



AWS
re:Start
CHALLENGE LAB

Python Scripting Exercise



WEEK 6





Overview

Python scripting streamlines tasks on different platforms, enabling efficient file and directory manipulation, text processing, and system administration. Mastering Python scripts empowers users with automation tools for enhanced productivity, workflow optimization, and scalable solutions across various domains like development, data science, and system management. Python's versatility and extensive library ecosystem make it a valuable tool for creating automation solutions, driving innovation, and improving efficiency.

Your Challenge

Write a Python script based on the following requirements:

- Display all the prime numbers between 1 to 250.
- Store the results in a results.txt file.
- Test the script. Verify that it produced the expected results in the results.txt file.
- Save the script and make a note of its location (absolute path) for future reference.

Note: Both Python 2 and Python 3 are installed on the Linux Host. It is recommended to use Python 3. To run a Python script using version 3, run the command `python3 file.py` by replacing file.py with your file name.



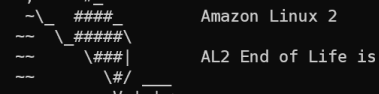
Use SSH to connect to the Linux Host

In the AWS Management Console, select the EC2 instance and make note of the **Public IPv4 address**.

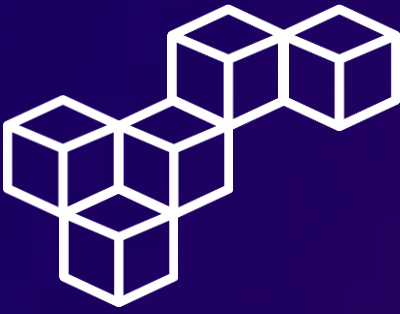
Connect to the instance using SSH

Establish a connection to the EC2 instance using the `ssh` command, the key and the instance's public IPv4 address.

```
support@HP-Pavilion-Laptop:~/Downloads$ ssh -i labsuser.pem ec2-user@54.190.158.131
The authenticity of host '54.190.158.131 (54.190.158.131)' can't be established.
ED25519 key fingerprint is SHA256:PWJqHKfZZpbWBPww6do/VzPIfrJJiL4VvbAGVIWuD0M.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '54.190.158.131' (ED25519) to the list of known hosts.
```



```
#  
###  
#####\n#####|  
#####/  
#/  
V-'  
->  
  
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-10-1-11-72 ~]$
```



Task 2

Your Challenge

Import the os module

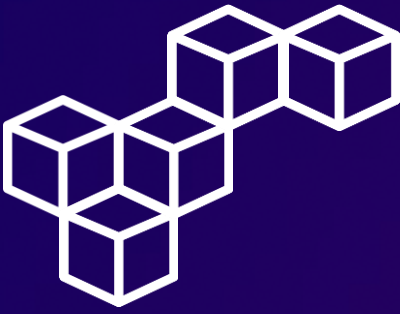
```
import os
```

This line imports the `os` module, which provides a way to interact with the operating system. It allows us to perform various system-related operations such as file manipulation, directory operations, and executing system commands.

Create the results.txt file

```
os.system("touch result.txt")
```

Here, we use the `os.system()` function to execute a system command. The command "touch results.txt" is a Unix/Linux command that creates a new empty file named **results.txt** in the current directory. The `touch` command is commonly used to create files or update file timestamps.



Task 2

Your Challenge

Display the prime numbers

```
for number in range(1,251):  
    prime = True  
    for divider in range(2,number):  
        if number % divider == 0:  
            prime = False  
    if prime == True:  
        os.system("echo " + str(number) + " | tee -a results.txt")
```

- **for number in range(1,251):** Iterate through numbers from 1 to 250.
- **prime = True:** Assume the number is prime initially.
- **for divider in range(2,number):** Iterate through dividers from 2 to number-1.
- **if number % divider == 0:** Check if number is divisible by any divider.
- **prime = False:** If divisible, the number is not prime.
- **if prime == True:** Check if the number remains prime.
- **os.system("echo " + str(number) + " | tee -a results.txt"):** Print and append the prime number to results.txt.





Task 2

Your Challenge

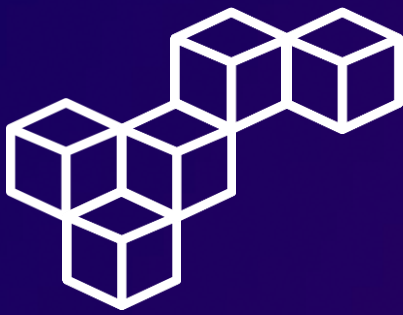
Testing the script

Test the script by executing it using [python3](#). All the prime numbers between 1 to 250 are displayed.

```
[ec2-user@ip-10-1-11-72 ~]$ ls
challenge-lab.py
[ec2-user@ip-10-1-11-72 ~]$ python3 challenge-lab.py
1
2
3
5
7
11
13
17
19
23
29
31
37
41
43
47
```

After execution, view the content of the **results.txt** file, the results were correctly stored in the file.

```
227
229
233
239
241
[ec2-user@ip-10-1-11-72 ~]$ ls
challenge-lab.py  results.txt
[ec2-user@ip-10-1-11-72 ~]$ cat results.txt
1
2
3
5
7
11
13
17
19
23
29
31
37
41
43
47
```



Conclusions

The import statement

The import statement allows Python code to access functionality from external modules, expanding the language's capabilities.

The `os.system()` function

The `os.system()` function enables Python scripts to execute system commands, providing flexibility for system-related operations.

The for loop

The for loop in Python iterates over a sequence of elements, making it efficient for tasks that require repeated actions or processing.

The `range()` function

The `range()` function generates a sequence of numbers, commonly used in conjunction with loops like for to control the iteration flow.

The if statement

The if statement in Python allows conditional execution of code blocks, facilitating decision-making and branching logic within programs.



Cristhian Becerra



[cristhian-becerra-espinoza](https://www.linkedin.com/in/cristhian-becerra-espinoza)



+51 951 634 354



cristhianbecerra99@gmail.com



Lima, Peru

