Ansible Project: Uptime and Disk Usage

Ansible Project:

# Verify server uptime and disk usage Information using Ansible

# Problem Statement:

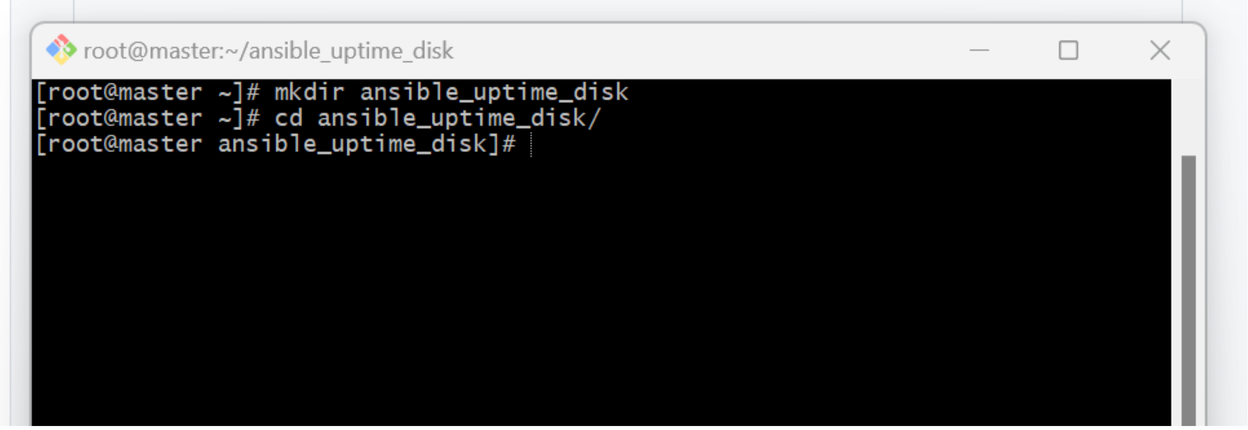
-It's a good practice to make sure servers are up and running all the time. Companies usually have monitoring tools to monitor servers and application uptime, but from time to time; the monitoring tools fail, and they need someone to verify a server or servers' status. It can take a lot of time to verify each server's uptime and disk usage manually, especially, if you are working with a lot of servers. A company submitted a ticket regarding automating the task of checking uptime and disk usage of servers using Ansible, so they can save time whenever the automated monitoring tools fail. Also, they need the logs output to be saved in a file.

# Solution Statement:

-For demonstration purposes, I am going to use three servers to solve the problem. The concept is the same if we are working with 3 servers or 100 servers. One of my servers will be used as the master node and the other two as the target nodes. I am going to create an Ansible playbook that would the uptime and disk usage of the target servers and saves the result in a text file.

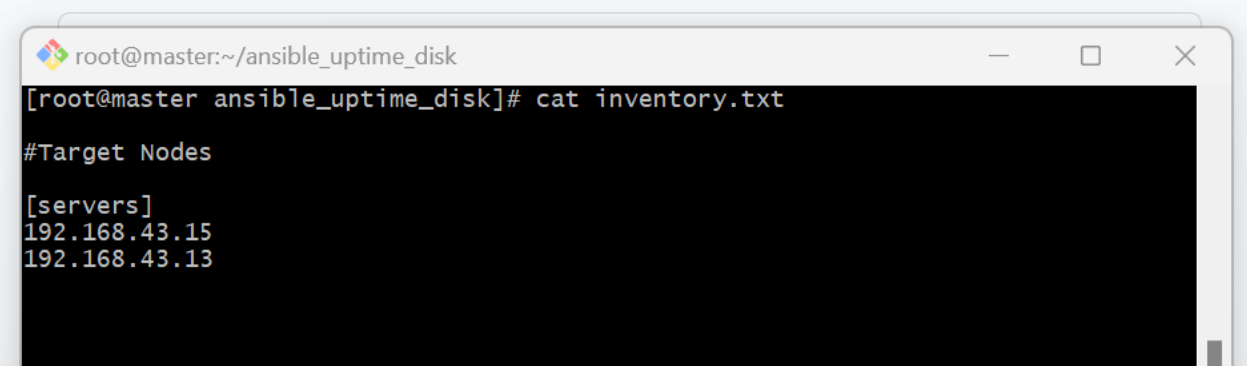
- All my servers are Linux Centos7 and I used a custom Vagrantfile to launch all of them at once.

# Creating a directory for Ansible files including the playbook



```
[root@master ~]# mkdir ansible_uptime_disk
[root@master ~]# cd ansible_uptime_disk/
[root@master ansible_uptime_disk]#
```

# Creating an inventory file to keep track of all the hosts.

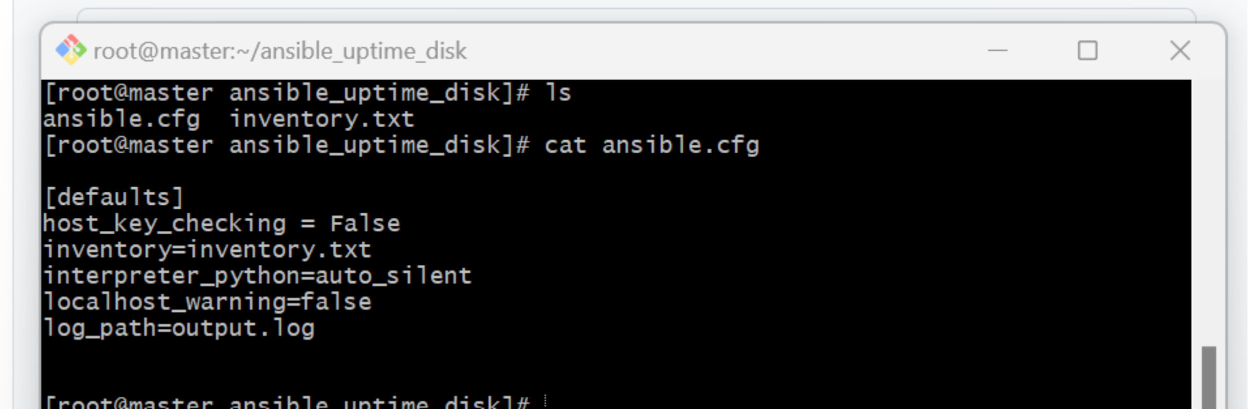


```
[root@master ansible_uptime_disk]# cat inventory.txt

#Target Nodes

[servers]
192.168.43.15
192.168.43.13
```

# Creating a configuration for ansible



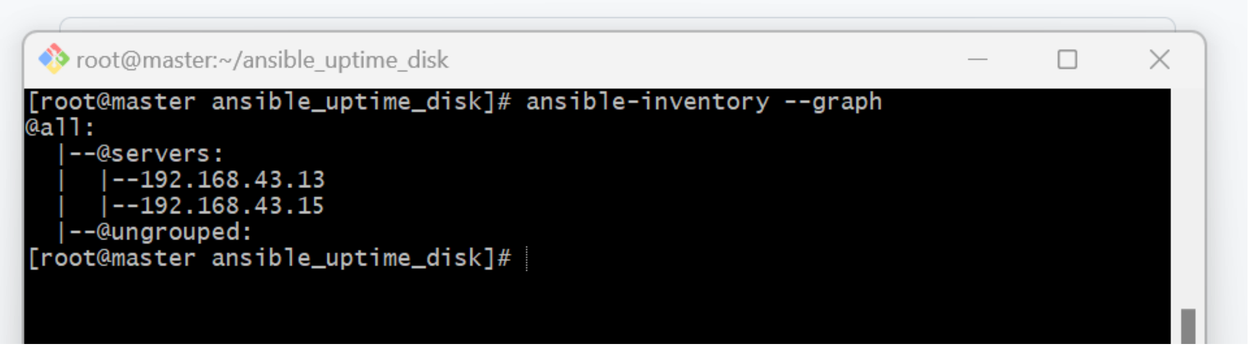
```
[root@master ansible_uptime_disk]# ls
ansible.cfg  inventory.txt
[root@master ansible_uptime_disk]# cat ansible.cfg

[defaults]
host_key_checking = False
inventory=inventory.txt
interpreter_python=auto_silent
localhost_warning=false
log_path=output.log


[root@master ansible_uptime_disk]#
```

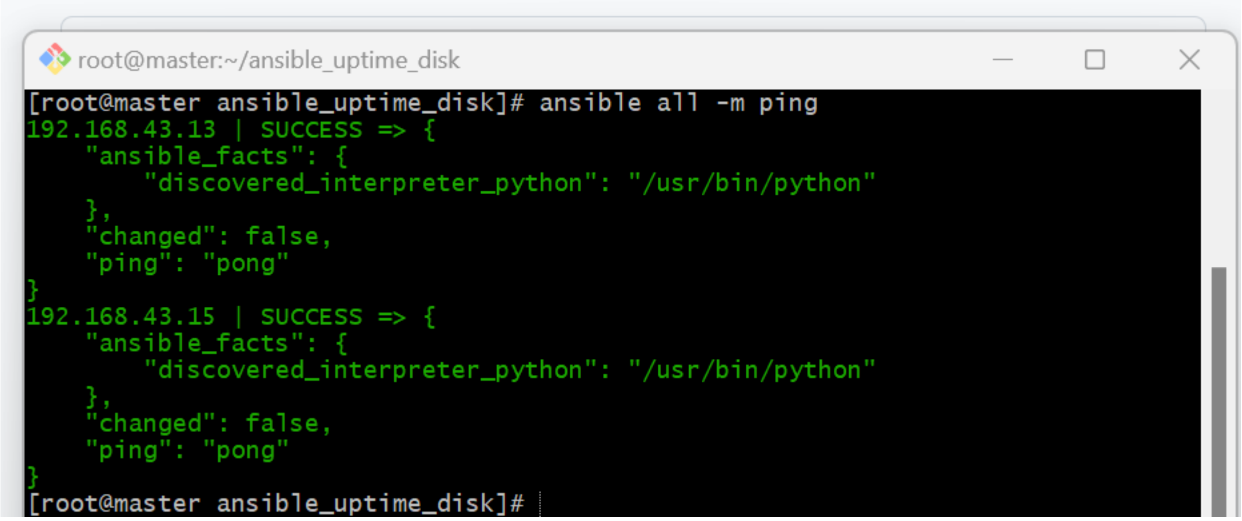
The output.log file will store the output logs.

# Validate the inventory



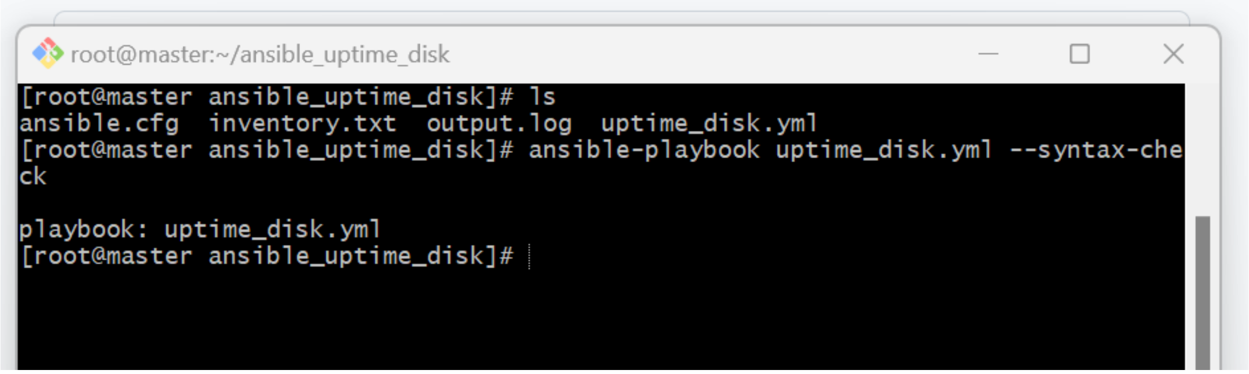
```
[root@master ansible_uptime_disk]# ansible-inventory --graph
@all:
  |--@servers:
  |  |--192.168.43.13
  |  |--192.168.43.15
  |--@ungrouped:
[root@master ansible_uptime_disk]#
```

# Checking the connectivity of all the target nodes



```
[root@master ansible_uptime_disk]# ansible all -m ping
192.168.43.13 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
192.168.43.15 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
[root@master ansible_uptime_disk]#
```

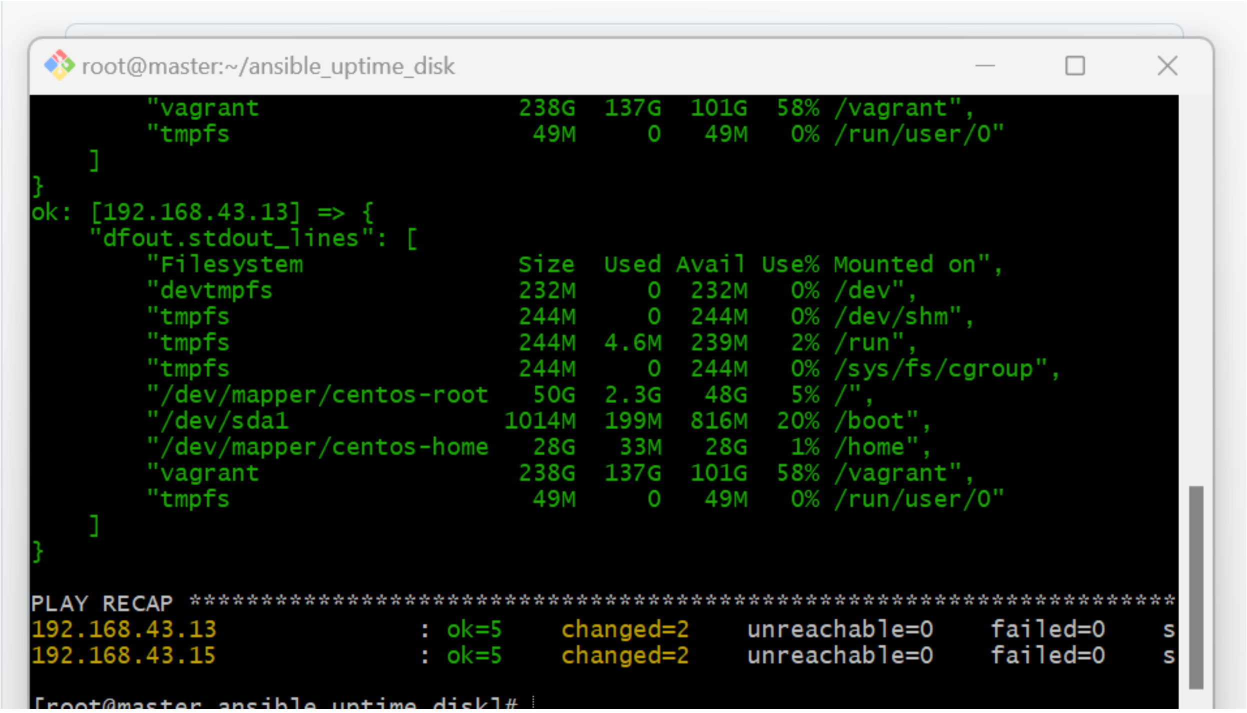
# Checking the playbook syntax before running it



```
[root@master ansible_uptime_disk]# ls
ansible.cfg  inventory.txt  output.log  uptime_disk.yml
[root@master ansible_uptime_disk]# ansible-playbook uptime_disk.yml --syntax-che
ck

playbook: uptime_disk.yml
[root@master ansible_uptime_disk]#
```

# Playbook output

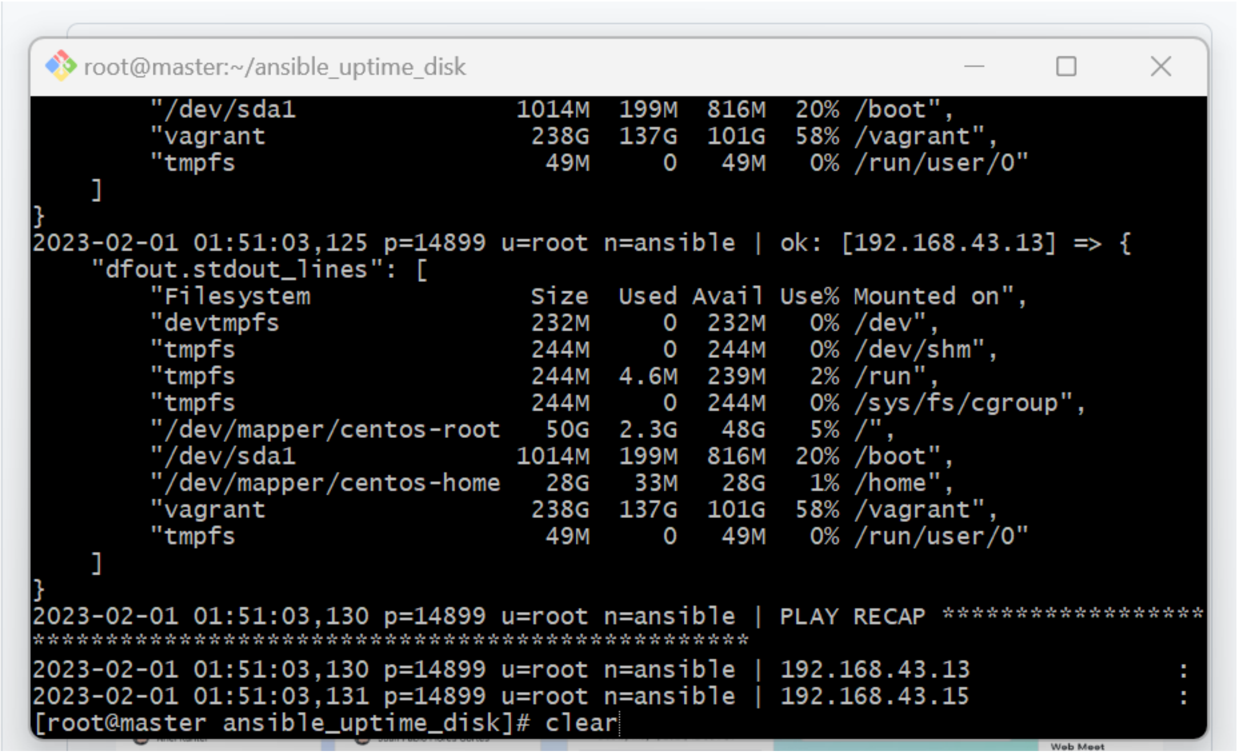


```
        "vagrant                      238G  137G  101G  58% /vagrant",
        "tmpfs                         49M     0   49M   0% /run/user/0"
    ]
}
ok: [192.168.43.13] => {
    "dfout.stdout_lines": [
        "Filesystem                   Size  Used Avail Use% Mounted on",
        "devtmpfs                     232M     0  232M   0% /dev",
        "tmpfs                        244M     0  244M   0% /dev/shm",
        "tmpfs                        244M  4.6M  239M   2% /run",
        "tmpfs                        244M     0  244M   0% /sys/fs/cgroup",
        "/dev/mapper/centos-root       50G  2.3G   48G   5% /",
        "/dev/sda1                   1014M  199M  816M  20% /boot",
        "/dev/mapper/centos-home       28G   33M   28G   1% /home",
        "vagrant                      238G  137G  101G  58% /vagrant",
        "tmpfs                         49M     0   49M   0% /run/user/0"
    ]
}

PLAY RECAP ****************************************************************************
192.168.43.13              : ok=5    changed=2    unreachable=0    failed=0    s
192.168.43.15              : ok=5    changed=2    unreachable=0    failed=0    s

[root@master ansible_uptime_disk]#
```

# Checking the output.log file to see the output logs



```
        "/dev/sda1                   1014M  199M  816M  20% /boot",
        "vagrant                      238G  137G  101G  58% /vagrant",
        "tmpfs                         49M     0   49M   0% /run/user/0"
    ]
}
2023-02-01 01:51:03,125 p=14899 u=root n=ansible | ok: [192.168.43.13] => {
    "dfout.stdout_lines": [
        "Filesystem                   Size  Used Avail Use% Mounted on",
        "devtmpfs                     232M     0  232M   0% /dev",
        "tmpfs                        244M     0  244M   0% /dev/shm",
        "tmpfs                        244M  4.6M  239M   2% /run",
        "tmpfs                        244M     0  244M   0% /sys/fs/cgroup",
        "/dev/mapper/centos-root       50G  2.3G   48G   5% /",
        "/dev/sda1                   1014M  199M  816M  20% /boot",
        "/dev/mapper/centos-home       28G   33M   28G   1% /home",
        "vagrant                      238G  137G  101G  58% /vagrant",
        "tmpfs                         49M     0   49M   0% /run/user/0"
    ]
}
2023-02-01 01:51:03,130 p=14899 u=root n=ansible | PLAY RECAP ******************
*************************************************
2023-02-01 01:51:03,130 p=14899 u=root n=ansible | 192.168.43.13              :
2023-02-01 01:51:03,131 p=14899 u=root n=ansible | 192.168.43.15              :
[root@master ansible_uptime_disk]# clear
```

Now, the company can be able to view the uptime and disk usage of the target servers.