

Maria del Pilar Morales Avila

Ivan Castillo

## Laboratorio: [Reto] Ejercicio Python

### Tu desafío

- Escriba un script de Python para:
- Muestra todos los números primos entre 1 y 250 .
- Guarde los resultados en un archivo results.txt .
- Pruebe el guión. Verifique que produjo los resultados esperados en el archivo results.txt .
- Guarde el script y anote su ubicación (ruta absoluta) para referencia futura.

### Tarea 1: utilizar SSH para conectarse a una instancia EC2 de Amazon Linux

En esta tarea, se conectará a una instancia EC2 de Amazon Linux. Utilizará una utilidad SSH para realizar todas estas operaciones. Las siguientes instrucciones varían ligeramente dependiendo de si está utilizando Windows o Mac/Linux.

#### USUARIOS DE WINDOWS: USO DE SSH PARA CONECTARSE

Estas instrucciones son específicamente para usuarios de Windows. Si está utilizando macOS o Linux, pase a la siguiente sección.

3. En el panel **Información del laboratorio** , seleccione el enlace **PPK** y guarde el archivo. El nombre del archivo será similar a *Ec2KeyPair-PPK.ppk* . Normalmente su navegador lo guardará en el directorio de Descargas.
4. Tome nota de la dirección **PublicIP** .
5. Descargue **PuTTY** a SSH en la instancia de Amazon EC2.
6. Abrir **PuTTY .exe**

7. Configure el tiempo de espera de PuTTY para mantener abierta la sesión de PuTTY durante un período de tiempo más largo:


- Seleccionar **conexión**
- Establecer **segundos entre keepalives** en 30

The image shows a screenshot of a lab instruction page with a sidebar on the left and a main content area. The sidebar contains links like 'AWS service restrictions', 'Scenario', 'Start lab', 'Task 1: Use SSH to connect to an Amazon Linux EC2 Instance', 'Task 2: Exercise - Explore the Linux man pages', 'End lab', and 'Additional Resources'. The main content area has a heading 'Machine Image (AMI) (EN)' and a 'Start Lab' button. Below this, it says 'Task, you will connect to a Amazon Linux EC2 instance. You will use an SSH utility to perform all of these operations. The following instructions vary slightly depending on whether you are using Windows or Mac/Linux.' It then has a section 'WINDOWS USERS: USING SSH TO CONNECT' with instructions for connecting via SSH. A 'Copied' tooltip is visible over the '30' value in the 'Seconds between keepalives' field.

8. Configure su sesión PuTTY:

- Seleccionar **sesión**
- **Nombre de host (o dirección IP):** pegue el **DNS público** o la **dirección IPv4** de la instancia que anotó anteriormente. Alternativamente, regrese a la Consola EC2 y seleccione **Instancias**. Marque la casilla junto a la instancia a la que desea conectarse y en la pestaña **Descripción** copie el valor de **IP pública IPv4**.

10. Cuando se le solicite **iniciar sesión como** , ingrese: **ec2-user** Esto lo conectará a la instancia EC2.



training and certification

## Introduction to a

### Información del laboratorio

1 hora

Idiomas disponibles

Valoración

### Recursos

PEM de par de claves de EC2

**Descargar PEM**

PPK de par de claves de EC2

**Descargar PPK**

LabRegion

35.89.75.219 - PuTTY

```
Unable to use certificate file "C:\Users\miguel iligaray\Downloads\Ec2KeyPair-PPK.ppk" (PuTTY SSH-2 private key)
login as: ec2-user
```

- Select **Open** again.

9. Select **Yes**, to trust and connect.

10. When prompted **login as**, enter: **ec2-user** This will connect you to the EC2 instance.

11. Windows Users, skip ahead to the next task.

Panel de EC2

Vista global de EC2

Eventos

▼ Instancias

Instancias

Tipos de instancia

Plantillas de lanzamiento

Solicitudes de spot

Savings Plans

Instancias reservadas

Alojamientos dedicados

Reservas de capacidad

▼ Imágenes

AMI

Catálogo de AMI

▼ Elastic Block Store

Volúmenes

Instantáneas

Administrador del ciclo de vida

▼ Red y seguridad

Security Groups

Direcciones IP elásticas

Grupos de ubicación

Dirección IPv6

Estado de la instancia: **En ejecución**

Nombre DNS de IP privada (solo IPv4): **ip-10-0-10-77.us-west-2.compute.internal**

Dirección IP asignada automáticamente: **35.89.75.219 [IP pública]**

Rol de IAM: **Optional**

IMDSv2: **Optional**

EC2 recommends setting IMDSv2 to required | Más información

Detalles | Status and alarms New | Monitoreo | Seguridad

▼ Detalles de redes Información

Dirección IPv4 pública: **35.89.75.219 [dirección abierta]**

DNS de IPv4 pública: **ec2-35-89-75-219.us-west-2.compute.amazonaws.com [dirección abierta]**

ID de subred: **subnet-0427ca3d3fdbc2ac4 (Public Subnet 1)**

Zona de disponibilidad: **us-west-2a**

Utilizar RBN como nombre de host del SO invitado: **desactivado**

▼ Interfaces de red (1) Información

Direcciones IP elásticas

Hallazgo de AWS Compute Optimizer

Suscribirse a AWS Compute Optimizer para recibir recomendaciones. | Más información

Nombre del grupo de Auto Scaling

ID de VPC: **vpc-0512e421c51e7da2e (Lab VPC)**

Direcciones IPv4 privadas secundarias

ID de Outpost

## Escriba un script de Python

### 1. Crear el Script:

- Utiliza un editor de texto para crear el script: **nano script\_numeros\_primos.py**
- Contenido del script en el editor de texto.
- Guarda y cierra el editor. En nano, puedes hacerlo presionando Ctrl + X, luego Y para confirmar que deseas guardar, y finalmente Enter.

```
ec2-user@ip-10-1-11-229:~  
login as: ec2-user  
Authenticating with public key "imported-openssh-key"  
#  
##### Amazon Linux 2  
#####  
##### AL2 End of Life is 2025-06-30.  
#####  
#####  
##### A newer version of Amazon Linux is available!  
#####  
##### Amazon Linux 2023, GA and supported until 2028-03-15.  
#####  
##### https://aws.amazon.com/linux/amazon-linux-2023/  
#####  
[ec2-user@ip-10-1-11-229 ~]$ nano script_numeros_primos.py  
GNU nano 2.9.8 script_numeros_primos.py Modified  
  
    return primos  
  
def guardar_en_archivo(primos, nombre_archivo):  
    with open(nombre_archivo, 'w') as archivo:  
        for primo in primos:  
            archivo.write(str(primo) + '\n')  
  
if __name__ == "__main__":  
    inicio_rango = 1  
    fin_rango = 250  
    archivo_salida = "resultados.txt"  
  
    numeros_primos = encontrar_primos(inicio_rango, fin_rango)  
    guardar_en_archivo(numeros_primos, archivo_salida)  
  
    print(f"Números primos entre {inicio_rango} y {fin_rango} guardados en {archivo_salida}")
```

```
ec2-user@ip-10-1-11-229:~  
GNU nano 2.9.8 script_numeros_primos.py  
  
def es_primo(num):  
    if num < 2:  
        return False  
    for i in range(2, int(num**0.5) + 1):  
        if num % i == 0:  
            return False  
    return True  
  
def encontrar_primos(inicio, fin):  
    primos = [num for num in range(inicio, fin + 1) if es_primo(num)]  
    return primos  
  
def guardar_en_archivo(primos, nombre_archivo):  
    with open(nombre_archivo, 'w') as archivo:  
        for primo in primos:  
            archivo.write(str(primo) + '\n')  
  
if __name__ == "__main__":  
    inicio_rango = 1  
    fin_rango = 250  
    archivo_salida = "resultados.txt"  
  
    numeros_primos = encontrar_primos(inicio_rango, fin_rango)  
    guardar_en_archivo(numeros_primos, archivo_salida)  
  
    print(f"Números primos entre {inicio_rango} y {fin_rango} guardados en {archivo_salida}")
```

```
ec2-user@ip-10-1-11-229:~  
login as: ec2-user  
Authenticating with public key "imported-openssh-key"  
#_#  
##### Amazon Linux 2  
#####  
##### AL2 End of Life is 2025-06-30.  
#####  
#####  
##### A newer version of Amazon Linux is available!  
#####  
##### Amazon Linux 2023, GA and supported until 2028-03-15.  
#####  
##### https://aws.amazon.com/linux/amazon-linux-2023/  
#####  
[ec2-user@ip-10-1-11-229 ~]$ nano script_numeros_primos.py  
[ec2-user@ip-10-1-11-229 ~]$ python3 script_numeros_primos.py  
Números primos entre 1 y 250 guardados en resultados.txt  
[ec2-user@ip-10-1-11-229 ~]$ cat resultados.txt  
2  
3  
5  
7  
11  
13  
17  
19  
23  
29  
31  
37  
41  
43  
47  
53  
59  
61  
67  
71  
73  
79  
83  
89  
97  
101  
103  
107  
109  
113  
127
```

1. Ejecutar el Script:

- Ejecuta el script con el siguiente comando: `python3 script_numeros_primos.py`
- Si Python 3 no está instalado en tu máquina, puedes usar python en lugar de python3.

2. Verificar el Resultado:

- Verifica que se haya creado el archivo de resultados (resultados.txt) y que contenga la lista de números primos. `cat resultados.txt`

Esto imprimirá el contenido del archivo en la pantalla.

## script

```
ec2-user@ip-10-1-11-229:~  
GNU nano 2.9.8 script_numeros_primos.py  
  
def es_primo(num):  
    if num < 2:  
        return False  
    for i in range(2, int(num**0.5) + 1):  
        if num % i == 0:  
            return False  
    return True  
  
def encontrar_primos(inicio, fin):  
    primos = [num for num in range(inicio, fin + 1) if es_primo(num)]  
    return primos  
  
def guardar_en_archivo(primos, nombre_archivo):  
    with open(nombre_archivo, 'w') as archivo:  
        for primo in primos:  
            archivo.write(str(primo) + '\n')  
  
if __name__ == "__main__":  
    inicio_rango = 1  
    fin_rango = 250  
    archivo_salida = "resultados.txt"  
  
    numeros_primos = encontrar_primos(inicio_rango, fin_rango)  
    guardar_en_archivo(numeros_primos, archivo_salida)  
  
    print(f"Números primos entre {inicio_rango} y {fin_rango} guardados en {archivo_salida}")
```

1. Función `es_primo(num)`: Esta función determina si un número dado `num` es primo o no. Retorna `True` si es primo y `False` en caso contrario. Para verificar si un número es primo, se itera desde 2 hasta la raíz cuadrada de `num` (más 1) y se verifica si `num` es divisible por algún número en ese rango.

```
def es_primo(num):
```

```
    if num < 2:
```

```
        return False
```

```
    for i in range(2, int(num**0.5) + 1):
```

```
        if num % i == 0: return False
```

```
    return True
```

2. Función `encontrar_primos(inicio, fin)`: Esta función encuentra todos los números primos en un rango dado desde `inicio` hasta `fin`. Utiliza la función `es_primo(num)` para verificar si cada número en el rango es primo.

```
def encontrar_primos(inicio, fin):
```

```
    primos = [num for num in range(inicio, fin + 1) if es_primo(num)]
```

```
    return primos
```

3. Función `guardar_en_archivo(primos, nombre_archivo)`: Esta función guarda una lista de números primos en un archivo de texto. Abre el archivo especificado en modo escritura ('w') y escribe cada número primo seguido de un salto de línea.

```
def guardar_en_archivo(primos, nombre_archivo):
```

```
    with open(nombre_archivo, 'w') as archivo:
```

```
        for primo in primos: archivo.write(str(primo) + '\n')
```

4. Bloque `if __name__ == "__main__":`: Este bloque se ejecuta cuando el script se ejecuta directamente, no cuando se importa como un módulo en otro script. Aquí se establecen los parámetros (`inicio_rango`, `fin_rango`, `archivo_salida`), se llaman a las funciones anteriores para encontrar y guardar los números primos, y se imprime un mensaje indicando el rango y el nombre del archivo.

```
if __name__ == "__main__":
```

```
    inicio_rango = 1
```

```
    fin_rango = 250
```

```
    archivo_salida = "resultados.txt"
```

```
    numeros_primos = encontrar_primos(inicio_rango, fin_rango)
    guardar_en_archivo(numeros_primos, archivo_salida)
```

```
    print (f"Números primos entre {inicio_rango} y {fin_rango} guardados en {archivo_salida}")
```

Este script encuentra todos los números primos en un rango específico (1 a 250), los guarda en un archivo de texto llamado `resultados.txt`, y muestra un mensaje indicando el rango y el nombre del archivo donde se guardaron los resultados.