Optional Assignment 3 - Setup a webserver and load balancer with Ansible

Muhannad Sinan

> In this assignment, we are going to create a playbook to deploy a web application hosted on two servers and a load balancer in between to manage the load.
>
> Steps consist of three parts:
>
> 1. *Provisioning (Package Management):* Install all the packages required on the server
> 2. *Configure Infrastructure:* Configure our system with necessary files or configuration files needed to configure system.
> 3. *Service Handler:* Create a service handler to start, stop, and restart our system when changes are done.

After creating four EC2 instances, one Linux server as an Ansible master, two Linux web servers and one Linux web server as a load balancer using this CloudFormation template: `loadbalanced.yml`

```
AWSTemplateFormatVersion: "2010-09-09"
Metadata:
  License: Apache-2.0
Description: "Optional Assignment 3 - Setup a webserver and load balancer with
Ansible"
Parameters:
  KeyName:
    Description: Name of an existing EC2 KeyPair to enable SSH access to the
instance
    Type: AWS::EC2::KeyPair::KeyName
    Default: main
    ConstraintDescription: must be the name of an existing EC2 KeyPair.
  InstanceType:
    Description: WebServer EC2 instance type
    Type: String
    Default: t3.micro
    AllowedValues: [t3.nano, t3.micro, t3.small, t3.medium]
    ConstraintDescription: must be a valid EC2 instance type.
  SSHLocation:
    Description: The IP address range that can be used to SSH to the EC2 instances
    Type: String
    MinLength: 9
```

```
      MaxLength: 18
      Default: 0.0.0.0/0
      AllowedPattern: (\d{1,3})\.(\d{1,3})\.(\d{1,3})\.(\d{1,3})/(\d{1,2})
      ConstraintDescription: must be a valid IP CIDR range of the form x.x.x.x/x.
    HTTPLocation:
      Description: The IP address range that can be used to SSH to the EC2 instances
      Type: String
      MinLength: 9
      MaxLength: 18
      Default: 0.0.0.0/0
      AllowedPattern: (\d{1,3})\.(\d{1,3})\.(\d{1,3})\.(\d{1,3})/(\d{1,2})
      ConstraintDescription: must be a valid IP CIDR range of the form x.x.x.x/x.
    LatestAmzLinuxAmiId:
      Type: "AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>"
      Default: "/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2"
    LatestUbuntuAmiId:
      Type:  'AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>'
      Default:
  '/aws/service/canonical/ubuntu/server/20.04/stable/current/amd64/hvm/ebs-gp2/ami-
  id'
    InstanceCount:
      Description: Number of EC2 instances (must be between 1 and 5).
      Type: Number
      Default: 1
      MinValue: 1
      MaxValue: 5
      ConstraintDescription: Must be a number between 1 and 5.

  Resources:
    MasterSecurityGroup:
      Type: AWS::EC2::SecurityGroup
      Properties:
        GroupName: ansbile-master-sg
        GroupDescription: Enable SSH access via port 22
        SecurityGroupIngress:
          - IpProtocol: tcp
            FromPort: 22
            ToPort: 22
            CidrIp: !Ref "SSHLocation"

    NodeSecurityGroup:
      Type: AWS::EC2::SecurityGroup
      Properties:
        GroupName: ansbile-node-sg
        GroupDescription: Enable SSH access via port 22 and HTTP access via port 80
        SecurityGroupIngress:
          - IpProtocol: tcp
            FromPort: 22
            ToPort: 22
            CidrIp: !Ref "SSHLocation"
```

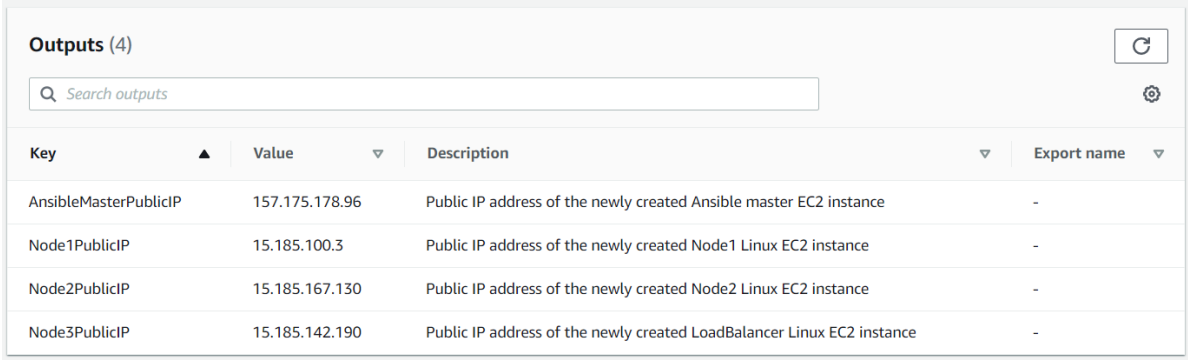
```
        - IpProtocol: tcp
          FromPort: 80
          ToPort: 80
          CidrIp: !Ref "HTTPLocation"

  EC2InstanceMaster:
    Type: AWS::EC2::Instance
    Properties:
      InstanceType: !Ref "InstanceType"
      SecurityGroups: [!Ref "MasterSecurityGroup"]
      KeyName: !Ref "KeyName"
      ImageId: !Ref "LatestAmzLinuxAmiId"
      UserData:
        Fn::Base64: !Sub |
          #!/bin/bash
          sudo yum update -y
          sudo amazon-linux-extras install ansible2 -y
      Tags:
        - Key: Name
          Value: ansible-master
  EC2InstanceNode1:
    Type: AWS::EC2::Instance
    Properties:
      InstanceType: !Ref "InstanceType"
      SecurityGroups: [!Ref "NodeSecurityGroup"]
      KeyName: !Ref "KeyName"
      ImageId: !Ref "LatestUbuntuAmiId"
      Tags:
        - Key: Name
          Value: Node1
  EC2InstanceNode2:
    Type: AWS::EC2::Instance
    Properties:
      InstanceType: !Ref "InstanceType"
      SecurityGroups: [!Ref "NodeSecurityGroup"]
      KeyName: !Ref "KeyName"
      ImageId: !Ref "LatestUbuntuAmiId"
      Tags:
        - Key: Name
          Value: Node2
  EC2InstanceNode3:
    Type: AWS::EC2::Instance
    Properties:
      InstanceType: !Ref "InstanceType"
      SecurityGroups: [!Ref "NodeSecurityGroup"]
      KeyName: !Ref "KeyName"
      ImageId: !Ref "LatestUbuntuAmiId"
      Tags:
        - Key: Name
          Value: LoadBalancer
```

```
Outputs:
  AnsibleMasterPublicIP:
    Description: Public IP address of the newly created Ansible master EC2 instance
    Value: !GetAtt [EC2InstanceMaster, PublicIp]

  Node1PublicIP:
    Description: Public IP address of the newly created Node1 Linux EC2 instance
    Value: !GetAtt [EC2InstanceNode1, PublicIp]

  Node2PublicIP:
    Description: Public IP address of the newly created Node2 Linux EC2 instance
    Value: !GetAtt [EC2InstanceNode2, PublicIp]

  Node3PublicIP:
    Description: Public IP address of the newly created LoadBalancer Linux EC2
instance
    Value: !GetAtt [EC2InstanceNode3, PublicIp]
```

`aws cloudformation create-stack --stack-name LoadBalancer --template-body file://loadbalanced.yml`

```
~\..\..\..\..\..\..\ansible aws cloudformation create-stack --stack-name LoadBalancer --template-body file://loadbalanced.yml
StackId: arn:aws:cloudformation:me-south-1:568935291733:stack/LoadBalancer/d0e793b0-fe02-11ec-843c-067eb04fc994
```

The outputs:



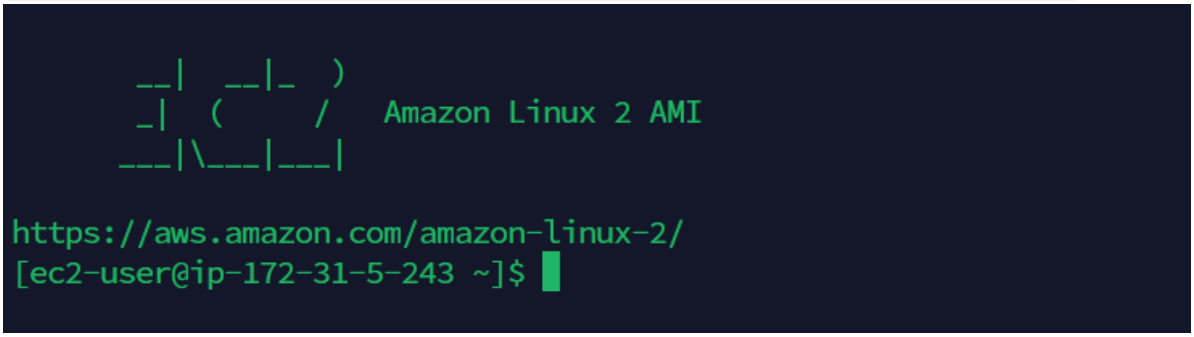
Outputs (4)

Search outputs

Key	Value	Description	Export name
AnsibleMasterPublicIP	157.175.178.96	Public IP address of the newly created Ansible master EC2 instance	-
Node1PublicIP	15.185.100.3	Public IP address of the newly created Node1 Linux EC2 instance	-
Node2PublicIP	15.185.167.130	Public IP address of the newly created Node2 Linux EC2 instance	-
Node3PublicIP	15.185.142.190	Public IP address of the newly created LoadBalancer Linux EC2 instance	-

Connect to ansible-master server using ssh protocol:

`ssh -i "main.pem" ec2-user@ec2-157-175-178-96.me-south-1.compute.amazonaws.com`



Add my private ssh key to the Ansible master server to this directory `/home/ec2-user/main.pem` and change the permissions to 400:

`sudo vim main.pem`

`sudo chmod 400 main.pem`

```
[ec2-user@ip-172-31-5-243 ~]$ sudo vim main.pem
[ec2-user@ip-172-31-5-243 ~]$ sudo chmod 400 main.pem
[ec2-user@ip-172-31-5-243 ~]$
```

Creating an inventory file in the path `/etc/ansible/hosts`
`sudo vim /etc/ansible/hosts`
with the following content:

```
[webservers]
node1 ansible_host=15.185.100.3 ansible_user=ubuntu
node2 ansible_host=15.185.167.130 ansible_user=ubuntu

[loadbalancers]
node3 ansible_host=15.185.142.190 ansible_user=ubuntu

[all:vars]
ansible_ssh_private_key_file=/home/ec2-user/main.pem
```

```
[ec2-user@ip-172-31-5-243 ~]$ sudo vim /etc/ansible/hosts
[ec2-user@ip-172-31-5-243 ~]$ sudo cat /etc/ansible/hosts
[webservers]
node1 ansible_host=15.185.100.3 ansible_user=ubuntu
node2 ansible_host=15.185.167.130 ansible_user=ubuntu

[loadbalancers]
node3 ansible_host=15.185.142.190 ansible_user=ubuntu

[all:vars]
ansible_ssh_private_key_file=/home/ec2-user/main.pem
[ec2-user@ip-172-31-5-243 ~]$
```

Let us create a new directory with name `playbooks` under `ansible` folder
`mkdir ansible`
`cd ansible`
`mkdir playbooks`
`cd playbooks`

```
[ec2-user@ip-172-31-5-243 ~]$ mkdir ansible
[ec2-user@ip-172-31-5-243 ~]$ cd ansible
[ec2-user@ip-172-31-5-243 ansible]$ mkdir playbooks
[ec2-user@ip-172-31-5-243 ansible]$ cd playbooks
[ec2-user@ip-172-31-5-243 playbooks]$
```

Installing Apache Server on all the servers (Webserver and Load Balancer)

Create an `ini` file that has all the hosts configured.
Install Apache2 (webserver) and php on all webservers machines and apache2 on the load balancer machine
Create a playbook to define all the services to install

Add the following steps to in the `install-services.yml` file
`install-services.yml`

```
---
  - hosts: loadbalancers
    become: true
    tasks:
      - name: Installing apache
        apt: name=apache2 state=present update_cache=yes
      - name: Ensure apache starts
        service: name=apache2 state=started enabled=yes

  - hosts: webservers
    become: true
    tasks:
      - name: Installing services
        apt:
          name:
            - apache2
            - php
          state: present
          update_cache: yes
      - name: Ensure apache starts
        service: name=apache2 state=started enabled=yes
```

Run the command to install all the services define above.

Create a playbook file in playbooks folder to define all the services to install:

`vim install-services.yml`

```
[ec2-user@ip-172-31-5-243 playbooks]$ vim install-services.yml
[ec2-user@ip-172-31-5-243 playbooks]$ cat install-services.yml
---
  - hosts: loadbalancers
    become: true
    tasks:
      - name: Installing apache
        apt: name=apache2 state=present update_cache=yes
      - name: Ensure apache starts
        service: name=apache2 state=started enabled=yes

  - hosts: webservers
    become: true
    tasks:
      - name: Installing services
        apt:
          name:
            - apache2
            - php
          state: present
          update_cache: yes
      - name: Ensure apache starts
        service: name=apache2 state=started enabled=yes
[ec2-user@ip-172-31-5-243 playbooks]$
```

Run the following command in the folder where `install-services.yml` file is saved

`sudo ansible-playbook install-services.yml`

```
[ec2-user@ip-172-31-5-243 playbooks]$ sudo ansible-playbook install-services.yml

PLAY [loadbalancers] ***********************************************************************************

TASK [Gathering Facts] *********************************************************************************
ok: [node3]

TASK [Installing apache] *******************************************************************************
changed: [node3]

TASK [Ensure apache starts] ****************************************************************************
ok: [node3]

PLAY [webservers] **************************************************************************************

TASK [Gathering Facts] *********************************************************************************
ok: [node2]
ok: [node1]

TASK [Installing services] *****************************************************************************
changed: [node2]
changed: [node1]

TASK [Ensure apache starts] ****************************************************************************
ok: [node1]
ok: [node2]

PLAY RECAP *********************************************************************************************
node1                      : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0
 ignored=0
node2                      : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0
 ignored=0
node3                      : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0
 ignored=0
```

Configure Server - Deploying application

Let us verify that the server is installed successfully and server is up and running
Create a file with name index.html under `ansible/config` folder

`vim index.html`

Copy paste the below code in `index.html` file

```
<?php
echo "<h1>Hello, World! This is my Ansible page.</h1>";
?>
```

Create a playbook to copy this index.html file in the webservers

`vim setup-app.yml`

Add the steps to copy index.html file to both the webservers

`# setup-app.yml`

```
---
  - hosts: webservers
    become: true
    tasks:
      - name: Upload application file
        copy:
          src: ../config/index.html
          dest: /var/www/html
          mode: 0755
```

Execute the ansible-playbook command to run `setup-app.yml` file and copy the index.html file to the webservers

`vim index.html`

```
[ec2-user@ip-172-31-5-243 config]$ vim index.html
[ec2-user@ip-172-31-5-243 config]$ cat index.html
<?php
echo "<h1>Hello, World! This is my Ansible page.</h1>";
?>
[ec2-user@ip-172-31-5-243 config]$
```

`vim setup-app.yml`

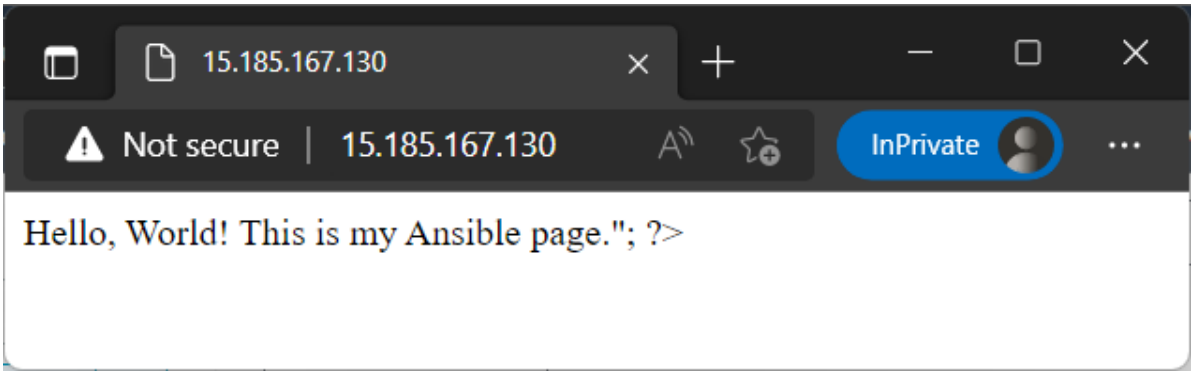
```
[ec2-user@ip-172-31-5-243 playbooks]$ vim setup-app.yml
[ec2-user@ip-172-31-5-243 playbooks]$ cat setup-app.yml
---
  - hosts: webservers
    become: true
    tasks:
      - name: Upload application file
        copy:
          src: ../config/index.html
          dest: /var/www/html
          mode: 0755
[ec2-user@ip-172-31-5-243 playbooks]$
```

`sudo ansible-playbook playbooks/setup-app.yml`

```
[ec2-user@ip-172-31-5-243 ansible]$ sudo ansible-playbook playbooks/setup-app.yml

PLAY [webservers] **************************************************************************************

TASK [Gathering Facts] *********************************************************************************
ok: [node2]
ok: [node1]

TASK [Upload application file] **************************************************************************
changed: [node1]
changed: [node2]

PLAY RECAP *********************************************************************************************
node1                      : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0
 ignored=0
node2                      : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0
 ignored=0
```

Node1

Node2



Configure Load Balancer

Create a new file with name `lb.conf` under `ansible/config` folder

```
ProxyRequests off
<Proxy balancer://webcluster >

 BalancerMember http://"Ip address of server 1"
 BalancerMember http://"Ip address of server 2"

 ProxySet lbmethod=byrequests
</Proxy>

# Optional
<Location /balancer-manager>
SetHandler balancer-manager
</Location>

ProxyPass /balancer-manager !
ProxyPass / balancer://webcluster/
```

Let us write a playbook to move this configuration file to `/etc/httpd/conf.d/lb.conf`

Create a new file with name `setup-lb.yml` under playbooks folder and add below code:-

`# setup-lb.yml`

```
---
  - hosts: loadbalancers
    become: true
    tasks:
      - name: Creating template
        copy:
          src: ../config/lb.conf
          dest: /etc/apache2/conf.d/
          mode: 0755

      - name: restart apache
        service: name=apache2 state=restarted
```

Execute the ansible-playbook command to run `setup-lb.yml` file

`vim lb.conf`

```
[ec2-user@ip-172-31-5-243 config]$ vim lb.conf
[ec2-user@ip-172-31-5-243 config]$ cat lb.conf
ProxyRequests off
<Proxy balancer://webcluster >

 BalancerMember http://"Ip address of server 1"
 BalancerMember http://"Ip address of server 2"

 ProxySet lbmethod=byrequests
</Proxy>

# Optional
<Location /balancer-manager>
SetHandler balancer-manager
</Location>

ProxyPass /balancer-manager !
ProxyPass / balancer://webcluster/
[ec2-user@ip-172-31-5-243 config]$
```

`vim setup-lb.yml`

```
[ec2-user@ip-172-31-5-243 playbooks]$ vim setup-lb.yml
[ec2-user@ip-172-31-5-243 playbooks]$ cat setup-lb.yml
---
  - hosts: loadbalancers
    become: true
    tasks:
      - name: Creating template
        copy:
          src: ../config/lb.conf
          dest: /etc/apache2/conf.d/
          mode: 0755

      - name: restart apache
        service: name=apache2 state=restarted
[ec2-user@ip-172-31-5-243 playbooks]$
```

`sudo ansible-playbook playbooks/setup-lb.yml`

```
[ec2-user@ip-172-31-5-243 ansible]$ sudo ansible-playbook playbooks/setup-lb.yml

PLAY [loadbalancers] ***********************************************************************************

TASK [Gathering Facts] *********************************************************************************
ok: [node3]

TASK [Creating template] ********************************************************************************
changed: [node3]

TASK [restart apache] ***********************************************************************************
changed: [node3]

PLAY RECAP *********************************************************************************************
node3                      : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0
 ignored=0
```

So far, we have created all the servers through different playbooks

Let us now create one single file to run all these playbooks in sequential order.

Let us create a file with name `all-playbooks.yml`

`# all-playbooks.yml`

```
---
  - import_playbook: install-services.yml
  - import_playbook: setup-app.yml
  - import_playbook: setup-lb.yml
```

Execute the ansible-playbook command to run `all-playbooks.yml` file

`vim all-playbooks.yml`

```
[ec2-user@ip-172-31-5-243 playbooks]$ vim all-playbooks.yml
[ec2-user@ip-172-31-5-243 playbooks]$ cat all-playbooks.yml
---
  - import_playbook: install-services.yml
  - import_playbook: setup-app.yml
  - import_playbook: setup-lb.yml
[ec2-user@ip-172-31-5-243 playbooks]$
```

`sudo ansible-playbook playbooks/all-playbooks.yml`

```
[ec2-user@ip-172-31-5-243 ansible]$ sudo ansible-playbook playbooks/all-playbooks.yml

PLAY [loadbalancers] ***********************************************************************************

TASK [Gathering Facts] *********************************************************************************
ok: [node3]

TASK [Installing apache] *******************************************************************************
ok: [node3]

TASK [Ensure apache starts] ****************************************************************************
ok: [node3]

PLAY [webservers] **************************************************************************************

TASK [Gathering Facts] *********************************************************************************
ok: [node2]
ok: [node1]

TASK [Installing services] *****************************************************************************
ok: [node2]
ok: [node1]

TASK [Ensure apache starts] ****************************************************************************
ok: [node1]
ok: [node2]

PLAY [webservers] **************************************************************************************

TASK [Gathering Facts] *********************************************************************************
ok: [node2]
ok: [node1]

TASK [Upload application file] **************************************************************************
ok: [node2]
ok: [node1]

PLAY [loadbalancers] ***********************************************************************************

TASK [Gathering Facts] *********************************************************************************
ok: [node3]

TASK [Creating template] ********************************************************************************
ok: [node3]

TASK [restart apache] **********************************************************************************
changed: [node3]

PLAY RECAP *********************************************************************************************
node1                      : ok=5    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
 ignored=0
node2                      : ok=5    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
 ignored=0
node3                      : ok=6    changed=1    unreachable=0    failed=0    skipped=0    rescued=0
 ignored=0
```