

HW1 Ansible

- Utilize the Dcloud DevNet Express Data Center v2 lab, or deploy your own three (3) Virtual Machines
- Configure Ansible server on **VM 1** to deploy a web server to **VM2** and **VM3** on port 8080 that displays the message: "Hello World from SJSU-X ", where X is 1 or 2 depending on which web server.
- Include in the Ansible playbook, plays to **deploy** and **undeploy** all the web server resources

GitHub Link: https://github.com/SaiPranaviKurapati/HW1_Ansible_TeamAlterEgos

Team AlterEgos

- Divija Choudhary
- Dhrithi Vishwanath Gulannavar
- Vaibhavi Hiteshkumar Savani
- Kurapati Sai Pranavi

Creating an instance in AWS

We have created three VMs with a Linux operating system:

- VM1: 172.31.3.177 (Server1)
- VM2: 172.31.1.16 (Server2)
- VM3: 172.31.8.87 (AnsibleServer)

During the VM creation process, we generated a key pair named 'testkeypair' and selected the 'launch wizard' security group.

- Creating an instance for VM1

The screenshot shows the 'Launch an instance' wizard in the AWS Management Console. The interface is divided into several sections:

- Name and tags:** A text box contains 'VM1'. There is a link 'Add additional tags'.
- Application and OS Images (Amazon Machine Image):** A search bar with the placeholder text 'Search our full catalog including 1000s of application and OS images'. Below the search bar are tabs for 'Recents' and 'Quick Start'. Under 'Quick Start', there are several AMI cards: Amazon Linux, Ubuntu, Windows, Red Hat, SUSE Linux, and Debian. A link 'Browse more AMIs' is also present.
- Summary:** A sidebar on the right containing:
 - Number of instances:** A dropdown menu set to '1'.
 - Software Image (AMI):** 'Amazon Linux 2023 AMI 2023.1.2...' with a 'read more' link and the ID 'ami-03f2f5212f24db70a'.
 - Virtual server type (instance type):** 't2.micro'.
 - Firewall (security group):** 'New security group'.
 - Storage (volumes):** '1 volume(s) - 8 GiB'.
- Free tier notification:** A blue box with a close button (X) stating: 'Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.'
- Buttons:** 'Cancel', 'Launch instance' (in orange), and 'Review commands'.

aws

Services

Search

[Alt+S]

▼ Instance type Info

Instance type

t2.micro

Family: t2 1 vCPU 1 GiB Memory Current generation: true
On-Demand RHEL base pricing: 0.0738 USD per Hour
On-Demand SUSE base pricing: 0.0138 USD per Hour
On-Demand Windows base pricing: 0.0184 USD per Hour
On-Demand Linux base pricing: 0.0138 USD per Hour

Free tier eligible

All generations

Compare instance types

Additional costs apply for AMIs with pre-installed software

▼ Key pair (login) Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - required

testkeypair

Create new key pair

▼ Network settings Info

Network Info

vpc-0115dc9c1b5a96b7b

Subnet Info

No preference (Default subnet in any availability zone)

Auto-assign public IP Info

Edit

▼ Summary

Number of instances Info

1

Software Image (AMI)

Amazon Linux 2023 AMI 2023.1.2...read more

ami-03f2f5212f24db70a

Virtual server type (instance type)

t2.micro

Firewall (security group)

New security group

Storage (volumes)

1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million IOs, 1 GiB of snapshots, and 100 GB of bandwidth to the internet.

Cancel

Launch instance

Review commands

aws

Services

Search

[Alt+S]

Key pair name - required

testkeypair

Create new key pair

▼ Network settings Info

Network Info

vpc-0115dc9c1b5a96b7b

Subnet Info

No preference (Default subnet in any availability zone)

Auto-assign public IP Info

Enable

Firewall (security groups) Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group

Select existing security group

Common security groups Info

Select security groups

launch-wizard-1 sg-02c9fc6e709190305

Compare security group rules

Security groups that you add or remove here will be added to or removed from all of your network interfaces.

▼ Configure storage Info

1x 8 GiB gp3

Root volume (Not encrypted)

Advanced

▼ Summary

Number of instances Info

1

Software Image (AMI)

Amazon Linux 2023 AMI 2023.1.2...read more

ami-03f2f5212f24db70a

Virtual server type (instance type)

t2.micro

Firewall (security group)

launch-wizard-1

Storage (volumes)

1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million IOs, 1 GiB of snapshots, and 100 GB of bandwidth to the internet.

Cancel

Launch instance

Review commands

Similarly, we created instances for VM2 and VM3.

Creating SSH key pair

SSH key-pairs are generated to establish communication between hosts and ansible server.

Generating SSH key pair :

ssh-keygen

```
[root@ip-172-31-11-45 ~]# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:vNM6IguvT+vGsYyVLGqg9yTz9MX7NC8oai/Zcp8OkTnc root@ip-172-31-11-45.us-west-1.compute.internal
The key's randomart image is:
---[RSA 3072]-----+
|
|.
|. S
| o. o o
| . +o=oE +o.
| .OoX*X.=o+o
|=o&X%*=o....
|
----[SHA256]-----+
[root@ip-172-31-11-45 ~]#
```

Copying the keys to the hosts :

```
ssh-copy-id ansible@172.31.1.16
```

```
[root@ip-172-31-11-45 .ssh]# ssh-copy-id ansible@172.31.1.16
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
The authenticity of host '172.31.1.16 (172.31.1.16)' can't be established.
ED25519 key fingerprint is SHA256:qi0KvM99y3yOq73W+37nHYRoAV8VCKUrpTaTuA//4H9E.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
ansible@172.31.1.16's password:

Number of key(s) added: 1

Now try logging into the machine, with:  "ssh 'ansible@172.31.1.16'"
and check to make sure that only the key(s) you wanted were added.
```

```
ssh-copy-id ansible@172.31.3.177
```

```

root@ip-172-31-11-45 .ssh)# ssh-copy-id ansible@172.31.3.177
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
The authenticity of host '172.31.3.177 (172.31.3.177)' can't be established.
ED25519 key fingerprint is SHA256:MlA7coNHr/4Zg/lemustHBVl7XblT9nz22AG9N4yUSE.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
ansible@172.31.3.177's password:

Number of key(s) added: 1

Now try logging into the machine, with:  "ssh 'ansible@172.31.3.177'"
and check to make sure that only the key(s) you wanted were added.

```

Verifying SSH

Verify that the keys has been copied :

```
ssh ansible@172.31.3.177
```

```
ansible@ip-172-31-8-87 ~]$ ssh ansible@172.31.3.177
#_
~\####_ Amazon Linux 2023
~~\_#####\
~~ \###|
~~ \|/ https://aws.amazon.com/linux/amazon-linux-2023
~~ V~'-'>
    ~~~
      ~~-.-
        /-/
       /m/'-
```

Last login: Wed Sep 6 22:05:54 2023 from 172.31.8.87

```
ansible@ip-172-31-3-177 ~]$ █
```

```
[ansible@ip-172-31-8-87 ~]$ ssh ansible@172.31.1.16
```

```
#_
~\   ###_      Amazon Linux 2023
~~ \_  #####\_
~~   \###|
~~    \#/      https://aws.amazon.com/linux/amazon-linux-2023
~~     v~' '->
~~~~
~~ . - 
~~ /m/' -
```

```
Last login: Thu Sep  7 04:17:27 2023 from 172.31.8.87
[ansible@ip-172-31-1-16 ~]$
```

Creating Ansible Inventory

The IP addresses of both servers, VM1 and VM2, are included in the Ansible server's hosts file. You can verify the connection to both VMs from the Ansible server using the following command.

Ansible all -m ping

```
[ansible@ip-172-31-8-87 ~]$ ansible all -m ping
WEBSERVER1 | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
WEBSERVER2 | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
```

Deploy the web server

After successfully establishing the connection, we create .yaml, .j2, and .html files. The .yaml file contains commands to:

- Install the Apache server.
- Launch the server after successful installation.
- Add firewall rules.
- Reload the firewall.
- Display the message

ansible-playbook httpd.yaml

```
[ansible@ip-172-31-8-87 ~]$ ansible-playbook httpd.yaml
PLAY [Configuring Apache server] *****

TASK [Gathering Facts] *****
ok: [VM1]
ok: [VM2]

TASK [Install Apache Server] *****
changed: [VM1]
changed: [VM2]

TASK [Start Apache Server] *****
changed: [VM1]
changed: [VM2]

TASK [Adding config file] *****
changed: [VM1]
changed: [VM2]

TASK [Copy index.html file on server] *****
changed: [VM2]
changed: [VM1]

TASK [Adding firewall rules] *****
changed: [VM1]
changed: [VM2]

TASK [reload firewall] *****
changed: [VM1]
changed: [VM2]

RUNNING HANDLER [server restart] *****
changed: [VM1]
changed: [VM2]

PLAY RECAP *****
VM1 : ok=8    changed=7    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
VM2 : ok=8    changed=7    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

Validation from workstation

Logging on to ansible server(VM3) on AWS server running the IP address will printout the below message after successful validation of curl command.

```
curl ansible@172.31.3.177:8080
```

```
curl ansible@172.31.1.16:8080
```

```
[ansible@ip-172-31-8-87 ~]$ curl ansible@172.31.3.177:8080
Hello World From SJSU- VM1
[ansible@ip-172-31-8-87 ~]$ curl ansible@172.31.1.16:8080
Hello World From SJSU- VM2
```

Un-deploy the web server

To un-deploy the web server, we uninstalled the Apache server on both servers and removed all its dependencies. After completing this task, we also disabled port 8080.

Ansible-playbook httpdundeploy.yaml

```
[ansible@ip-172-31-8-87 ~]$ ansible-playbook httpdundeploy.yaml
PLAY [deconfiguring Apache server] *****
TASK [Gathering Facts] *****
ok: [VM1]
ok: [VM2]
TASK [Stop Apache Server] *****
changed: [VM1]
changed: [VM2]
TASK [Uninstall Apache Server] *****
changed: [VM1]
changed: [VM2]
TASK [Remove config file] *****
changed: [VM1]
changed: [VM2]
TASK [remove index.html file on server] *****
changed: [VM1]
changed: [VM2]
TASK [block firewall port] *****
changed: [VM1]
changed: [VM2]
TASK [reload firewall] *****
changed: [VM1]
changed: [VM2]
PLAY RECAP *****
VM1      : ok=7    changed=6    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
VM2      : ok=7    changed=6    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

Validate after un-deployment

After un-deploying, we validate that the servers are no longer reachable using the following commands.

Curl ansible@172.31.1.16:8080

Curl ansible@172.31.3.177:8080

```
[ansible@ip-172-31-8-87 ~]$ curl ansible@172.31.1.16:8080
curl: (7) Failed to connect to 172.31.1.16 port 8080 after 0 ms: Couldn't connect to server
[ansible@ip-172-31-8-87 ~]$ curl ansible@172.31.3.177:8080
curl: (7) Failed to connect to 172.31.3.177 port 8080 after 0 ms: Couldn't connect to server
[ansible@ip-172-31-8-87 ~]$
```