

## ANSIBLE LOOP AND CONDITION

**Task-1:** To provision 2 ubuntu instances and 1 amazon linux instance on AWS cloud using Ansible.

### Pre-requisites:

1) Install the boto-3 module to interact with AWS API

```
pip install boto3
```

2) Install the AWS ansible collection:

```
ansible-galaxy collection install amazon.aws
```

3) Create ansible vault and save aws credentials securely:

```
openssl rand -base64 2048 > vault.pass
```

```
ansible-vault create group_vars/all/pass.yml --  
vault-password-file vault.pass
```

➞ Lets write a playbook for above task using **Ansible loop**:

```
---  
- hosts: localhost  
  connection: local  
  tasks:  
    - name: start 2 ubuntu and 1 amazon linux  
instances with a public IP address  
      amazon.aws.ec2_instance:  
        name: "ansible-instance"  
        key_name: "Rayeez_AWS"  
        instance_type: t2.micro  
        region: us-east-1  
        security_group: default  
        aws_access_key: "{{ec2_access_key}}"  
        aws_secret_key: "{{ec2_secret_key}}"  
        network_interfaces:  
          - assign_public_ip: true
```

```

    image_id: "{{item}}"
  loop:
    - "ami-0341d95f75f311023"
    - "ami-0360c520857e3138f"
    - "ami-0360c520857e3138f"

```

Here AMI-id of ubuntu instances = ami-0360c520857e3138f

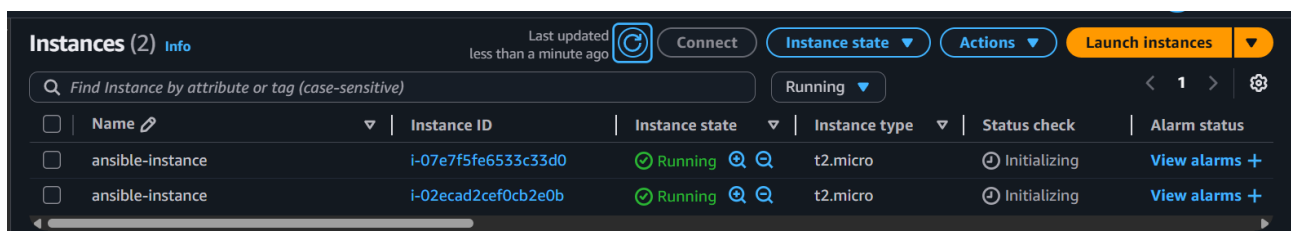
AMI-id of Amazon-linux-instance = ami-0341d95f75f311023

```

ansible-playbook ec2-create.yaml --vault-
password-file vault.pass

```

The problem here is , It will create only 2 instances – **one ubuntu instance** and one **amazon linux instance**.



Name	Instance ID	Instance state	Instance type	Status check	Alarm status
ansible-instance	i-07e7f5fe6533c33d0	Running	t2.micro	Initializing	View alarms +
ansible-instance	i-02ecad2cef0cb2e0b	Running	t2.micro	Initializing	View alarms +

## → Why this has happened?

When we tried to launch two Ubuntu instances with the same AMI ID and name, Ansible created only one, treating the second as a duplicate and skipping it. This known as **Idempotency nature** of Ansible

Unlike Linux, which throws an error when creating an existing directory, Ansible simply ignores the repeated task without showing an error.

Lets make some changes in ansible playbook and execute it again:

```

---
- hosts: localhost
  connection: local
  tasks:
    - name: start 2 ubuntu and 1 amazon linux
      instances with a public IP address
      amazon.aws.ec2_instance:

```

```

name: "{{item.name}}"
key_name: "Rayeez_AWS"
instance_type: t2.micro
region: us-east-1
security_group: default
aws_access_key: "{{ec2_access_key}}"
aws_secret_key: "{{ec2_secret_key}}"
network_interfaces:
  - assign_public_ip: true
image_id: "{{item.image}}"
tags:
  environment: "{{item.name}}"
loop:
  - { image: "ami-0341d95f75f311023", name:
"amazon-linux-server" }
  - { image: "ami-0360c520857e3138f", name:
"ubuntu-server-1" }
  - { image: "ami-0360c520857e3138f", name:
"ubuntu-server-2" }

```

```

ansible-playbook ec2-create.yaml --vault-password-
file vault.pass

```

This time it will create **2 ubuntu** instances and **1 amazon linux** instance:

Instances (3) <a href="#">Info</a>							
Last updated less than a minute ago <a href="#">Refresh</a> <a href="#">Connect</a>				<a href="#">Instance state</a>	<a href="#">Actions</a>	<a href="#">Launch instances</a>	
<input type="text" value="Find Instance by attribute or tag (case-sensitive)"/>				<a href="#">All states</a>		< 1 > <a href="#">Settings</a>	
<input type="checkbox"/>	Name <a href="#">↗</a>	Instance ID	Instance state	Instance type	Status check	Alarm status	
<input type="checkbox"/>	amazon-linux-server	i-09bc510c993cc3e3f	<span>Running</span> <a href="#">🔍</a> <a href="#">🔍</a>	t2.micro	<span>2/2 checks passed</span>	<a href="#">View alarms</a> <a href="#">+</a>	
<input type="checkbox"/>	ubuntu-server-1	i-002cc1cb6b92e3e2e	<span>Running</span> <a href="#">🔍</a> <a href="#">🔍</a>	t2.micro	<span>2/2 checks passed</span>	<a href="#">View alarms</a> <a href="#">+</a>	
<input type="checkbox"/>	ubuntu-server-2	i-0b463b82056ff98b5	<span>Running</span> <a href="#">🔍</a> <a href="#">🔍</a>	t2.micro	<span>2/2 checks passed</span>	<a href="#">View alarms</a> <a href="#">+</a>	

**Task-2:** Set up passwordless authentication with above created instances.

```

ssh-copy-id -f "-o IdentityFile <PATH TO PEM
FILE>" ubuntu@<INSTANCE-PUBLIC-IP>

```

Make sure to download a pemfile from aws console.

➡ Using above command , we can set up passwordless authentication with each server.

```
mdrayeez@DevOps:~$ sudo ssh-copy-id -f "-o IdentityFile Rayeez_AWS.pem" ubuntu@3.239.104.228
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
The authenticity of host '3.239.104.228 (3.239.104.228)' can't be established.
ED25519 key fingerprint is SHA256:CJx/tL7WsSiuQSNT+twRYL3BeL7zM2yBpq5yieTx8Fk.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes

Number of key(s) added: 1

Now try logging into the machine, with: "ssh -o 'IdentityFile Rayeez_AWS.pem' 'ubuntu@3.239.104.228'"
and check to make sure that only the key(s) you wanted were added.

mdrayeez@DevOps:~$ sudo ssh -o 'IdentityFile Rayeez_AWS.pem' 'ubuntu@3.239.104.228'
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-1011-aws x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/pro
```

### Task-3: Now shutdown only ubuntu instances.

Lets write a playbook for this task using a **Ansible condition**:

```
---
- hosts: all
  become: true
  tasks:
    - name: to shutdown only ubuntu instances
      ansible.builtin.command: /sbin/shutdown -t
      now
      when:
        ansible_facts['os_family'] == "Debian"
```

This will shutdown only ubuntu instances:

Instances (3) <span>Info</span>						
Last updated less than a minute ago <span>Refresh</span> <span>Connect</span> <span>Instance state</span> <span>Actions</span> <span>Launch instances</span>						
<input type="text" value="Find Instance by attribute or tag (case-sensitive)"/> <span>All states</span> <span>&lt; 1 &gt;</span> <span>Settings</span>						
<input type="checkbox"/>	Name <span>↗</span>	Instance ID	Instance state	Instance type	Status check	Alarm status
<input type="checkbox"/>	amazon-linux-server	i-09bc510c993cc3e3f	<span>Running</span>	t2.micro	<span>2/2 checks passed</span>	<span>View alarms</span> <span>+</span>
<input type="checkbox"/>	ubuntu-server-1	i-002cc1cb6b92e3e2e	<span>Stopping</span>	t2.micro	-	<span>View alarms</span> <span>+</span>
<input type="checkbox"/>	ubuntu-server-2	i-0b463b82056ff98b5	<span>Stopping</span>	t2.micro	-	<span>View alarms</span> <span>+</span>

✓ **Conclusion:** Using concept of ansible loops and ansible condition able to perform above tasks successfully.