualmente acompañadas con un conjunto de reglas de seguridad que permiten que el dueño de la máquina pueda prevenir el uso indebido. Básicamente, lo que se busca es que estén controlados los puertos en la VM. Se trata de:

- Reglas de entrada (Inbound rules): sirven para permitir o denegar el tráfico entrante a través de ciertos puertos de red. Ellas controlan el acceso a la VM desde fuentes externas, bien sea desde otras VM, servidores web, computadores de usuarios, dispositivos iot, etc.
- Reglas de salida (Outbound rules): es similar a las reglas de entrada, pero para el control del tráfico de salida.
- La configuración por defecto de esa regla no es suficiente para permitir la entrada de los datos que envían dispositivos iot.

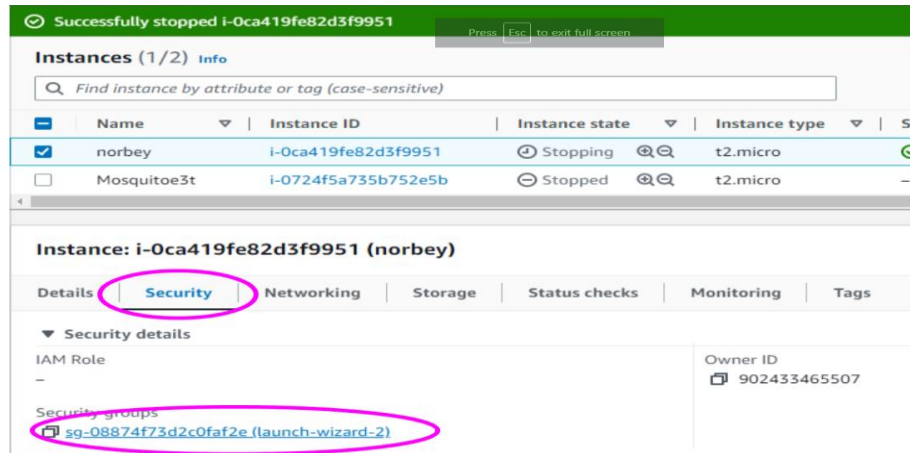
Paso 1. Revisar las reglas de seguridad de tu instancia o VM

Si te saliste del panel que lista de las instancias creadas, vuelve all. Si no sabes como, puede ser así:

<https://aws.amazon.com/> > Sign in > EC2 > Menu lateral > Instances > Instances

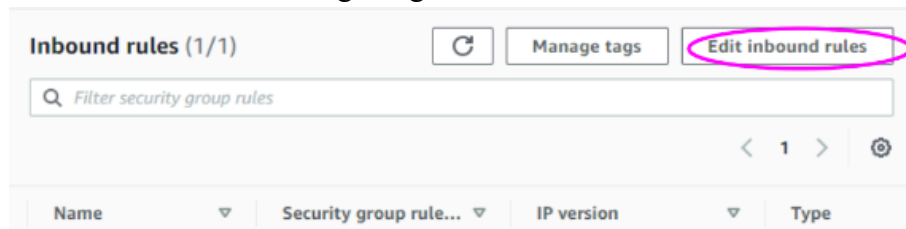
Paso 2. La configuración en sí de las reglas de seguridad

Chequeas la instancia de interés > **asegurate que la VM esté apagada** > Security > busca “Security Groups” > haz click en el enlace que aparece debajo de “Security groups”

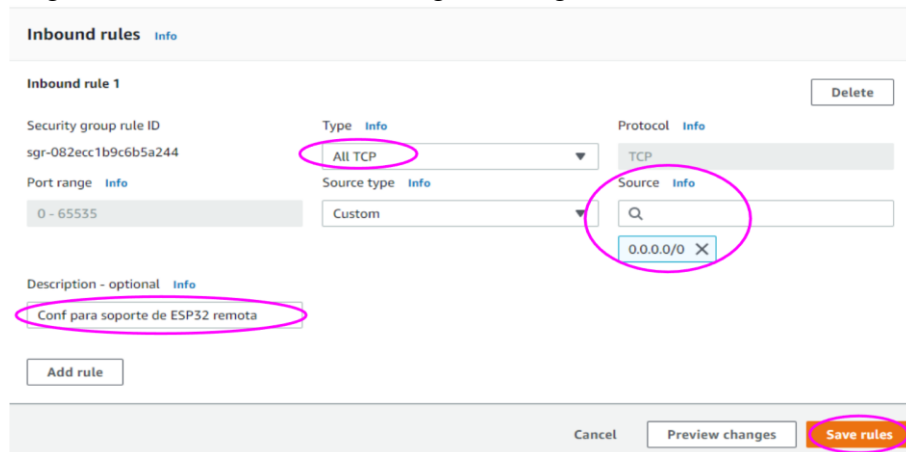


Editar la regla de seguridad de entradas y dejarla activa

En la nueva ventana selecciona la regla vigente > “Edit inbound rules”



Edita la configuración con los datos de la siguiente figura:



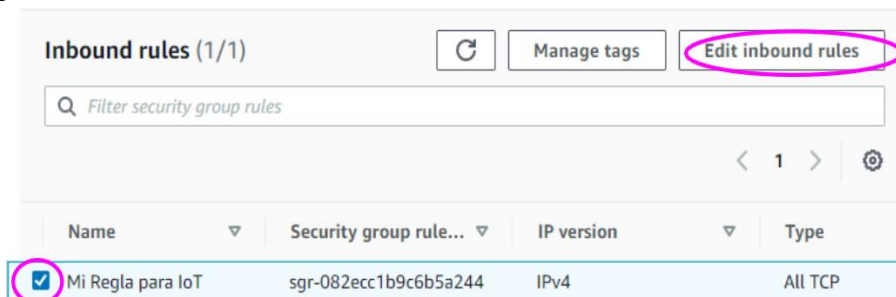
Observa la advertencia que aparece y dice:

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

Es algo que debemos recordar para tomar futuras acciones en pro de la seguridad. Por ahora está bien porque nos interesa que las comunicaciones fluyan desde cualquier IP de manera que una tarjeta ESP-32 no tenga problemas para enviarle datos a esta VM.

Oprime “Save rules”


Deja la regla activa así cuando desees acciones adicionales

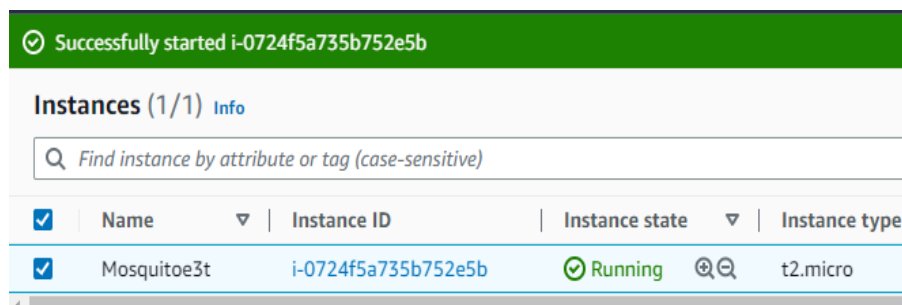


Puedes pasar a tu Instancia así: menú lateral izquierdo > Instances > Instances

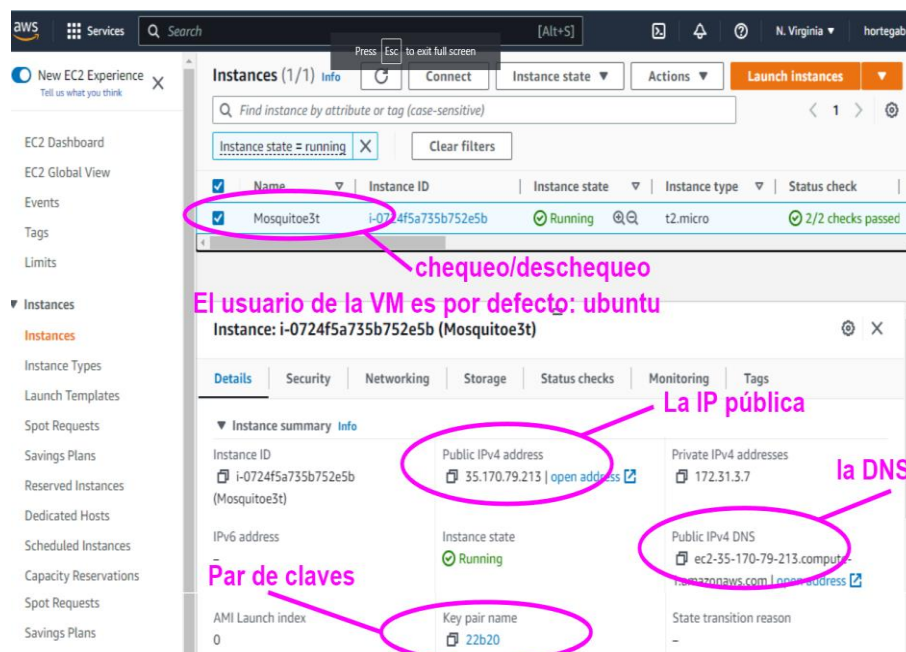
Tarea 4. El día a día para encender la máquina y contar con una terminal de trabajo

Paso 1. Encendemos la VM así:

Te logeas con tu cuenta AWS y quedas en la consola de gestión > entras a “EC2” > Menu lateral > Instances, Instances > chequeas la instancia de interés > Menu horizontal > Instance state > start instance > constatamos que la VM quede corriendo (puede tomar hasta un minuto o más. A veces la instancia ya está corriendo, pero la página web de AWS se demora en refrescar la información. Puedes refrescar la página manualmente con ). Deberá quedar corriendo como en la Fig siguiente



Es importante identificar datos de relevancia de la VM, como se muestra en la siguiente figura



Paso 2. Abrimos una terminal

La opción de usar una terminal en línea (la recomendada)

- Es la recomendada porque no requiere instalar nada en un computador ni usar keys. Simplemente, en un navegador aparece el recuadro negro, osea el equivalente a una terminal, para escribir los comandos.

La secuencia es:

- Menu horizontal > Connect > Connect
- Se abrirá una nueva pestaña en el navegador con una terminal de comandos en línea

```
Last login: Sat Feb 25 21:19:34 2023 from 18.206.107.28
ubuntu@ip-172-31-3-7:~$
```

La opción de usar una terminal putty

- Se puede seguir al pie de la letra el video de este enlace, excepto que en el minuto 6:45 dice que el usuario es ec2-user, pero hay que colocar allí el usuario de nuestra máquina, que últimamente se llama por defecto “Ubuntu”.

Tarea 5. Instalación de Node.js

Node.js representa una revolución en el desarrollo de aplicaciones web y servicios del lado del servidor, destacándose por su enfoque asincrónico basado en javascript. Su eficiencia en la gestión de múltiples solicitudes concurrentes lo posiciona como la elección ideal para aplicaciones en tiempo real y servicios que requieren actualizaciones continuas, como chats y juegos en línea.

Para aquellos familiarizados con Google Apps Script, Node.js comparte similitudes valiosas. Ambas plataformas utilizan javascript, permiten el desarrollo de servicios web, apis, y webhooks, y facilitan el cómputo en la nube, incluyendo el desarrollo tanto del backend como del frontend. Sin embargo, la distinción principal radica en que Node.js se instala en un servidor propio, otorgando un control más profundo sobre la infraestructura.

Cuando exploramos alternativas, Python, con su capacidad para crear servicios restful, se presenta como un competidor significativo. Mientras Node.js se destaca por su eficiencia y capacidad para manejar eventos en tiempo real, Python brilla en el desarrollo restful, ofreciendo un marco robusto y ampliamente utilizado llamado Flask. Este marco proporciona herramientas poderosas para la construcción de apis restful y servicios web eficientes.

En la elección entre Node.js y Python, se trata de sopesar las fortalezas específicas de cada tecnología en función de los requisitos particulares del proyecto. Ambas ofrecen soluciones sólidas para el desarrollo web y del lado del servidor, permitiendo a los desarrolladores elegir la herramienta que mejor se adapte a sus necesidades y preferencias.

Las siguientes son las instrucciones tomadas del archivo Readme que aparece en el [repositorio de github](#) en el año 2023, semestre 2 (2023-2). Este proceso también instala npm que es un gestor de paquetes para node. Es decir para realizar instalaciones adicionales a Node se usará npm. Hacemos esta aclaración porque las instrucciones pueden llegar a cambiar con el tiempo y muchas veces es importante ir a la fuente, sobre todo porque las versiones van cambiando, En el 2023-2, todo indica que la versión más madura y más apropiada para nuestra versión de Ubuntu es la 20.09, aunque ya existe la versión 21x.

Paso 1. Bajar el código fuente y la clave de seguridad
Entrega todos los siguientes comandos de un solo golpe
<pre>Sudo apt-get update && sudo apt-get install -y ca-certificates curl gnupg && Sudo mkdir -p /etc/apt/keyrings && curl -fsSL https://deb.nodesource.com/gpgkey/nodesource- repo.gpg.key sudo gpg --dearmor -o /etc/apt/keyrings/nodesource.gpg</pre>
Paso 2. Crear el repositorio deb

Nota: con “NODE_MAJOR=20” estarás indicando que deseas instalar la versión 20.x. Otras opciones son

NODE_MAJOR=16

NODE_MAJOR=18

NODE_MAJOR=20

NODE_MAJOR=21

```
NODE_MAJOR=20 && echo "deb [signed-by=/etc/apt/keyrings/nodesource.gpg]
https://deb.nodesource.com/node_$NODE_MAJOR.x nodistro main" | sudo tee
/etc/apt/sources.list.d/nodesource.list
```

Paso 3. Correr las actualizaciones e instalar Node.js

```
Sudo apt-get update && sudo apt-get install nodejs -y
```

Tarea 6. Instalación de Node Red

Node Red es una herramienta que se puede instalar en cualquier computadora, pero el verdadero sentido está en instalarlo en un servidor como es el caso de la nuestra VM en AWS. De manera similar a cuando Moodle se instala en un servidor, es posible invocar Node Red desde cualquier computadora como una página web. Desde otro ángulo, Node-RED también es comparable, en la forma en que se crean soluciones, con GNU radio en el sentido que ambos usan la programación visual, mediante flujogramas. Pero en Node-RED es más común el término nodo para referirse no sólo a los elementos que se usan sino también a un flujograma, que surge como producto de unir varios nodos para una solución. En este sentido hay una similitud con la programación orientada a objetos, donde varios objetos se pueden unir para componer un objeto mayor. Pero la concepción detrás de Node-RED va más allá de todo lo existente porque está pensada más bien para soluciones en la nube a nivel de servidor.

Más allá de todo eso, con Node-RED es posible crear, en línea, interfaces para visualización gráfica en línea. En este sentido es una competencia para otras herramientas que se usan en la UIS llamada Ignition con MQTT Engine, pero que tiene un alto costo para su uso.

[Node-RED](#). Podemos decir que Node-RED busca llegar a ser el equivalente gráfico de Node.js, lo cual es mucho decir si recordamos que Node.js está entre las herramientas más competitivas para los desarrollos en la nube.

Recomendaciones iniciales:

- Visita la página de Node-RED para conocer novedades: <https://nodered.org/>

- Vemos que Node-RED tiene muchas opciones de uso, desde la instalación local, en PC, en una raspberry, en android y en la nube. Nos interesa esta última y que sea en AWS, de manera que, en la página <https://nodered.org/> nos interesa explorar más bien el enlace: [Amazon Web Services](#)
- El enlace anterior es el recomendado para conocer las últimas novedades en la instalación. Los comandos de la siguiente tabla es una experiencia de instalación vivida en febrero 2023, siguiendo este [video](#)

Sudo npm install -g --unsafe-perm node-red	Usando nmp hacemos la instalación de Node-RED
Sudo systemctl daemon-reload	Recarga la configuración del demonio del sistema para asegurarse de que los cambios realizados en la configuración o unidad systemd, hechos por la instalación de Node-RED, se apliquen correctamente.
Sudo apt-get upgrade	
Node-red -v	Lanzamiento de Node Red. Es como cuando uno abre el editor de python o gnuRadio. Entonces la consola se queda en espera. No se debe cerrar la consola porque se cerrará node red.

Tarea 7. Pruebas de reconocimiento a Node Red

Pruebas en la consola de node-RED

Abre node-RED con lo cual entras a la consola. Para ello usa el comando:
Node-red -v
Como resultado verás un prompt que te indica que ya no estás en Ubuntu sino en Node-RED
<pre>28 Nov 03:43:47 - [warn] Encrypted credentials not found 28 Nov 03:43:47 - [info] Server now running at http://127.0.0.1:1880/ 28 Nov 03:43:47 - [info] Starting flows 28 Nov 03:43:47 - [info] Started flows</pre>
Ve a AWS y revisa cual es la IP pública de tu instancia

Instances (1/12) Info

Find Instance by attribute or tag (case-sensitive)

Name	Instance ID	Instance state	Instance type
grIoT.Homero3	i-0b23cf16ab4eadae5	Running	t2.micro
e3tiot	i-0015528e7baf5b704	Stopped	t2.micro
Mosquitoe3t	i-0724f5a735b752e5b	Stopped	t2.micro
grIoT.CapstoneD	i-0b0b0074d525ed7b5	Stopped	t2.micro

Instance: i-0b23cf16ab4eadae5 (grIoT.Homero3)

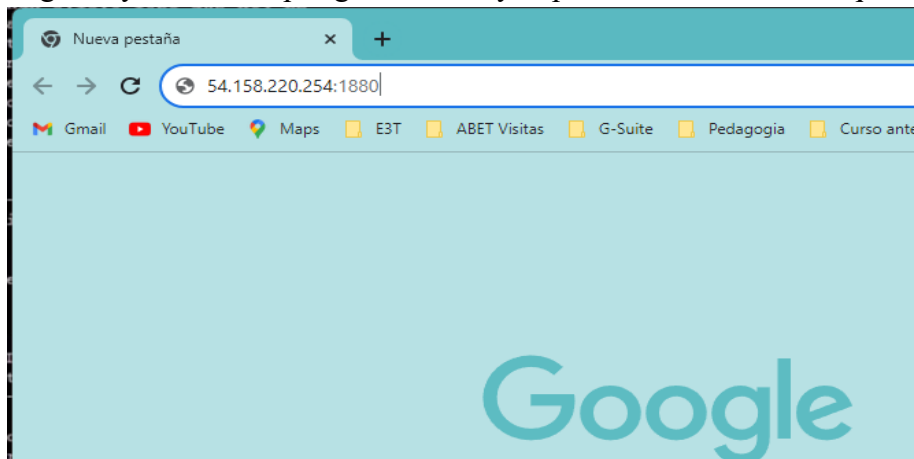
Details Status and alarms New Monitoring Security Networking Storage Tags

▼ Instance summary Info

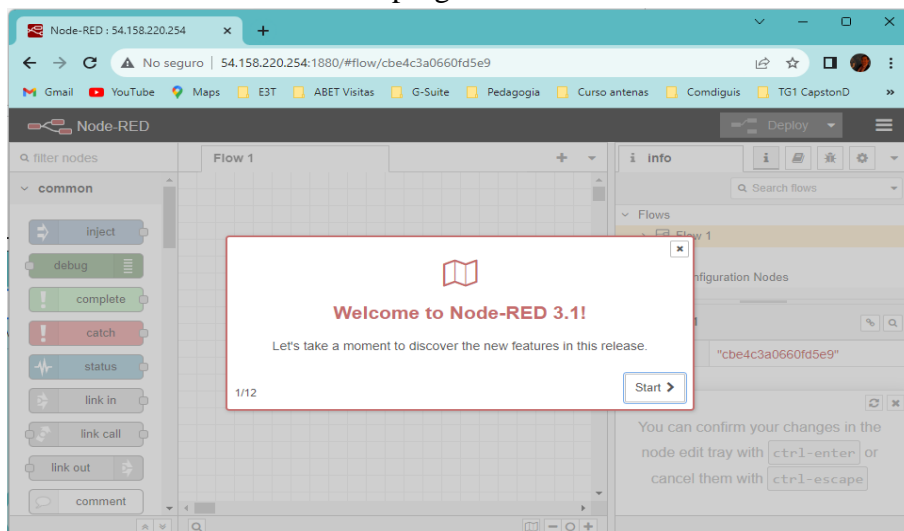
Instance ID
i-0b23cf16ab4eadae5 (grIoT.Homero3)

Public IPv4 address
54.158.220.254 [Open address](#)

Ve a un navegador y escribe la ip seguida de “:” y el puerto de este servicio que es 1880



Como resultado estarás en la interfaz de programación visual de Node-RED



¿Comprueba que pasa si en tu instancia sales de la consola Node-RED con Ctrl+C. Puedes seguir viendo tu interfaz gráfica en el navegador como se hizo en el punto anterior? Debe refrescar el navegador para conocer qué pasa. Verás que ya no podrás trabajar en la interfaz gráfica, porque se ha cerrado Node-RED

En Ubuntu, comprueba si hay una instancia abierta de Node-RED, con el siguiente comando:

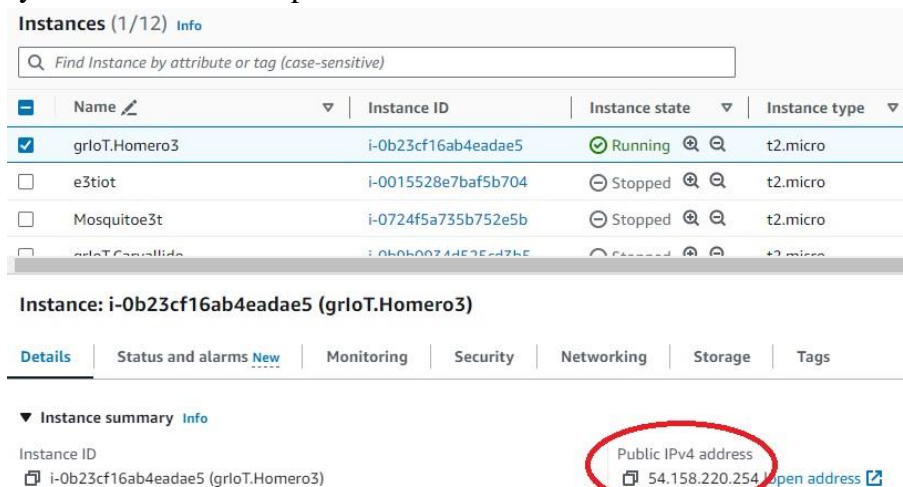
Ps aux | grep node-red

La siguiente respuesta indica que hay una instancia de grep, pero no de node-RED:

```
ubuntu@ip-172-31-43-14:~$ ps aux | grep node-red
ubuntu    1201  0.0  0.2   7004  2304 pts/0    S+   04:01   0:00 grep --color=auto node-red
ubuntu@ip-172-31-43-14:~$
```

Pruebas de funcionamiento de la interfaz gráfica

Ve a AWS y revisa cual es la IP pública de tu instancia



The screenshot shows the AWS Management Console 'Instances' page. A table lists several instances, with 'grIoT.Homero3' selected. Below the table, the details for instance 'i-0b23cf16ab4eadae5' are shown. The 'Public IPv4 address' is highlighted with a red circle, showing the IP '54.158.220.254'.

Name	Instance ID	Instance state	Instance type
grIoT.Homero3	i-0b23cf16ab4eadae5	Running	t2.micro
e3tiot	i-0015528e7baf5b704	Stopped	t2.micro
Mosquito3t	i-0724f5a735b752e5b	Stopped	t2.micro
edIoT.Casallida	i-0b0b0074d525ed7b5	Stopped	t2.micro

Instance: i-0b23cf16ab4eadae5 (grIoT.Homero3)

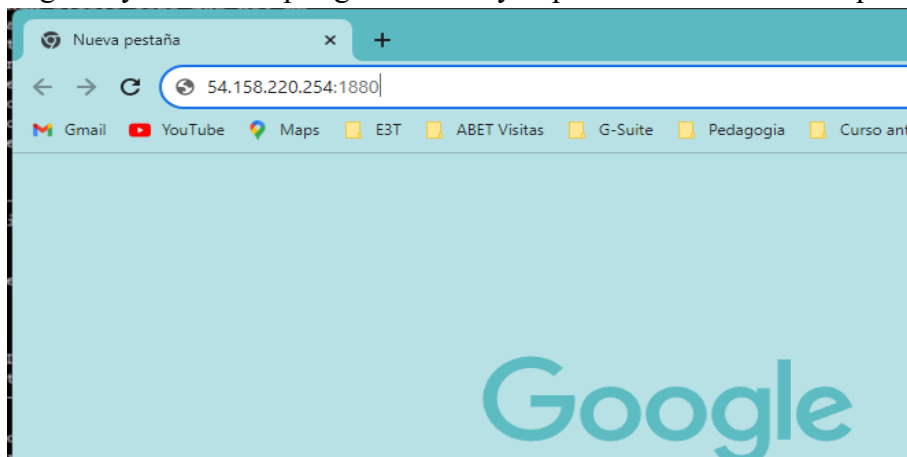
Details | Status and alarms New | Monitoring | Security | Networking | Storage | Tags

▼ Instance summary Info

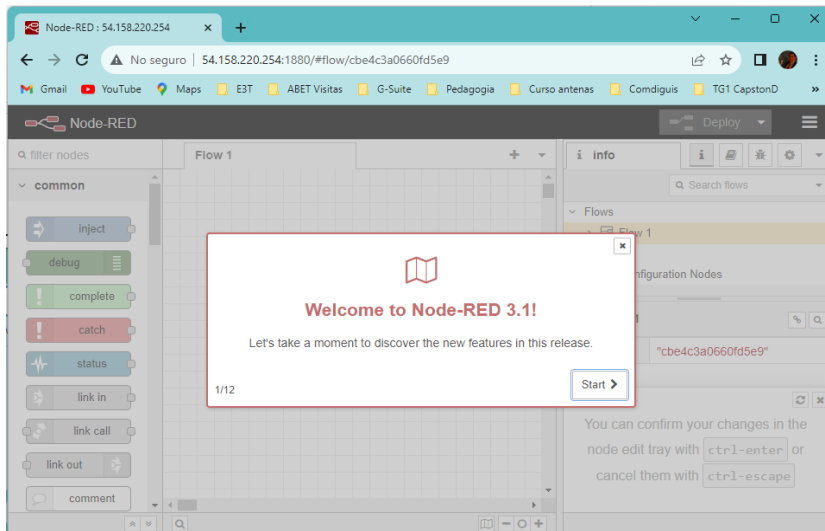
Instance ID
i-0b23cf16ab4eadae5 (grIoT.Homero3)

Public IPv4 address
54.158.220.254 [Open address](#)

Ve a un navegador y escribe la ip seguida de “:” y el puerto de este servicio que es 1880



Como resultado estarás en la interfaz de programación visual de Node-RED



¿Comprueba que pasa si en tu instancia sales de la consola Node-RED con Ctrl+C. Puedes seguir viendo tu interfaz gráfica en el navegador como se hizo en el punto anterior? Debe refrescar el navegador para conocer qué pasa. Verás que ya no podrás trabajar en la interfaz gráfica, porque se ha cerrado Node-RED

Averiguar si Node-RED está corriendo. Es decir, si tu máquina tiene una instancia de Node-RED

En Ubuntu, comprueba si hay una instancia abierta de Node-RED, con el siguiente comando:

Ps aux | grep node-red

La siguiente respuesta indica que hay una instancia de grep, pero no de node-RED:

```
ubuntu@ip-172-31-43-14:~$ ps aux | grep node-red
ubuntu      1201  0.0  0.2   7004  2304 pts/0    S+   04:01   0:00 grep --color=auto node-red
ubuntu@ip-172-31-43-14:~$
```

Conclusión: si no estás en la consola, has salido de ella con Ctrl+C. No tendrás una instancia de Node-RED. La interfaz gráfica remota no funcionará. Pero es posible hacer una reconfiguración para que esto no ocurra como se explica en el siguiente capítulo

Averiguar qué pasa si tienes Node-RED corriendo, reinicias tu máquina, ¿con ello Node-RED también se reinicia?

Inicia node-RED con:

Node-red -v

Reinicia tu máquina desde la consola de AWS
Con el siguiente comando comprueba si hay una instancia de Node-RED corriendo:
Ps aux grep node-red
La respuesta es que no, veremos esto, pero eso solo indica que hay una instancia de grep
<pre>ubuntu@ip-172-31-43-14:~\$ ps aux grep node-red ubuntu 1431 0.0 0.2 7004 2304 pts/0 S+ 04:17 0:00 grep --color=auto node-red ubuntu@ip-172-31-43-14:~\$</pre>

Tarea 8. Configuración de Node Red.

Paso 1. Configuración de Node-Red para que arranque automáticamente

En el paso anterior vimos que, con la instalación hecha, Node-RED no se reinicia automáticamente. Supongamos que tienes un servicio al público y por alguna razón se fue se apagó la máquina, entonces en adelante, el servicio no estará disponible, aunque la máquina se reinicie. De allí la importancia de una configuración para lograr que Node-RED arranque automáticamente. La siguiente es la configuración que recomienda chatgpt para lograr que Node-RED arranque automáticamente cuando se enciende la VM:

Comando	Explicación
Sudo nano /etc/systemd/system/node-red.service	Se crea un archivo para el control del servicio para Node-RED por parte de systemd. El archivo resulta abierto con el editor nano
<pre>[Unit] Description=Node-RED After=network.target [Service] Execstart=/usr/bin/node-red Restart=on-failure User=ubuntu [Install] Wantedby=multi-user.target</pre>	Es el texto que se debe pegar en el archivo creado. En nuestro caso el usuario se llama ubuntu y será con ese usuario que arrancará el servidor. Esto representa una orden a systemd para iniciar Node-RED como un servicio después de que la red esté disponible. Si Node-RED falla, systemd intentará reiniciarlo automáticamente. El servicio se ejecutará con el usuario ubuntu, y se instalará para arrancar automáticamente en el nivel de ejecución multiusuario. Parece ser un prerequisite para hacer que Node-

	RED arranque automáticamente cuando arranca el sistema. Hay que guardar y salir con Ctrl+O, Ctrl+X
Sudo systemctl daemon-reload	Recargar la configuración de systemd
Sudo systemctl enable node-red.service	Habilita el servicio de Node-RED para que se inicie automáticamente al arrancar el sistema

Pruebas de arranque automático

Reinicia la máquina
Envía el comando para revisar si hay instancias corriendo
Ps aux grep node-red
Resultado
<pre>ubuntu@ip-172-31-43-14:~\$ ps aux grep node-red ubuntu 365 8.7 9.2 11740356 89672 ? Ssl 04:29 0:01 node-red ubuntu 1153 0.0 0.2 7004 2304 pts/0 S+ 04:30 0:00 grep --color=auto node-red ubuntu@ip-172-31-43-14:~\$</pre>
Conclusión: Node-RED arranca automáticamente, el comando anterior muestra una instancia de node-red corriendo, la cual tiene el PID 365

Paso 2. Comprueba que, sin tocar la máquina, el navegador de internet que permite el acceso remoto a Node-RED. En el navegador escribe la IP pública seguida de “: 1880”, por ejemplo:

[Http://54.236.56.149:1880](http://54.236.56.149:1880)

Una falla común es que: uno toma la IP que aparece en la instancia, pero resulta que puede ser la IP de una corrida anterior, debido a que al navegador le toma tiempo refrescar los datos.

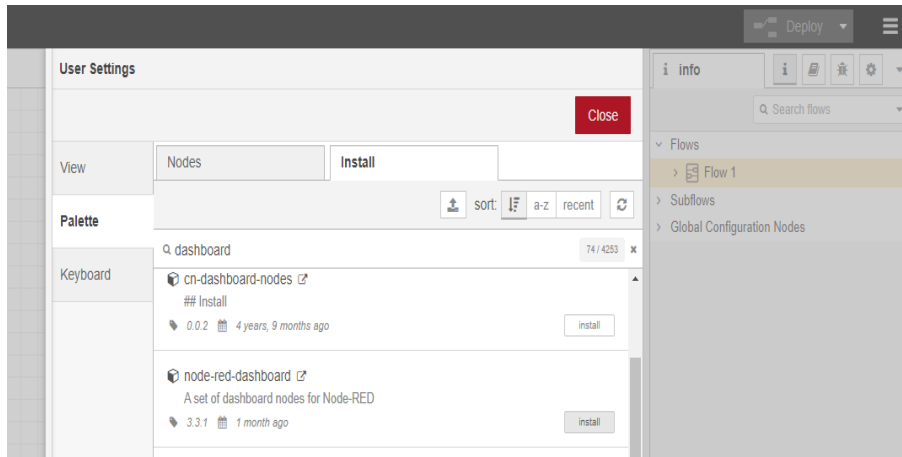
Para descartar esta falla, usa el botón de refrescar 

Paso 3. Instalar complementos de Node-RED



> Manage Palette > Install > nombre de lo que desea instalar > Install> Install

- Node-red-dashboard



- De manera similar instalar: node-red-contrib-ui-led
- Instalar tambien: node-red-mysql

Tarea 9. Mantenimiento de Node RED.

Esta tarea se dedica a conquistar habilidades para hacerle mantenimiento a estos servidores en aspectos como: saber si se está presentando alguna falla, si hay que hacer configuraciones, etc. Básicamente, dominar el uso de comandos útiles

Node Red	
Comando	Explicación
node-red -log	Permite conocer el estado del servidor Node Red. Por ejemplo, si está en ejecución
sudo systemctl start node-red.service	Iniciar el servicio de Node-RED
sudo systemctl stop node-red.service	Detener el servicio de Node-RED
sudo systemctl restart node-red.service	Reiniciar el servicio de Node-RED

REFERENCIAS

N. U. Khuzairi, S. Saon, A. K. Mahamad, M. S. M. Zainordin, S. Yamaguchi and M. A. Bin Ahmadon. (s/f). Weather Station Monitoring System with Node-RED,.

Nodered.org. (s/f). <https://nodered.org/docs/user-guide/>.