

Author : Brijesh Yadav

Gmail: bkumaryadav096@gmail.com

Date: 6 Oct 2021

!/usr/bin/env python

```
In [1]: import os
import subprocess
import re
import shutil
```

Physical and Logical CPU , Memory (RAM) & Network

```
In [2]: def get_statistics():
    statistics = {}
    matcher = re.compile('\d+')

    # CPU Usages

    top_command = subprocess.run(['top', '-l 1', '-n 0'], stdout=subprocess.PIPE).stdout
    physical_and_logical_cpu_count = os.cpu_count()
    statistics['physical_and_logical_cpu_count'] = physical_and_logical_cpu_count
    cpu_load = [x / os.cpu_count() * 100 for x in os.getloadavg()][-1]
    statistics['cpu_load'] = round(cpu_load)

    #Memory usages
    # Used memory = wired_memory + inactive + active
    total_ram = subprocess.run(['sysctl', 'hw.memsize'], stdout=subprocess.PIPE).stdout
    vm = subprocess.Popen(['vm_stat'], stdout=subprocess.PIPE).communicate()[0].decode()
    vmLines = vm.split('\n')

    wired_memory = (int(matcher.search(vmLines[6]).group()) * 4096) / 1024 ** 3
    free_memory = (int(matcher.search(vmLines[1]).group()) * 4096) / 1024 ** 3
    active_memory = (int(matcher.search(vmLines[2]).group()) * 4096) / 1024 ** 3
    inactive_memory = (int(matcher.search(vmLines[3]).group()) * 4096) / 1024 ** 3

    statistics['ram'] = dict({
        'total_ram': int(matcher.search(total_ram).group()) / 1024 ** 3,
        'used_ram': round(wired_memory + active_memory + inactive_memory, 2),
    })

    ## Disk usage - total disk size, used disk space, and free disk

    total, used, free = shutil.disk_usage("/")

    read_written = top_command[9].split(':')[1].split(',')
    read = read_written[0].split(' ')[1]
    written = read_written[1].split(' ')[1]
```

```
statistics['disk'] = dict(
    {
        'total_disk_space': round(total / 1024 ** 3, 1),
        'used_disk_space': round(used / 1024 ** 3, 1),
        'free_disk_space': round(free / 1024 ** 3, 1),
        'read_write': {
            'read': read,
            'written': written
        }
    }
)

# Network Latency
ping_result = subprocess.run(['ping', '-i 5', '-c 5', 'google.com'], stdout=subprocess.
    'utf-8').split('\n')

min, avg, max = ping_result[-2].split('=')[-1].split('/')[ :3]
statistics['network_latency'] = dict(
    {
        'min': min.strip(),
        'avg': avg.strip(),
        'max': max.strip(),
    }
)

return statistics
statistics = get_statistics()
```

Thank You