**Running Ansible Playbooks in Docker on Windows**

Create a Playbook Directory on Windows

1. Open PowerShell and create a directory for playbooks:

Powershell

mkdir C:\ansible-playbooks

2. Inside this directory, create a playbook file (`my\_playbook.yml`):

Powershell

notepad C:\ansible-playbooks\my\_playbook.yml

3. Paste the following playbook (Example: Installing Apache):

yaml

- name: Install Apache Web Server

hosts: webservers

become: yes

tasks:

- name: Install Apache

ansible.builtin.apt:

name: apache2

state: present

4. Save the file.

Run Ansible in a Docker Container

Option 1: Run Once (Without Persisting Changes)

Powershell

docker run --rm -it -v C:\ansible-playbooks:/ansible ghcr.io/linuxserver/ansible bash

- `--rm` → Removes the container after exit.

- `-it` → Runs it interactively.

- `-v C:\ansible-playbooks:/ansible` → Mounts the playbooks folder inside the container.

Option 2: Run a Persistent Ansible Container

To keep a running Ansible container, use:

Powershell

docker run -dit --name ansible-container -v C:\ansible-playbooks:/ansible ghcr.io/linuxserver/ansible

Then, access it anytime:

powershell

docker exec -it ansible-container bash

Run the Playbook

Once inside the container:

1. Change directory to playbooks:

bash

cd /ansible

2. Run the playbook:

bash

ansible-playbook -i inventory.ini my\_playbook.yml

(Make sure to add your `inventory.ini` file in the same directory.)\*

Verify Execution

Check if the web server is running on the remote machine:

bash

curl http://your-server-ip

Ansible Playbook example for each task category.

**1. System Configuration & Management**

Task: Create a User and Set SSH Key

yaml

- name: Create a user and set SSH key

hosts: all

become: yes

tasks:

- name: Create user 'devops'

ansible.builtin.user:

name: devops

state: present

shell: /bin/bash

- name: Set up SSH key for 'devops'

ansible.builtin.authorized\_key:

user: devops

state: present

key: "ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEAr..."

**2. Application Deployment**

Task: Install Apache Web Server & Deploy Website

yaml

- name: Install Apache and deploy a website

hosts: webservers

become: yes

tasks:

- name: Install Apache (Ubuntu/Debian)

ansible.builtin.apt:

name: apache2

state: present

update\_cache: yes

when: ansible\_os\_family == "Debian"

- name: Start and enable Apache

ansible.builtin.service:

name: apache2

state: started

enabled: yes

- name: Deploy custom index.html

ansible.builtin.copy:

content: "<h1>Welcome to Ansible Deployment</h1>"

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

**3. Security & Compliance**

Task: Configure UFW (Firewall)

yaml

- name: Configure UFW Firewall

hosts: all

become: yes

tasks:

- name: Allow SSH and HTTP traffic

ansible.builtin.ufw:

rule: allow

port: "{{ item }}" #Specifies the port (loop variable)

proto: tcp

loop:

- 22 #SSH port

- 80 # HTTP Port

- name: Enable UFW

ansible.builtin.ufw:

state: enabled

**4. Windows Management**

Task: Install Notepad++ Using Chocolatey

yaml

- name: Install Notepad++ on Windows

hosts: windows

tasks:

- name: Install Notepad++ via Chocolatey

win\_chocolatey:

name: notepadplusplus

state: present

**7. Monitoring & Logging**

Task: Install and Configure Prometheus

yaml

- name: Install Prometheus

hosts: monitoring

become: yes

tasks:

- name: Download Prometheus

ansible.builtin.get\_url:

url: https://github.com/prometheus/prometheus/releases/latest/download/prometheus-2.41.0.linux-amd64.tar.gz

dest: /tmp/prometheus.tar.gz

- name: Extract Prometheus

ansible.builtin.unarchive:

src: /tmp/prometheus.tar.gz

dest: /opt/

remote\_src: yes

- name: Start Prometheus Service

ansible.builtin.command:

cmd: nohup /opt/prometheus/prometheus --config.file=/opt/prometheus/prometheus.yml &

**8. Backup & Recovery**

Task: Backup MySQL Database

yaml

- name: Backup MySQL Database

hosts: database\_servers

become: yes

tasks:

- name: Create MySQL dump

ansible.builtin.command:

cmd: mysqldump -u root -p'yourpassword' mydatabase > /backup/mydatabase.sql