To configure an \*\*Ansible Master and Slave\*\* (Managed Node) on the same system using \*\*Docker\*\* in Windows, follow these steps:

Prerequisites\*\*

- Install \*\*Docker Desktop\*\* on Windows.

- Enable \*\*WSL 2 Backend\*\* in Docker Desktop settings.

- Install \*\*Ansible\*\* inside a container (as Windows doesn’t support Ansible natively).

Step 1: Pull the Base Image\*\*

We will use an \*\*Ubuntu-based container\*\* for Ansible.

Open \*\*PowerShell\*\* or \*\*Command Prompt\*\* and run:

docker pull ubuntu:latest

Step 2: Create Ansible Master (Control Node)\*\*

Now, create a container for the Ansible master.

docker run -dit --name ansible-master ubuntu:latest

Once the container is running, \*\*access it\*\*:

docker exec -it ansible-master bash

Inside the container, update packages:

apt update && apt install -y software-properties-common

Then, install Ansible:

apt update && apt install -y python3 python3-pip

apt install -y ansible

Verify installation:

ansible --version

Step 3: Create Ansible Slave (Managed Node)\*\*

Now, create another container for the Ansible-managed node:

docker run -dit --name ansible-slave ubuntu:latest

Step 4: Configure SSH for Ansible Master\*\*

Inside the \*\*ansible-master\*\* container, install OpenSSH:

apt install -y openssh-client openssh-server

Generate an SSH key:

ssh-keygen -t rsa -b 4096

(Enter file in which to save the key (C:\Users\ELCOT/.ssh/id\_rsa): rrr (can be any name)

Enter passphrase (empty for no passphrase):

Enter same passphrase again:

Your identification has been saved in rrr

Your public key has been saved in rrr.pub)

Step 5: Install and Configure SSH on the Slave\*\*

Access the \*\*ansible-slave\*\* container:

docker exec -it ansible-slave bash

Inside the container, install OpenSSH:

apt update && apt install -y openssh-server

mkdir -p /root/.ssh

chmod 700 /root/.ssh

Step 6: Set Up SSH Key-Based Authentication\*\*

Copy the SSH public key from \*\*ansible-master\*\* to \*\*ansible-slave\*\*.

From the \*\*ansible-master\*\* container:

bash

docker cp ~/rrr.pub ansible-slave:/root/.ssh/authorized\_keys

(~ should be replaced with full path (eg: C:/Users/ELCOT/)

Now, inside the \*\*ansible-slave\*\* container:

bash

chmod 600 /root/.ssh/authorized\_keys

If the previous command shows error the follow these steps

Check if the file is present:

bash

CopyEdit

ls -l /root/.ssh/

If the file is named incorrectly (e.g., authorized\_keysc instead of authorized\_keys), rename it:

bash

CopyEdit

mv /root/.ssh/authorized\_keysc /root/.ssh/authorized\_keys

service ssh start

Test SSH connection from \*\*ansible-master\*\*:

bash

ssh root@ansible-slave

If prompted about authenticity, type \*\*yes\*\* and press Enter.

In ansible-slave set passoword

# passwd root

Then change password authentication permission to “yes”

sed -i 's/^#\?PermitRootLogin .\*/PermitRootLogin yes/' /etc/ssh/sshd\_config

sed -i 's/^#\?PasswordAuthentication .\*/PasswordAuthentication yes/' /etc/ssh/sshd\_config

Step 7: Configure Ansible Inventory\*\*

Inside \*\*ansible-master\*\*, edit the inventory file:

bash

echo "ansible-slave ansible\_host=ansible-slave ansible\_user=root" > /etc/ansible/hosts

Test connectivity:

bash

ansible ansible-slave -m ping

If successful, you will see \*\*pong\*\* responses.

Step 8: Run an Ansible Command\*\*

Try running a simple command:

bash

ansible ansible-slave -m shell -a "uptime"

SAMPLE Ansible Playbook to test your setup.

Step 1: Create a Playbook File\*\*

Inside the \*\*ansible-master\*\* container, create a new playbook file:

nano /root/setup.yml

YAML code:

yaml

- name: Test Ansible Playbook

hosts: ansible-slave

become: yes

tasks:

- name: Update package cache

apt:

update\_cache: yes

- name: Install Nginx

apt:

name: nginx

state: present

- name: Start Nginx service

service:

name: nginx

state: started

Save the file (Ctrl + X → Y → Enter).

Step 2: Run the PlaybookExecute the playbook using:

ansible-playbook /root/setup.yml

If everything is set up correctly, Ansible will:

✅ Update the package cache

✅ Install Nginx on the \*\*ansible-slave\*\*

✅ Start the Nginx service

Step 3: Verify the Installation

On the ansible-slave, check if Nginx is running:

docker exec -it ansible-slave bash

systemctl status nginx