**Ansible Project To Monitor VMs Health**

**Create EC2 instance**

**Ansible-master – t2 medium with 20 gb**

**Worker – e2 small with 15 gb**

**🔹 Step 1: Update the System**

sudo apt update && sudo apt upgrade -y

**🔹 Step 2: Add the Ansible PPA**

Ansible provides an official maintained PPA (for latest versions):

sudo add-apt-repository --yes --update ppa:ansible/ansible

**🔹 Step 3: Install Ansible**

sudo apt install ansible –y

ansible --version

**# Install AWS CLI**

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 configure

**Tagging Script:**

#!/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

**create ssh key in ansible-master**

ssh-keygen -t rsa -b 4096 -C "ansible-master"

**ansible.cfg (time to ssh conform yes , this conf check is disable)**

vi ansible.cfg

[defaults]

inventory = ./inventory/aws\_ec2.yaml

host\_key\_checking = False

[ssh\_connection]

ssh\_args = -o StrictHostKeyChecking=no -o UserKnownHostsFile=/dev/null

**Dynamic Inventory**

inventory/aws\_ec2.yaml

plugin: amazon.aws.aws\_ec2

regions:

- ap-south-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

# Step 1: Install venv module if not already present

sudo apt install python3-venv -y

# Step 2: Create a virtual environment

python3 -m venv ansible-env

# Step 3: Activate it

source ansible-env/bin/activate

# Step 4: Install required Python packages

pip install boto3 botocore docker

ansible-galaxy collection install amazon.aws

ansible-inventory -i inventory/aws\_ec2.yaml --graph

**Copy Public Key in local <nvkey.pem> and paste in new file nvkey.pem in ansible-master instance**

sudo chmod 400 nvkey.pem

**Copy public key ansible-master to all workers**

vi copy-public-key.sh

#!/bin/bash

# Define vars

PEM\_FILE="DevOps-Shack.pem"

PUB\_KEY=$(cat ~/.ssh/id\_rsa.pub)

USER="ubuntu" # or ec2-user

INVENTORY\_FILE="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

**check ssh is correct**

ssh Ubuntu@ <ec2-3-86-193-164.compute-1.amazonaws.com> (copy)

**Create the Project:**

create directory

mkdir vm-monitor

copy ansible.cfg and inventory into vm-monitor directory

mkdir /home/ubuntu/vm-monitor/group\_vars

vi /home/ubuntu/vm-monitor/group\_vars/all.yaml

smpt-server: “smtp.gmail.com”

smtp\_port: 587

email\_user: “yourname@gmail.com”

email\_pass: “emailpassword” <create app password in google account https://myaccount.google.com/u/2/apppasswords >

alert\_recipient: “other email address”

**vi /home/ubuntu/vm-monitor/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}'" (alternative <mpstat 1 1 | grep -i average | awk '{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) }}"

**vi /home/ubuntu/vm-monitor/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

**mkdir templates**

**vi /home/ubuntu/vm-monitor/templates/report\_email\_animated.html.j2**

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<style>

body {

font-family: 'Segoe UI', sans-serif;

background-color: #f9fbfc;

padding: 32px;

color: #333;

max-width: 960px;

margin: auto;

}

h2 {

text-align: center;

color: #2e7d32;

font-size: 28px;

margin-bottom: 32px;

border-bottom: 2px solid #c8e6c9;

padding-bottom: 10px;

}

.summary {

text-align: center;

margin-bottom: 30px;

background-color: #e3f2fd;

border: 1px solid #bbdefb;

padding: 10px 18px;

border-radius: 8px;

font-size: 14px;

color: #0d47a1;

}

table {

width: 100%;

border-collapse: collapse;

box-shadow: 0 4px 12px rgba(0,0,0,0.08);

background-color: #ffffff;

border-radius: 10px;

overflow: hidden;

}

th {

background-color: #1565c0;

color: white;

padding: 14px;

font-size: 14px;

text-align: center;

}

td {

padding: 14px;

font-size: 13px;

text-align: center;

border-bottom: 1px solid #f0f0f0;

}

tr:hover td {

background-color: #f5faff;

}

.hostname a {

color: #1565c0;

text-decoration: none;

font-weight: bold;

}

.bar-container {

width: 100%;

background-color: #eee;

border-radius: 6px;

height: 10px;

overflow: hidden;

margin-top: 4px;

}

.bar {

height: 100%;

border-radius: 6px;

}

.cpu-bar { background-color: #ef5350; }

.mem-bar { background-color: #42a5f5; }

.disk-bar { background-color: #66bb6a; }

.label {

margin-top: 4px;

font-size: 12px;

color: #555;

}

.badge {

font-size: 11px;

padding: 3px 8px;

border-radius: 12px;

display: inline-block;

font-weight: 600;

color: #fff;

}

.healthy { background-color: #43a047; }

.warning { background-color: #fbc02d; color: #000; }

.critical { background-color: #d32f2f; }

.footer {

text-align: center;

font-size: 12px;

margin-top: 24px;

color: #666;

}

</style>

</head>

<body>

<h2>📊 Consolidated VM Health Report</h2>

<div class="summary">

📅 <b>{{ collected\_metrics | length }} VMs</b> | 🔥 Avg CPU: {{ ((collected\_metrics | map(attribute='cpu') | map('float') | sum | float) / (collected\_metrics | length)) | round(2) }}% | 📥 Avg Mem: {{ ((collected\_metrics | map(attribute='mem') | map('float') | sum | float) / (collected\_metrics | length)) | round(2) }}% | 📦 Avg Disk: {{ ((collected\_metrics | map(attribute='disk') | map('float') | sum | float) / (collected\_metrics | length)) | round(2) }}%

</div>

<table>

<tr>

<th>🌐 Hostname</th>

<th>🔥 CPU Usage</th>

<th>📥 Memory Usage</th>

<th>📦 Disk Usage</th>

</tr>

{% for vm in collected\_metrics %}

<tr>

<td class="hostname">

<a href="http://{{ vm.hostname }}" target="\_blank">{{ vm.hostname }}</a>

</td>

<td>

{{ vm.cpu }}%

<div class="bar-container"><div class="bar cpu-bar" style="width: {{ vm.cpu }}%"></div></div>

{% if vm.cpu|float < 50 %}

<div class="badge healthy">Healthy</div>

<div class="label">Low</div>

{% elif vm.cpu|float < 80 %}

<div class="badge warning">Warning</div>

<div class="label">Moderate</div>

{% else %}

<div class="badge critical">Critical</div>

<div class="label">High</div>

{% endif %}

</td>

<td>

{{ vm.mem }}%

<div class="bar-container"><div class="bar mem-bar" style="width: {{ vm.mem }}%"></div></div>

{% if vm.mem|float < 50 %}<div class="label">Low</div>

{% elif vm.mem|float < 80 %}<div class="label">Moderate</div>

{% else %}<div class="label">High</div>{% endif %}

</td>

<td>

{{ vm.disk }}%

<div class="bar-container"><div class="bar disk-bar" style="width: {{ vm.disk }}%"></div></div>

{% if vm.disk|float < 50 %}<div class="label">Ample</div>

{% elif vm.disk|float < 80 %}<div class="label">Monitor</div>

{% else %}<div class="label">Full</div>{% endif %}

</td>

</tr>

{% endfor %}

</table>

<div class="footer">

⏱️ Report Generated on: {{ timestamp }}

</div>

</body>

</html>

**Run ansible playbook :**

ansible-playbook playbook.yaml

**Create the Project & Run below Command to execute**

Project Repo: https://github.com/jaiswaladi246/Ansible-VM-Monitor.git

ansible-playbook playbook.yaml