# Laboritorio: [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

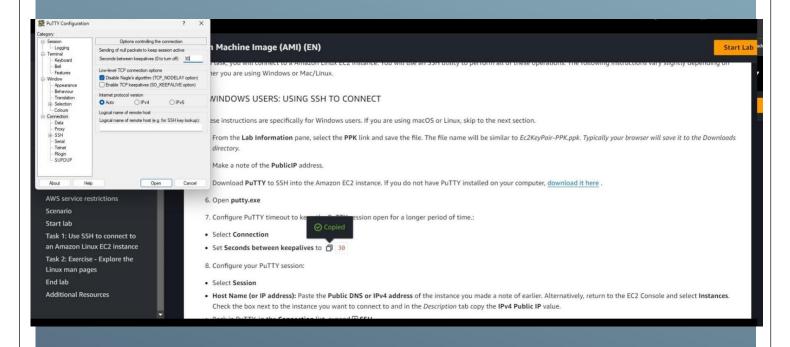
En esta tarea, se conectará a una instancia EC2 de Amazon Linux. Utilizará una utilidad SSH para realizar todas estas operaciones. Las siguientes instrucciones varían ligeramente dependiendo de si está utilizando Windows o Mac/Linux.

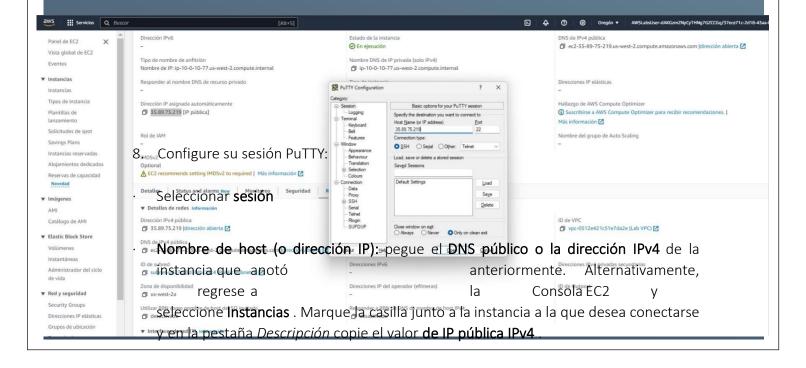
#### USUARIOS DE WINDOWS: USO DE SSH PARA CONECTARSE

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





10. Cuando se le solicite iniciar sesión como , ingrese: ec2-user Esto lo conectará a la instancia EC2. aws training and certification ₫ 35.89.75.219 - PuTTY Información del laboratorio a estas instrucciones. ① 1 hora (i) Idiomas disponibles □ Valoración ed Recursos PEM de par de claves de EC2 · Select Open again. **⊘** Copied Descargar PEM **₩** 9. Select Yes, to trust and con PPK de par de claves de EC2 10. When prompted login as, enter: dec2-user This will connect you to the EC2 instance. Descargar PPK **₩** LabRegion 11. Windows Users, skip ahead to the next task. aws Servicios Q Busca D ⊕ Oregón ▼ AWSLabsUser-dAKGzmZNyCyTHNg7GZCCGq/37ece71c-2d18-43 Dirección IPv6 Estado de la instancia Panel de EC2 🗗 ec2-35-89-75-219.us-west-2.compute.amazonaws.com |dirección abierta 🗹 Vista global de EC2 Nombre DNS de IP privada (solo IPv4)

| ip-10-0-10-77.us-west-2.compute.internal Nombre de IP: ip-10-0-10-77.us-west-2.compute.internal **▼** Instancias Responder al nombre DNS de recurso privado Instancias Tipos de instancia allazgo de AWS Compute Optimizer Basic options for your PuTTY session Plantillas de 35.89.75.219 [IP pública] Suscribirse a AWS Compute Optimizer para recibir recomendaciones. | Specify the destination you want to connect to Host Name (or IP address) Port 35.89.75.219 22 Más información 🖸 Rol de IAM Nombre del grupo de Auto Scaling Savings Plans SSH Serial Other: Telnet Instancias reservadas Load, save or delete a stored ses Alojamientos dedicados Saved Sessions ⚠ EC2 recommends setting IMDSv2 to required | Más información 🖸 Reservas de capacidad Default Settings Detalles Status and alarms New Monitoreo Seguridad ▼ Imágenes AMI ID de VPC

10 vpc-0512e421c51e7da2e (Lab VPC) Catálogo de AMI ☐ 35.89.75.219 |dirección abierta 🖸 Close window on exit:

Aways

Never

Only on clean exit dec2-35-89-75-219.us-west-2.compute.amazonaws.com |dirección abiert Instantáneas Administrador del ciclo Direcciones IP del operador (efimeras) ID de Outpost ▼ Red y seguridad r us-west-2a Security Groups Utilizar RBN como nombre de host del SO invitado Responder a RBN de DNS de nombre de host IPv4 Direcciones IP elásticas desactivado desactivado Grupos de ubicación ▼ Interfaces de red (1) Información

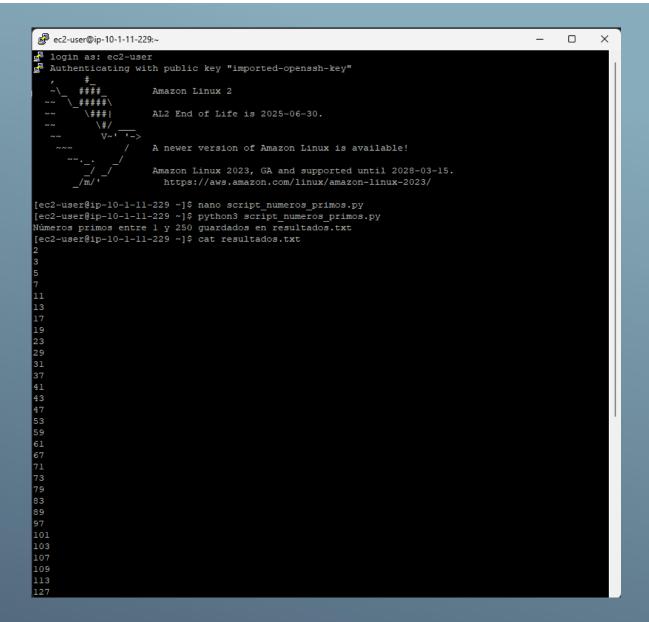
### Escriba un script de Python

#### 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
        #####\
         \###1
                     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
                                                                                               Modified
                                           script numeros primos.py
   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}")
```

```
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 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. 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
ef es primo(num):
    if num < 2:
       return False
    for i in range(2, int(num**0.5) + 1):
    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.

```
if num < 2:
    return False
    for i in range(2, int(num**0.5) + 1):
    if num % i == 0: return False
    return True</pre>
```

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.

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.