# \*\*Study Material: Day 2 - Passwordless Authentication, Ansible Inventory, and Ad Hoc Commands\*\*

---

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

1. \*\*Passwordless Authentication\*\*

- What is Passwordless Authentication?

- Why is it a Prerequisite for Ansible?

- Two Ways to Implement Passwordless Authentication

2. \*\*Ansible Inventory\*\*

- What is Ansible Inventory?

- Inventory File Formats

- Grouping Servers in Inventory

3. \*\*Ansible Ad Hoc Commands\*\*

- What are Ad Hoc Commands?

- Use Cases for Ad Hoc Commands

- Examples of Ad Hoc Commands

4. \*\*Conclusion and Next Steps\*\*

---

## \*\*1. Passwordless Authentication\*\*

### \*\*What is Passwordless Authentication?\*\*

Passwordless authentication is a mechanism that allows a control node (where Ansible is installed) to connect to managed nodes (servers being managed) without requiring a password or SSH key every time. This is achieved by setting up SSH keys or enabling password-based authentication for the first time and then allowing future connections without manual intervention.

### \*\*Why is it a Prerequisite for Ansible?\*\*

- \*\*Automation\*\*: Ansible needs to connect to multiple servers to execute tasks. If it asks for a password or SSH key every time, automation is blocked.

- \*\*Efficiency\*\*: Passwordless authentication ensures that Ansible can run tasks without human intervention, making the process seamless.

### \*\*Two Ways to Implement Passwordless Authentication\*\*

1. \*\*Using SSH Keys\*\*:

- Generate SSH keys on the control node.

- Copy the public key to the managed nodes using the `ssh-copy-id` command.

- Example:

```bash

ssh-copy-id -i ~/.ssh/id\_rsa.pub ubuntu@<public-ip>

```

- After this, the control node can SSH into the managed node without a password.

2. \*\*Using Password\*\*:

- Enable password authentication on the managed node (if disabled by default, e.g., on AWS EC2 instances).

- Create a password for the user (e.g., `ubuntu`).

- Use `ssh-copy-id` with the password:

```bash

ssh-copy-id ubuntu@<public-ip>

```

- Enter the password when prompted.

- After this, the control node can SSH into the managed node without a password.

---

## \*\*2. Ansible Inventory\*\*

### \*\*What is Ansible Inventory?\*\*

Ansible inventory is a file that lists the managed nodes (servers) that Ansible will manage. It tells Ansible which servers to connect to and execute tasks on.

### \*\*Inventory File Formats\*\*

- \*\*INI Format\*\*:

```ini

[app]

ubuntu@<public-ip-1>

ubuntu@<public-ip-2>

[db]

ubuntu@<public-ip-3>

ubuntu@<public-ip-4>

```

- \*\*YAML Format\*\*:

```yaml

app:

hosts:

ubuntu@<public-ip-1>

ubuntu@<public-ip-2>

db:

hosts:

ubuntu@<public-ip-3>

ubuntu@<public-ip-4>

```

### \*\*Grouping Servers in Inventory\*\*

- You can group servers in the inventory file to target specific sets of servers (e.g., app servers, database servers).

- Example:

```ini

[app]

ubuntu@<public-ip-1>

ubuntu@<public-ip-2>

[db]

ubuntu@<public-ip-3>

ubuntu@<public-ip-4>

```

- To run a command on all app servers:

```bash

ansible -i inventory.ini -m ping app

```

---

## \*\*3. Ansible Ad Hoc Commands\*\*

### \*\*What are Ad Hoc Commands?\*\*

Ad hoc commands are one-liner commands used to perform quick tasks on managed nodes without writing a full playbook. They are useful for simple tasks like checking connectivity, installing packages, or managing files.

### \*\*Use Cases for Ad Hoc Commands\*\*

- \*\*Ping Test\*\*: Check connectivity to managed nodes.

```bash

ansible -i inventory.ini -m ping all

```

- \*\*Install Packages\*\*: Install software on managed nodes.

```bash

ansible -i inventory.ini -m apt -a "name=openjdk-11-jdk state=present" all

```

- \*\*Manage Files\*\*: Create, delete, or modify files.

```bash

ansible -i inventory.ini -m file -a "path=/tmp/test.txt state=touch" all

```

- \*\*Run Shell Commands\*\*: Execute shell commands on managed nodes.

```bash

ansible -i inventory.ini -m shell -a "ls /etc" all

```

### \*\*Examples of Ad Hoc Commands\*\*

1. \*\*Ping Test\*\*:

```bash

ansible -i inventory.ini -m ping all

```

2. \*\*Install Apache\*\*:

```bash

ansible -i inventory.ini -m apt -a "name=apache2 state=present" all

```

3. \*\*List Files in `/etc`\*\*:

```bash

ansible -i inventory.ini -m shell -a "ls /etc" all

```

---

## \*\*4. Conclusion and Next Steps\*\*

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

- \*\*Passwordless Authentication\*\*: Essential for Ansible to automate tasks without manual intervention.

- \*\*Ansible Inventory\*\*: A file that lists managed nodes and groups them for targeted tasks.

- \*\*Ad Hoc Commands\*\*: Quick, one-liner commands for simple tasks.

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

- \*\*Day 3\*\*: Writing your first Ansible playbook.

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

---

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

### \*\*Ansible Inventory Example\*\*

```ini

[app]

ubuntu@192.168.1.1

ubuntu@192.168.1.2

[db]

ubuntu@192.168.1.3

ubuntu@192.168.1.4

```

### \*\*Ad Hoc Command Syntax\*\*

```bash

ansible -i <inventory-file> -m <module> -a "<arguments>" <target>

```

---

## \*\*References\*\*

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

- [SSH Key Authentication](https://www.ssh.com/academy/ssh/key)

- [Ansible Ad Hoc Commands](https://docs.ansible.com/ansible/latest/user\_guide/intro\_adhoc.html)

---

This study material provides a detailed explanation of passwordless authentication, Ansible inventory, and ad hoc commands. It includes examples, use cases, and next steps to help you master these concepts.

**Study Material: Day 2 - Passwordless Authentication, Ansible Inventory, and Ad Hoc Commands**

**Table of Contents**

1. **Passwordless Authentication**
   * What is Passwordless Authentication?
   * Why is it a Prerequisite for Ansible?
   * Two Ways to Implement Passwordless Authentication
2. **Ansible Inventory**
   * What is Ansible Inventory?
   * Inventory File Formats
   * Grouping Servers in Inventory
3. **Ansible Ad Hoc Commands**
   * What are Ad Hoc Commands?
   * Use Cases for Ad Hoc Commands
   * Examples of Ad Hoc Commands
4. **Conclusion and Next Steps**

**1. Passwordless Authentication**

**What is Passwordless Authentication?**

Passwordless authentication is a mechanism that allows a control node (where Ansible is installed) to connect to managed nodes (servers being managed) without requiring a password or SSH key every time. This is achieved by setting up SSH keys or enabling password-based authentication for the first time and then allowing future connections without manual intervention.

**Why is it a Prerequisite for Ansible?**

* **Automation**: Ansible needs to connect to multiple servers to execute tasks. If it asks for a password or SSH key every time, automation is blocked.
* **Efficiency**: Passwordless authentication ensures that Ansible can run tasks without human intervention, making the process seamless.

**Two Ways to Implement Passwordless Authentication**

1. **Using SSH Keys**:
   * Generate SSH keys on the control node.
   * Copy the public key to the managed nodes using the ssh-copy-id command.
   * Example:
   * ssh-copy-id -i ~/.ssh/id\_rsa.pub ubuntu@<public-ip>
   * After this, the control node can SSH into the managed node without a password.
2. **Using Password**:
   * Enable password authentication on the managed node (if disabled by default, e.g., on AWS EC2 instances).
   * Create a password for the user (e.g., ubuntu).
   * Use ssh-copy-id with the password:
   * ssh-copy-id ubuntu@<public-ip>
   * Enter the password when prompted.
   * After this, the control node can SSH into the managed node without a password.

**2. Ansible Inventory**

**What is Ansible Inventory?**

Ansible inventory is a file that lists the managed nodes (servers) that Ansible will manage. It tells Ansible which servers to connect to and execute tasks on.

**Inventory File Formats**

* **INI Format**:
* [app]
* ubuntu@<public-ip-1>
* ubuntu@<public-ip-2>
* [db]
* ubuntu@<public-ip-3>
* ubuntu@<public-ip-4>
* **YAML Format**:
* app:
* hosts:
* ubuntu@<public-ip-1>
* ubuntu@<public-ip-2>
* db:
* hosts:
* ubuntu@<public-ip-3>
* ubuntu@<public-ip-4>

**Grouping Servers in Inventory**

* You can group servers in the inventory file to target specific sets of servers (e.g., app servers, database servers).
* Example:
* [app]
* ubuntu@<public-ip-1>
* ubuntu@<public-ip-2>
* [db]
* ubuntu@<public-ip-3>
* ubuntu@<public-ip-4>
* To run a command on all app servers:
* ansible -i inventory.ini -m ping app

**3. Ansible Ad Hoc Commands**

**What are Ad Hoc Commands?**

Ad hoc commands are one-liner commands used to perform quick tasks on managed nodes without writing a full playbook. They are useful for simple tasks like checking connectivity, installing packages, or managing files.

**Use Cases for Ad Hoc Commands**

* **Ping Test**: Check connectivity to managed nodes.
* ansible -i inventory.ini -m ping all
* **Install Packages**: Install software on managed nodes.
* ansible -i inventory.ini -m apt -a "name=openjdk-11-jdk state=present" all
* **Manage Files**: Create, delete, or modify files.
* ansible -i inventory.ini -m file -a "path=/tmp/test.txt state=touch" all
* **Run Shell Commands**: Execute shell commands on managed nodes.
* ansible -i inventory.ini -m shell -a "ls /etc" all

**Examples of Ad Hoc Commands**

1. **Ping Test**:
2. ansible -i inventory.ini -m ping all
3. **Install Apache**:
4. ansible -i inventory.ini -m apt -a "name=apache2 state=present" all
5. **List Files in /etc**:
6. ansible -i inventory.ini -m shell -a "ls /etc" all

**4. Conclusion and Next Steps**

**What We Learned**

* **Passwordless Authentication**: Essential for Ansible to automate tasks without manual intervention.
* **Ansible Inventory**: A file that lists managed nodes and groups them for targeted tasks.
* **Ad Hoc Commands**: Quick, one-liner commands for simple tasks.

**Next Steps**

* **Day 3**: Writing your first Ansible playbook.
* **Day 4**: Deep dive into Ansible roles and their folder structure.

**Images and Tables**

**Ansible Inventory Example**

[app]

ubuntu@192.168.1.1

ubuntu@192.168.1.2

[db]

ubuntu@192.168.1.3

ubuntu@192.168.1.4

**Ad Hoc Command Syntax**

ansible -i <inventory-file> -m <module> -a "<arguments>" <target>

**References**

* [Ansible Documentation](https://docs.ansible.com/)
* [SSH Key Authentication](https://www.ssh.com/academy/ssh/key)
* [Ansible Ad Hoc Commands](https://docs.ansible.com/ansible/latest/user_guide/intro_adhoc.html)

This study material provides a detailed explanation of passwordless authentication, Ansible inventory, and ad hoc commands. It includes examples, use cases, and next steps to help you master these concepts.