**JINI ZACHARIAS - 500221149**

**Group 1: Installing and Configuring Ansible**

* **Objective**: Set up Ansible on a Linux machine and understand basic configuration.
* **Tasks**:
  1. **Research Ansible Installation**: Understand Ansible’s purpose and prerequisites, especially on Linux-based systems.
  2. **Install Ansible**:
     + **On Ubuntu/Debian**:

bash

Copy code

sudo apt update

sudo apt install ansible -y

A screenshot of a computer

Description automatically generated

**A screenshot of a computer

Description automatically generated**

* + - **On CentOS/RHEL**:

bash

Copy code

sudo yum install epel-release -y

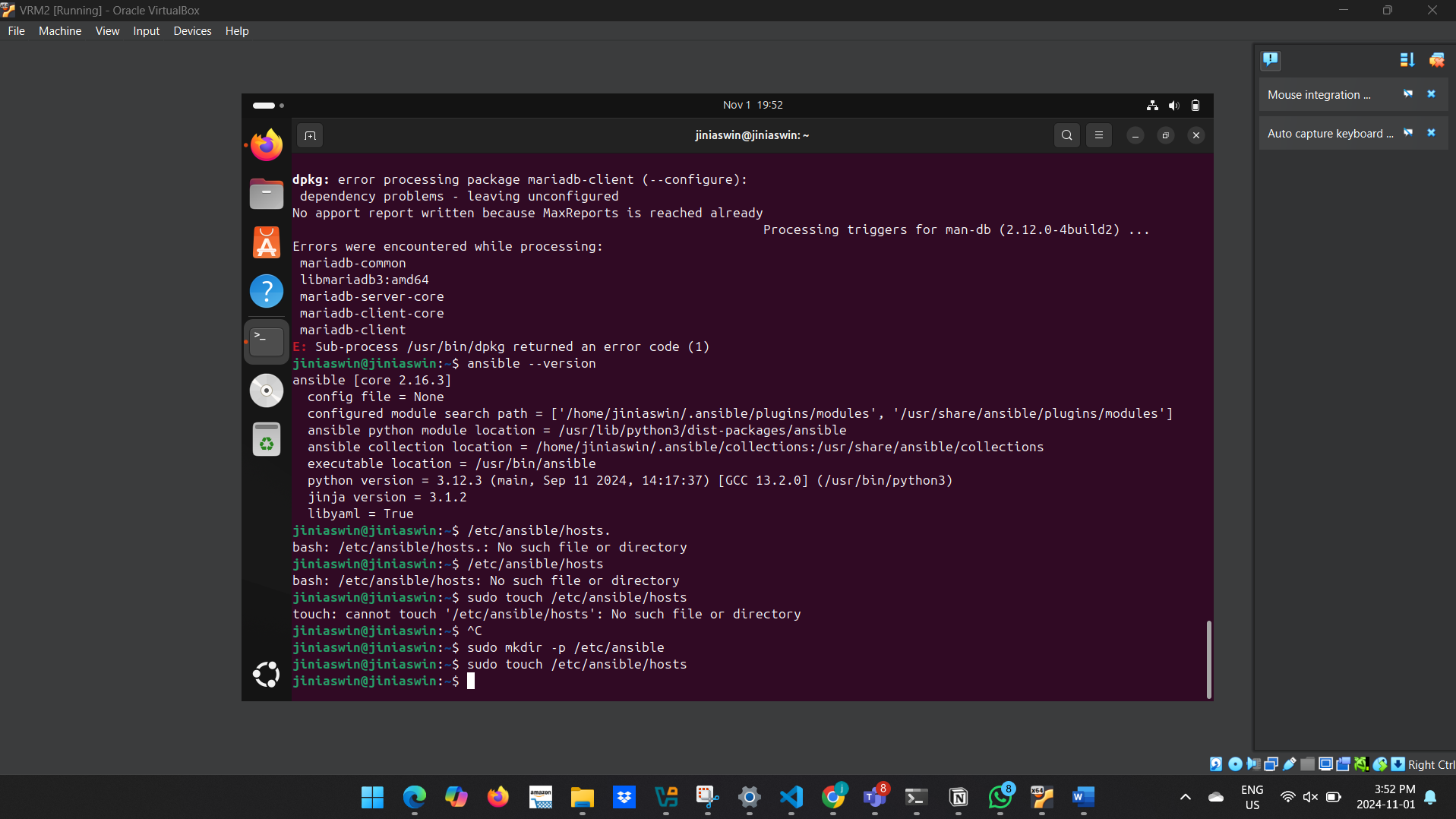
sudo yum install ansible -y

* 1. **Verify Installation**: Run ansible --version to confirm Ansible is correctly installed.

**A screenshot of a computer program

Description automatically generated**

* 1. **Set Up Inventory File**:
     + Create an inventory file at /etc/ansible/hosts.



* + - Add localhost to the inventory for local testing:

ini

Copy code

[local]

localhost ansible\_connection=local

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

* 1. **Document the Installation**:
     + Note any issues or system requirements.
     + Prepare a guide on the steps taken and why each step is essential for using Ansible.
* **Output**: A documented process of Ansible installation and configuration with troubleshooting notes.

**Group 2: Writing a Simple Python Application**

* **Objective**: Develop a basic Python application that will be deployed via Ansible.
* **Tasks**:
  1. **Write a Simple Python Script**:
     + Create a file called app.py with this code:

python

Copy code

print("Hello from Ansible!")

A screenshot of a computer

Description automatically generated

* + - Test the script by running:

bash

Copy code

python3 app.py

A screenshot of a computer

Description automatically generated

* 1. **Extend Functionality (Optional)**:
     + Add features such as accepting user input or reading data from a file.
     + Example (optional):

python

Copy code

name = input("Enter your name: ")

print(f"Hello, {name}! This is an Ansible-deployed app.")

A screenshot of a computer

Description automatically generated

* 1. **Test the Application**: Ensure the script runs correctly on the group’s Linux system.

A screenshot of a computer program

Description automatically generated

* 1. **Share with Other Groups**: Pass the Python script file to the next group for deployment setup.
* **Output**: A working Python script file (app.py) ready for Ansible deployment.

**Group 3: Writing Ansible Playbook for Application Deployment**

* **Objective**: Write an Ansible playbook to deploy the Python application created by Group 2.
* **Tasks**:
  1. **Set Up the Playbook**:
     + Create a file called deploy\_app.yml for the playbook.

A computer screen shot of a computer screen

Description automatically generated

* + - Write the playbook as follows:

yaml

Copy code

---

- name: Deploy Python Application

hosts: local

tasks:

- name: Ensure Python is installed

apt:

name: python3

state: present

become: true

- name: Copy Python script to the target machine

copy:

src: ./app.py

dest: /home/{{ ansible\_user }}/app.py

- name: Run the Python script

command: python3 /home/{{ ansible\_user }}/app.py

* 1. **Explanation of Each Task**:
     + **Ensure Python is Installed**: Installs Python if it's not present (for Debian-based systems; use yum for CentOS).
     + **Copy Python Script**: Transfers the script created by Group 2 to the home directory on the target machine.
     + **Run the Python Script**: Executes the script to display output.
  2. **Test the Playbook**:
     + Run the playbook using ansible-playbook deploy\_app.yml.
     + If issues arise, update and refine the playbook based on testing.

A screenshot of a computer

Description automatically generated

* **Output**: A completed Ansible playbook (deploy\_app.yml) with documentation on each task.