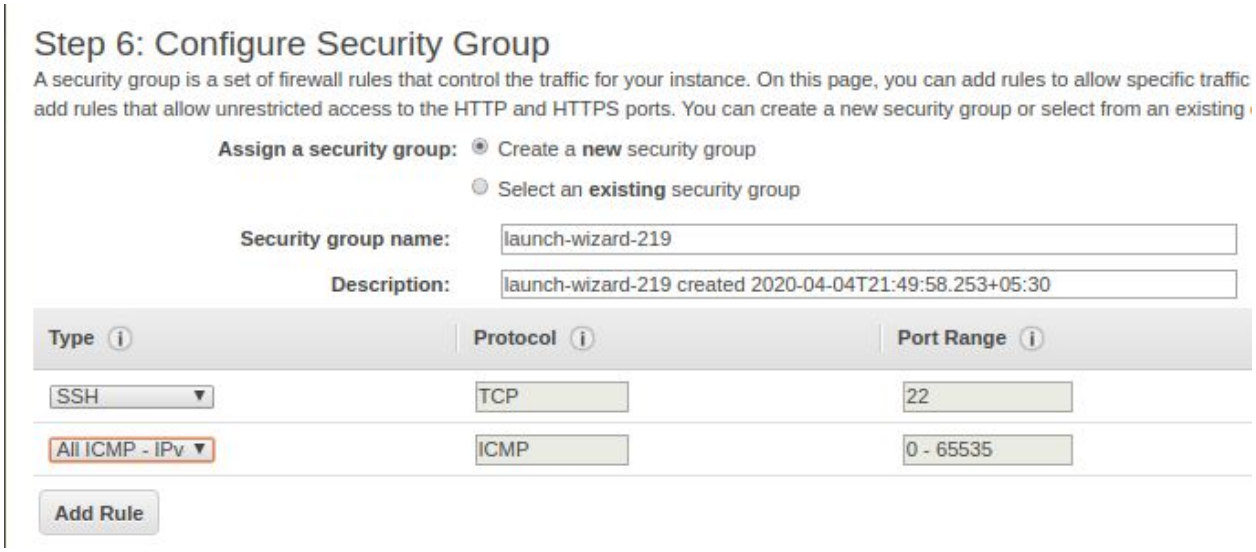
ASSESSMENT ON: ANSIBLE (2)

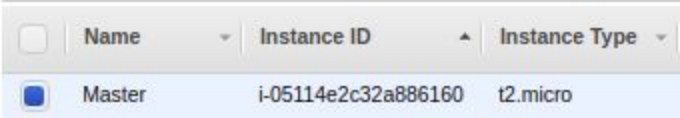
TO THE NEW

1. Create two nodes with tag:key role and tag:value master & slave respectively. Setup the dynamic inventory on ansible control nodes.

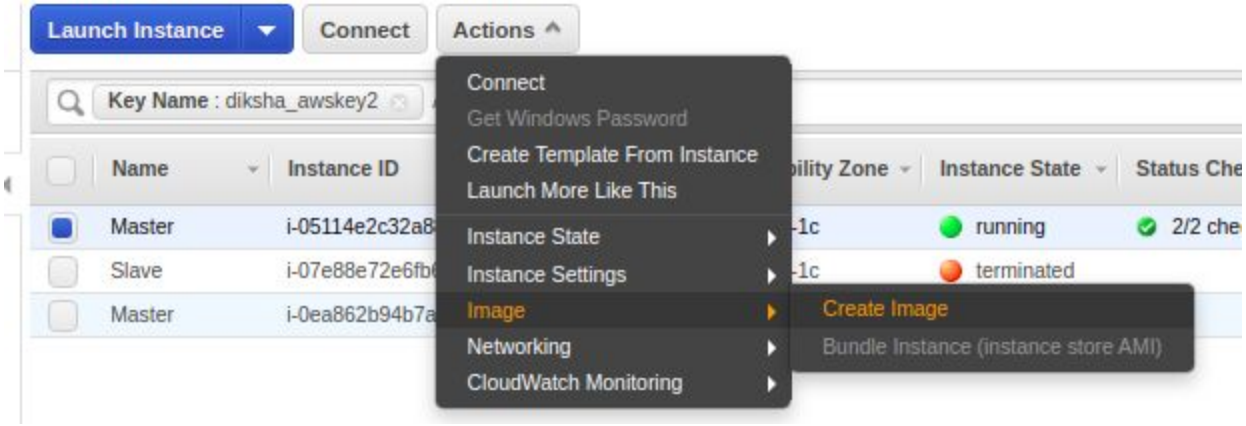
STEP 1: Launch an ec2 instance



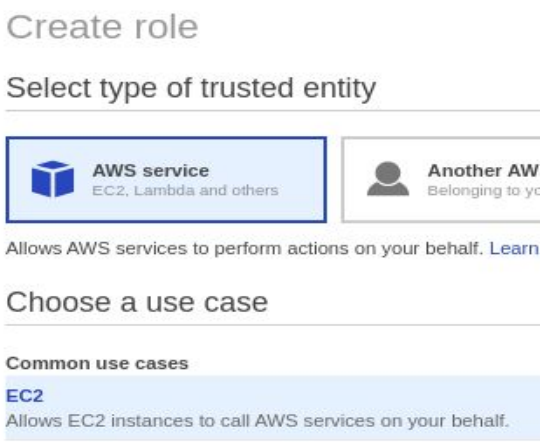
STEP 2: SSH into the instance and create a ssh key and add it to master and slaves authorised_key file.

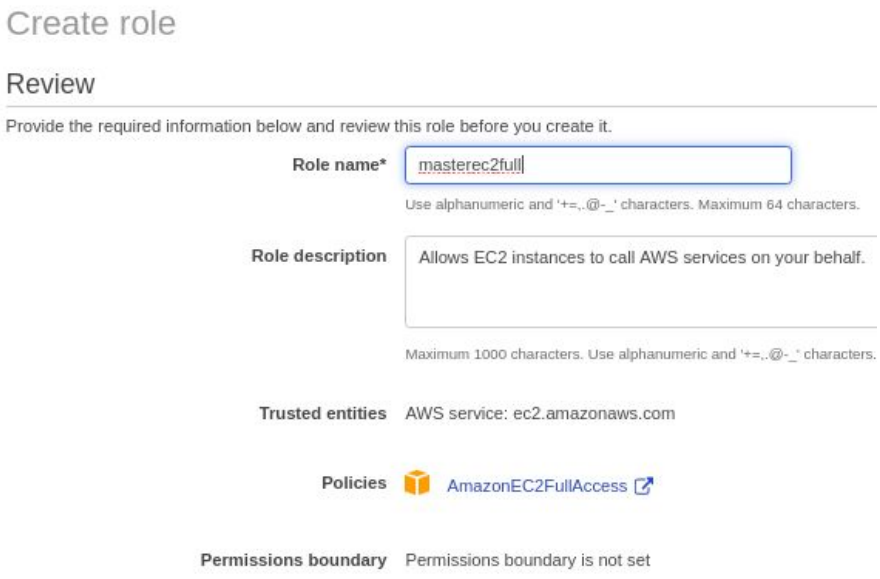


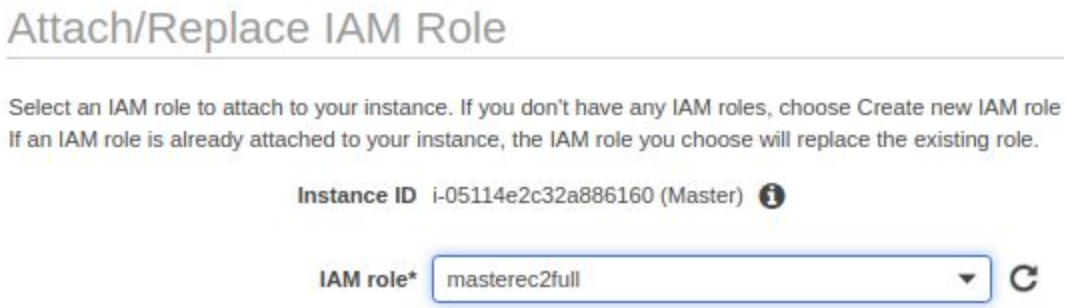
STEP 3: Now launch another instance from that ami.



<u>STEP 4:</u> Then create a role for ec2 full access and attach it to the ansible-master node.







STEP 5: Download the ec2.py and ec2.ini files and put them in /etc/ansible directory and give ec2.py the execute permissions.

```
ubuntu@ip-172-31-46-248:/etc/ansible$ ls
ansible.cfg  ec2.ini  ec2.py  hosts
ubuntu@ip-172-31-46-248:/etc/ansible$ sudo chmod +x ec2.py
```

STEP 6: Set RDS and Elasticache to false in ec2.ini

```
# To exclude RDS instanc
rds = False
elasticache = False
```

* Pre-requisites:

- **Boto** must be installed on the master and slave server to make AWS API Calls.
- Ansible servers should be able to do SSH on nodes.

Run the following modules using tag key-value:
1.1 Ping master node and slave node separately.

```
ubuntu@ip-172-31-46-248:/etc/ansible$ ansible -i ec2.py tag_role_master -m ping
 [ERROR]:

The authenticity of host '3.92.50.160 (3.92.50.160)' can't be established.
ECDSA key fingerprint is SHA256:egitsu1YsttgWR6bizjlvEOAflid0DamCl0M9gFxd4s.
Are you sure you want to continue connecting (yes/no)? 18.208.156.61 | SUCCESS =
> {
    "changed": false,
    "ping": "pong"
}
```

```
ubuntu@ip-172-31-46-248:/etc/ansible$ ansible -i ec2.py tag_role_slave -m ping
 [ERROR]:

3.92.225.200 | UNREACHABLE! => {
    "changed": false,
    "msg": "Failed to connect to the host via ssh: ubuntu@3.92.225.200: Permissi
on denied (publickey,gssapi-keyex,gssapi-with-mic).\r\n",
    "unreachable": true
}
3.83.42.122 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
```

1.2 To check all running processes on slave node.

```
ubuntu@ip-172-31-46-248:/etc/ansible$ ansible -i ec2.py tag_role_slave -m shell
-a "ps -aux"
 [ERROR]:

3.92.225.200 | UNREACHABLE! => {
    "changed": false,
    "msg": "Failed to connect to the host via ssh: ubuntu@3.92.225.200: Permissi
on denied (publickey,gssapi-keyex,gssapi-with-mic).\r\n",
    "unreachable": true
}
3.83.42.122 | SUCCESS | rc=0 >>
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.9 225408  9220 ?        Ss   16:27   0:05 /lib/systemd/sy
stemd --system --deserialize 19
root         2  0.0  0.0      0     0 ?        S    16:27   0:00 [kthreadd]
root         4  0.0  0.0      0     0 ?        I<   16:27   0:00 [kworker/0:0H]
root         6  0.0  0.0      0     0 ?        I<   16:27   0:00 [mm_percpu_wq]
root         7  0.0  0.0      0     0 ?        S    16:27   0:00 [ksoftirqd/0]
root         8  0.0  0.0      0     0 ?        I    16:27   0:00 [rcu_sched]
root         9  0.0  0.0      0     0 ?        I    16:27   0:00 [rcu_bh]
root        10  0.0  0.0      0     0 ?        S    16:27   0:00 [migration/0]
root        11  0.0  0.0      0     0 ?        S    16:27   0:00 [watchdog/0]
```

1.3 To copying files to both nodes concurrently.

```
ubuntu@ip-172-31-46-248:/etc/ansible$ ansible -i ec2.py ec2 -m copy -a "src=/etc
/hosts dest=/home/ubuntu"
```

```
18.208.156.61 | SUCCESS => {
    "changed": true,
    "checksum": "a777be51c79c607edd61c8e12cd9b775ddc8a6c6",
    "dest": "/home/ubuntu/hosts",
    "gid": 1000,
    "group": "ubuntu",
    "md5sum": "1463508f28edb4d6d5ae349b20e00409",
    "mode": "0664",
    "owner": "ubuntu",
    "size": 221,
    "src": "/home/ubuntu/.ansible/tmp/ansible-tmp-1586024595.55-6373230439234/so
urce",
    "state": "file",
    "uid": 1000
}
```

```
3.83.42.122 | SUCCESS => {
    "changed": true,
    "checksum": "a777be51c79c607edd61c8e12cd9b775ddc8a6c6",
    "dest": "/home/ubuntu/hosts",
    "gid": 1000,
    "group": "ubuntu",
    "md5sum": "1463508f28edb4d6d5ae349b20e00409",
    "mode": "0664",
    "owner": "ubuntu",
    "size": 221,
    "src": "/home/ubuntu/.ansible/tmp/ansible-tmp-1586024597.19-203267895781573/
source",
    "state": "file",
    "uid": 1000
}
```

```
ubuntu@ip-172-31-46-248:~$ ls
hosts
ubuntu@ip-172-31-46-248:~$
```

```
ubuntu@ip-172-31-218-181:~$ ls
hosts
ubuntu@ip-172-31-218-181:~$
```

2. Setup nginx on both nodes with a single custom configuration template, on master nginx should run on 8000 while on slave nginx would listen on port 80. [Jinja2+conditional]

STEP 1: Create a role named “custom”

```
ubuntu@ip-172-31-46-248:~$ cd /etc/ansible/
ubuntu@ip-172-31-46-248:/etc/ansible$ sudo mkdir roles
ubuntu@ip-172-31-46-248:/etc/ansible$ cd roles/
ubuntu@ip-172-31-46-248:/etc/ansible/roles$ ls
ubuntu@ip-172-31-46-248:/etc/ansible/roles$ sudo ansible-galaxy init /etc/ansibl
e/roles/custom
- /etc/ansible/roles/custom was created successfully
ubuntu@ip-172-31-46-248:/etc/ansible/roles$
```

```
ubuntu@ip-172-31-46-248:/etc/ansible/roles$ tree custom/
custom/
├── README.md
├── defaults
│   └── main.yml
├── files
├── handlers
│   └── main.yml
├── meta
│   └── main.yml
├── tasks
│   └── main.yml
├── templates
├── tests
│   ├── inventory
│   └── test.yml
└── vars
    └── main.yml

8 directories, 8 files
```

STEP 2: Create a template for nginx configuration file

```
ubuntu@ip-172-31-46-248:/etc/ansible/roles$ cat custom/templates/nginx_temp.j2
{% if ec2_tag_role == 'master' %}
server{
        listen 8000;
        server_name _;
        index index.nginx-debian.html index.html;
        root /var/www/html;
}
{% else %}
server{
        listen 80;
        server_name _;
        index index.nginx-debian.html index.html;
        root /var/www/html;
}
{% endif %}
```

```
ubuntu@ip-172-31-46-248:/etc/ansible/roles$ cat custom/tasks/main.yml
---
# tasks file for /etc/ansible/roles/custom
- name: Updating apt
  apt:
          update_cache: yes
- name: Installing nginx
  apt:
          name: nginx
          state: latest
- name: Copy custom conf files (use template)
  templates:
          src: nginx_temp.j2
          dest: /etc/nginx/sites-available/nginx
- name: Remove unnecessary file in sites-enabled
  file:
          state: absent
          path: "/etc/nginx/sites-enabled/default"
- name: Creating soft link
  raw: test -L /etc/nginx/sites-enabled/nginx || ln -s /etc/nginx/sites-available/nginx /etc/nginx/sites-enabled/nginx

- name: Reloading service
  service:
          name: nginx
          state: reloaded
```

STEP 3: Create Playbook

```
ubuntu@ip-172-31-46-248:/etc/ansible/roles$ cat playbook.yml
- name: PLAYBOOK
  hosts: ec2
  become: yes
  gather_facts: yes
  roles:
```

STEP 4: Run playbook

```
ubuntu@ip-172-31-46-248:/etc/ansible/roles$ ansible-playbook -i /etc/ansible/ec2.py playbook.yml
[ERROR]:


PLAY [PLAYBOOK] ***************************************************************************************

TASK [Gathering Facts] ********************************************************************************
The authenticity of host '3.83.220.82 (3.83.220.82)' can't be established.
ECDSA key fingerprint is SHA256:1yl6O8UMss5W4Vz8kKsgZcIQHGsEJLKtmn89XmN39Ho.
Are you sure you want to continue connecting (yes/no)? ok: [3.83.42.122]
ok: [18.208.156.61]
```

```
TASK [custom : Updating apt] ***********************************************************************************
changed: [3.83.42.122]
changed: [18.208.156.61]

TASK [custom : Installing nginx] *******************************************************************************
changed: [3.83.42.122]
changed: [18.208.156.61]

TASK [custom : Copy custom conf files (use template)] ******************************************************
changed: [3.83.42.122]
changed: [18.208.156.61]

TASK [custom : Remove unnecessary file in sites-enabled] ***************************************************
changed: [3.83.42.122]
changed: [18.208.156.61]

TASK [custom : Creating soft link] *****************************************************************************
changed: [18.208.156.61]
changed: [3.83.42.122]

TASK [custom : Reloading service] ******************************************************************************
 [WARNING]: Consider using the service module rather than running service.  If you need to use command because service
is insufficient you can add warn=False to this command task or set command_warnings=False in ansible.cfg to get rid of
this message.

changed: [3.83.42.122]
changed: [18.208.156.61]
 [WARNING]: Could not create retry file '/etc/ansible/roles/playbook.retry'.         [Errno 13] Permission denied:
u'/etc/ansible/roles/playbook.retry'


PLAY RECAP *****************************************************************************************************
18.208.156.61              : ok=7    changed=6    unreachable=0    failed=0
3.83.220.82                : ok=0    changed=0    unreachable=1    failed=0
3.83.42.122                : ok=7    changed=6    unreachable=0    failed=0
```

STEP 5: Check for file on master node

```
ubuntu@ip-172-31-46-248:/etc/ansible/roles$ cd ../../nginx/sites-enabled/
ubuntu@ip-172-31-46-248:/etc/nginx/sites-enabled$ ls
nginx
ubuntu@ip-172-31-46-248:/etc/nginx/sites-enabled$ cat nginx
server{
        listen 8000;
        server_name _;
        index index.nginx-debian.html index.html;
        root /var/www/html;
}
```

Welcome to nginx!

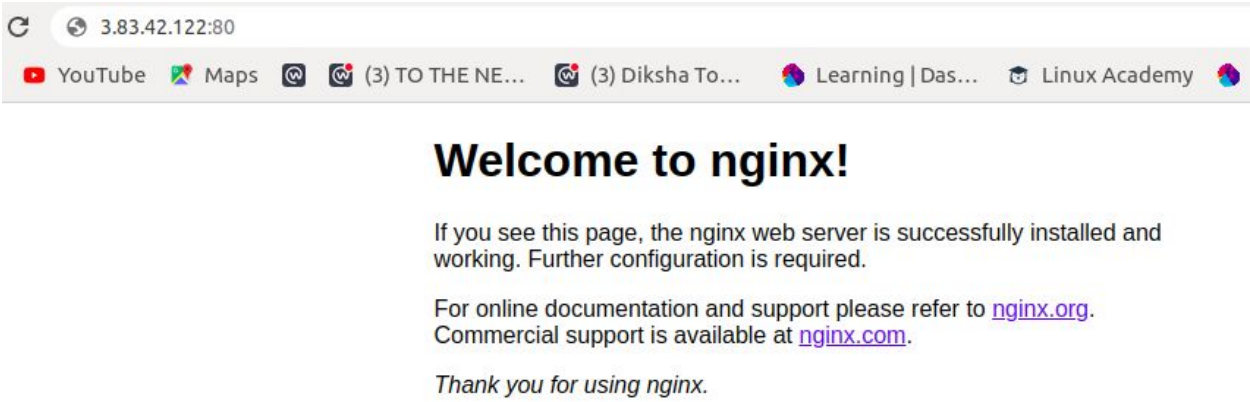
If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

STEP 6: Similarly on slave node

```
ubuntu@ip-172-31-218-181:~$ cd /etc/nginx/sites-enabled
ubuntu@ip-172-31-218-181:/etc/nginx/sites-enabled$ ls
nginx
ubuntu@ip-172-31-218-181:/etc/nginx/sites-enabled$ cat nginx
server{
        listen 80;
        server_name _;
        index index.nginx-debian.html index.html;
        root /var/www/html;
}
```



3. Setup mysql on a remote server, create a user with password. Passwords should be encrypted using Ansible vault. Verify the setup by log in to mysql.

STEP 1: Create password

```
ubuntu@ip-172-31-46-248:/etc/ansible$ sudo ansible-vault encrypt_string diksha
New Vault password:
Confirm New Vault password:
!vault |
          $ANSIBLE_VAULT;1.1;AES256
          36356339616139623230646264653934653333353635323335366362363935376566306532613263
          34643361373365316665383931656434616165303034626 60a633436316339653664633566303562
          6434626133653338353463383433656334366332386234613938366236313263353762333864653 2
          6464613464393961390a66323335633730643038623034363463333661666332333134626132316 5
          3262
Encryption successful
```

STEP 2: Make entry of both master and slave in hosts file

```
ubuntu@ip-172-31-46-248: /etc/ansible
File  Edit  View  Search  Terminal  Tabs  Help
ubuntu@ip-172-31-46-248: /etc...  ×   ubuntu@ip-172-31-218-181: /et...  ×   diksha@diksha: /etc/ansible/
# This is the default ansible 'hosts' file.
#
# It should live in /etc/ansible/hosts
#
#   - Comments begin with the '#' character
#   - Blank lines are ignored
#   - Groups of hosts are delimited by [header] elements
#   - You can enter hostnames or ip addresses
#   - A hostname/ip can be a member of multiple groups

# Ex 1: Ungrouped hosts, specify before any group headers.
ubuntu@18.208.156.61
ubuntu@3.83.42.122
```

STEP 3: Create a playbook

```
ubuntu@ip-172-31-46-248:/etc/ansible$ cat mysqlplaybook.yml
- hosts: all
  gather_facts: no
  become: yes
  vars:
          db_password: !vault |
          $ANSIBLE_VAULT;1.1;AES256
          36356339616139623230646264653934653333353635323335366362363935376566306532613263
          34643361373365316665383931656434616165303034626 60a633436316339653664633566303562
          64346261336533383534633834336563343663323862346139383662363132633537623338646532
          64646134643939613 90a663233356337306430386230343634633336616663323331346261323165
          3262
  tasks:
          - name:  Installing mysql
            apt:
                    name: mysql-server
                    state: present
                    update-cache: yes

          - name : Installing dependencies for db
            apt:
                    name: python2.7-mysqldb
                    state: present
                    update-cache: yes

          - name: wait 10 sec
            pause:
                    seconds: 10

          - name: create db user
            mysql_user:
                    name: wordpress
                    password: '{{ db_password }}'
                    priv: '*.*:ALL'
                    state: present
```

STEP 4: Run playbook

```
ubuntu@ip-172-31-46-248:/etc/ansible$ ansible-playbook  mysqlplaybook.yml  --ask-vault-pass

Vault password:

PLAY [all] *********************************************************************

TASK [Installing mysql] ********************************************************
changed: [ubuntu@18.208.156.61]
changed: [ubuntu@3.83.42.122]

TASK [Installing dependencies for db] ******************************************
changed: [ubuntu@18.208.156.61]
changed: [ubuntu@3.83.42.122]

TASK [wait 10 sec] *************************************************************
Pausing for 10 seconds
(ctrl+C then 'C' = continue early, ctrl+C then 'A' = abort)
ok: [ubuntu@18.208.156.61]

TASK [create db user] **********************************************************
changed: [ubuntu@18.208.156.61]
changed: [ubuntu@3.83.42.122]

PLAY RECAP *********************************************************************
ubuntu@18.208.156.61       : ok=4    changed=3    unreachable=0    failed=0
ubuntu@3.83.42.122         : ok=3    changed=3    unreachable=0    failed=0
```

STEP 5: Verify the setup by log in to mysql.

```
ubuntu@ip-172-31-46-248:~$ sudo mysql -u wordpress -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3
Server version: 5.7.29-0ubuntu0.18.04.1 (Ubuntu)

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```