# **Ansible Complete Notes**

## **1. Introduction to Ansible**

### **What is Ansible?**

Ansible is an open-source **IT automation tool** that automates **configuration management, application deployment, orchestration, and provisioning**.

### **Why Use Ansible?**

* Agentless (No need to install software on target machines)
* Simple YAML-based automation (Uses Playbooks)
* Scalable and efficient
* Works over SSH (No additional ports required)

## **2. Installing Ansible**

### **On Ubuntu/Debian**

sudo apt update

sudo apt install ansible -y

### **On RHEL/CentOS**

sudo yum install epel-release -y

sudo yum install ansible -y

### **On macOS (via Homebrew)**

brew install ansible

### **Check Ansible Version**

ansible --version

## **3. Ansible Inventory**

Ansible manages hosts via an **inventory file** (usually hosts or inventory).

### **Default Inventory Location**

/etc/ansible/hosts

### **Example Inventory File (**inventory.ini**)**

[web\_servers]

192.168.1.10

192.168.1.11

[db\_servers]

192.168.1.20 ansible\_user=root ansible\_password=pass

### **Test Connection to Hosts**

ansible all -m ping

## **4. Ansible Ad-hoc Commands**

### **Check Connectivity**

ansible all -m ping

### **Check Disk Usage**

ansible all -m shell -a 'df -h'

### **Check Memory Usage**

ansible all -m shell -a 'free -m'

### **Create a Directory on Remote Machine**

ansible all -m file -a "path=/opt/demo state=directory mode=0755"

## **5. Ansible Playbooks**

Ansible Playbooks are YAML-based configuration files used to define automation tasks.

### **Basic Playbook Example (**setup.yml**)**

- name: Install Apache and Start Service

hosts: web\_servers

become: yes

tasks:

- name: Install Apache

apt:

name: apache2

state: present

- name: Start Apache Service

service:

name: apache2

state: started

### **Run the Playbook**

ansible-playbook setup.yml

## **6. Ansible Modules**

Modules are used to perform automation tasks.

### **Commonly Used Modules**

| **Module** | **Purpose** |
| --- | --- |
| ping | Checks connectivity |
| copy | Copies files to remote machines |
| file | Creates/deletes files and directories |
| service | Manages system services |
| apt/yum | Installs software packages |
| user | Manages users |
| cron | Schedules cron jobs |

### **Using the Copy Module**

- name: Copy File to Remote Server

hosts: web\_servers

tasks:

- name: Copy index.html

copy:

src: /home/user/index.html

dest: /var/www/html/index.html

## **7. Ansible Variables**

Variables allow dynamic configuration.

### **Example of Variables in a Playbook**

- name: Install Nginx with Variables

hosts: web\_servers

vars:

package\_name: nginx

tasks:

- name: Install Web Server

apt:

name: "{{ package\_name }}"

state: present

## **8. Ansible Handlers**

Handlers run only when notified by a task.

### **Example with Handlers**

- name: Restart Apache when Configuration Changes

hosts: web\_servers

tasks:

- name: Copy Config File

copy:

src: apache.conf

dest: /etc/apache2/apache2.conf

notify: Restart Apache

handlers:

- name: Restart Apache

service:

name: apache2

state: restarted

## **9. Ansible Loops**

Loops allow tasks to run multiple times.

### **Example Loop in Playbook**

- name: Install Multiple Packages

hosts: web\_servers

tasks:

- name: Install Packages

apt:

name: "{{ item }}"

state: present

loop:

- nginx

- git

- curl

## **10. Ansible Roles**

Roles organize playbooks into reusable components.

### **Creating a Role**

ansible-galaxy init my\_role

### **Role Directory Structure**

my\_role/

├── tasks/

│ └── main.yml

├── handlers/

│ └── main.yml

├── templates/

├── files/

├── vars/

│ └── main.yml

├── defaults/

│ └── main.yml

### **Using a Role in a Playbook**

- name: Apply Web Server Role

hosts: web\_servers

roles:

- my\_role

## **11. Ansible Facts**

Ansible gathers system information using **facts**.

### **List System Facts**

ansible all -m setup

### **Using Facts in Playbooks**

- name: Print Hostname

hosts: all

tasks:

- name: Display Hostname

debug:

msg: "Hostname is {{ ansible\_hostname }}"

## **12. Ansible Vault (Secrets Management)**

Ansible Vault encrypts sensitive data like passwords.

### **Encrypt a File**

ansible-vault encrypt secret.yml

### **Decrypt a File**

ansible-vault decrypt secret.yml

### **Edit an Encrypted File**

ansible-vault edit secret.yml

### **Use Vault in Playbooks**

vars\_files:

- secret.yml

## **Conclusion**

Ansible is a powerful automation tool that simplifies IT infrastructure management. 🚀 Let me know if you need any modifications or additional topics! 😊