Ansible – Dynamic – Inventory

Step1 – Create a VM instance (Master-node) With Compute-Admin permission to Service Account and Ubuntu as OS

Step2 – Login to VM Instance SSH Terminal and create SSH keys.

Follow the below commands for the process

- ssh-keygen (press - enter – enter – enter – enter)
- cd .ssh

Step3 – Copy the Public key (id_rsa.pub) and paste it in Project level Metadata: SSH keys.

Run the below commands:

- cat ida_rsa.pub
- Goto GCP console Compute Engine –> Metadata –> Click SSH-Key –> Edit –> Add item and paste SSH Public key –> Save.

Step4 – Copy and Paste Private-key (id_rsa) in local (/root/key.pem)

Run the below commands:

- cat id_rsa (copy the private key)
- cd
- Vim key.pem (paste the key here)
- chmod 600 key.pem

Step5 – Install ansible

- sudo apt-get update
- sudo apt-get install software-properties-common
- sudo add-apt-repository --yes --update ppa:ansible/ansible
- sudo apt install ansible –y
- ansible --version

Step6 – install pip and apache-libcloud

Run these commands manually:

- sudo apt-get install python3-pip -y
- pip3 install apache-libcloud

Step7 – Generate the Service Account .JSON key and paste it in Local (/root/key.json)

- Root: vim key.json (paste the JSON key here)

Step8 – Run the Following Commands and get gce.py (file to get GCP compute engine information) and gce.ini (inventory file)

- wget https://raw.githubusercontent.com/ansible/ansible/stable-2.8/contrib/inventory/gce.py
- wget https://raw.githubusercontent.com/ansible/ansible/stable-2.8/contrib/inventory/gce.ini

Note: You will get gce.py and gce.ini files in local (/root/)

Step9 – In gce.py file change the first line as below

- Vim gce.py (root)
- #!/usr/bin/env python3

Step10 – Copy and paste the below content in gce.ini file and modify the required data.

- vim gce.ini (delete the content in it and paste the below content)

```
[gce]

# GCE Service Account configuration information

gce_service_account_email_address = your-service-account-email@your-project-
id.iam.gserviceaccount.com # CHANGE THIS

gce_service_account_pem_file_path = /path/to/your/service-account-key.json # CHANGE THIS

gce_project_id = your-project-id # CHANGE THIS

gce_zone = your-gce-zone # CHANGE THIS

# Filter inventory based on state (optional)

# instance_states = RUNNING,PROVISIONING

# Filter inventory based on instance tags (optional)

# instance_tags = http-server,https-server

[inventory]

# Specify whether 'ansible_ssh_host' should contain the instance internal or external address

# Values: 'internal' or 'external' (optional)

# inventory_ip_type =

[cache]

# Directory in which cache should be created

cache_path = ~/.ansible/tmp

# The number of seconds a cache file is considered valid. After this many

# seconds, a new API call will be made, and the cache file will be updated.

# To disable the cache, set this value to 0

cache_max_age = 300
```

Step11 – Replace the hosts file content with gce.py content.

Run below commands one by one:

- sudo mv gce.py /etc/ansible/hosts

Step12 – Move the gce.ini (inventory file) to /etc/ansible

- sudo mv gce.ini /etc/ansible

Step13 – Run the below commands:

- eval "$(ssh-agent -s)"

# used to start the SSH agent in Unix-like operating systems

- ssh-keygen -p -m PEM -f /Path/to/key.pem (/root/key.pem)

- -p = used to change the passphrase of an existing private key,
- -m = used to specify the format for the key file (-m PEM to ensure the key is in PEM format),
- -f = file name of the key file you want to operate on.

- export ANSIBLE_HOST_KEY_CHECKING=False

Checking if able to connect to all the instances:

- ansible all -m ping --private-key=/Path/to/key.pem (/root/key.pem)

NOTE: This method is most used for frequently Scalable VM instances.

Method2: Ansible Dynamic Inventory Set-Up

2a – Using Cloud Shell as a Master Node:

Step1 – Login to Cloud Shell SSH Terminal and create SSH keys.

Follow the below commands for the process

- ssh-keygen (press - enter – enter – enter – enter)
- cd .ssh

Step2 – Copy the Public key (id_rsa.pub) and paste it in Project level Metadata: SSH keys.

Run the below commands:

- cat ida_rsa.pub
- Goto GCP console Compute Engine –> Metadata –> Click SSH-Key –> Edit –> Add item and paste SSH Public key –> Save.

Step3 – Install ansible

- sudo apt-get update
- sudo apt-get install software-properties-common
- sudo add-apt-repository --yes --update ppa:ansible/ansible
- sudo apt install ansible –y
- ansible --version

Step4 – Create a Service account with Compute Admin Permission

Step5 – Generate the Service Account .JSON key and paste it in Local (/root/key.json)

- Root: vim key.json (paste the JSON key here)

Step6 – Create a File gcp.yml and write the below command.

- vim gcp.yml

```
plugin: gcp_compute

projects:

 - $YOUR_PROJECT_ID

filters: []

auth_kind: serviceaccount

service_account_file: /root/keys.json  #Note:Make sure to a dd the correct path.
```

Step7 – Add the below content in ansible.cfg file

- vim /etc/ansible/ansible.cfg

```
[inventory]

enable_plugins = gcp_compute

[defaults]

inventory = /Path/to/gcp.yml

#/root/gcp.yml

[ssh_connection]

ssh_args = -o strictHostKeyChecking=no
```

Step8 – Create 2 VM instances with Ubuntu as OS.

Checking if inventory is working properly:

- ansible-inventory gcp.yml --list

Checking if able to connect to all the instances:

- `ansible -i gcp.yaml all -m ping`

2b – Using VM – Instance as a Master Node:

Step1 – Create a VM instance (Master-node) With Compute-Admin permission to Service Account and Ubuntu as OS

Step2 – Login to VM Instance SSH Terminal and create SSH keys.

Follow the below commands for the process

- ssh-keygen (press - enter – enter – enter – enter)
- cd .ssh

Step3 – Copy the Public key (id_rsa.pub) and paste it in Project level Metadata: SSH keys.

Run the below commands:

- cat ida_rsa.pub
- Goto GCP console Compute Engine –> Metadata –> Click SSH-Key –> Edit –> Add item and paste SSH Public key –> Save.

Step4 – Copy and Paste Private-key (id_rsa) in local (/root/key.pem)

Run the below commands:

- cat id_rsa (copy the private key)
- cd
- Vim key.pem (paste the key here)
- chmod 600 key.pem

Step5 – Install ansible

- sudo apt-get update
- sudo apt-get install software-properties-common
- sudo add-apt-repository --yes --update ppa:ansible/ansible
- sudo apt install ansible –y
- ansible --version

Step6 – Installing "google-auth" Python library:

Run the below commands one by one

- sudo apt install python3-pip -y
- pip install google-auth
- pip install google-auth-oauthlib
- pip install google-auth-httplib2
- pip install google-api-python-client

Step7 – Create a File gcp.yml and write the below command.

vim gcp.yml

```
plugin: gcp_compute
projects:
 - $YOUR_PROJECT_ID
filters: []
auth_kind: serviceaccount
service_account_file: /root/keys.json  #Note: Make sure to a dd the correct path.
```

Step8 – Add the below content in ansible.cfg file

- vim /etc/ansible/ansible.cfg

```
[inventory]
enable_plugins = gcp_compute
[defaults]
inventory = /Path/to/gcp.yml
#/root/gcp.yml
[ssh_connection]
ssh_args = -o strictHostKeyChecking=no
```

Step9 – Generate the VM Instance - Service Account .JSON key and paste it in Local (/root/key.json)

- Root: vim key.json (paste the JSON key here)

Step10 – Create 2 VM instances with Ubuntu as OS.

Checking if inventory is working properly:

- ansible-inventory gcp.yml --list

Checking connection:

- ansible all --private-key=/root/key.pem -m ping