**Ansible Project To Monitor VMs Health**

**Instance Setup**

Name: Server

Type: Ubuntu

Instance Type: t2.medium

Key Pair: Roopam(create)

SG: 22,587, 80, 443, 3000-9000,8000

Storage: 25

Connect the instance

$sudo apt update

Git clone

**Worker Nodes**

Number of Instance: 10

Name and tags:

Key: Web

Additional tags:

Key: environment

Value: dev

Type: Ubuntu

Instance Type: t2.small

Key Pair: Roopam(create)

SG: 22,587, 80, 443, 3000-9000

Storage: 15

**Install Ansible on Master**

**🔹 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

**# 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

$mkdir inventory

**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

**Create an virtual environment to isolate all ansible related dependencies**

# 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

**To show the list of IP address**

ansible-galaxy collection install amazon.aws

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

**Copy the pem file inside the virtual env we created**

vi Ansible-VM.pem

copy and paste the key

sudo chmod 400 Ansible-VM.pem

**ansible.cfg**

vi ansible.cfg

[defaults]

inventory = ./inventory/aws\_ec2.yaml

host\_key\_checking = False

[ssh\_connection]

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

**Copy Pub Key**

vi copy-public-key.sh

#!/bin/bash

# Define vars

PEM\_FILE=" Ansible-VM.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

mkdir vm-monitor

cd vm-monitor