

Vijay's Blog

 Follow

Expense project using Ansible



vijaykumar guntireddy

Sep 4, 2024 •  7 min read

Table of contents

Ansible connection to AWS and create an Instance

- › AWS IAM settings
- › Ansible r53 settings

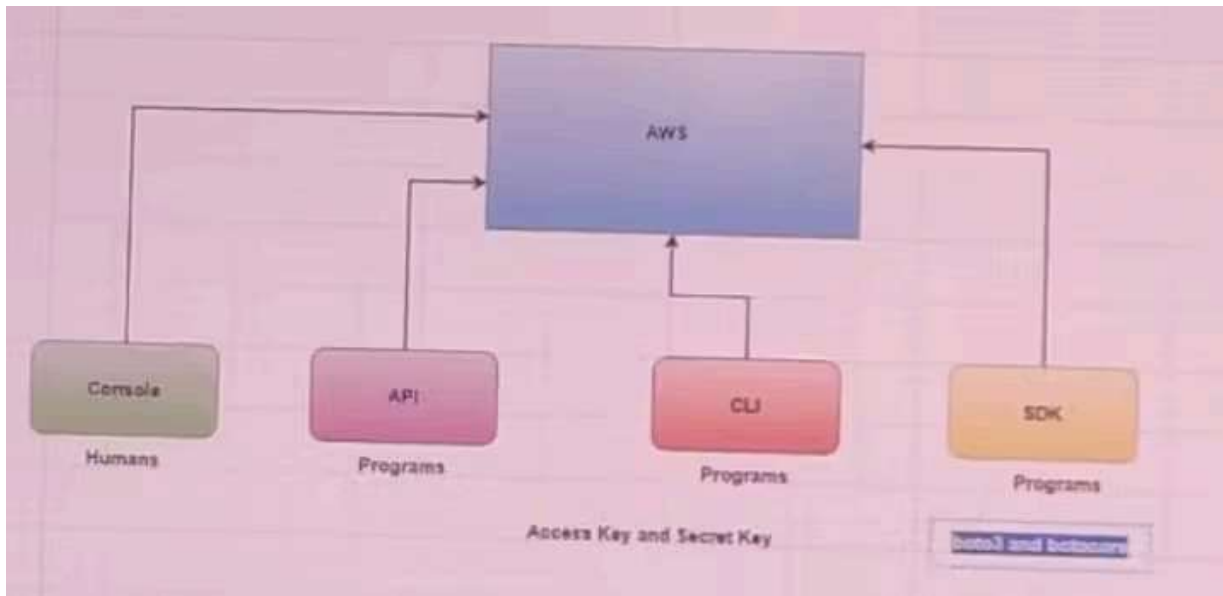
Create a Expense project repository

- › Create playbook for mysql server
- › Create a playbook for Backend server
- › Create a playbook for Frontend server

Challenges and Solutions

- Expense project requirements are 3 servers , 3 records.

- Ansible is not CM tool , it can connect to any machine if module is available.



- AWS can be connected in many ways , with ansible we can connect to AWS through boto3 and botocore.
- Boto3 and Botocore are both Python libraries used to interact with Amazon Web Services (AWS).
- https://docs.ansible.com/ansible/latest/collections/amazon/aws/ec2_instance_module.html

Ansible connection to AWS and create an Instance

Lets find out mandatory fields in amazon.aws.ec2_instance

COPY

```
- name: start an instance with a public IP address
  amazon.aws.ec2_instance:
    name: "public-compute-instance"
    key_name: "prod-ssh-key"
    vpc_subnet_id: subnet-5ca1ab1e
    instance_type: c5.large
```



```
security_group: default
network:
  assign_public_ip: true
image_id: ami-123456
tags:
  Environment: Testing
```

- name
- vpc_subnet_id

i-0474019559accff8a (server)

Details | Status and alarms | Monitoring | Security | **Networking** | Storage | Tags

▼ Networking details [Info](#)

Public IPv4 address
34.203.209.44 | [open address](#)

Public IPv4 DNS
compute-1.amazonaws.com | [open address](#)

Subnet ID copied

subnet-00b0898fc1fd11e34

Private IPv4 address
172.31.38.1

Private IP DNS
ip-172-31-38-1

IPv6 addresses
-

- instance_type : "t3.micro"
- security_group

Details | Status and alarms | Monitoring | **Security** | Networking | Storage | Tags

▼ Security details

IAM Role
-

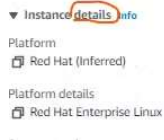
Owner ID
664418955401

Security groups
sg-0bc7f605c01cc2ed2 (allow-all)

Inbound rules

- image_id





Now we have mandatory fields information , lets develop the yaml file

COPY

```
- name: create ec2 and r53 records
hosts: local
connection: local
vars:
  subnet_id: subnet-00b0898fc1fd11e34 # replace with your subnet
  sg_id: "sg-0bc7f605c01cc2ed2" # replace with your security group
  ami_id: "ami-09c813fb71547fc4f" # replace with your image id
  instances:
    - mysql
    - backend
    - frontend
  zone: vijaydevops.store #replace your zone
tasks:
- name: create ec2 instance
  amazon.aws.ec2_instance:
    name: "{{ item }}"
    vpc_subnet_id: "{{ subnet_id }}"
    instance_type: "t3.micro"
    security_group: "{{ sg_id }}"
    image_id: "{{ ami_id }}"
  loop: "{{ instances }}"
  register: ec2_instances
```

- Above yaml file expectation to create 3 instances of mysql , backend, frontend
- ec2_instances → amazon.aws.ec2_instance with return execution output of this task.

Execute this yaml file and its given below error.

COPY

```
TASK [create ec2 instance] *****
Error : Failed to import the required Python library (botocore and

$pip3.9 install boto3 botocore
```



Execute the yaml file and its given below error

COPY

```
.NoCredentialsError: Unable to locate credentials
```

AWS connection requires authentication , for that we need to do settings in AWS IAM

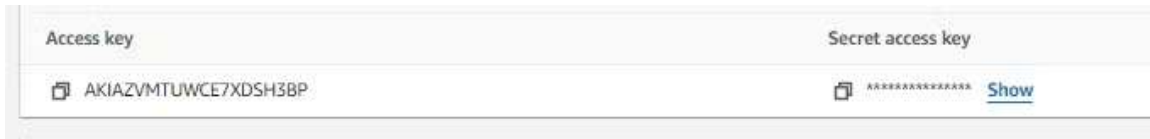
AWS IAM settings

- In AWS , search 'IAM'
- click users → create user → Enter <name> → click next
- select "attached policies directly" → click next
- click create user

User name	Path	Group	Last activity	MFA	Password age	Console last sign-in	Access key ID	Active key age	Access key last used
ansible	/	0	-	-	-	-	-	-	-

user "Ansible" is created as it is non human user we need to create access keys

- Click Ansible → select "Security credentials"
- click "Create Accesskey" → select "CLI" → next → create access key



copy these accesskey and secretkey as this window will not open again to see this information

- In Ansible server , execute below command provide accesskey and secret key

```
34.203.209.44 | 172.31.38.245 | t3.micro | https://github.com/qtivijay/devopsaws.git
[ root@ip-172-31-38-245 ~/devopsaws/ansible1 ]# aws configure
AWS Access Key ID [None]: 
AWS Secret Access Key [None]: 
Default region name [None]: us-east-1
Default output format [None]:
```

Execute the command again to create the three instances

COPY

```
ansible-playbook -i inventroy.ini 18-ec2instance.yaml
```

three instances you can find in EC2

<input type="checkbox"/>	frontend	i-0833211609bc361a5	Running	🔍	🔍
<input type="checkbox"/>	mysql	i-01e0b56aaa9dd5204	Running	🔍	🔍
<input type="checkbox"/>	server	i-0474019559accff8a	Running	🔍	🔍
<input type="checkbox"/>	backend	i-016c0423347370eb3	Running	🔍	🔍

Lets check the output of ec2_instances ([ec2instance playbook file](#))



COPY

```
- name: print the ec2_instances message
  ansible.builtin.debug:
    msg: "{{ec2_instances}}"
```

ec2_instance message generates big data

https://github.com/qtivijay/devopsaws/blob/master/ansible1/ec2_createinstance.json

If you observe it generates three results for three instances

```
{
  "msg": {
    "changed": false,
    "msg": "All items completed",
    "results": [
      > { ...
    },
      > { ...
    },
      > { ...
    ]
  },
  "skipped": false
}
```

you can find public and private ip address

```
{
  "private_ip_address": "172.31.45.98",
  "product_codes": [],
  "public_dns_name": "ec2-54-89-174-182.compute-1.amazonaws.com",
  "public_ip_address": "54.89.174.182",
  "root_device_name": "/dev/sda1",
  "root_device_type": "ebs",
}
```

we can loop ec2_instances.results to get these information.



Ansible r53 settings

- https://docs.ansible.com/ansible/latest/collections/amazon/aws/route53_module.html
- added below code in ec2 instance playbook ([ec2instance playbook file](#))

COPY

```
#private IP R53
- name: create r53 private records #<-- new code
  amazon.aws.route53:
    state: present
    zone: "{{ zone }}"
    record: "{{ item.item }}.{{ zone }}" #mysql.vijaydevops.store
    type: A
    ttl: 1
    value: "{{ item.instances[0].private_ip_address }}"
    wait: true
    overwrite: true
    loop: "{{ ec2_instances.results }}"
```

created three records

<input type="checkbox"/>	expense.vijaydevops.store	A	Simple	-	No	54.173.178.88
<input type="checkbox"/>	frontend.vijaydevops.store	A	Simple	-	No	172.31.41.37
<input type="checkbox"/>	mysql.vijaydevops.store	A	Simple	-	No	172.31.45.98
<input type="checkbox"/>	nodejs.vijaydevops.store	A	Simple	-	No	172.31.47.108

Add public record for user to access the front end

here we are adding a conditon to do the same operation as above only for "frontend"(when: item.item == "frontend") ([ec2instance playbook file](#))




```
- name: create r53 public record for frontend
amazon.aws.route53:
  state: present
  zone: "{{ zone }}"
  record: "{{ zone }}" #mysql.vijaydevops.store
  type: A
  ttl: 1
  value: "{{ item.instances[0].public_ip_address }}"
  wait: true
  overwrite: true
loop: "{{ ec2_instances.results }}"
when: item.item == "frontend"
```

created a public record.

Servers are ready , now we are good to create a expense project files

Create a Expense project repository

- First we need to create inventory.ini

```
[mysql]
mysql.vijaydevops.store

[backend]
backend.vijaydevops.store

[frontend]
frontend.vijaydevops.store

[all:vars]
```



```
ansible_user=ec2-user
ansible_password=DevOps321
```

Create playbook for mysql server

COPY

```
- name: mysql configuration
  hosts: mysql
  become: yes
  tasks:
    - name: install MySQL Server
      ansible.builtin.package:
        name: mysql-server
        state: present

    - name: start mysql server
      ansible.builtin.service:
        name: mysqld
        state: started
        enabled: yes
```

Execute the ansible playbook for mysql

COPY

```
# ansible-playbook -i inventory.ini mysql.yaml
```

```
PLAY [mysql configuration] *****
```

```
TASK [Gathering Facts] *****
```

```
ok: [mysql.vijaydevops.store]
```



```
TASK [install MySQL Server] *****
```

```
changed: [mysql.vijaydevops.store]
```

```
TASK [start mysql server] *****
```

```
changed: [mysql.vijaydevops.store]
```

```
PLAY RECAP *****
```

```
mysql.vijaydevops.store : ok=3    changed=2    unreachable=0
```

database server is created with mysql service.

To create a root user for mysql server , first we need to check if root user already there or not. i will do following steps in mysql playbook.

COPY

1. Provide the MySQL server host, username, and password to establish connection.
2. If the connection attempt fails due to the username not being set, create the user.
3. Create the necessary username and password.
4. If the initial connection attempt is successful, skip the user creation step.



you can use ansible module : [community.mysql.mysql_info](#)

updated mysql playbook with below code to print the mysql server info.
(sqlserver playbook [link](#))

COPY

```
- name: Connecting to mysql server
  community.mysql.mysql_info:
    login_user: root
    login_password: "{{mysql_password}}"
    login_host: "{{mysql_host}}"
    ignore_errors: true
    register: mysql_info
```



```
- name: print the mysql_info
  ansible.builtin.debug:
    msg: "{{mysql_info}}"
```

executing the mysql playbook results are

COPY

```
# thrown below error
```

```
"A MySQL module is required: for Python 2.7 either PyMySQL, or MySQLdb
or for Python 3.X mysqlclient or PyMySQL. Consider setting
ansible_python_interpreter to use the intended Python version."
```



Updated playbook to install PyMySQL (sqlserver playbook [link](#))

COPY

```
- name: install python lib PyMySQL
  ansible.builtin.pip:
    name:
      - PyMySQL
```

executing the mysql playbook results the mysqlserver information.
using this we can update the root user.

COPY

```
TASK [print the mysql_info] *****
ok: [mysql.vijaydevops.store] => {
  "msg": {
    "changed": false,
    "failed": true,
    "msg": "unable to connect to database using pymysql 1.1.1,"
```

```
}
}
```

Updated the mysql server playbook (sqlserver playbook [link](#))

COPY

```
- name: update my sql root password if it not set
  ansible.builtin.command: "mysql_secure_installation --set-root-
  when: mysql_info.failed == true
```

Executing the mysql playbook

COPY

```
TASK [print the mysql_info] *****
ok: [mysql.vijaydevops.store] => {
  "msg": {
    "changed": false,
    "failed": true,
    "msg": "unable to connect to database using pymysql 1.1.1,"
  }
}
```

```
TASK [update my sql root password if it not set] *****
changed: [mysql.vijaydevops.store]
```

you can root password is set , now we have to login in mysql server and check



In mysql server

```
mysql -h mysql.vijaydevops.store -u root -pExpenseApp@1
```

Afterwards if you run mysql playbook ,it is Successful (sqlserver
playbook [link](#))

by this mysql server setup is completed

```
you can see last step is skipping  
skipping: [mysql.vijaydevops.store]
```

Create a playbook for Backend server

Backend playbook :

<https://github.com/qtivijay/devopsaws/blob/master/expense-ansible/backend.yaml>

Create a playbook for Frontend server

Frontend playbook :

<https://github.com/qtivijay/devopsaws/blob/master/expense-ansible/frontend.yaml>

Execute with domain name (<http://vijaydevops.store/>)





COPY

```
#check the data in data server
mysql -h mysql.vijaydevops.store -u root -pExpenseApp@1

mysql> select * from transactions;
+----+-----+-----+
| id | amount | description |
+----+-----+-----+
| 1 | 200 | travel |
+----+-----+-----+
1 row in set (0.00 sec)
```

Challenges and Solutions

1. In Ansible server after installing Ansible and executing the 18-ec2instance.yaml playbook to create 3 servers and 3 records , if you are seeing below issue

COPY

```
NoCredentialsError: Unable to locate credentials
```

solution:



COPY

execute "aws configure" and update access key information

2. If you are repeating the project by creating a new ansible instance(Ansible server) ensure subnetid , sgid ,amcid are updated in 18-ec2instance.yaml file before execution.
3. After running the mysql playbook "mysql server password" has been set but if i'm not manually login through mysqlserver and try to run mysql playbook again in ansible server i see login still fails why ..? <I don't know the reason>

Subscribe to my newsletter

Read articles from **Vijay's Blog** directly inside your inbox. Subscribe to the newsletter, and don't miss out.

SUBSCRIBE

Written by



vijaykumar guntireddy

Results-oriented Global Management Professional with diverse experience of >19 years in different capacities primarily in end-to-end product management, quality, delivery of critical business applications, and project management.



Follow