Week 5: Monitoring services using Python

Preparation: Install httpd on AWS instance

- Create and launch a new AWS micro instance (or restart an existing one). Connect to it using SSH. You can connect via PuTTY or, better, from a local Linux terminal as follows:

```
ssh -i mykeyfile.pem ec2-user@public-ip-address
```

- Install the web server on your instance. The command to do this is (Amazon Linux 2):

```
sudo yum -y install httpd
```

Preparation: Install Python 3 on the AWS instance

```
sudo yum -y install python3
```

Exercise 1 – (N.B. solution at this link for use in Assignment 1)

- Write a Python script `check_webserver.py` to check if the web server is running.

  Outline solution:
  - Import the subprocess module into Python
  - Run the Linux shell command

    ```
    ps -A | grep httpd
    ```
  - Check the status returned by this command execution, or use exception handling to detect if there is an issue.
  - Print an appropriate message

Exercise 2

- Test the script above by running it both before and after starting httpd. Extend the script to start httpd if it is not running. The command to start httpd is:

```
sudo systemctl start httpd
```

Exercise 3

- Write a Python script to perform a few basic system monitoring tasks. For example, report to the user on the number of processes running and the percentage CPU utilization. You could use the shell commands such as 'ps', 'top', 'netstat', 'vmstat', etc.

Exercise 4

- Install and start a MariaSQL database server (similar to MySQL), as follows:

```
sudo yum install -y mariadb-server
sudo systemctl start mariadb
```

  Write a Python script `check_db.py` to check if mariadb is running.

Exercise 5

- Run your *check_webserver* script (ex 1 above) from a **local** Linux machine. You can do this using SSH remote command execution.
  The general syntax for SSH remote command execution on an EC2 instance is:

```
ssh -i mykeyfile.pem ec2-user@public-ip-address 'command'
```

  e.g.

```
ssh -i mykey.pem ec2-user@54.7.8.9 'pwd'
ssh -i mykey.pem ec2-user@54.7.8.9 'python3 check_webserver.py'
```

Exercise 6

- Run a remote SSH command (see exercise 5) from a local Python script (using the *subprocess* module)