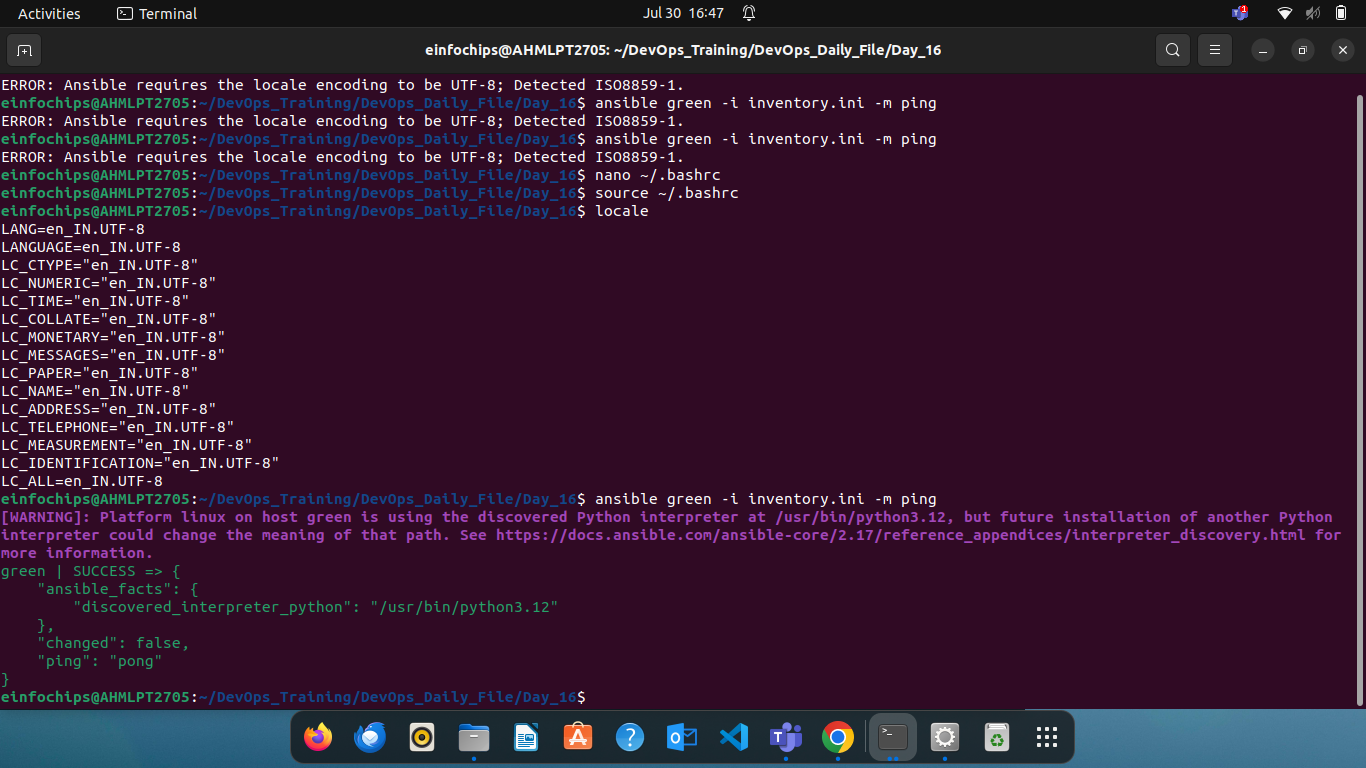
**Project is on last page**

**Problem Statement:** You are tasked with deploying Ansible in a multi-node environment consisting of multiple Linux servers. The goal is to set up Ansible on a control node and configure it to manage several managed nodes. This setup will be used for automating system administration tasks across the network.

**Deliverables:**

1. **Control Node Setup:**
2. **Managed Nodes Configuration:**



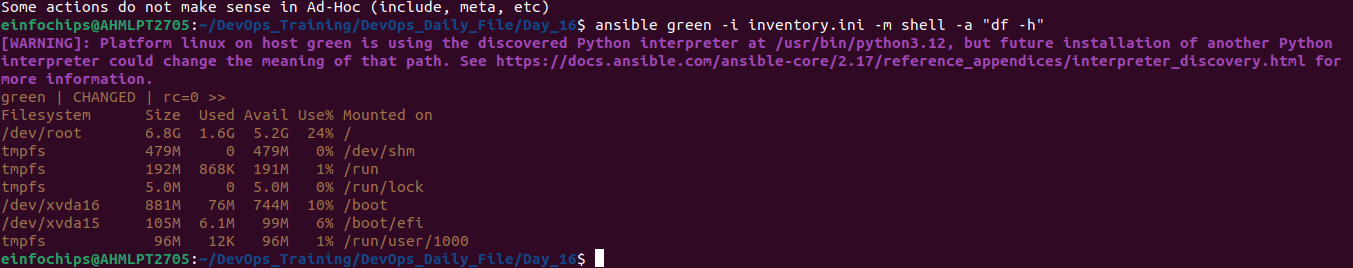
### **Project 2: Ad-Hoc Ansible Commands**

**Problem Statement:** Your organization needs to perform frequent, one-off administrative tasks across a fleet of servers. These tasks include checking disk usage, restarting services, and updating packages. You are required to use Ansible ad-hoc commands to accomplish these tasks efficiently.

**Deliverables:**

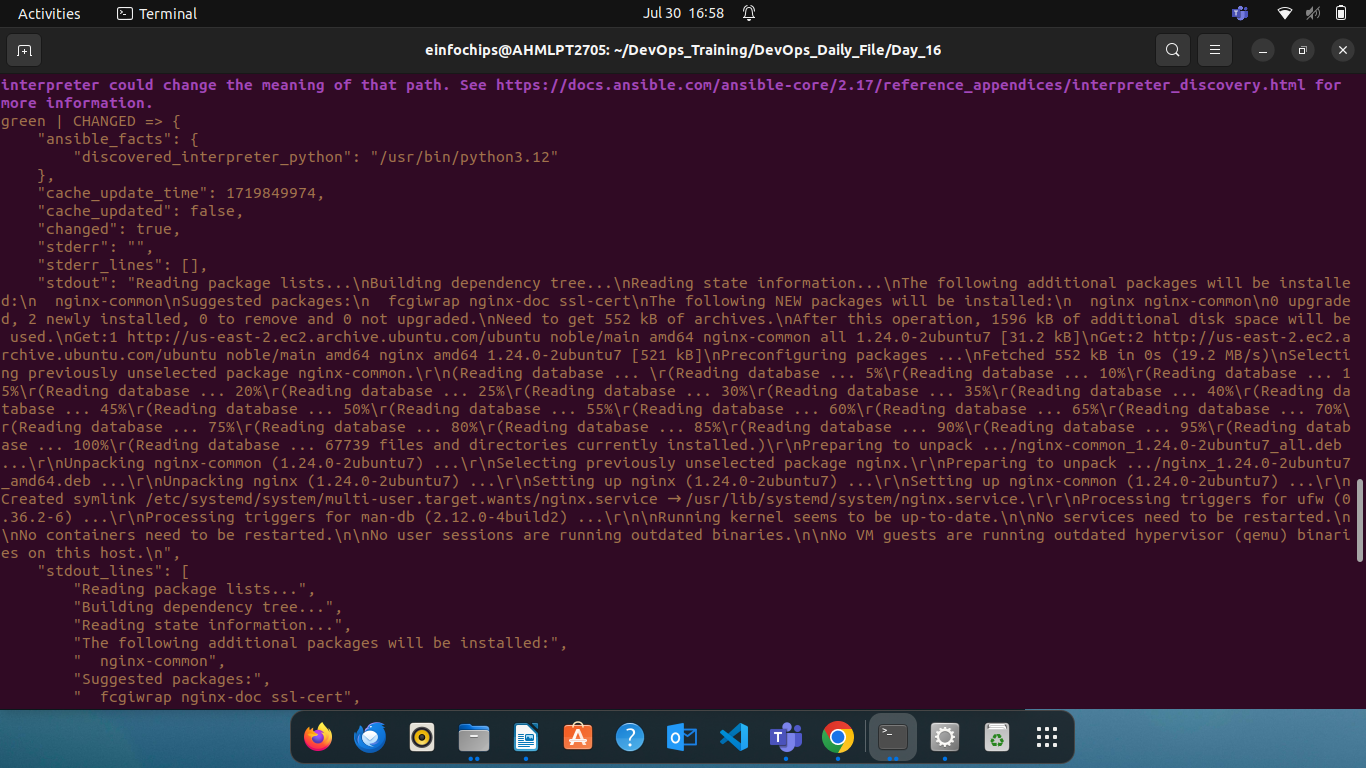
1. **Task Execution:**

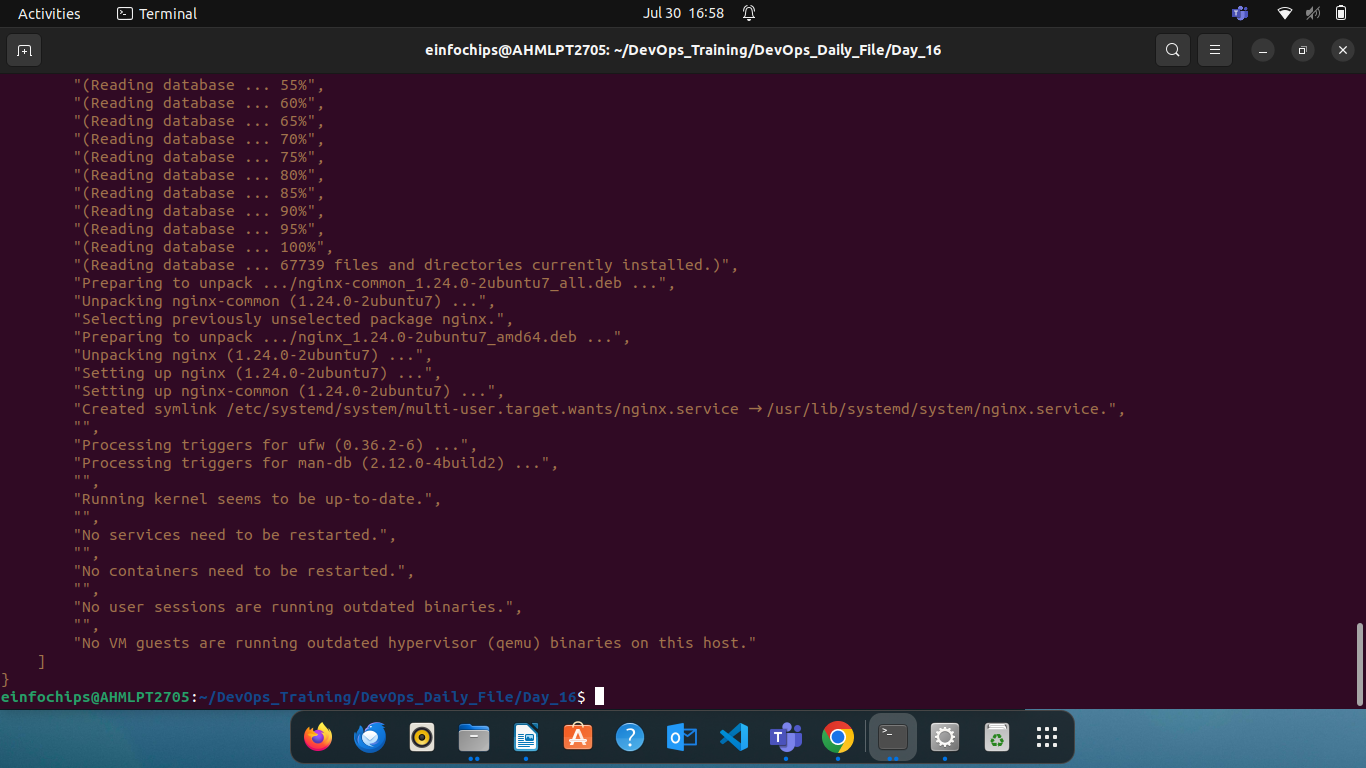
**ansible green -i inventory.ini -m shell -a “df -h”**



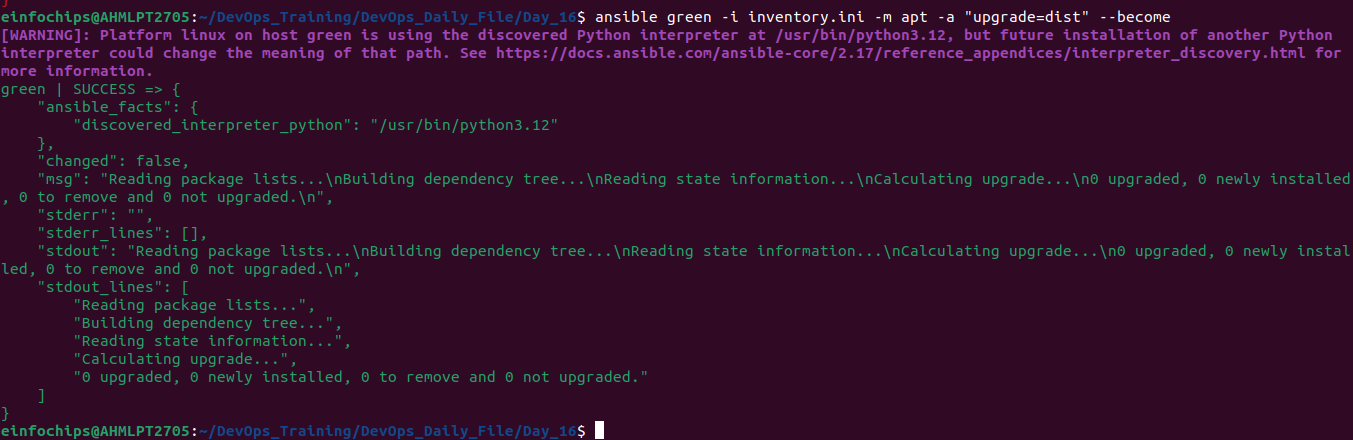
* + Restart a specific service on all managed nodes.

**Ansible green -i inventory.ini -m shell -a “**

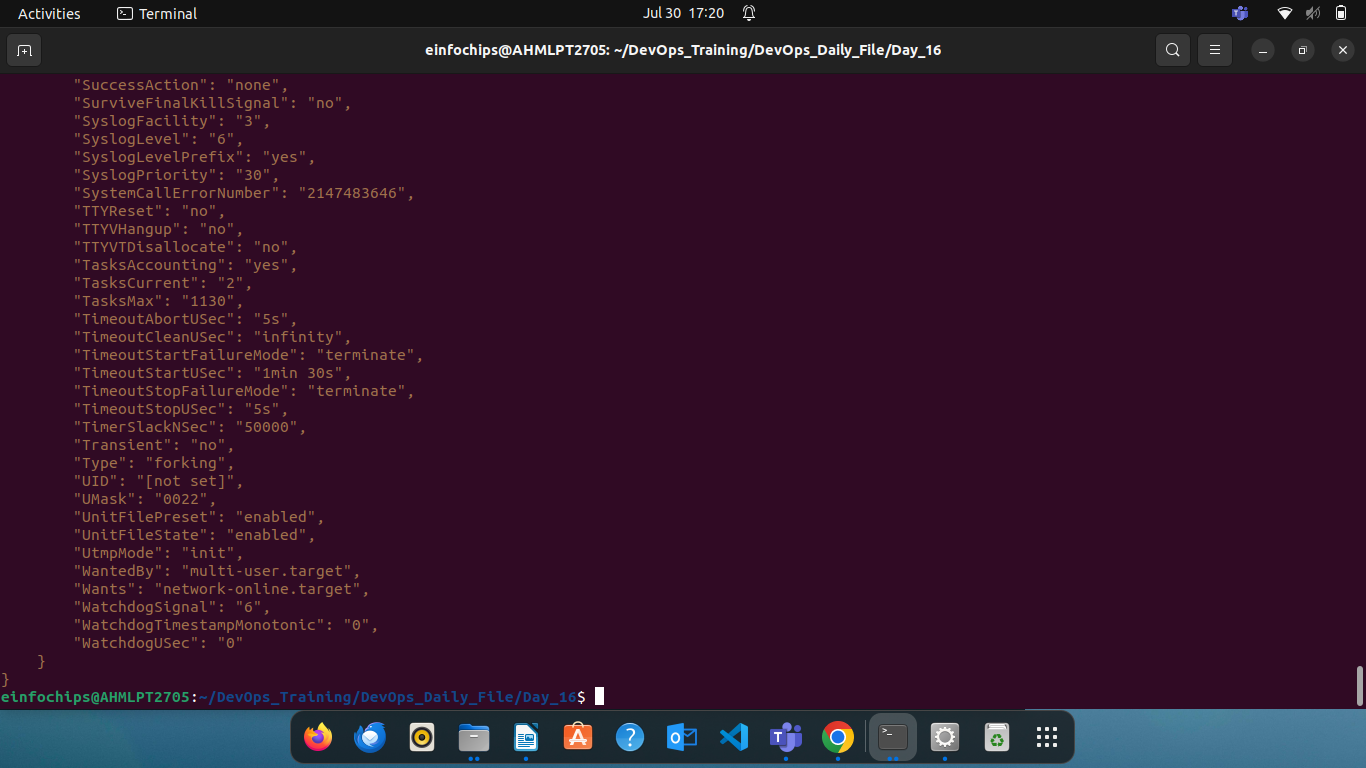




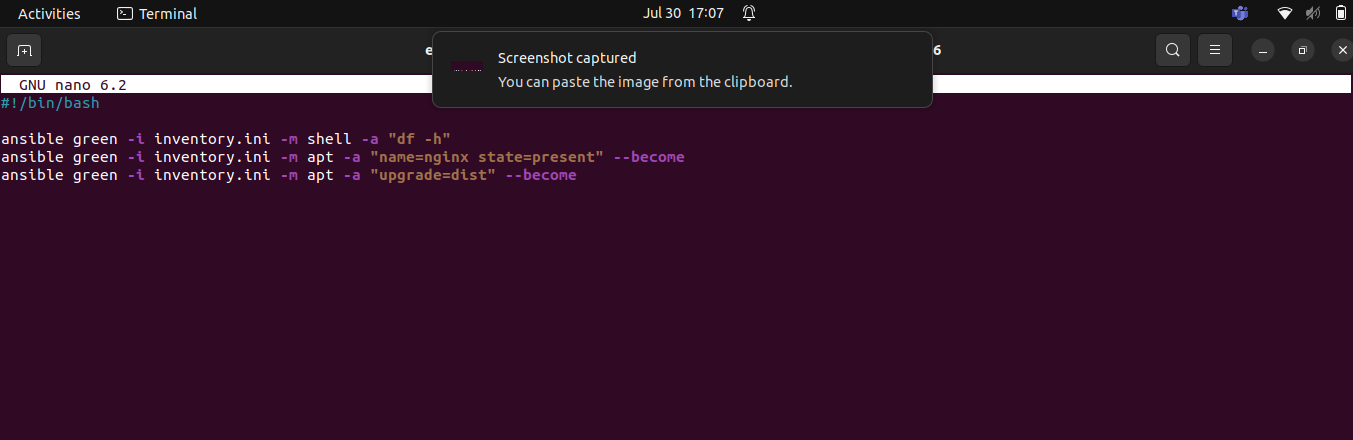
* + Update all packages on a subset of managed nodes.



ansible all -i inventory.ini -m service -a "name=nginx state=restarted" –become



1. **Command Scripts:**
   * Create a script or documentation for each task, detailing the ad-hoc command used and its output.

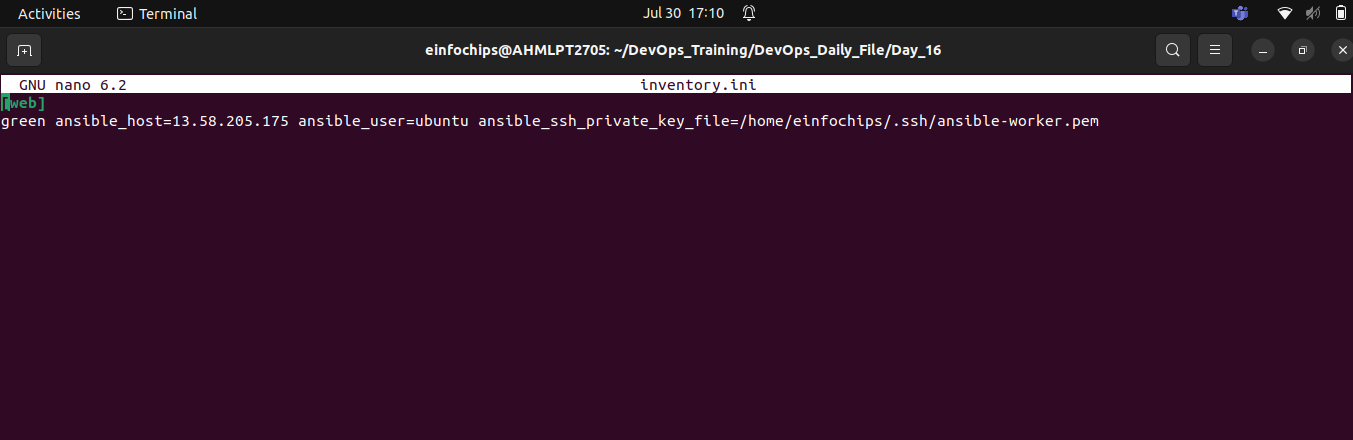


### **Project 3: Working with Ansible Inventories**

**Problem Statement:** You need to manage a dynamic and diverse set of servers, which requires an organized and flexible inventory system. The project involves creating static and dynamic inventories in Ansible to categorize servers based on different attributes such as environment (development, staging, production) and roles (web servers, database servers).

**Deliverables:**

1. **Static Inventory:**
   * Create a static inventory file with different groups for various environments and roles.
   * Verify that the inventory is correctly structured and accessible by Ansible.

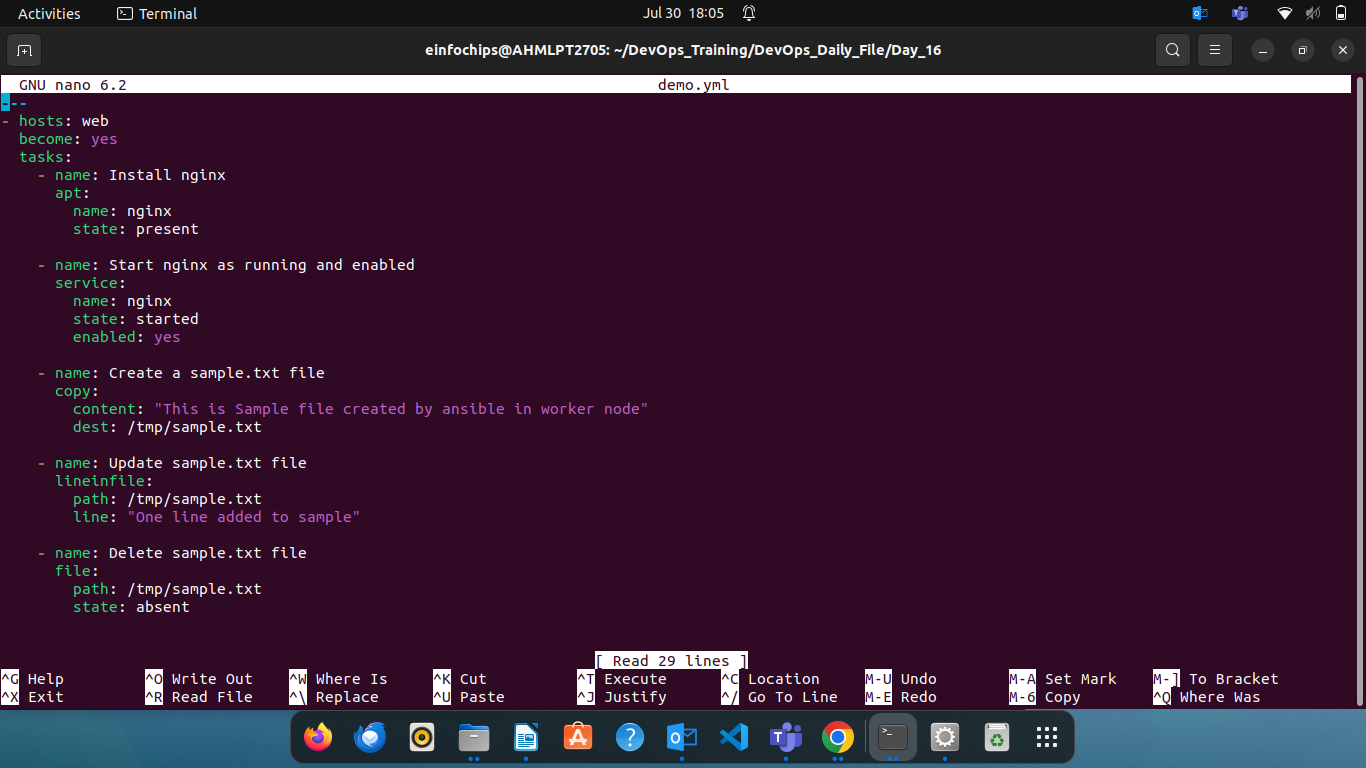


### **Project 4: Ansible Playbooks: The Basics**

**Problem Statement:** Your team needs to automate repetitive tasks such as installing packages, configuring services, and managing files on multiple servers. The project involves writing basic Ansible playbooks to automate these tasks, ensuring consistency and efficiency in the operations.

**Deliverables:**

1. **Playbook Creation:**
   * Write a playbook to install a specific package on all managed nodes.
   * Create a playbook to configure a service with specific parameters.
   * Develop a playbook to manage files, such as creating, deleting, and modifying files on managed nodes.



1. **Testing and Verification:**
   * Test the playbooks to ensure they run successfully and perform the intended tasks.
   * Validate the changes made by the playbooks on the managed nodes.

