# \*\*Study Material: Day 3 - Writing Your First Ansible Playbook\*\*

---

## \*\*Table of Contents\*\*

1. \*\*Introduction to Ansible Playbooks\*\*

- What is an Ansible Playbook?

- Why Use Playbooks?

2. \*\*Anatomy of a Playbook\*\*

- Playbook Structure

- Key Components: Hosts, Tasks, Modules

3. \*\*Writing Your First Playbook\*\*

- Example: Install Apache on Managed Nodes

- Explanation of Each Step

4. \*\*Running a Playbook\*\*

- Command to Execute a Playbook

- Verifying the Output

5. \*\*Best Practices for Writing Playbooks\*\*

- Use of Variables

- Idempotency

- Error Handling

6. \*\*Conclusion and Next Steps\*\*

---

## \*\*1. Introduction to Ansible Playbooks\*\*

### \*\*What is an Ansible Playbook?\*\*

An Ansible playbook is a YAML file that defines a set of tasks to be executed on managed nodes. Playbooks are used to automate complex tasks, such as configuring servers, deploying applications, and managing infrastructure.

### \*\*Why Use Playbooks?\*\*

- \*\*Reusability\*\*: Playbooks can be reused across different environments and projects.

- \*\*Idempotency\*\*: Playbooks ensure that tasks are only executed if necessary, making them safe to run multiple times.

- \*\*Collaboration\*\*: Playbooks can be version-controlled and shared among team members.

---

## \*\*2. Anatomy of a Playbook\*\*

### \*\*Playbook Structure\*\*

A playbook consists of one or more "plays." Each play defines:

- \*\*Hosts\*\*: The managed nodes where the tasks will be executed.

- \*\*Tasks\*\*: The actions to be performed on the hosts.

- \*\*Modules\*\*: The tools used to perform the tasks.

### \*\*Key Components\*\*

- \*\*Hosts\*\*: Specifies the target hosts (e.g., `all`, `app`, `db`).

- \*\*Tasks\*\*: A list of actions to be performed.

- \*\*Modules\*\*: Predefined tools (e.g., `apt`, `yum`, `copy`) used to perform tasks.

---

## \*\*3. Writing Your First Playbook\*\*

### \*\*Example: Install Apache on Managed Nodes\*\*

```yaml

---

- name: Install Apache on Managed Nodes

hosts: all

become: yes

tasks:

- name: Ensure Apache is installed

apt:

name: apache2

state: present

- name: Ensure Apache is running

service:

name: apache2

state: started

enabled: yes

```

### \*\*Explanation of Each Step\*\*

1. \*\*`name`\*\*: A description of the playbook.

2. \*\*`hosts`\*\*: Specifies the target hosts (e.g., `all` for all hosts in the inventory).

3. \*\*`become`\*\*: Allows the playbook to run with elevated privileges (e.g., `sudo`).

4. \*\*`tasks`\*\*: A list of tasks to be executed.

- \*\*Task 1\*\*: Install Apache using the `apt` module.

- \*\*Task 2\*\*: Ensure Apache is running and enabled on boot using the `service` module.

---

## \*\*4. Running a Playbook\*\*

### \*\*Command to Execute a Playbook\*\*

```bash

ansible-playbook -i inventory.ini playbook.yml

```

### \*\*Verifying the Output\*\*

- Check if Apache is installed and running on the managed nodes:

```bash

ansible -i inventory.ini -m shell -a "systemctl status apache2" all

```

---

## \*\*5. Best Practices for Writing Playbooks\*\*

### \*\*Use of Variables\*\*

- Variables make playbooks more flexible and reusable.

- Example:

```yaml

- name: Install Apache on Managed Nodes

hosts: all

become: yes

vars:

apache\_package: apache2

tasks:

- name: Ensure Apache is installed

apt:

name: "{{ apache\_package }}"

state: present

```

### \*\*Idempotency\*\*

- Ensure that tasks are idempotent (i.e., running the playbook multiple times does not change the system state if the desired state is already achieved).

- Example: The `apt` module ensures that Apache is installed only if it is not already present.

### \*\*Error Handling\*\*

- Use the `ignore\_errors` directive to handle errors gracefully.

- Example:

```yaml

- name: Attempt to start a non-existent service

service:

name: non-existent-service

state: started

ignore\_errors: yes

```

---

## \*\*6. Conclusion and Next Steps\*\*

### \*\*What We Learned\*\*

- \*\*Playbooks\*\*: YAML files that define tasks to be executed on managed nodes.

- \*\*Anatomy of a Playbook\*\*: Hosts, tasks, and modules.

- \*\*Writing a Playbook\*\*: Example of installing Apache on managed nodes.

- \*\*Running a Playbook\*\*: Command to execute a playbook and verify the output.

- \*\*Best Practices\*\*: Use of variables, idempotency, and error handling.

### \*\*Next Steps\*\*

- \*\*Day 4\*\*: Deep dive into Ansible roles and their folder structure.

- \*\*Day 5\*\*: Advanced playbook techniques, including loops and conditionals.

---

## \*\*Images and Tables\*\*

### \*\*Playbook Example\*\*

```yaml

---

- name: Install Apache on Managed Nodes

hosts: all

become: yes

tasks:

- name: Ensure Apache is installed

apt:

name: apache2

state: present

- name: Ensure Apache is running

service:

name: apache2

state: started

enabled: yes

```

### \*\*Command to Run a Playbook\*\*

```bash

ansible-playbook -i inventory.ini playbook.yml

```

---

## \*\*References\*\*

- [Ansible Documentation](https://docs.ansible.com/)

- [Ansible Playbooks Guide](https://docs.ansible.com/ansible/latest/user\_guide/playbooks.html)

- [Ansible Modules](https://docs.ansible.com/ansible/latest/modules/modules\_by\_category.html)

---

This study material provides a comprehensive guide to writing your first Ansible playbook, including examples, best practices, and next steps. It is designed to help you understand the core concepts and apply them in real-world scenarios.  
  
  
  
  
**Study Material: Day 3 - Writing Your First Ansible Playbook**

**Table of Contents**

1. **Introduction to Ansible Playbooks**
   * What is an Ansible Playbook?
   * Why Use Playbooks?
2. **Anatomy of a Playbook**
   * Playbook Structure
   * Key Components: Hosts, Tasks, Modules
3. **Writing Your First Playbook**
   * Example: Install Apache on Managed Nodes
   * Explanation of Each Step
4. **Running a Playbook**
   * Command to Execute a Playbook
   * Verifying the Output
5. **Best Practices for Writing Playbooks**
   * Use of Variables
   * Idempotency
   * Error Handling
6. **Conclusion and Next Steps**

**1. Introduction to Ansible Playbooks**

**What is an Ansible Playbook?**

An Ansible playbook is a YAML file that defines a set of tasks to be executed on managed nodes. Playbooks are used to automate complex tasks, such as configuring servers, deploying applications, and managing infrastructure.

**Why Use Playbooks?**

* **Reusability**: Playbooks can be reused across different environments and projects.
* **Idempotency**: Playbooks ensure that tasks are only executed if necessary, making them safe to run multiple times.
* **Collaboration**: Playbooks can be version-controlled and shared among team members.

**2. Anatomy of a Playbook**

**Playbook Structure**

A playbook consists of one or more "plays." Each play defines:

* **Hosts**: The managed nodes where the tasks will be executed.
* **Tasks**: The actions to be performed on the hosts.
* **Modules**: The tools used to perform the tasks.

**Key Components**

* **Hosts**: Specifies the target hosts (e.g., all, app, db).
* **Tasks**: A list of actions to be performed.
* **Modules**: Predefined tools (e.g., apt, yum, copy) used to perform tasks.

**3. Writing Your First Playbook**

**Example: Install Apache on Managed Nodes**

---

- name: Install Apache on Managed Nodes

hosts: all

become: yes

tasks:

- name: Ensure Apache is installed

apt:

name: apache2

state: present

- name: Ensure Apache is running

service:

name: apache2

state: started

enabled: yes

**Explanation of Each Step**

1. **name**: A description of the playbook.
2. **hosts**: Specifies the target hosts (e.g., all for all hosts in the inventory).
3. **become**: Allows the playbook to run with elevated privileges (e.g., sudo).
4. **tasks**: A list of tasks to be executed.
   * **Task 1**: Install Apache using the apt module.
   * **Task 2**: Ensure Apache is running and enabled on boot using the service module.

**4. Running a Playbook**

**Command to Execute a Playbook**

ansible-playbook -i inventory.ini playbook.yml

**Verifying the Output**

* Check if Apache is installed and running on the managed nodes:
* ansible -i inventory.ini -m shell -a "systemctl status apache2" all

**5. Best Practices for Writing Playbooks**

**Use of Variables**

* Variables make playbooks more flexible and reusable.
* Example:
* - name: Install Apache on Managed Nodes
* hosts: all
* become: yes
* vars:
* apache\_package: apache2
* tasks:
* - name: Ensure Apache is installed
* apt:
* name: "{{ apache\_package }}"
* state: present

**Idempotency**

* Ensure that tasks are idempotent (i.e., running the playbook multiple times does not change the system state if the desired state is already achieved).
* Example: The apt module ensures that Apache is installed only if it is not already present.

**Error Handling**

* Use the ignore\_errors directive to handle errors gracefully.
* Example:
* - name: Attempt to start a non-existent service
* service:
* name: non-existent-service
* state: started
* ignore\_errors: yes

**6. Conclusion and Next Steps**

**What We Learned**

* **Playbooks**: YAML files that define tasks to be executed on managed nodes.
* **Anatomy of a Playbook**: Hosts, tasks, and modules.
* **Writing a Playbook**: Example of installing Apache on managed nodes.
* **Running a Playbook**: Command to execute a playbook and verify the output.
* **Best Practices**: Use of variables, idempotency, and error handling.

**Next Steps**

* **Day 4**: Deep dive into Ansible roles and their folder structure.
* **Day 5**: Advanced playbook techniques, including loops and conditionals.

**References**

* [Ansible Documentation](https://docs.ansible.com/)
* [Ansible Playbooks Guide](https://docs.ansible.com/ansible/latest/user_guide/playbooks.html)
* [Ansible Modules](https://docs.ansible.com/ansible/latest/modules/modules_by_category.html)

This study material provides a comprehensive guide to writing your first Ansible playbook, including examples, best practices, and next steps. It is designed to help you understand the core concepts and apply them in real-world scenarios.