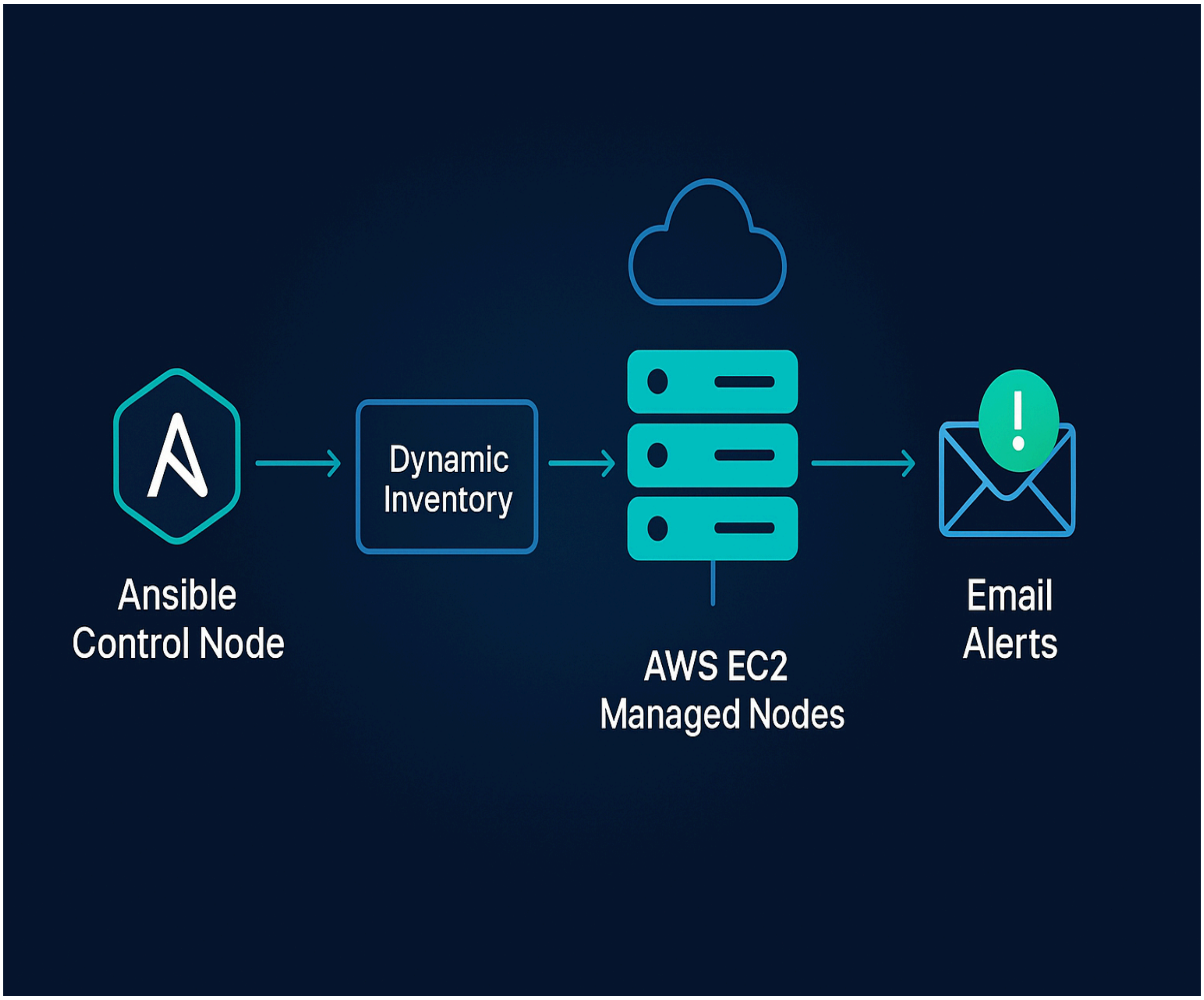
# VM Health Monitoring with Ansible



# Overview

This project provides a fully automated solution for monitoring the health of AWS EC2 instances using Ansible. It leverages Dynamic Inventory to automatically discover instances, collects vital system metrics (CPU, RAM, Disk), and sends formatted HTML email alerts detailing the health status of your infrastructure.

## Key Features

- Dynamic Discovery: Automatically finds EC2 instances tagged with Environment=dev.
- Automated Setup: Scripts to tag instances and inject SSH keys.
- Health Metrics: Monitors CPU, Memory, and Disk usage.
- Reporting: Sends visual HTML email reports with health status badges.

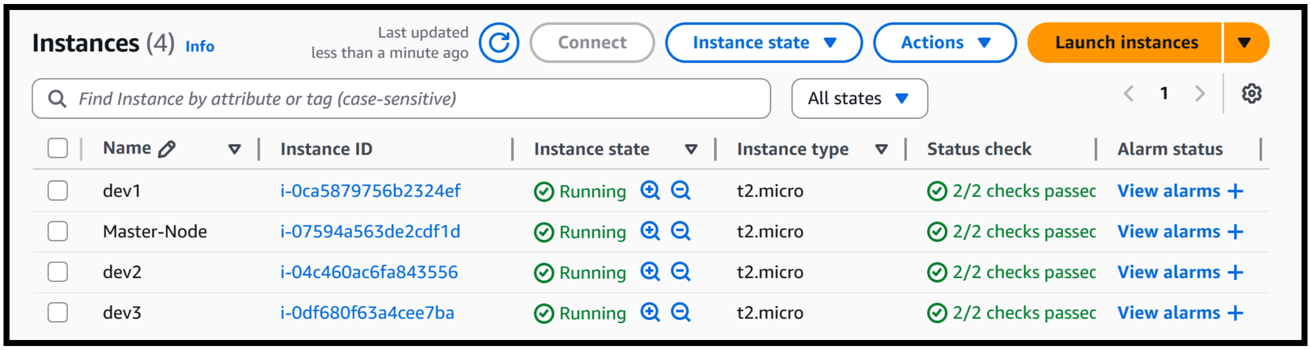
# Prerequisites

- OS: Ubuntu 20.04/22.04 (Control Node)
- AWS Account: With access to create/read EC2 instances.
- IAM User: With Programmatic Access (Access Key & Secret Key).
- SSH Key Pair: A .pem file valid for the target instances.

# Part 1: System Installation & Setup

Perform these steps on your Ansible Control Node.

## 1.1 Creating  Virtual Machines (ec2-instances)

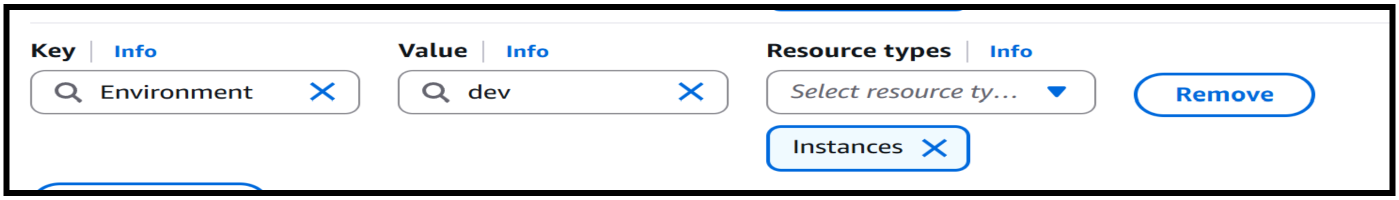

Instances (4) Info
Last updated less than a minute ago
Connect
Instance state ▼
Actions ▼
Launch instances ▼
Find Instance by attribute or tag (case-sensitive)
All states ▼
1

| □ | Name | Instance ID | Instance state | Instance type | Status check | Alarm status |
|---|---|---|---|---|---|---|
| □ | dev1 | i-0ca5879756b2324ef | ✓ Running | t2.micro | ✓ 2/2 checks passec | View alarms + |
| □ | Master-Node | i-07594a563de2cdf1d | ✓ Running | t2.micro | ✓ 2/2 checks passec | View alarms + |
| □ | dev2 | i-04c460ac6fa843556 | ✓ Running | t2.micro | ✓ 2/2 checks passec | View alarms + |
| □ | dev3 | i-0df680f63a4cee7ba | ✓ Running | t2.micro | ✓ 2/2 checks passec | View alarms + |

In this Master-Node VM is my Control node and other are my control nodes

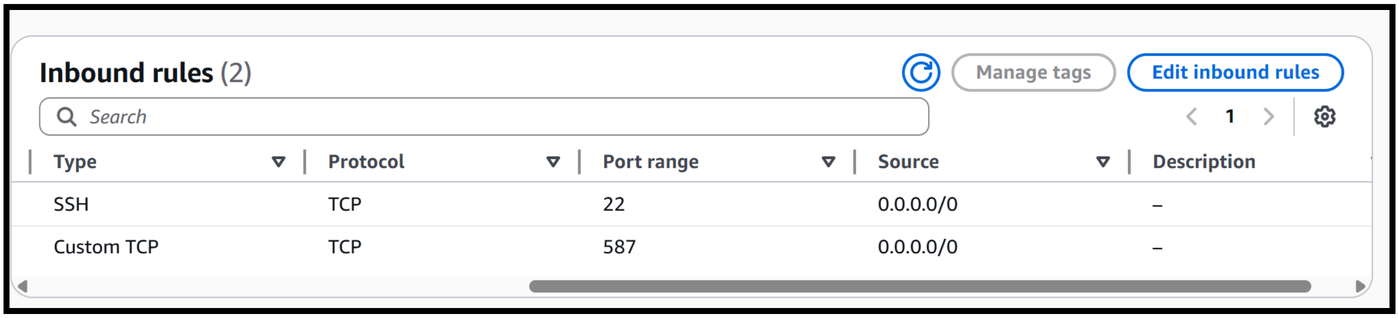
.

For managed nodes give tagging “Environment = dev”



| Key | Value | Resource types |
|---|---|---|
| Environment | dev | Select resource ty…<br>Instances |

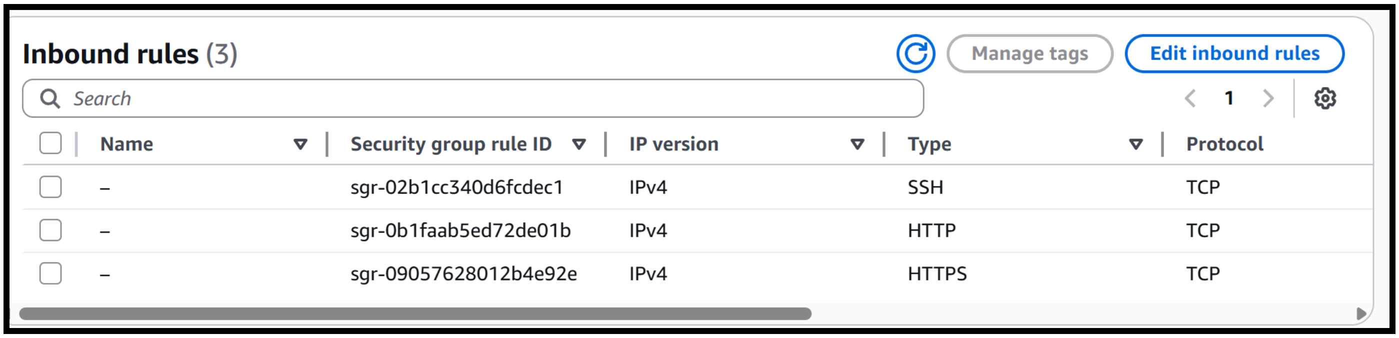
Inbound rules for Master-node (control-node)



Inbound rules (2)

| Type | Protocol | Port range | Source | Description |
|---|---|---|---|---|
| SSH | TCP | 22 | 0.0.0.0/0 | – |
| Custom TCP | TCP | 587 | 0.0.0.0/0 | – |

Inbound rule for managed nodes



Inbound rules (3)

| □ | Name | Security group rule ID | IP version | Type | Protocol |
|---|---|---|---|---|---|
| □ | – | sgr-02b1cc340d6fcdec1 | IPv4 | SSH | TCP |
| □ | – | sgr-0b1faab5ed72de01b | IPv4 | HTTP | TCP |
| □ | – | sgr-09057628012b4e92e | IPv4 | HTTPS | TCP |

## 1.2 Connecting to Master Node , Update System & Install Ansible

**Update the package repositories and install the official Ansible PPA for the latest version.**

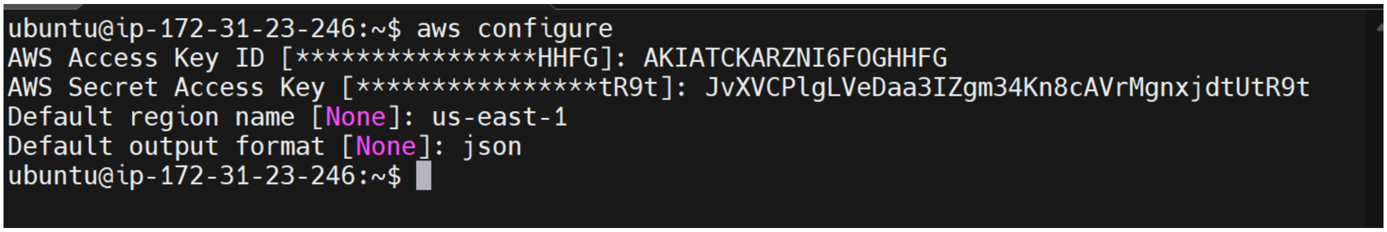
```
#sudo apt update && sudo apt upgrade -y
#sudo apt install software-properties-common -y
#sudo add-apt-repository --yes --update ppa:ansible/ansible
#sudo apt install ansible -y
```

## 1.2 Install AWS CLI

The AWS CLI is required for the dynamic inventory plugin to interact with the AWS API.

```
#curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"

#sudo apt install unzip
#unzip awscliv2.zip
#sudo ./aws/install
aws --version
```

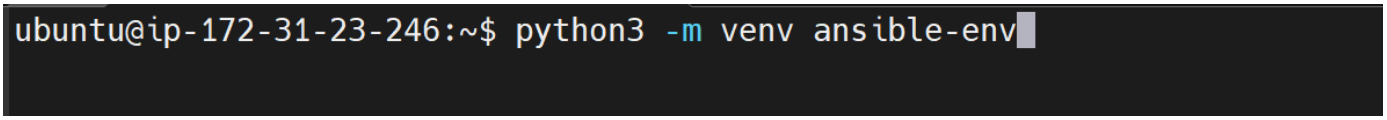
## 1.3 Configure AWS Credentials



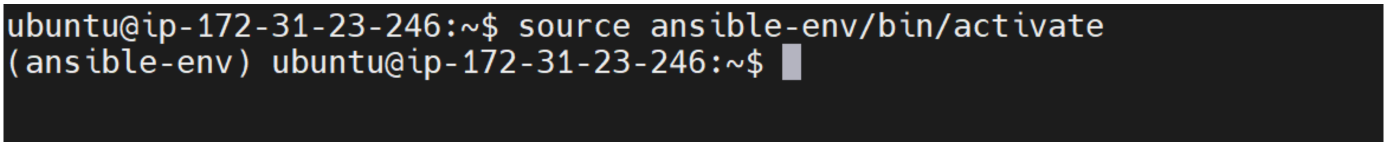
```
ubuntu@ip-172-31-23-246:~$ aws configure
AWS Access Key ID [****************HHFG]: AKIATCKARZNI6FOGHHFG
AWS Secret Access Key [****************tR9t]: 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
Default region name [None]: us-east-1
Default output format [None]: json
ubuntu@ip-172-31-23-246:~$
```

## 1.4 Set Up Python Environment

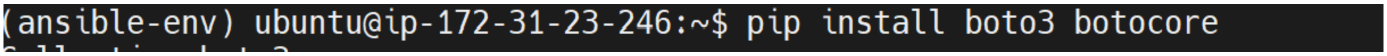
Ansible requires specific Python libraries (boto3, botocore) to talk to AWS. We will use a virtual environment to keep the system clean.



```
ubuntu@ip-172-31-23-246:~$ python3 -m venv ansible-env
```



```
ubuntu@ip-172-31-23-246:~$ source ansible-env/bin/activate
(ansible-env) ubuntu@ip-172-31-23-246:~$
```

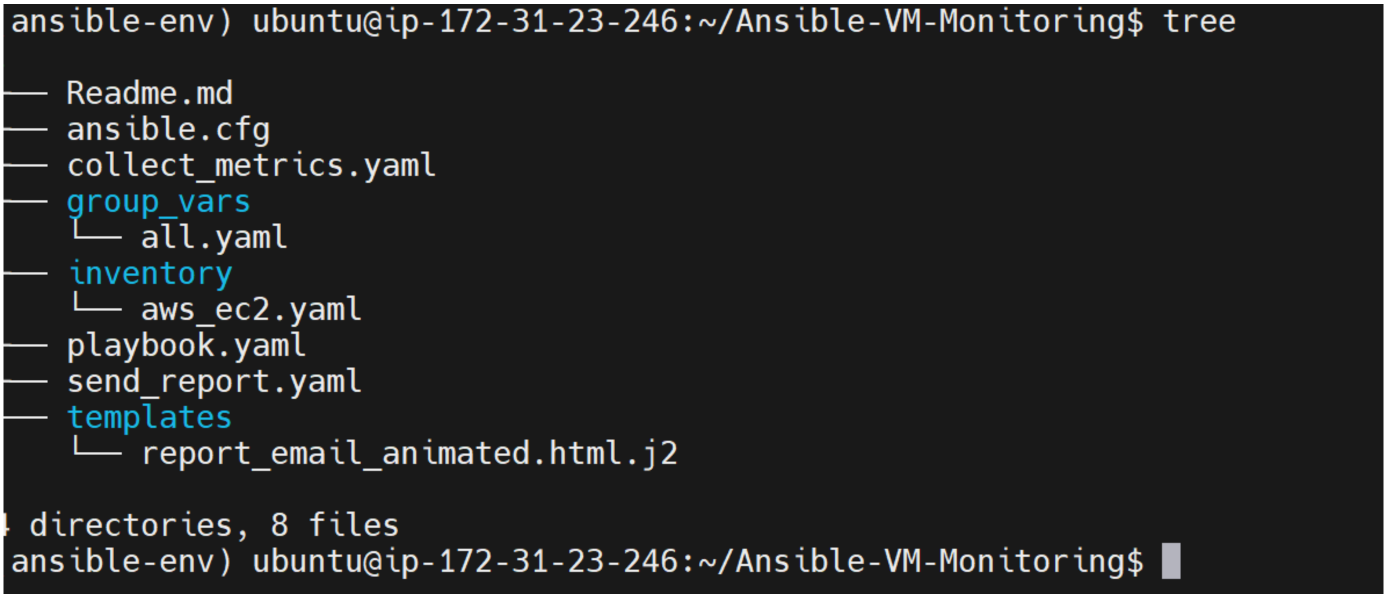


```
(ansible-env) ubuntu@ip-172-31-23-246:~$ pip install boto3 botocore
```

## Part 2: Project Configuration

Create the project structure and configuration files.

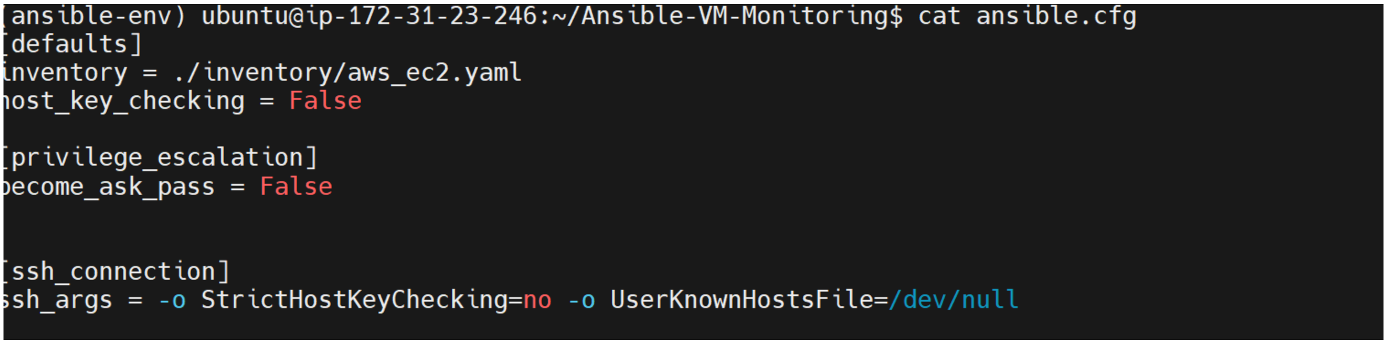
### 2.1 Project Directory Structure



```
ansible-env) ubuntu@ip-172-31-23-246:~/Ansible-VM-Monitoring$ tree

── Readme.md
── ansible.cfg
── collect_metrics.yaml
── group_vars
   └── all.yaml
── inventory
   └── aws_ec2.yaml
── playbook.yaml
── send_report.yaml
── templates
   └── report_email_animated.html.j2

4 directories, 8 files
ansible-env) ubuntu@ip-172-31-23-246:~/Ansible-VM-Monitoring$
```

### 2.2 Ansible Configuration (ansible.cfg)

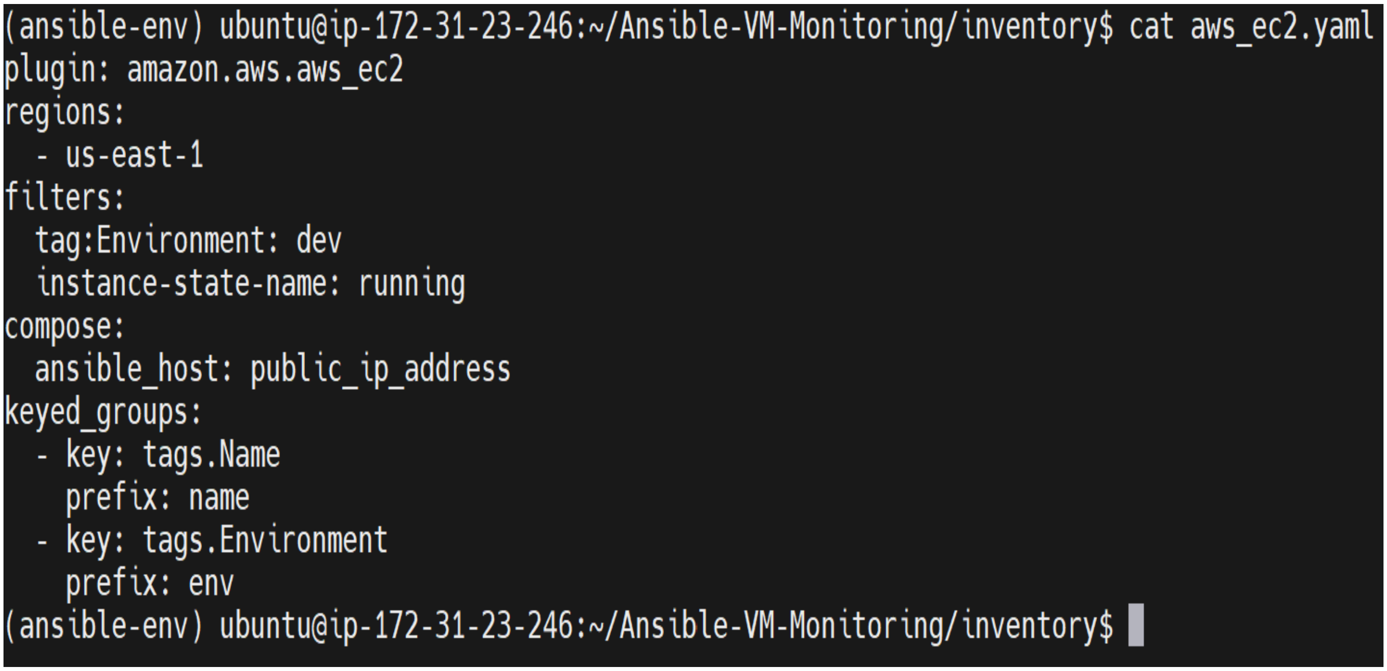
Create ansible.cfg in the project root. This tells Ansible to use the dynamic inventory and ignore host key checking (essential for cloud environments).



```
(ansible-env) ubuntu@ip-172-31-23-246:~/Ansible-VM-Monitoring$ cat ansible.cfg
[defaults]
inventory = ./inventory/aws_ec2.yaml
host_key_checking = False

[privilege_escalation]
become_ask_pass = False


[ssh_connection]
ssh_args = -o StrictHostKeyChecking=no -o UserKnownHostsFile=/dev/null
```

## 2.3 Dynamic Inventory (inventory/aws_ec2.yaml)

This file defines how Ansible finds your EC2 instances.



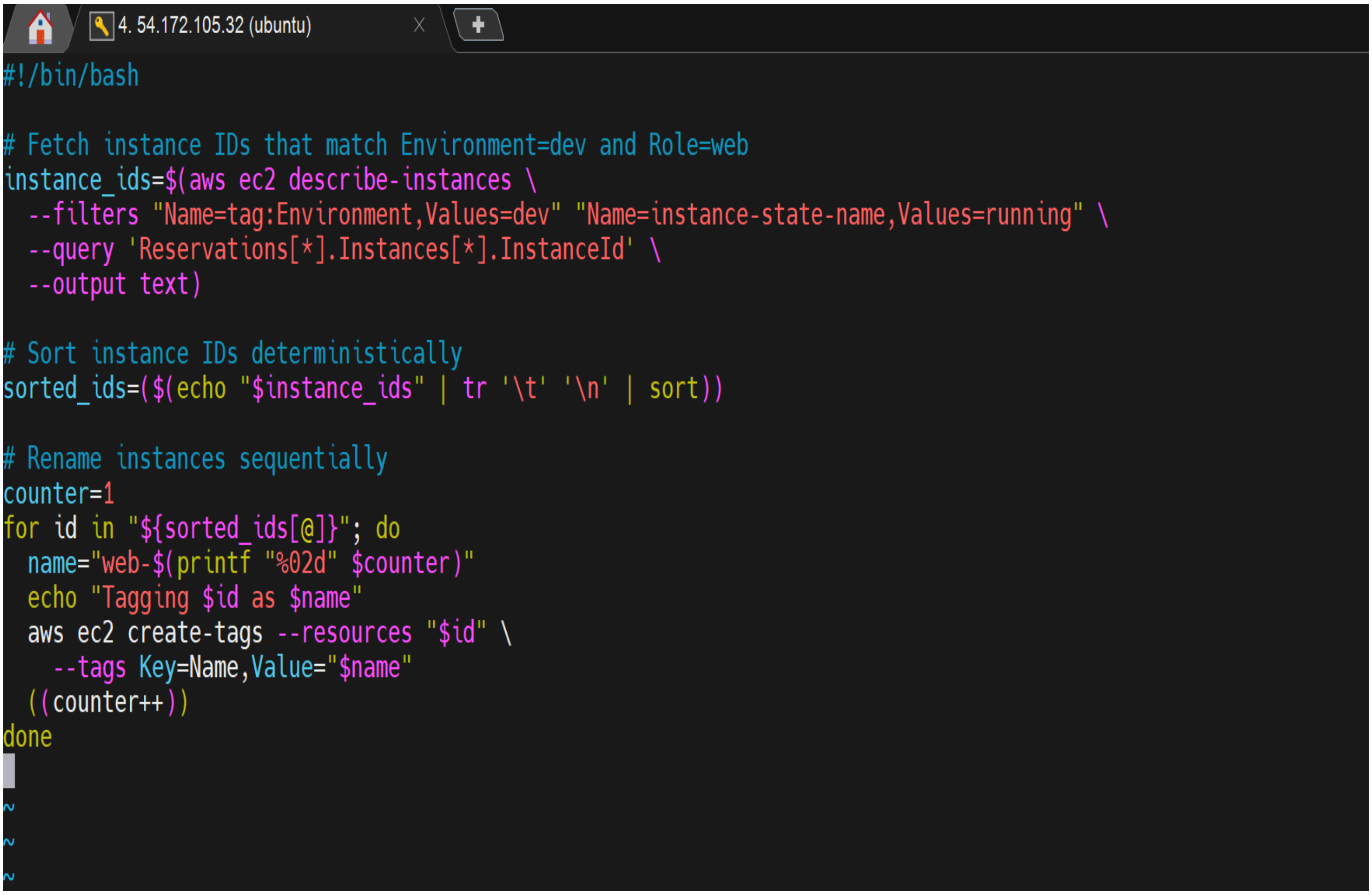
```
(ansible-env) ubuntu@ip-172-31-23-246:~/Ansible-VM-Monitoring/inventory$ cat aws_ec2.yaml
plugin: amazon.aws.aws_ec2
regions:
  - us-east-1
filters:
  tag:Environment: dev
  instance-state-name: running
compose:
  ansible_host: public_ip_address
keyed_groups:
  - key: tags.Name
    prefix: name
  - key: tags.Environment
    prefix: env
(ansible-env) ubuntu@ip-172-31-23-246:~/Ansible-VM-Monitoring/inventory$
```

# Part 3: Infrastructure Preparation

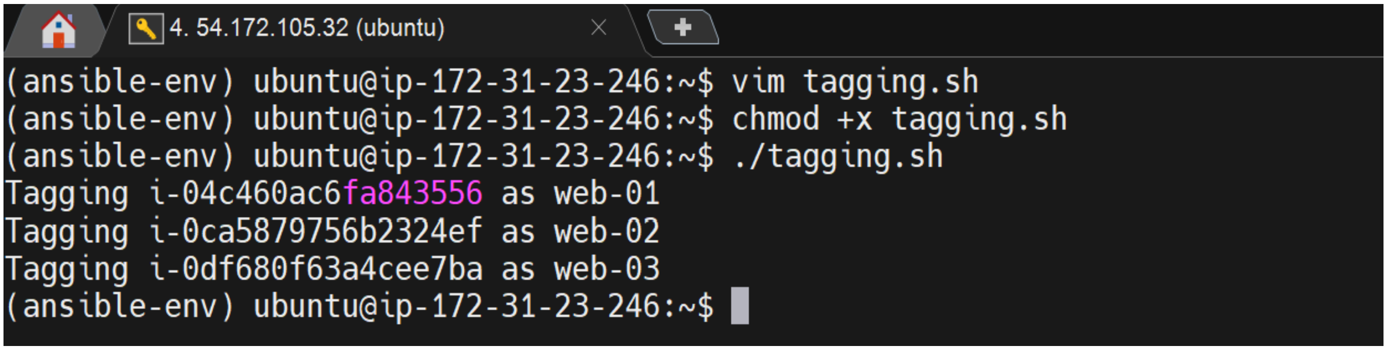
Before running the monitoring playbooks, we need to prepare the AWS instances.

## 3.1 Tag EC2 Instances

Run this script to verify your instances are running and assign them sequential names (e.g .web-01, web-02).



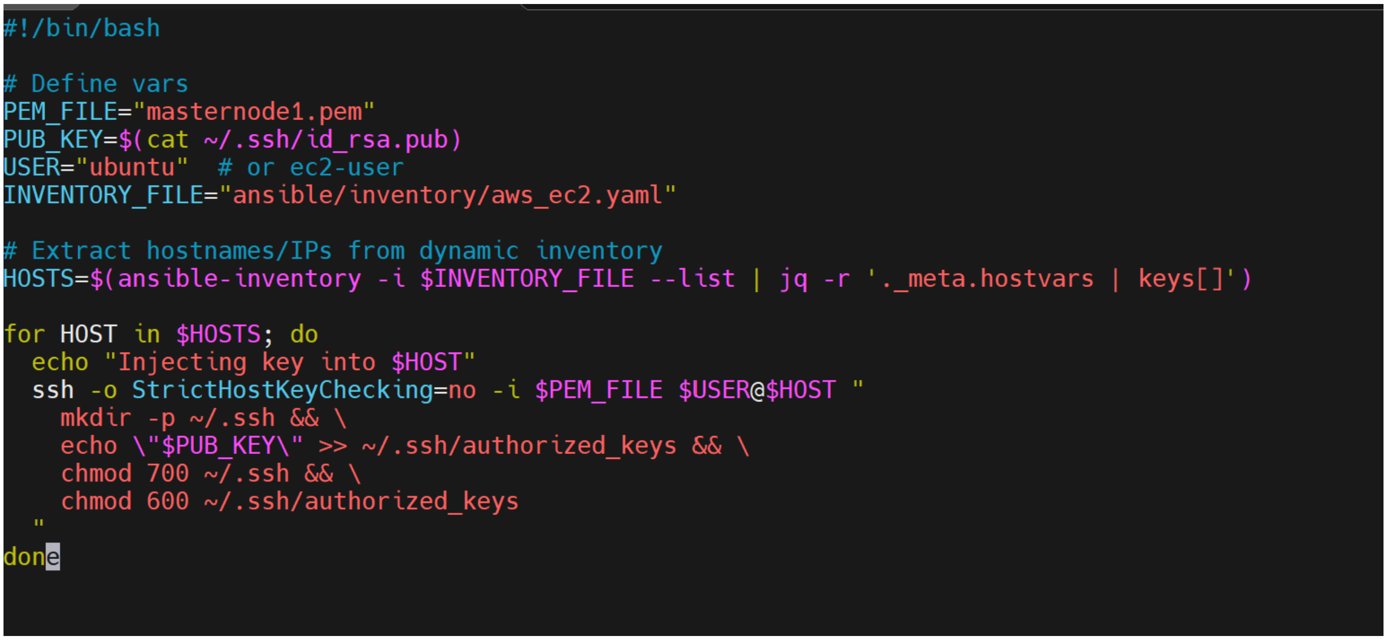
```
#!/bin/bash

# Fetch instance IDs that match Environment=dev and Role=web
instance_ids=$(aws ec2 describe-instances \
  --filters "Name=tag:Environment,Values=dev" "Name=instance-state-name,Values=running" \
  --query 'Reservations[*].Instances[*].InstanceId' \
  --output text)

# Sort instance IDs deterministically
sorted_ids=($(echo "$instance_ids" | tr '\t' '\n' | sort))

# Rename instances sequentially
counter=1
for id in "${sorted_ids[@]}"; do
  name="web-$(printf "%02d" $counter)"
  echo "Tagging $id as $name"
  aws ec2 create-tags --resources "$id" \
    --tags Key=Name,Value="$name"
  ((counter++))
done

~
~
~
```



```
(ansible-env) ubuntu@ip-172-31-23-246:~$ vim tagging.sh
(ansible-env) ubuntu@ip-172-31-23-246:~$ chmod +x tagging.sh
(ansible-env) ubuntu@ip-172-31-23-246:~$ ./tagging.sh
Tagging i-04c460ac6fa843556 as web-01
Tagging i-0ca5879756b2324ef as web-02
Tagging i-0df680f63a4cee7ba as web-03
(ansible-env) ubuntu@ip-172-31-23-246:~$
```

### 3.2 Inject SSH Keys

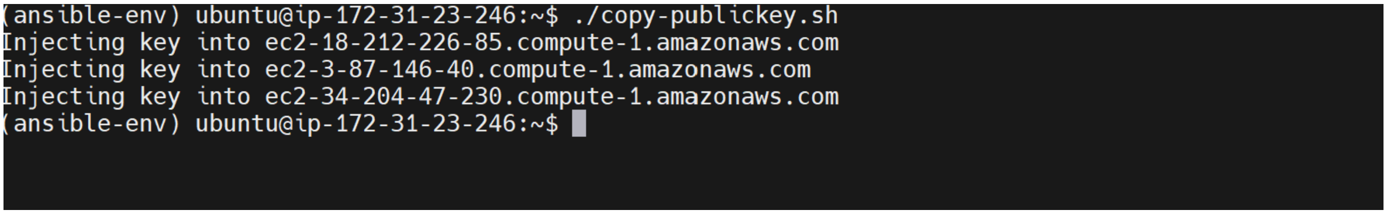
This script uses the dynamic inventory to find the public IPs and copies your local SSH public key to the remote servers, allowing passwordless Ansible execution.



```
#!/bin/bash

# Define vars
PEM_FILE="masternode1.pem"
PUB_KEY=$(cat ~/.ssh/id_rsa.pub)
USER="ubuntu"  # or ec2-user
INVENTORY_FILE="ansible/inventory/aws_ec2.yaml"

# Extract hostnames/IPs from dynamic inventory
HOSTS=$(ansible-inventory -i $INVENTORY_FILE --list | jq -r '._meta.hostvars | keys[]')

for HOST in $HOSTS; do
  echo "Injecting key into $HOST"
  ssh -o StrictHostKeyChecking=no -i $PEM_FILE $USER@$HOST "
    mkdir -p ~/.ssh && \
    echo \"$PUB_KEY\" >> ~/.ssh/authorized_keys && \
    chmod 700 ~/.ssh && \
    chmod 600 ~/.ssh/authorized_keys
  "
done
```

**Giving Execution permission to the key**

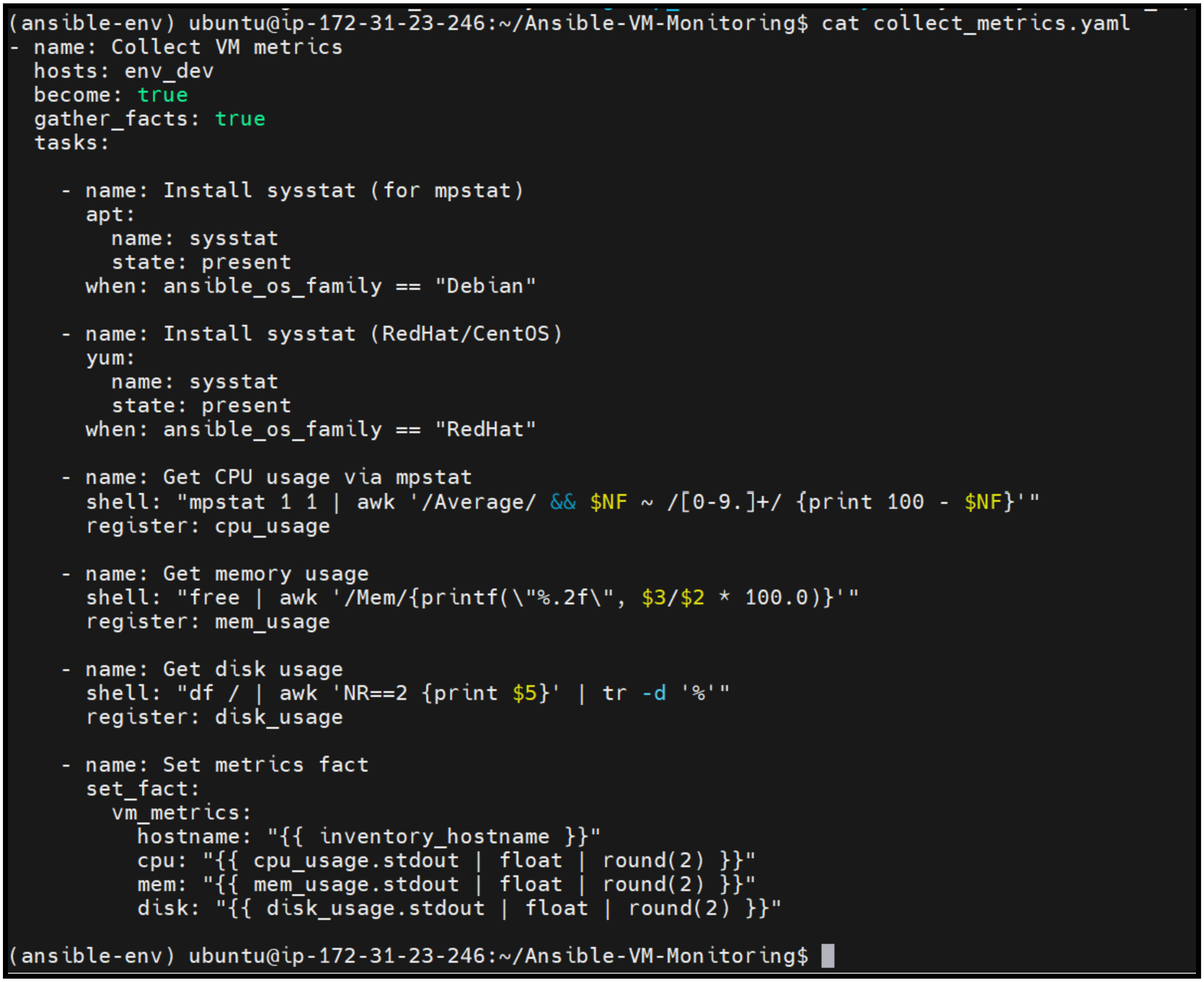
```
#chmod +x scripts/copy-publickey.sh
```



```
(ansible-env) ubuntu@ip-172-31-23-246:~$ ./copy-publickey.sh
Injecting key into ec2-18-212-226-85.compute-1.amazonaws.com
Injecting key into ec2-3-87-146-40.compute-1.amazonaws.com
Injecting key into ec2-34-204-47-230.compute-1.amazonaws.com
(ansible-env) ubuntu@ip-172-31-23-246:~$
```

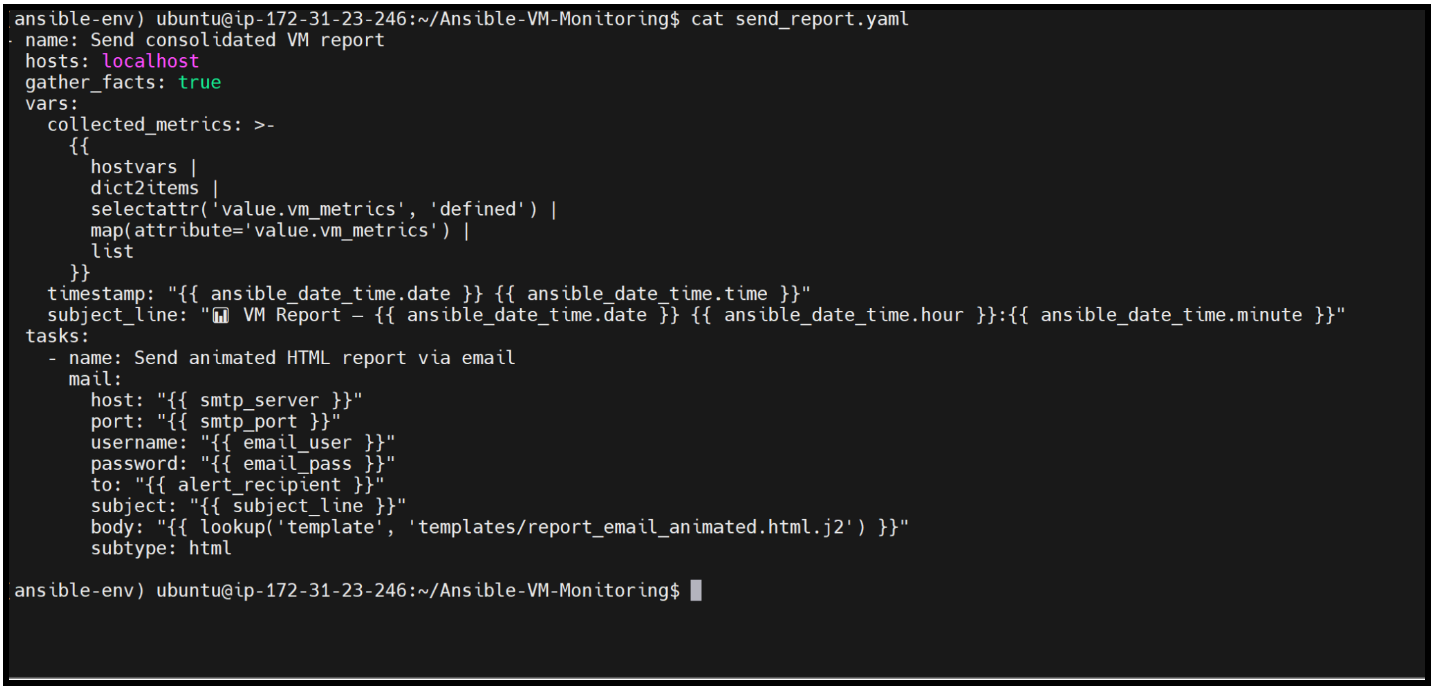
# Part4: creating playbook files

## 1.collect_matrix.yml



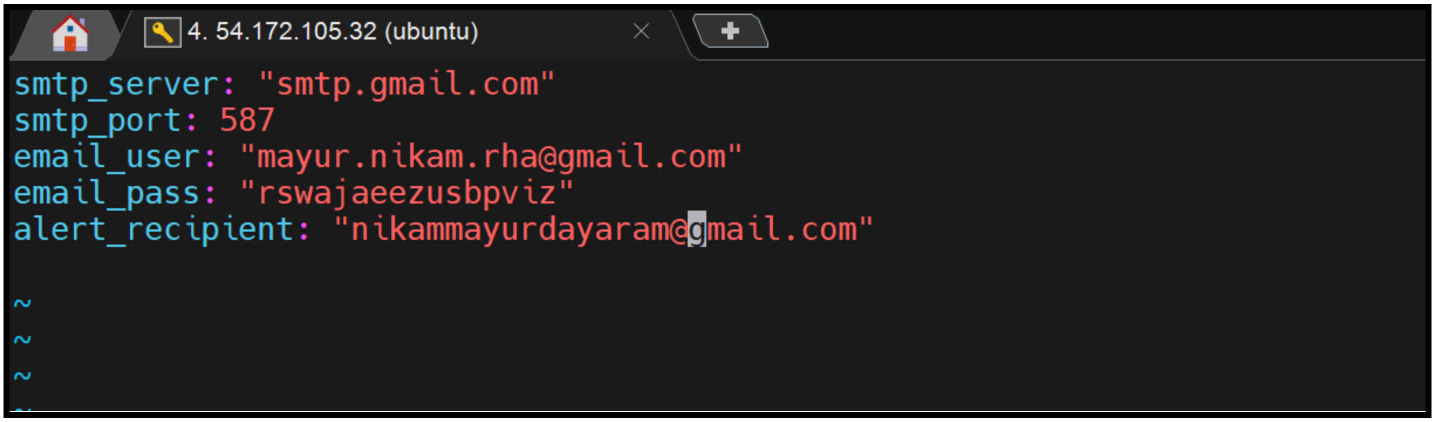
```
(ansible-env) ubuntu@ip-172-31-23-246:~/Ansible-VM-Monitoring$ cat collect_metrics.yaml
- name: Collect VM metrics
  hosts: env_dev
  become: true
  gather_facts: true
  tasks:

    - name: Install sysstat (for mpstat)
      apt:
        name: sysstat
        state: present
      when: ansible_os_family == "Debian"

    - name: Install sysstat (RedHat/CentOS)
      yum:
        name: sysstat
        state: present
      when: ansible_os_family == "RedHat"

    - name: Get CPU usage via mpstat
      shell: "mpstat 1 1 | awk '/Average/ && $NF ~ /[0-9.]+/ {print 100 - $NF}'"
      register: cpu_usage

    - name: Get memory usage
      shell: "free | awk '/Mem/{printf(\"%.2f\", $3/$2 * 100.0)}'"
      register: mem_usage

    - name: Get disk usage
      shell: "df / | awk 'NR==2 {print $5}' | tr -d '%'"
      register: disk_usage

    - name: Set metrics fact
      set_fact:
        vm_metrics:
          hostname: "{{ inventory_hostname }}"
          cpu: "{{ cpu_usage.stdout | float | round(2) }}"
          mem: "{{ mem_usage.stdout | float | round(2) }}"
          disk: "{{ disk_usage.stdout | float | round(2) }}"

(ansible-env) ubuntu@ip-172-31-23-246:~/Ansible-VM-Monitoring$
```

## 2.send_report.yml



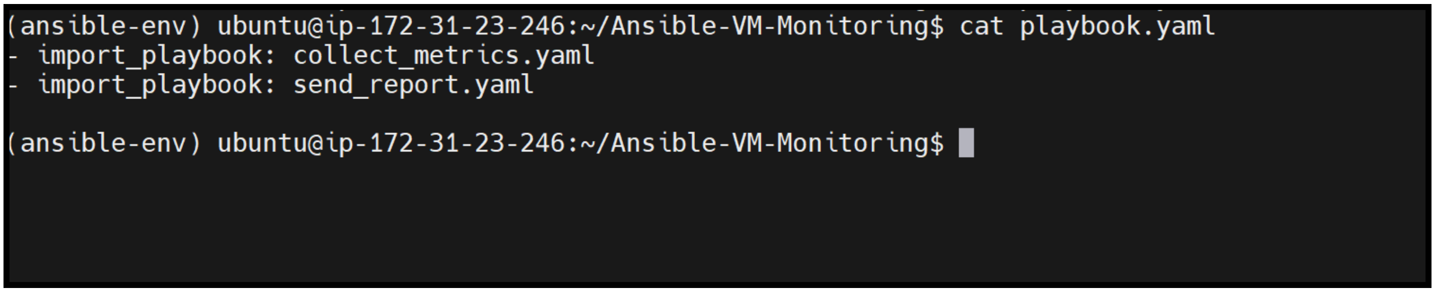
```
ansible-env) ubuntu@ip-172-31-23-246:~/Ansible-VM-Monitoring$ cat send_report.yaml
- name: Send consolidated VM report
  hosts: localhost
  gather_facts: true
  vars:
    collected_metrics: >-
      {{
        hostvars |
        dict2items |
        selectattr('value.vm_metrics', 'defined') |
        map(attribute='value.vm_metrics') |
        list
      }}
    timestamp: "{{ ansible_date_time.date }} {{ ansible_date_time.time }}"
    subject_line: "📊 VM Report – {{ ansible_date_time.date }} {{ ansible_date_time.hour }}:{{ ansible_date_time.minute }}"
  tasks:
    - name: Send animated HTML report via email
      mail:
        host: "{{ smtp_server }}"
        port: "{{ smtp_port }}"
        username: "{{ email_user }}"
        password: "{{ email_pass }}"
        to: "{{ alert_recipient }}"
        subject: "{{ subject_line }}"
        body: "{{ lookup('template', 'templates/report_email_animated.html.j2') }}"
        subtype: html

ansible-env) ubuntu@ip-172-31-23-246:~/Ansible-VM-Monitoring$
```

## 3.Ansible-VM-Monitoring/group_vars/all.yml



```
smtp_server: "smtp.gmail.com"
smtp_port: 587
email_user: "mayur.nikam.rha@gmail.com"
email_pass: "rswajaeezusbpviz"
alert_recipient: "nikammayurdayaram@gmail.com"

~
~
~
```

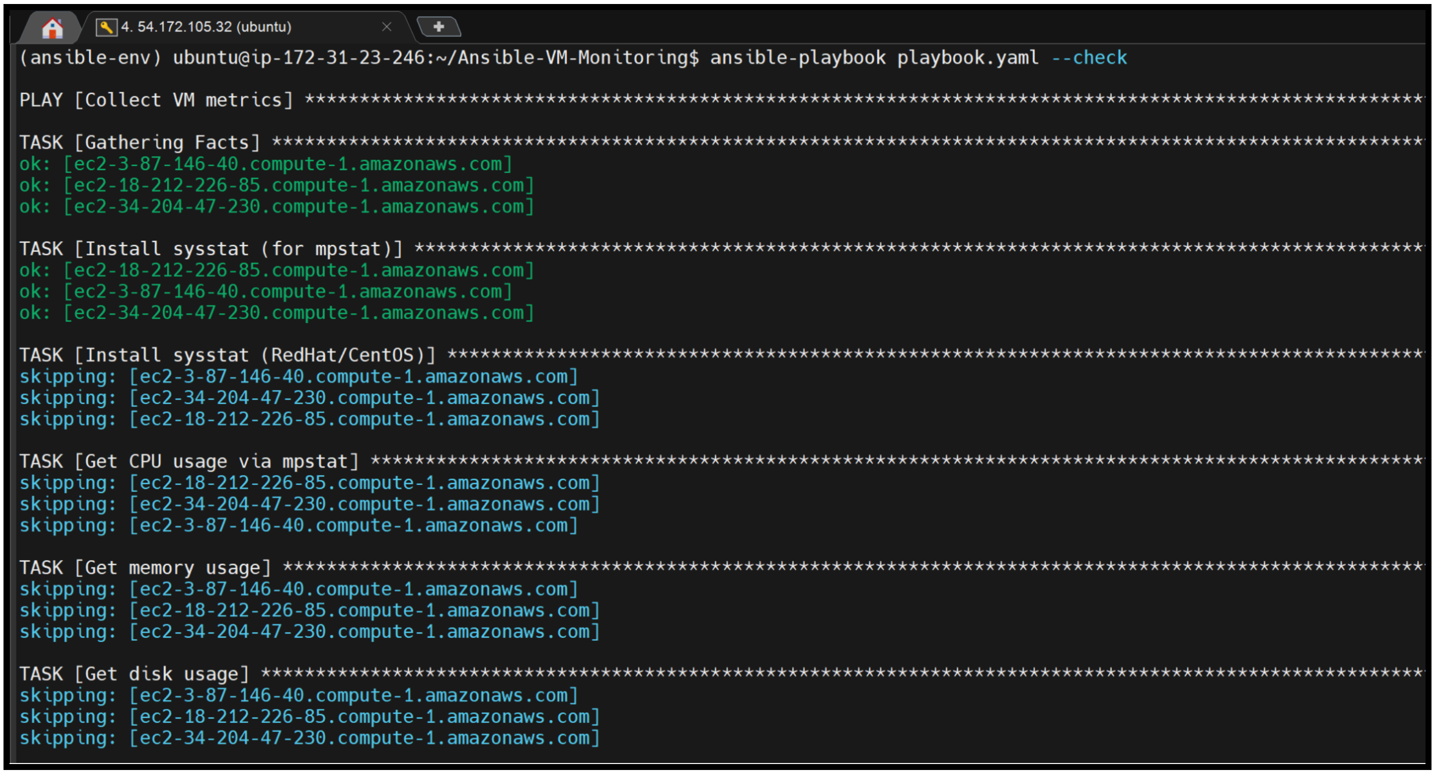
## 3.playbook.yml



```
(ansible-env) ubuntu@ip-172-31-23-246:~/Ansible-VM-Monitoring$ cat playbook.yaml
- import_playbook: collect_metrics.yaml
- import_playbook: send_report.yaml

(ansible-env) ubuntu@ip-172-31-23-246:~/Ansible-VM-Monitoring$
```

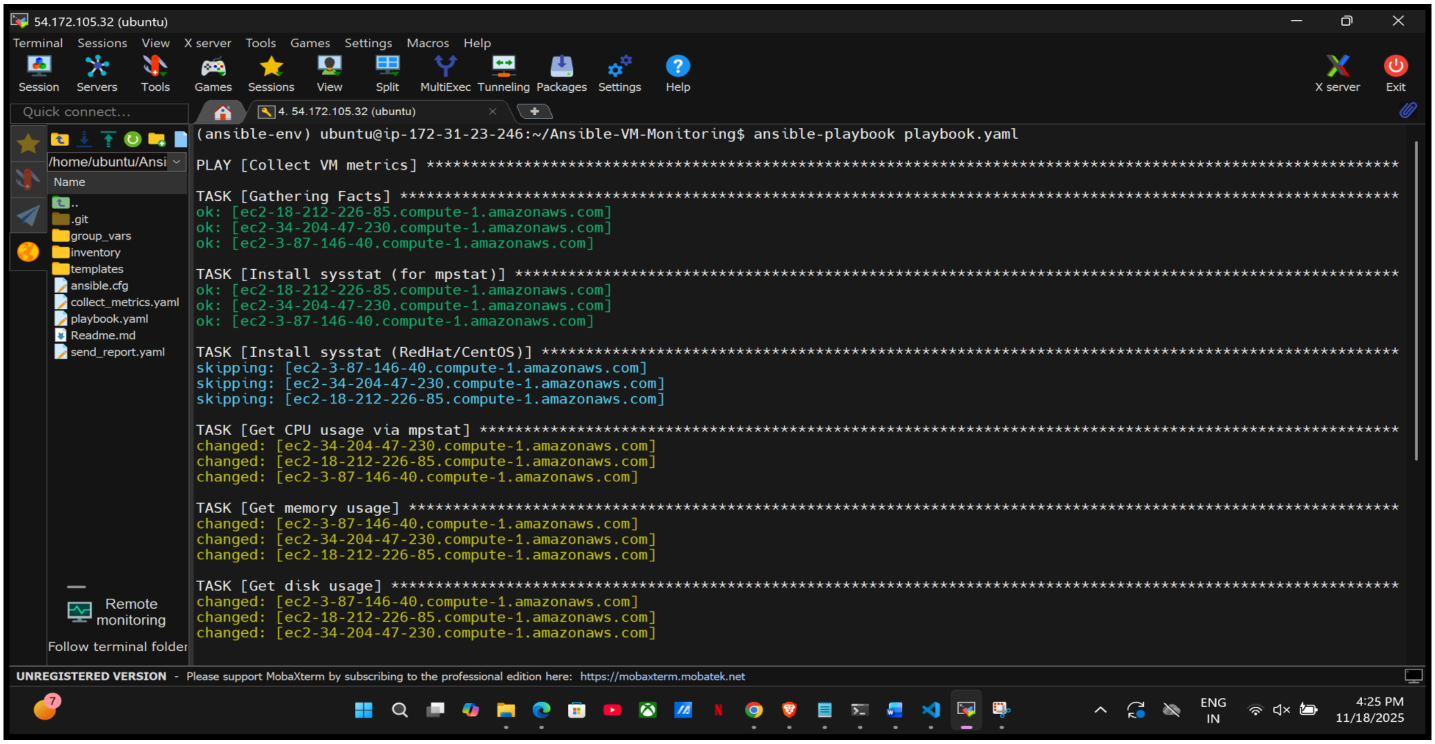
# Part 5: Running the Monitor

## 5.1 Verify by “—check”



```
(ansible-env) ubuntu@ip-172-31-23-246:~/Ansible-VM-Monitoring$ ansible-playbook playbook.yaml --check

PLAY [Collect VM metrics] *****************************************************************************

TASK [Gathering Facts] ********************************************************************************
ok: [ec2-3-87-146-40.compute-1.amazonaws.com]
ok: [ec2-18-212-226-85.compute-1.amazonaws.com]
ok: [ec2-34-204-47-230.compute-1.amazonaws.com]

TASK [Install sysstat (for mpstat)] *******************************************************************
ok: [ec2-18-212-226-85.compute-1.amazonaws.com]
ok: [ec2-3-87-146-40.compute-1.amazonaws.com]
ok: [ec2-34-204-47-230.compute-1.amazonaws.com]

TASK [Install sysstat (RedHat/CentOS)] ****************************************************************
skipping: [ec2-3-87-146-40.compute-1.amazonaws.com]
skipping: [ec2-34-204-47-230.compute-1.amazonaws.com]
skipping: [ec2-18-212-226-85.compute-1.amazonaws.com]

TASK [Get CPU usage via mpstat] ***********************************************************************
skipping: [ec2-18-212-226-85.compute-1.amazonaws.com]
skipping: [ec2-34-204-47-230.compute-1.amazonaws.com]
skipping: [ec2-3-87-146-40.compute-1.amazonaws.com]

TASK [Get memory usage] *******************************************************************************
skipping: [ec2-3-87-146-40.compute-1.amazonaws.com]
skipping: [ec2-18-212-226-85.compute-1.amazonaws.com]
skipping: [ec2-34-204-47-230.compute-1.amazonaws.com]

TASK [Get disk usage] *********************************************************************************
skipping: [ec2-3-87-146-40.compute-1.amazonaws.com]
skipping: [ec2-18-212-226-85.compute-1.amazonaws.com]
skipping: [ec2-34-204-47-230.compute-1.amazonaws.com]
```

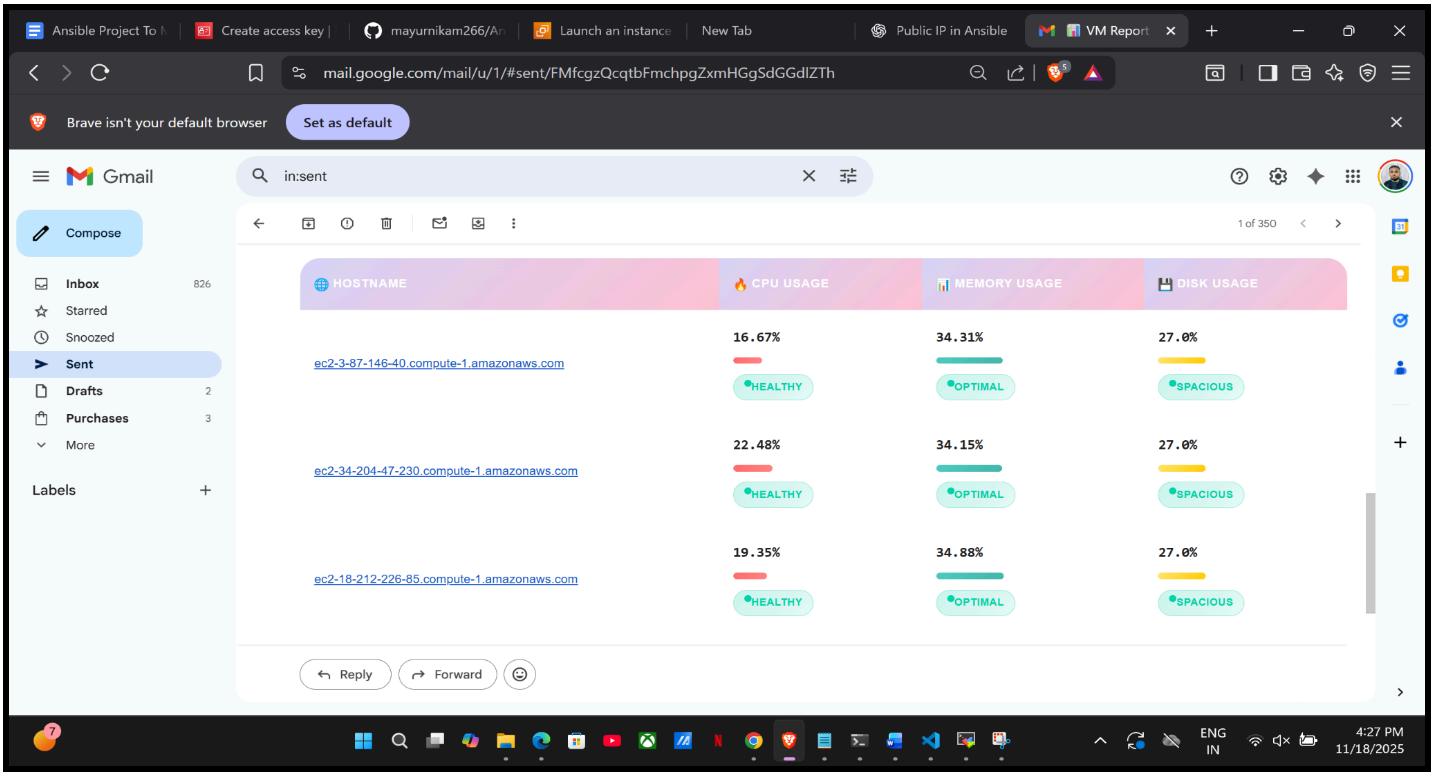
## 5.2 Running the Playbook



```
(ansible-env) ubuntu@ip-172-31-23-246:~/Ansible-VM-Monitoring$ ansible-playbook playbook.yaml
PLAY [Collect VM metrics] *****************************************************************************************
TASK [Gathering Facts] *******************************************************************************************
ok: [ec2-18-212-226-85.compute-1.amazonaws.com]
ok: [ec2-34-204-47-230.compute-1.amazonaws.com]
ok: [ec2-3-87-146-40.compute-1.amazonaws.com]

TASK [Install sysstat (for mpstat)] ******************************************************************************
ok: [ec2-18-212-226-85.compute-1.amazonaws.com]
ok: [ec2-34-204-47-230.compute-1.amazonaws.com]
ok: [ec2-3-87-146-40.compute-1.amazonaws.com]

TASK [Install sysstat (RedHat/CentOS)] ***************************************************************************
skipping: [ec2-3-87-146-40.compute-1.amazonaws.com]
skipping: [ec2-34-204-47-230.compute-1.amazonaws.com]
skipping: [ec2-18-212-226-85.compute-1.amazonaws.com]

TASK [Get CPU usage via mpstat] **********************************************************************************
changed: [ec2-34-204-47-230.compute-1.amazonaws.com]
changed: [ec2-18-212-226-85.compute-1.amazonaws.com]
changed: [ec2-3-87-146-40.compute-1.amazonaws.com]

TASK [Get memory usage] ******************************************************************************************
changed: [ec2-3-87-146-40.compute-1.amazonaws.com]
changed: [ec2-34-204-47-230.compute-1.amazonaws.com]
changed: [ec2-18-212-226-85.compute-1.amazonaws.com]

TASK [Get disk usage] ********************************************************************************************
changed: [ec2-3-87-146-40.compute-1.amazonaws.com]
changed: [ec2-18-212-226-85.compute-1.amazonaws.com]
changed: [ec2-34-204-47-230.compute-1.amazonaws.com]
```

**UNREGISTERED VERSION** - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net

# 6. EMAIL REPORT



| 🌐 HOSTNAME | 🔥 CPU USAGE | 📊 MEMORY USAGE | 💾 DISK USAGE |
|---|---|---|---|
| ec2-3-87-146-40.compute-1.amazonaws.com | 16.67% HEALTHY | 34.31% OPTIMAL | 27.0% SPACIOUS |
| ec2-34-204-47-230.compute-1.amazonaws.com | 22.48% HEALTHY | 34.15% OPTIMAL | 27.0% SPACIOUS |
| ec2-18-212-226-85.compute-1.amazonaws.com | 19.35% HEALTHY | 34.88% OPTIMAL | 27.0% SPACIOUS |

# Troubleshooting

| Issue | Solution |
|---|---|
| boto3 not found | Ensure you activated the virtual env: source ansible-env/bin/activate |
| Permission Denied (Public Key) | Run copy-publickey.sh again. Ensure your local id_rsa.pub exists. |
| No hosts found | Check AWS Console. Ensure instances have tag Environment: dev and are Running. |
| Email Authentication Failed | If using Gmail, ensure you are using an App Password, not your login password. |