Python script to verify disk space

```
[user1@vm1 ~]$ df -h
Filesystem                  Size  Used Avail Use% Mounted on
devtmpfs                    307M     0  307M   0% /dev
tmpfs                       342M     0  342M   0% /dev/shm
tmpfs                       342M   35M  307M  11% /run
tmpfs                       342M     0  342M   0% /sys/fs/cgroup
/dev/mapper/ocivolume-root   36G  5.1G   31G  15% /
/dev/mapper/ocivolume-oled   10G  123M  9.9G   2% /var/oled
/dev/sda2                  1014M  298M  717M  30% /boot
/dev/sda1                   100M  5.1M   95M   6% /boot/efi
tmpfs                        69M     0   69M   0% /run/user/0
tmpfs                        69M     0   69M   0% /run/user/987
tmpfs                        69M     0   69M   0% /run/user/1003
tmpfs                        69M     0   69M   0% /run/user/1005
tmpfs                        69M     0   69M   0% /run/user/1001
[user1@vm1 ~]$ cat diskspace.py
import os


def getDfDescription():
    df = os.popen("df -h /")
    i = 0
    while True:
        i = i + 1
        line = df.readline()
        if i==1:
            return(line.split()[0:6])
```

```
 def getDf():
    df = os.popen("df -h /")
    i = 0
    while True:
        i = i + 1
        line = df.readline()
        if i==2:
            return(line.split()[0:6])

 # Disk information
description = getDfDescription()
disk_root = getDf()
 print(disk_root)
print(description[0] + " : " + disk_root[0])
print(description[1] + " : " + disk_root[1])
print(description[2] + " : " + disk_root[2])
print(description[3] + " : " + disk_root[3])
print(description[4] + " : " + disk_root[4])
print(description[5] + " : " + disk_root[5])
```

```
 [user1@vm1 ~]$ python diskspace.py
['/dev/mapper/ocivolume-root', '36G', '5.1G', '31G', '15%', '/']
Filesystem : /dev/mapper/ocivolume-root
Size : 36G
Used : 5.1G
Avail : 31G
Use% : 15%
Mounted : /
```

Python Script to monitor disk space and send an email in case threshold reached(gmail as provider)

Python Script to monitor disk space usage

```
threshold = 90
partition = “/”df = subprocess.Popen([“df”,”-h”], stdout=subprocess.PIPE)
 for line in df.stdout:
 splitline = line.decode().split()
 if splitline[5] == partition:
 if int(splitline[4][:-1]) > threshold:
```

Script :

```
import subprocess

import smtplib

from email.mime.text import MIMEText


threshold = 90

partition = “/”


def report_via_email():

msg = MIMEText(“Server running out of disk space”)

msg[“Subject”] = “Low disk space warning”

msg[“From”] = “admin@example.com”

msg[“To”] = “test@gmail.com”

with smtplib.SMTP(“smtp.gmail.com”, 587) as server:

server.ehlo()

server.starttls()

server.login(“gmail_user”,”gmail_password)

server.sendmail(“admin@example.com”,”test@gmail.com”,msg.as_string())


def check_once():

df = subprocess.Popen([“df”,”-h”], stdout=subprocess.PIPE)

for line in df.stdout:

splitline = line.decode().split()

if splitline[5] == partition:

if int(splitline[4][:-1]) > threshold:

report_via_email()

check_once()
```