

## | Ansible Project for Beginners (Asked in Interviews) | Provisioning + Configuration Management

Ansible Realtime project

### Task 1

Create three(3) EC2 instances on AWS using Ansible loops

- 2 Instances with Ubuntu Distribution
- 1 Instance with Centos Distribution

Hint: Use `connection: local` on Ansible Control node.

### Task 2

Set up passwordless authentication between Ansible control node and newly created instances.

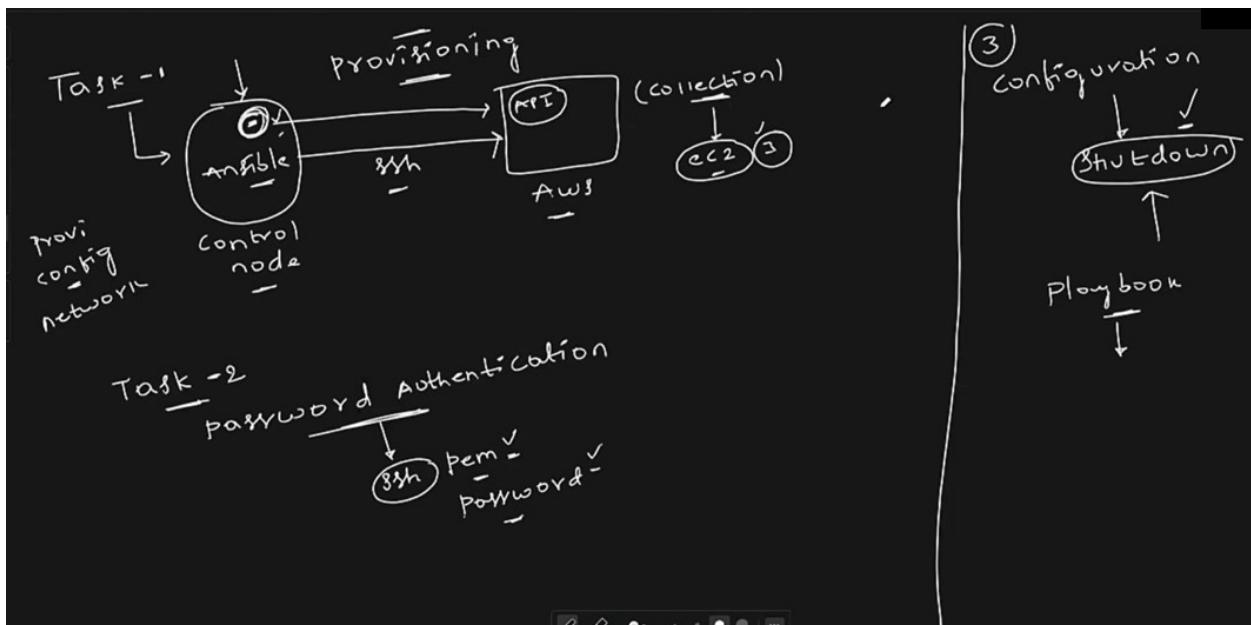
### Task 3

Automate the shutdown of Ubuntu Instances only using Ansible Conditionals

Hint: Use `when` condition on ansible `gather_facts`



# we use AWS Linux .. ignore centos



**IAM Dashboard**

**Security recommendations** (2)

- Add MFA for root user
- Deactivate or delete access keys for root user

**IAM resources**

User groups	Users	Roles	Policies	Identity providers
0	0	2	0	0

**AWS Account**

Account ID: 381491961949  
Account Alias: Create  
Sign-in URL: https://381491961949.signin.aws.amazon.com/console

**Quick Links**

My security credentials: Manage your access keys, multi-factor authentication (MFA) and other credentials.

## # Create a user

**Set permissions**

Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. [Learn more](#)

**Permissions options**

- Add user to group: Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.
- Copy permissions: Copy all group memberships, attached managed policies, and inline policies from an existing user.
- Attach policies directly: Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

**Permissions policies (1204)**

Choose one or more policies to attach to your new user.

Policy name	Type	Attached entities
<a href="#">AccessAnalyzerServiceRolePolicy</a>	AWS managed	0
<a href="#">AdministratorAccess</a>	AWS managed - job function	0

Step 2  
[Set permissions](#)

Step 3  
**Review and create**

User details		
User name	Console password type	Require password reset
ansible-user	None	No

**Permissions summary**

Name	Type	Used as
<a href="#">AmazonEC2FullAccess</a>	AWS managed	Permissions policy

**Tags - optional**  
Tags are key-value pairs you can add to AWS resources to help identify, organize, or search for resources. Choose any tags you want to associate with this user.

No tags associated with the resource.

[Add new tag](#)  
You can add up to 50 more tags.

[Cancel](#) [Previous](#) [Create user](#)

**Identity and Access Management (IAM)**

[View user](#) [X](#)

**User created successfully**  
You can view and download the user's password and email instructions for signing in to the AWS Management Console.

[IAM](#) > [Users](#)

**Users (1) Info**  
An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

User name	Path	Group	Last activity	MFA	Password age	Conso
<a href="#">ansible-user</a>	/	0	-	-	-	-

# go to Security Credentials

**Identity and Access Management (IAM)**

Search IAM

ARN: arn:aws:iam::381491961949:user/ansible-user

Created: June 20, 2024, 19:38 (UTC+05:30)

Console access: Disabled

Last console sign-in: -

Access key 1: Create access key

Permissions | Groups | Tags | **Security credentials** | Access Advisor

### Console sign-in

Console sign-in link: https://381491961949.signin.aws.amazon.com/console

Console password: Not enabled

**Enable console access**

### Multi-factor authentication (MFA) (0)

Use MFA to increase the security of your AWS environment. Signing in with MFA requires an authentication code from an MFA device. Each user can have a maximum of 1 MFA device assigned. [Learn more](#)

Type	Identifier	Certifications	Created on
No MFA devices. Assign an MFA device to improve the security of your AWS environment			
<a href="#">Assign MFA device</a>			

### Access key best practices & alternatives Info

Avoid using long-term credentials like access keys to improve your security. Consider the following use cases and alternatives.

**Step 2 - optional**

Set description tag

**Step 3**

Retrieve access keys

**Use case**

- Command Line Interface (CLI)**  
You plan to use this access key to enable the AWS CLI to access your AWS account.
- Local code**  
You plan to use this access key to enable application code in a local development environment to access your AWS account.
- Application running on an AWS compute service**  
You plan to use this access key to enable application code running on an AWS compute service like Amazon EC2, Amazon ECS, or AWS Lambda to access your AWS account.
- Third-party service**  
You plan to use this access key to enable access for a third-party application or service that monitors or manages your AWS resources.
- Application running outside AWS**  
You plan to use this access key to authenticate workloads running in your data center or other infrastructure outside of AWS that needs to access your AWS resources.

**Access key created**  
This is the only time that the secret access key can be viewed or downloaded. You cannot recover it later. However, you can create a new access key any time.

IAM > Users > ansible-user > Create access key

Step 1  
Access key best practices & alternatives

Step 2 - optional  
Set description tag

Step 3  
Retrieve access keys

**Retrieve access keys**

**Access key**  
If you lose or forget your secret access key, you cannot retrieve it. Instead, create a new access key and make the old key inactive.

Access key	Secret access key
AKIAVRUVRMBOREADEH4E	***** Show

**Access key best practices**

- Never store your access key in plain text, in a code repository, or in code.
- Disable or delete access key when no longer needed.
- Enable least-privilege permissions.
- Rotate access keys regularly.

For more details about managing access keys, see the [best practices for managing AWS access keys](#).

[Download .csv file](#) [Done](#)

## VScode

EXTENSIONS: MARKETPLACE

**Ansible**

**Ansible** v24.6.0

658K | ★ 3.5

Ansible language support

This publisher has verified ownership of redhat.com

Activation time: 69ms

This extension is enabled globally.

This extension has a Pre-Release version available

**{DEPRECATED} Ansible** v24.6.0

69K | ★ 5

Ansible language support

Tomasz Maciążek

**ansible-vault** v24.6.0

76K | ★ 3.5

Encrypt/decrypt ansible-vault

Eric Ho

**VSCode snippets** v24.6.0

28K | ★ 5

Provide an automatically built-in snippets for Ansible

Mattias Baake

**Azure REST for...** v24.6.0

14K | ★ 5

Create, update or delete an Azure REST API resource

Zim Kalinowski

**ansible-vault-i...** v24.6.0

22K | ★ 4

Encrypt/decrypt ansible-vault

wolfmah

**Ansible Snippets** v24.6.0

2K | ★ 5

Snippets for Ansible Playbooks

Travis Michette

**Ansible Variable...** v24.6.0

1K | ★ 5

Helps finding the definition of Ansible variables

IkBenGeenRobot

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS AZURE

abhishekveeramalla@aveerama-mac: conditionals %

# Now .. We need Pre-requisites .. installation

# Set up EC2 Collection + Authentication

The screenshot shows a browser window with two tabs open. The active tab is titled "ansible-zero-to-hero/Day-06/0" and displays a GitHub page for a file named "main/prereq.md". The page content includes sections for installing boto3, installing the AWS Collection, and setting up Vault. Each section contains a command-line instruction in a code block.

**Install boto3**

```
pip install boto3
```

**Install AWS Collection**

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

**Setup Vault**

1. Create a password for vault

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

2. Add your AWS credentials using the below vault command

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

The screenshot shows the AWS EC2 Instances details page for an instance named 'ansible-instance'. The instance ID is i-0630b079c943805a5. The instance is running, with a public IPv4 address of 23.20.88.127 and a private IPv4 address of 172.31.34.79. It has an elastic IP assigned. The instance type is t2.micro, and it's associated with a VPC ID (vpc-0af77a95d619f75fb) and a subnet ID (subnet-0953d45727baeacd8). The instance ARN is arn:aws:ec2:us-east-1:381491961949:instance/i-0630b079c943805a5. The instance was created using an AMI and an IAM role. The instance summary table includes columns for Instance ID, Public IPv4 address, Private IPv4 addresses, IPv6 address, Instance state, Hostname type, Private IP DNS name (IPv4 only), Answer private resource DNS name, Instance type, VPC ID, Elastic IP addresses, Auto-assigned IP address, IAM Role, Subnet ID, AWS Compute Optimizer finding, IMDSv2, Required, and Instance ARN.

```
# Same thing .. Go to inventory file .. app / db ... you can create password/ any varabile in the  
group_vars
```

The screenshot shows the VS Code interface with the following details:

- EXPLORER**: Shows a tree view of files and folders under **PLAYBOOKS**, including `.vscode`, `ec2`, `group_vars` (with `all`, `! pass.yml`, `! app.yaml`, and `ec2_create.yaml`), `inventory.ini` (selected), `vault.pass`, and `first-playbook`.
- TERMINAL**: Displays the command `zsh - ec2` and the output of an Ansible task:

```
TASK [ec2 : start an instance with a public IP address] ****
changed: [localhost]
PLAY RECAP ****
localhost : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```
- PROBLEMS**: Shows no problems.
- OUTPUT**: Shows no output.
- DEBUG CONSOLE**: Shows no debug console.
- AZURE**: Shows no Azure resources.

The screenshot shows the VS Code interface with the following details:

- EXPLORER**: Shows a tree view of files and folders under **PLAYBOOKS**, including `.vscode`, `ec2`, `group_vars` (with `all`, `! pass.yml`, and `! app.yaml` selected), `ec2_create.yaml`, `inventory.ini`, `vault.pass`, and `first-playbook`.
- TERMINAL**: Displays the command `zsh - ec2` and the output of an Ansible task:

```
TASK [ec2 : start an instance with a public IP address] ****
changed: [localhost]
PLAY RECAP ****
localhost : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```
- PROBLEMS**: Shows no problems.
- OUTPUT**: Shows no output.
- DEBUG CONSOLE**: Shows no debug console.
- AZURE**: Shows no Azure resources.

In the `app.yaml` file, there is a line of code highlighted in blue:

```
password: asdfsdfsdfsdg
```

# Setup EC2 Collection and Authentication

## Install boto3

```
pip install boto3
```

## Install AWS Collection

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

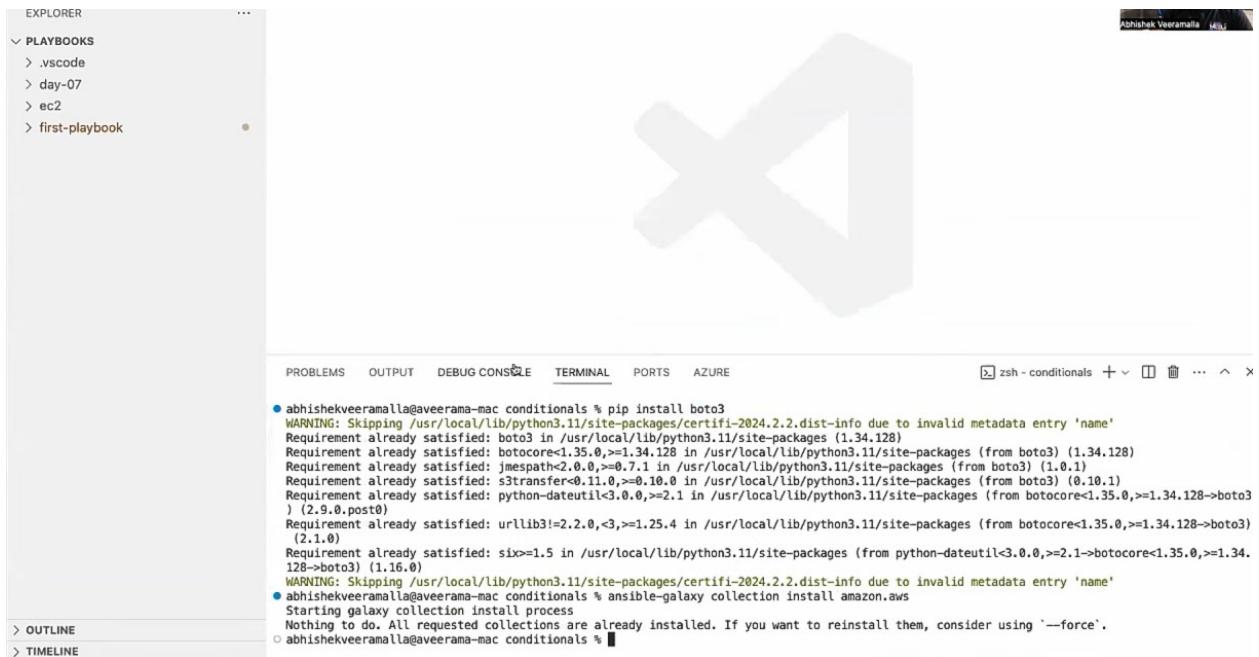
## Setup Vault

- ## 1. Create a password for vault

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

2. Add your AWS credentials using the below vault command

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



## Set up vault

## Setup Vault

### 1. Create a password for vault

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

### 2. Add your AWS credentials using the below vault command

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

The screenshot shows a terminal window with the following content:

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS AZURE zsh - day-07

● abhishekveeramalla@aveerama-mac day-07 % openssl rand -base64 2048 > vault.pass  
● abhishekveeramalla@aveerama-mac day-07 % ls vault.pass  
○ abhishekveeramalla@aveerama-mac day-07 % ansible-vault create group\_vars/all/pass.yml --vault-password-file vault.pass

Access key created  
This is the only time that the secret access key can be viewed or downloaded. You cannot recover it later. However, you can create a new access key any time.

IAM > Users > ansible-user > Create access key

Step 1  
Access key best practices & alternatives

Step 2 - optional  
Set description tag

Step 3  
Retrieve access keys

Retrieve access keys Info

Access key  
If you lose or forget your secret access key, you cannot retrieve it. Instead, create a new access key and make the old key inactive.

Access key Copied Secret access key

AKIAVVRUVRMBOREADEH4E \*\*\*\*\* Show

Access key best practices

- Never store your access key in plain text, in a code repository, or in code.
- Disable or delete access key when no longer needed.
- Enable least-privilege permissions.
- Rotate access keys regularly.

For more details about managing access keys, see the [best practices for managing AWS access keys](#).

Download .csv file Done

# Create a file .. ec2\_create.yaml file .. Start writing playbook

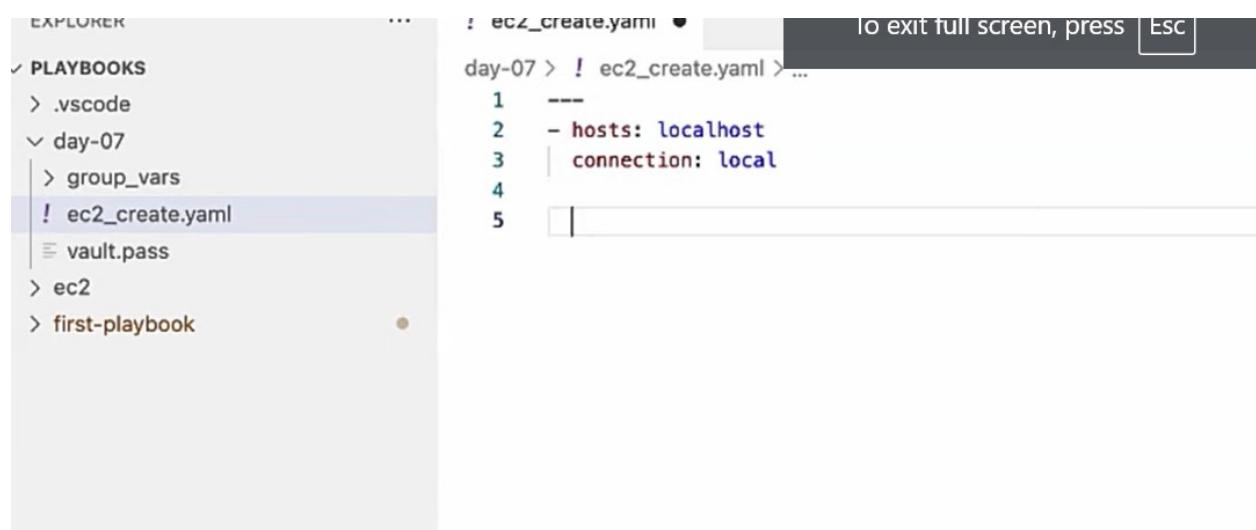
```

---
- hosts: localhost
  connection: local

  tasks:
    - name: Create EC2 instances
      amazon.aws.ec2_instance:
        name: "{{ item.name }}"
        key_name: "abhi-aws-keypair"
        instance_type: t2.micro
        security_group: default
        region: ap-south-1
        aws_access_key: "{{ec2_access_key}}" # From vault as defined
        aws_secret_key: "{{ec2_secret_key}}" # From vault as defined
        network:
          assign_public_ip: true
        image_id: "{{ item.image }}"
        tags:
          environment: "{{ item.name }}"
    loop:
      - { image: "ami-0e1d06225679bc1c5", name: "manage-node-1" } # Update AMI ID according
      - { image: "ami-0f58b397bc5c1f2e8", name: "manage-node-2" } # to your account
      - { image: "ami-0f58b397bc5c1f2e8", name: "manage-node-3" }

```

# executing the ansible in the Control node itself



```

EXPLORER               ...
PLAYBOOKS
> .vscode
day-07
  > group_vars
  ! ec2_create.yaml
  = vault.pass
> ec2
> first-playbook

```

```

: ec2_create.yaml  ▾ To exit full screen, press Esc
day-07 > ! ec2_create.yaml >...
1  ---
2  - hosts: localhost
3  | connection: local
4
5

```

# now how to write the task – Collection .. go to Google .. Ansible Doc .. collection

## # Create a Module Link

[https://docs.ansible.com/ansible/latest/collections/amazon/aws/ec2\\_instance\\_module.html#ansible-collections-amazon-aws-ec2-instance-module](https://docs.ansible.com/ansible/latest/collections/amazon/aws/ec2_instance_module.html#ansible-collections-amazon-aws-ec2-instance-module)

## # Example

The screenshot shows a browser window with the URL [docs.ansible.com/ansible/latest/collections/amazon/aws/ec2\\_instance\\_module.html#ansible-collections-amazon-aws-ec2-instance-module](https://docs.ansible.com/ansible/latest/collections/amazon/aws/ec2_instance_module.html#ansible-collections-amazon-aws-ec2-instance-module). The page title is "Ansible Community Documentation". On the left, there's a sidebar with a list of collections in various namespaces. The main content area shows Ansible playbooks for managing EC2 instances using the `amazon.aws.ec2_instance` module. One example playbook is shown:

```
    TELLERS:
      instance_state_name: running
      - name: restart a particular instance by its ID
        amazon.aws.ec2_instance:
          state: restarted
          instance_ids:
            - i-12345678
      - 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-5ca1abfe
          instance_type: t5.large
          security_group: default
          network:
            assign_public_ip: true
          image_id: ami-123456
          tags:
            Environment: Testing
      - name: start an instance and Add EBS
        amazon.aws.ec2_instance:
          name: "public-with-ebs-instance"
          vpc_subnet_id: subnet-5ca1abfe
          instance_type: t2.micro
          key_name: "prod-ssh-key"
          security_group: default
          volumes:
            - device_name: /dev/sda1
              ebs:
                volume_size: 16
```

The screenshot shows the Visual Studio Code editor with an Ansible playbook file named `ec2_create.yaml` open. The file contains the following YAML code:

```
day-07 > ec2_create.yaml
1 --- 
2   - hosts: localhost
3     connection: local
4
5     tasks:
6       - name: Create EC2 instances
7         amazon.aws.ec2_instance:
8           name: "ansible-instance"
9             # key_name: "prod-ssh-key"
10            # vpc_subnet_id: subnet-013744e41e8088axx
11            instance_type: t2.micro
12            security_group: default
13            region: us-east-1
14            aws_access_key: "{{ec2_access_key}}" # From vault as defined
15            aws_secret_key: "{{ec2_secret_key}}" # From vault as defined
16            network:
17              assign_public_ip: true
18              image_id: ami-04b70fa74e45c3917
19            tags:
20              Environment: Testing
21
```

# We need 3 ec2 instance , , name should be unique ..

# key pair

Key pairs (1/2) Info						
<input type="button" value="C"/> Actions ▾ <input type="button" value="Create key pair"/>						
<input type="text" value="Find Key Pair by attribute or tag"/>						
Name	Type	Created	Fingerprint	ID		
ansible-key-pair	rsa	2024/06/20 20:03 GMT+5:30	7c:33:a3:9c:dd:a5:62:46:c6:4b:c5:...	key-0702b46d4c3854b8f		
<input checked="" type="checkbox"/> abhi-aws-keypair	rsa	2024/05/28 18:08 GMT+5:30	e8:01:c9:81:63:81:96:76:ec:27:ee:...	key-06dceafdb17ff1077		

PLAYBOOKS

- > .vscode
- └ day-07
  - > group\_vars
  - └ ec2\_create.yaml
  - └ vault.pass
- > ec2
- > first-playbook

```
day-07 > ⌂ ec2_create.yaml
  3 | connection: local
  4 |
  5 | tasks:
  6 |   - name: Create EC2 instances
  7 |     amazon.aws.ec2_instance:
  8 |       name: "ansible-instance"
  9 |       key_name: "abhi-aws-keypair"
 10 |       instance_type: t2.micro
 11 |       security_group: default
 12 |       region: ap-south-1
 13 |       aws_access_key: "{{ec2_access_key}}" # From vault as defined
 14 |       aws_secret_key: "{{ec2_secret_key}}" # From vault as defined
 15 |       network:
 16 |         assign_public_ip: true
 17 |       image_id: ami-04b70fa74e45c3917
 18 |
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS AZURE

abhishekveeramalla@aveerama-mac day-07 %

# Loops .. eliminate repetitive task

The screenshot shows a code editor with a sidebar on the left containing a file tree. The tree includes a .vscode folder, a day-07 folder which contains group\_vars, ec2\_create.yaml (which is selected), vault.pass, ec2, and first-playbook. The main pane displays the contents of the ec2\_create.yaml file, which defines a task to create EC2 instances using the amazon.aws.ec2\_instance module. The terminal at the bottom shows the command abhishekveeramalla@aveerama-mac day-07 %

```
PLAYBOOKS
> .vscode
< day-07
| > group_vars
|   ec2_create.yaml
|   vault.pass
| > ec2
| > first-playbook

day-07 > ec2_create.yaml

4
5     tasks:
6       - name: Create EC2 instances
7         amazon.aws.ec2_instance:
8           name: "ansible-instance"
9           key_name: "abhi-aws-keypair"
10          instance_type: t2.micro
11          security_group: default
12          region: ap-south-1
13          aws_access_key: "{{ec2_access_key}}" # From vault as defined
14          aws_secret_key: "{{ec2_secret_key}}" # From vault as defined
15          network:
16            assign_public_ip: true
17            image_id: ami-04b70fa74e45c3917
18          loop:
19            - "ami-0e1d06225679bc1c5"
20            - ""
21            - ""

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS    AZURE
abhishekveeramalla@aveerama-mac day-07 %
```

# Go to AWS Console .. Lauch instance ..

1<sup>st</sup> Linux... Check the AMI ID '

**2<sup>nd</sup> & 3<sup>rd</sup> .. Check AMI ID in the ubuntu**

☰



Browse more AMIs 

Including AMIs from AWS, Marketplace and the Community

**Amazon Machine Image (AMI)**

Amazon Linux 2023 AMI ami-0e1d06225679bc1c5 (64-bit (x86), uefi-preferred) / ami-09db444d0eeafae0e (64-bit (Arm), uefi) Virtualization: hvm ENA enabled: true Root device type: ebs	Free tier eligible
---	--------------------

Description

Amazon Linux 2023 AMI 2023.4.20240611.0 x86\_64 HVM kernel-6.1

Architecture	Boot mode	AMI ID
64-bit (x86)	uefi-preferred	ami-0e1d06225679bc1c5 

▼ Instance type [Info](#) | [Get advice](#)

t2.micro	Free tier eligible
Family: t2 1 vCPU 1 GiB Memory Current generation: true On-Demand Linux base pricing: 0.0124 USD per Hour	

 All generations

▼ Summary

Number of instances	<a href="#">Info</a>
1	

Software Image (AMI)

Amazon Linux 2023 AMI 2023.4.2... [read more](#)  
ami-0e1d06225679bc1c5

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 X  
750 hours of t2.micro (or t3.micro is unavailable) instance usage on free

[Cancel](#) [Launch instance](#) [Review commands](#)

Virtualization: hvm ENA enabled: true Root device type: ebs

Description  
Canonical, Ubuntu, 24.04 LTS, amd64 noble image build on 2024-04-23

Architecture AMI ID  
64-bit (x86) ami-0f58b397bc5c1f2e8 Verified provider

▼ Instance type [Info](#) | [Get advice](#)

Instance type  
t2.micro Free tier eligible  
Family: t2 1 vCPU 1 GiB Memory Current generation: true  
On-Demand Linux base pricing: 0.0124 USD per Hour  
On-Demand Windows base pricing: 0.017 USD per Hour  
On-Demand RHEL base pricing: 0.0724 USD per Hour  
On-Demand SUSE base pricing: 0.0124 USD per Hour

All generations Compare instance types

Additional costs apply for AMIs with pre-installed software

▼ Key pair (login) [Info](#)

▼ Summary

Number of instances [Info](#)  
1

Software Image (AMI)  
Canonical, Ubuntu, 24.04 LTS, ...[read more](#)  
ami-0f58b397bc5c1f2e8

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 X

[Cancel](#) [Launch instance](#)

The screenshot shows the VS Code interface with the following details:

- EXPLORER**: Shows the project structure with `PLAYBOOKS` expanded, listing files like `.vscode`, `day-07`, `group_vars`, `ec2_create.yaml` (selected), `vault.pass`, `ec2`, and `first-playbook`.
- Editor Panel**: Displays the contents of `ec2_create.yaml` (line numbers 6-23). The code defines a task to create EC2 instances using the `amazon.aws.ec2_instance` module.
- Bottom Navigation Bar**: Includes tabs for PROBLEMS (3), OUTPUT, DEBUG CONSOLE, TERMINAL (selected), PORTS, and AZURE.
- Terminal**: Shows the command `abhishekveeramalla@aveerama-mac: day-07 %` in the terminal.

EXPLORER

PLAYBOOKS

- .vscode
- day-07
  - group\_vars
  - ec2\_create.yaml**
  - vault.pass
- ec2
- first-playbook

... **ec2\_create.yaml**

day-07 > **ec2\_create.yaml**

```
4
5   tasks:
6     - name: Create EC2 instances
7       amazon.aws.ec2_instance:
8         name: "ansible-instance"
9         key_name: "abhi-aws-keypair"
10        instance_type: t2.micro
11        security_group: default
12        region: ap-south-1
13        aws_access_key: "{{ec2_access_key}}" # From vault as defined
14        aws_secret_key: "{{ec2_secret_key}}" # From vault as defined
15        network:
16          assign_public_ip: true
17          image_id: "{{ item }}"
18
19      loop:
20        - "ami-0e1d06225679bc1c5"
21        - "ami-0f58b397bc5c1f2e8"
22        - "ami-0f58b397bc5c1f2e8"
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS AZURE

abhishekveeramalla@aveerama-mac day-07 %

The screenshot shows the VS Code interface with the following details:

- Project Structure:** On the left, there's a tree view with items: > group\_vars, ec2\_create.yaml (selected), vault.pass, > ec2, and > first-playbook.
- Code Editor:** The main area displays an Ansible playbook named `ec2_create.yaml`. The code defines a task to create EC2 instances using the `amazon.aws.ec2_instance` module. It specifies parameters like instance type (t2.micro), security group, region (ap-south-1), and network settings (assigning a public IP). It also uses variables from the `vault.pass` file for AWS access and secret keys.
- Terminal:** Below the editor, the terminal window shows the command: `abhishekveeramalla@aveerama-mac day-07 % ansible-playbook ec2_create.yaml --vault-password-file vault.pass`.
- Output:** The terminal output shows the play and task execution. The play runs on [localhost]. The first task, `[Gathering Facts]`, succeeds with `ok: [localhost]`. The second task, `[Create EC2 instances]`, also succeeds with `ok: [localhost]`. The output includes the changed status and item IDs for each host.

Instances (2) <a href="#">Info</a>		<a href="#">C</a>	Connect	Instance state	Actions	Launch instances	
<input type="text"/> Find Instance by attribute or tag (case-sensitive)		<a href="#">Running</a>					
<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Avg.
<input type="checkbox"/>	ansible-instance	i-0ae8da517ea12a98d	<span>Running</span> <a href="#">Q</a> <a href="#">Q</a>	t2.micro	<span>Initializing</span> <a href="#">I</a>	<a href="#">View alarms</a> <a href="#">+</a>	ap-southeast-1
<input type="checkbox"/>	ansible-instance	i-0b5480bb5cdb4d319	<span>Running</span> <a href="#">Q</a> <a href="#">Q</a>	t2.micro	<span>Initializing</span> <a href="#">I</a>	<a href="#">View alarms</a> <a href="#">+</a>	ap-southeast-1

# Only 2 instance Got executed . not the 3<sup>rd</sup> one...

Idempotency in Ansible is **the concept that an operation will produce the same result whether it's executed once or multiple times without any intervening actions**. This is because most Ansible modules check if the desired state has already been achieved, and exit without performing any actions if it has.

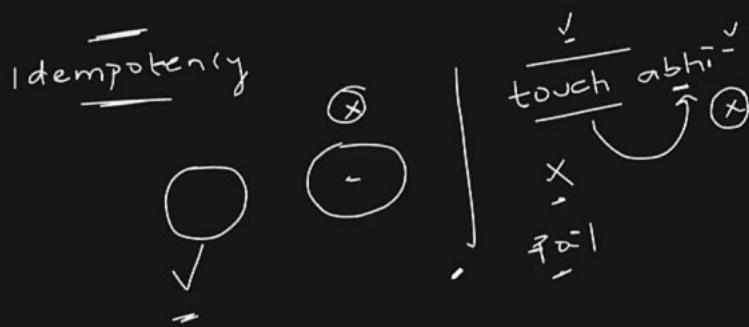
Here are some things to know about idempotency in Ansible:

- Idempotent modules: Modules that behave in this way are often called "idempotent".
- Testing: If you're unsure if your playbooks are idempotent, you can test them in a sandbox environment before running them in production.
- Not all modules and playbooks are idempotent: Not all Ansible modules and playbooks are idempotent.

Idempotency is a property of certain operations or API requests. It ensures that the outcome of an operation is the same, no matter how many times it's executed under identical conditions.

### Desired state and 'idempotency'

Most Ansible modules check whether the desired final state has already been achieved, and exit without performing any actions if that state has been achieved, so that repeating the task does not change the final state. Modules that behave this way are often called 'idempotent.'



EXPLORER

PLAYBOOKS

- .vscode
- day-07
- group\_vars
- ec2\_create.yaml**
- vault.pass
- ec2
- first-playbook

ec2\_create.yaml

```

1  ---
2  - hosts: localhost
3    connection: local
4
5  tasks:
6    - name: Create EC2 instances
7      amazon.aws.ec2_instance:
8        name: "{{ item.name }}"
9        key_name: "abhi-aws-keypair"
10       instance_type: t2.micro
11       security_group: default
12       region: ap-south-1
13       aws_access_key: "{{ec2_access_key}}" # From vault as defined
14       aws_secret_key: "{{ec2_secret_key}}" # From vault as defined
15       network:
16         assign_public_ip: true
17       image_id: "{{ item.image }}"
18
19   loop:
20     - { image: "ami-0e1d06225679bc1c5", name: "manage-node-1" }
21     - { image: "ami-0f58b397bc5c1f2e8", name: "manage-node-2" }
22     - { image: "ami-0f58b397bc5c1f2e8", name: "manage-node-3" }

```

# now terminate the instances

Instances (2/2) Info

Terminate instances?

On an EBS-backed instance, the default action is for the root EBS volume to be deleted when the instance is terminated. Storage on any local drives will be lost.

Are you sure you want to terminate these instances?

Instance ID	Termination protection
i-0ae8da517ea12a98d (ansible-instance)	Disabled
i-0b5480bb5cdb4d319 (ansible-instance)	Disabled

To confirm that you want to terminate the instances, choose the terminate button below. Instances with termination protection enabled will not be terminated. Terminating the instance cannot be undone.

Cancel    **Terminate**

# now run the Code

```
Code File Edit Selection View Go Run Terminal Window Help
← → playbooks

EXPLORER
PLAYBOOKS
  .vscode
  day-07
    group_vars
      ec2_create.yaml
    vault.pass
  ec2
  first-playbook

ec2_create.yaml ×
day-07 > ec2_create.yaml
1 ---
2   - hosts: localhost
3     connection: local
4
5     tasks:
6       - name: Create EC2 instances
7         amazon.aws.ec2_instance:
8           name: "{{ item.name }}"
9           key_name: "abhi-aws-keypair"
10          instance_type: t2.micro
11          security_group: default
12          region: ap-south-1
13          aws_access_key: "{{ec2_access_key}}" # From vault as defined
14          aws_secret_key: "{{ec2_secret_key}}" # From vault as defined
15          network:
16            assign_public_ip: true
17            image_id: "{{ item.image }}"
18          loop:
19            - { image: "ami-0e1d06225679hc1c5", name: "manage-node-1" }

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS AZURE
Python - day-07

abhishekveeramalla@aveerama-mac day-07 % ansible-playbook ec2_create.yaml --vault-password-file vault.pass
PLAY [localhost] ****
TASK [Gathering Facts] ****
```

@use Tags ... Vey ry important for image name & item

```

EXPLORER          ...  ec2_create.yaml X
PLAYBOOKS
> .vscode
< day-07
| > group_vars
|   ec2_create.yaml
|   vault.pass
> ec2
> first-playbook

day-07 >  ec2_create.yaml
  - hosts: localhost
    connection: local

  tasks:
    - name: Create EC2 instances
      amazon.aws.ec2_instance:
        name: "{{ item.name }}"
        key_name: "abhi-aws-keypair"
        instance_type: t2.micro
        security_group: default
        region: ap-south-1
        aws_access_key: "{{ec2_access_key}}" # From vault as defined
        aws_secret_key: "{{ec2_secret_key}}" # From vault as defined
        network:
          | assign_public_ip: true
          image_id: "{{ item.image }}"
        tags:
          | environment: "{{ item.name }}"
    loop:
      - { image: "ami-0e1d06225679bc1c5", name: "manage-node-1" }
      - { image: "ami-0f58b397bc5c1f2e8", name: "manage-node-2" }
      - { image: "ami-0f58b397bc5c1f2e8", name: "manage-node-3" }

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  AZURE
zsh - day-07

TASK [Create EC2 instances] *****
changed: [localhost] => (item={'image': 'ami-0e1d06225679bc1c5', 'name': 'manage-node-1'})
changed: [localhost] => (item={'image': 'ami-0f58b397bc5c1f2e8', 'name': 'manage-node-2'})
changed: [localhost] => (item={'image': 'ami-0f58b397bc5c1f2e8', 'name': 'manage-node-3'})

PLAY RECAP *****
localhost                  : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
abhishekveeramalla@aveerama-mac day-07 %

```

# 3 instances available

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Z.
manage-node-1	i-02898cdf3e7d6f38b	Running	t2.micro	2/2 checks passed	View alarms +	ap-south-1a
manage-node-2	i-0ad5217610a09989f	Running	t2.micro	2/2 checks passed	View alarms +	ap-south-1a
manage-node-3	i-0ff00fd85d3bf227	Running	t2.micro	2/2 checks passed	View alarms +	ap-south-1a

## Task 2

Set up passwordless authentication between Ansible control node and newly created instances.

The screenshot shows a Microsoft Edge browser window with several tabs open. The active tab is a GitHub page titled "How to setup Passwordless Authentication". The page content includes sections on "EC2 Instances" and "Using Public Key". A code block shows the command "ssh-copy-id -f -o IdentityFile <PATH TO PEM FILE>" followed by a list of bullet points explaining its parameters. Another section on "Using Password" lists steps to modify sshd\_config. The browser's toolbar and taskbar are visible at the bottom.

The screenshot shows a terminal window on a Mac OS X desktop. It displays the command "ssh-copy-id -f -o IdentityFile ~/Downloads/abhi-aws-keypair.pem" being run, followed by the output of the command showing it successfully installed a key. The terminal then prompts for a password, which is entered, and shows a successful SSH connection to an EC2 instance at 13.234.32.37. The session ends with a "Last login" message and a prompt for "[ec2-user@ip-172-31-37-206 ~]\$". The desktop background shows a person's face.

```
abhishekveeramalla@aveerama-mac day-07 % ssh ubuntu@43.205.127.222
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1008-aws x86_64)

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

System information as of Thu Jun 20 16:11:54 UTC 2024

System load:  0.0          Processes:           105
Usage of /:   23.2% of 6.71GB  Users logged in:    0
Memory usage: 19%          IPv4 address for enX0: 172.31.47.24
Swap usage:   0%          

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Thu Jun 20 15:51:42 2024 from 49.204.21.74
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
```

### Task 3

Automate the shutdown of Ubuntu Instances only using Ansible Conditionals

Hint: Use `when` condition on ansible `gather_facts`

```
# create a file that stop the instance automatically
```

The screenshot shows the VS Code interface with the following details:

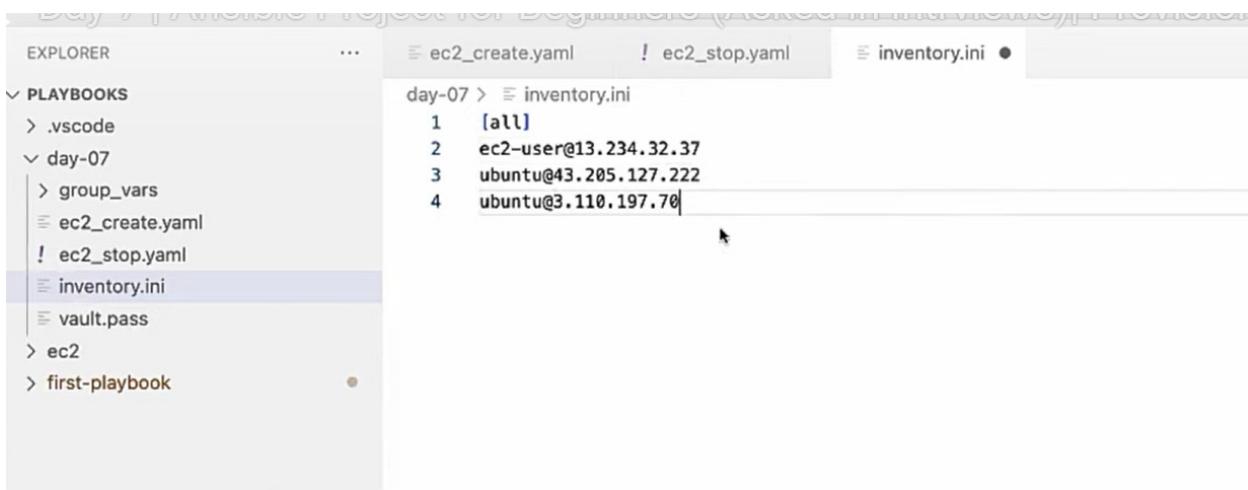
- EXPLORER:** Shows a tree view of files and folders. Under "PLAYBOOKS", there are "day-07", "group\_vars", "ec2\_create.yaml", "ec2\_stop.yaml", "vault.pass", "ec2", and "first-playbook".
- EC2 CREATE YML:** The file "ec2\_create.yaml" is open in the editor. It contains the following YAML code:

```
version: '2'
- hosts: all
  tasks:
    - name: Stop EC2 instance
      aws_ec2_stop:
        instance_ids: "{{ instance_ids }}"
        region: us-east-1
```
- EC2 STOP YML:** The file "ec2\_stop.yaml" is open in the editor. It contains the following YAML code:

```
version: '2'
- hosts: all
  tasks:
    - name: Stop EC2 instance
      aws_ec2_stop:
        instance_ids: "{{ instance_ids }}"
        region: us-east-1
```

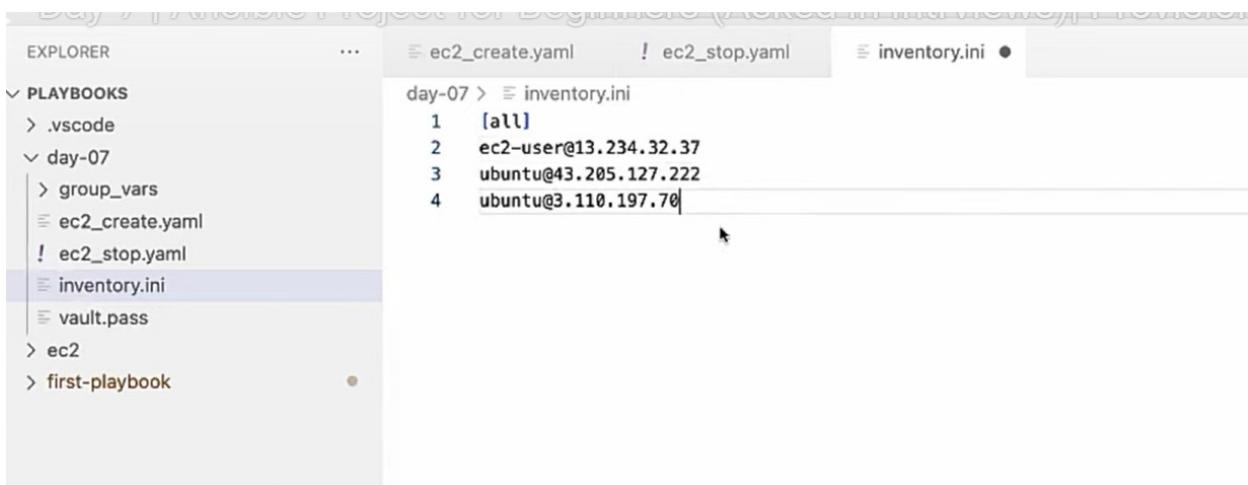
Instances (1/3) <a href="#">Info</a>		<a href="#">C</a>	<a href="#">Connect</a>	<a href="#">Instance state ▾</a>	<a href="#">Actions ▾</a>	<a href="#">Launch instances</a> ▾
<a href="#">Q. Find Instance by attribute or tag (case-sensitive)</a>				<a href="#">Running ▾</a>		
Instance type ▾	Status check	Alarm status	Availability Zone ▾	Public IPv4 DNS	Public IPv4 ... ▾	Elastic IP
t2.micro	<span>2/2 checks passed</span> <a href="#">View alarms</a> +		ap-south-1a	ec2-13-234-32-37.ap-s...	13.234.32.37	-
t2.micro	<span>2/2 checks passed</span> <a href="#">View alarms</a> +		ap-south-1a	ec2-43-205-127-222.ap...	43.205.127.222	-
t2.micro	<span>2/2 checks passed</span> <a href="#">View alarms</a> +		ap-south-1a	ec2-3-110-197-70.ap-s...	3.110.197.70	-

# Same Folder .. Create a Inventory file



```
day-07 > inventory.ini
1 [all]
2 ec2-user@13.234.32.37
3 ubuntu@43.205.127.222
4 ubuntu@3.110.197.70
```

# Now Playbook File .. Stop instances



```
day-07 > inventory.ini
1 [all]
2 ec2-user@13.234.32.37
3 ubuntu@43.205.127.222
4 ubuntu@3.110.197.70
```

### Task 3

Automate the shutdown of Ubuntu Instances only using Ansible Conditionals

Hint: Use `when condition` on ansible `gather_facts`

## When Condition

```
PLAYBOOKS
> .vscode
day-07 >  ec2_stop.yaml
1  ---
2  - hosts: all
3    become: true
4
5  tasks:
6    - name: Shutdown ubuntu instances only
7      ansible.builtin.command: /sbin/shutdown -t now
8

Execute commands on targets
Description
• The ansible.builtin.command module takes the command name followed by a list of space-delimited arguments.
• The given command will be executed on all selected nodes.
• The command(s) will not be processed through the shell, so variables like $ and operations like "", <, >, |, ;, and & will not work. Use the ansible.builtin.shell module if you need these features.
• To create tasks that are easier to read than the ones using space-delimited arguments, pass parameters using the task keyword or use parameter.
• Either a free form command or parameter is required, see the examples.
• For Windows targets, use the ansible.windows.win_command module instead.

Notes
• If you want to run a command through the shell (say you are using <, >, |, and so on), you actually want the
```

## When Clause is used

```
PLAY [all] ****
ERROR! Attempting to decrypt but no vault secrets found
```

The screenshot shows the VS Code interface with the following details:

- Left Sidebar (PLAYBOOKS):** Shows the project structure with files: .vscode, day-07, group\_vars/all, ec2\_create.yaml, ec2\_stop.yaml, inventory.ini, vault.pass, ec2, and first-playbook.
- Center Area:** A code editor window titled "day-07 > ec2\_stop.yaml" containing the following YAML code:

```
---  
- hosts: all  
  become: true  
  
tasks:  
  - name: Shutdown ubuntu instances only  
    ansible.builtin.command: /sbin/shutdown -t now  
    when:  
      - ansible_facts['os.family'] == "Debian"
```
- Bottom Terminal:** The terminal tab is active, showing the command: `ansible-playbook -i inventory.ini ec2_stop.yaml --vault-password-file vault.pass`. The output shows the playbook running on all hosts, gathering facts, and shutting down the Ubuntu instances.

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS AZURE  
Python - day-07 + ✎ ⌂ ⌂ ... ⌂  
abhishekveeramalla@averama-mac day-07 % ansible-playbook -i inventory.ini ec2_stop.yaml --vault-password-file vault.pass  
PLAY [all] *****  
TASK [Gathering Facts] *****  
PLAY [all] *****  
TASK [Gathering Facts] *****  
PLAY [all] *****  
TASK [Gathering Facts] *****  
[WARNING]: Platform linux on host ec2-user@13.234.32.37 is using the discovered Python interpreter at /usr/bin/python3.9, but future installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.16/reference_appendices/interpreter_discovery.html for more information.  
ok: [ec2-user@13.234.32.37]  
ok: [ubuntu@3.110.197.70]  
ok: [ubuntu@43.205.127.222]  
  
TASK [Shutdown ubuntu instances only] *****  
changed: [ubuntu@3.110.197.70]  
changed: [ubuntu@43.205.127.222]  
changed: [ec2-user@13.234.32.37]  
  
PLAY RECAP *****  
ec2-user@13.234.32.37 : ok=2 changed=1 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0  
ubuntu@3.110.197.70 : ok=2 changed=1 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0  
ubuntu@43.205.127.222 : ok=2 changed=1 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
```

# all are Running ..

EC2 Dashboard	Instances (1/3) <a href="#">Info</a>						
EC2 Global View	<a href="#">Find Instance by attribute or tag (case-sensitive)</a>						
Events	<a href="#">Running</a>						
Instances	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability
Instances	<input type="checkbox"/> manage-node-1	i-02898cdf3e7d6f58b	<span style="color: green;">Running</span>  	t2.micro	<span style="color: green;">2/2 checks passed</span>	<a href="#">View alarms +</a>	ap-south-1
Instances	<input type="checkbox"/> manage-node-2	i-0ad5217610a09989f	<span style="color: green;">Running</span>  	t2.micro	<span style="color: green;">2/2 checks passed</span>	<a href="#">View alarms +</a>	ap-south-1
Instances	<input checked="" type="checkbox"/> manage-node-3	i-04ff00fd85d3bf227	<span style="color: green;">Running</span>  	t2.micro	<span style="color: green;">2/2 checks passed</span>	<a href="#">View alarms +</a>	ap-south-1

```
# now check where is my Fault ???
```

Ansible collects pretty much all the information about the remote hosts as it runs a playbook. The task of collecting this remote system information is called as Gathering Facts by ansible and the details collected are generally known as facts or variables.

In Ansible, the `gather_facts` feature collects information from target systems before tasks or playbooks are executed. This information is stored in variables called facts, which can then be used in playbooks.

Here are some things to know about Ansible facts:

- When facts are gathered: Ansible gathers facts at the start of each play execution, unless the `gather_facts` option is disabled.
- What facts are gathered: Facts include information about a host's filesystem, network interfaces, operating system, and hardware specifications.
- Where facts are stored: Facts are stored in the `ansible_facts` variable.
- How to use facts: Facts can be accessed in playbooks.
- How to use the `gather_facts` module: The `gather_facts` module is included in all Ansible installations and can be used by its short module name, `gather_facts`. However, the fully qualified collection name (FQCN) `ansible.builtin.gather_facts` is recommended for easy linking to the module documentation.

```
TASK [Gathering Facts] ****
[WARNING]: Platform linux on host ec2-user@13.234.32.37 is using the discovered Python interpreter at /usr/bin/python3.9, but future
installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-
core/2.16/reference_appendices/interpreter_discovery.html for more information.
ok: [ec2-user@13.234.32.37]
ok: [ubuntu@3.110.197.70]
ok: [ubuntu@43.205.127.222]

TASK [Shutdown ubuntu instances only] ****
changed: [ubuntu@3.110.197.70]
changed: [ubuntu@43.205.127.222]
changed: [ec2-user@13.234.32.37]

PLAY RECAP ****
ec2-user@13.234.32.37      : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu@3.110.197.70        : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu@43.205.127.222     : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

○ abhishekveeramalla@aveerama-mac day-07 %
```

The screenshot shows the VS Code interface with the following details:

- EXPLORER:** Shows a tree view of the project structure:
  - PLAYBOOKS:
    - > .vscode
    - < day-07
      - > group\_vars
      - | ec2\_create.yaml
      - | ec2\_stop.yaml
      - | inventory.ini
      - | vault.pass
    - > ec2
    - > first-playbook
- CODE EDITOR:** The `ec2_stop.yaml` file is open, showing the following YAML code:

```
day-07 >  ec2_stop.yaml
1  ---
2  - hosts: all
3    become: true
4
5    tasks:
6      - name: Print all the ansible gathered ansible_facts
7        ansible.builtin.debug:
8          | var: ansible_facts
9
10   #- name: Shutdown ubuntu instances only
11   # ansible.builtin.command: /sbin/shutdown -t now
12   # when:
13   #   ansible_facts['os.family'] == "Debian"
```
- PROMPT:** The terminal shows the command `abhishekveeramalla@aveerama-mac day-07 %`.

```

> group_vars
└ ec2_create.yaml
  ec2_stop.yaml
  inventory.ini
  vault.pass
> ec2
> first-playbook

4 |   tasks:
5 |     - name: Print all the ansible gathered ansible_facts
6 |       ansible.builtin.debug:
7 |         var: ansible_facts
8 |
9 |
10 |     #-- name: Shutdown ubuntu instances only
11 |     #   ansible.builtin.command: /sbin/shutdown -t now
12 |     #   when:
13 |     #     ansible_facts['os.family'] == "Debian"

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS AZURE

abhishekveeramalla@aveerama-mac day-07 % ansible-playbook -i inventory.ini ec2\_stop.yaml --vault-password-file vault.pass

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

TASK [Gathering Facts] \*\*\*\*

```

< PLAYBOOKS
  > .vscode
  < day-07
    > group_vars
      ec2_create.yaml
      ec2_stop.yaml
      inventory.ini
      vault.pass
    > ec2
    > first-playbook

day-07 > ec2_stop.yaml
1 ---
2   - hosts: all
3     become: true
4

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS AZURE
zsh - day-07

        "block_used": 12491,
        "device": "/dev/xvda15",
        "fstype": "vfat",
        "inode_available": 0,
        "inode_total": 0,
        "inode_used": 0,
        "mount": "/boot/efi",
        "options": "rw,relatime,fmask=0077,dmask=0077,codepage=437,iocharset=iso8859-1,shortname=mixed,errors=1",
        "size_available": 103000064,
        "size_total": 109395456,
        "uuid": "3CC2-C362"
      },
      {
        "nodename": "ip-172-31-42-25",
        "os_family": "Debian",
        "pkg_mgr": "apt",
        "proc_cmdline": [
          {
            "BOOT_IMAGE": "/vmlinuz-6.8.0-1008-aws",
            "console": [
              "tty1",
              "tty50"
            ],
            "nvme_core.io_timeout": "4294967295",
            "panic": "-1",
            "ro": true,
            "root": "PARTUUID=227fdd7f-aae5-441a-80c5-ad2f1c6e5bb1"
          },
          {
            "processor": [
              {
                "GenuineIntel",
                "Intel(R) Xeon(R) CPU E5-2676 v3 @ 2.40GHz"
              }
            ],
            "processor_cores": 1,
            "processor_count": 1
          }
        ]
      }
    
```

# Search os.family

```

inventory.ini
vault.pass
ec2
first-playbook

    "block_available": 201172,
    "block_size": 512,
    "block_total": 213663,
    "block_used": 12491,
    "device": "/dev/xvda15",
    "fstype": "vfat",
    "inode_available": 0,
    "inode_total": 0,
    "inode_used": 0,
    "mount": "/boot/efi",
    "options": "rw,relatime,fmask=0077,dmask=0077,codepage=437,iocharset=iso8859-1,shortname=mixed(errors=remount-ro",
    "size_available": 103000064,
    "size_total": 109395456,
    "uuid": "3CC2-C362"
},
{
    "nodename": "ip-172-31-42-25",
    "os_family": "Debian",
    "pkg_mgr": "apt",
    "proc cmdline": {
        "BOOT_IMAGE": "/vmlinuz-6.8.0-1008-aws",
        "console": [
            "tty1",
            "tty58"
        ],
        "nvme_core.io_timeout": "4294967295",
        "panic": "-1",
        "ro": true,
        "root": "PARTUUID=227fd7f-aae5-441a-80c5-ad2f1c6e5bb1"
}
}

```

EXPLORER

- PLAYBOOKS
  - > .vscode
  - < day-07
    - > group\_vars
    - `ec2_create.yaml`
    - `ec2_stop.yaml` (selected)
    - `inventory.ini`
    - `vault.pass`
  - > ec2
  - > first-playbook

... `ec2_create.yaml` `ec2_stop.yaml` `inventory.ini`

```

day-07 > ec2_stop.yaml
1 ---
2   - hosts: all
3     become: true
4
5     tasks:
6       - name: Shutdown ubuntu instances only
7         ansible.builtin.command: /sbin/shutdown -t now
8         when:
9           | ansible_facts['os_family'] == "Debian"

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS AZURE

zsh - day-07 + □ ☰ ⌂ ⌄ ⌅ ⌆ ⌇ ⌈ ⌉ ⌊ ⌋

```

ok: [ubuntu@13.233.50.68]
[WARNING]: Platform linux on host ec2-user@15.206.117.242 is using the discovered Python interpreter at /usr/bin/python3.9, but future
installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-
core/2.16/reference_appendices/interpreter_discovery.html for more information.
ok: [ec2-user@15.206.117.242]
ok: [ubuntu@13.232.69.230]

TASK [Shutdown ubuntu instances only] ****
skipping: [ec2-user@15.206.117.242] 
changed: [ubuntu@13.233.50.68]
changed: [ubuntu@13.232.69.230]

PLAY RECAP ****
ec2-user@15.206.117.242 : ok=1    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
ubuntu@13.232.69.230   : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu@13.233.50.68   : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

```

abhishekveeramalla@aveerama-mac day-07 %

hamilton-devcluster-openshift-com:6443/kube:admin default [Azurite Queue Service] [Azurite Blob Service] Ln 9, Col 46 Spaces: 2 UTF-8 LF

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with navigation links like EC2 Dashboard, EC2 Global View, Events, Instances, Instance Types, Launch Templates, Spot Requests, and Savings Plans. The main area is titled 'Instances (1/1) Info' and shows a table with one row. The row contains the instance name 'manage-node-1', its ID 'i-02898cdf3e7d6f38b', its state 'Running', its type 't2.micro', and other details like status check and alarm status. A 'Launch instances' button is at the top right.

# now I can Terminate Linux too... Type – Redhart

This screenshot shows a terminal window with an Ansible playbook being run. The terminal interface includes tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, PORTS, and AZURE. The command run is 'ansible-playbook -i inventory.ini ec2\_stop.yaml --vault-password-file vault.pass'. The output shows the play is targeting all hosts and executing tasks to shutdown instances where the OS family is RedHat.

```

PLAY [all] *****
[...]
```

The screenshot shows the 'Instance summary for i-02898cdf3e7d6f38b (manage-node-1)' page. It provides detailed information about the instance, including its ID, state, IP addresses, and various identifiers. The instance is currently running. Key details include its public IP (15.206.117.242), private IP (172.31.37.206), and VPC ID (vpc-07b22e2a35b5ef472).

Attribute	Value
Instance ID	i-02898cdf3e7d6f38b (manage-node-1)
Public IPv4 address	15.206.117.242   open address
Private IPv4 addresses	172.31.37.206
IPv6 address	-
Instance state	Running
Public IPv4 DNS	ec2-15-206-117-242.ap-south-1.compute.amazonaws.com   open address
Hostname type	IP name: ip-172-31-37-206.ap-south-1.compute.internal
Private IP DNS name (IPv4 only)	ip-172-31-37-206.ap-south-1.compute.internal
Instance type	t2.micro
Elastic IP addresses	-
Answer private resource DNS name	-
VPC ID	vpc-07b22e2a35b5ef472
AWS Compute Optimizer finding	Opt-in to AWS Compute Optimizer for recommendations.   Learn more
Auto-assigned IP address	15.206.117.242 [Public IP]
Subnet ID	subnet-082ce1cba216c0775
IAM Role	-
Auto Scaling Group name	-
IMDSv2	Instance ARN

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS AZURE zsh - day-07 + □ 🗑 ... ⓘ
❶ abhishekveeramalla@aveerama-mac day-07 % ansible-playbook -i inventory.ini ec2_stop.yaml --vault-password-file vault.pass
PLAY [all] ****
TASK [Gathering Facts] ****
[WARNING]: Platform linux on host ec2-user@15.206.117.242 is using the discovered Python interpreter at /usr/bin/python3.9, but future installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.16/reference_appendices/interpreter_discovery.html for more information.
ok: [ec2-user@15.206.117.242]
fatal: [ubuntu@13.233.50.68]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: ssh: connect to host 13.233.50.68 port 22: Operation timed out", "unreachable": true}
fatal: [ubuntu@13.232.69.230]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: ssh: connect to host 13.232.69.230 port 22: Operation timed out", "unreachable": true}

TASK [Shutdown ubuntu instances only] ****
changed: [ec2-user@15.206.117.242]

PLAY RECAP ****
ec2-user@15.206.117.242 : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu@13.232.69.230   : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0    ignored=0
ubuntu@13.233.50.68   : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0    ignored=0
❷ abhishekveeramalla@aveerama-mac day-07 %
```

The screenshot shows the AWS EC2 Instances page. The left sidebar includes options like Dashboard, EC2 Global View, Events, Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity, Reservations (marked as New), and Images. The main content area has tabs for Instances and Info, with the Instances tab selected. It features a search bar with placeholder text "Find Instance by attribute or tag (case-sensitive)" and a status filter set to "All states". A button labeled "Instance state = running" is highlighted with a blue border. Below the search bar is a "Clear filters" button. The main table header includes columns for Name, Instance ID, Instance st..., Instance type, and Status check. A message "No matching instances found" is displayed at the bottom of the table area.