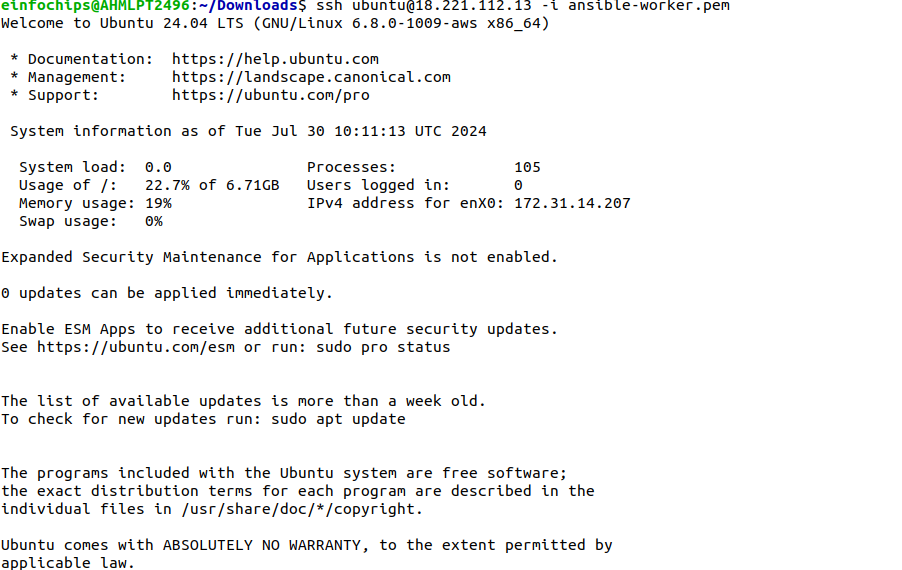
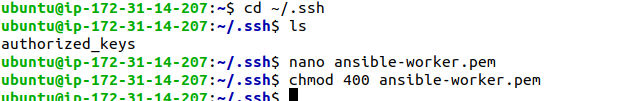
Day 16 task Documentation:

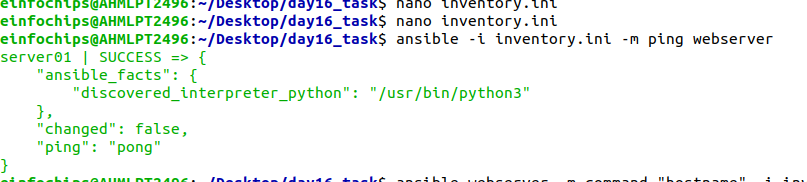
**Problem Statement:** You are tasked with deploying Ansible in a multi-node environment consisting of multiple Linux servers. The goal is to set up Ansible on a control node and configure it to manage several managed nodes. This setup will be used for automating system administration tasks across the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

**Deliverables:**

1. **Control Node Setup:**
   * Install Ansible on the control node.
   * Configure SSH key-based authentication between the control node and managed nodes.
2. **Managed Nodes Configuration:**
   * Ensure all managed nodes are properly configured to be controlled by Ansible.
   * Verify connectivity and proper setup between the control node and managed nodes.
3. **Documentation:**
   * Detailed installation and configuration steps.
   * Troubleshooting guide for common issues encountered during deployment.







### 

### 

### **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:**

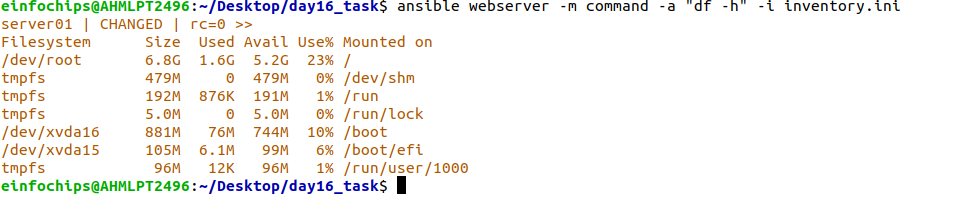
1. **Task Execution:**
   * Execute commands to check disk usage across all managed nodes.
   * Restart a specific service on all managed nodes.
   * Update all packages on a subset of managed nodes.
2. **Command Scripts:**
   * Create a script or documentation for each task, detailing the ad-hoc command used and its output.
3. **Documentation:**
   * Provide a comprehensive guide on using Ansible ad-hoc commands.
   * Include examples of common administrative tasks that can be performed with ad-hoc commands.

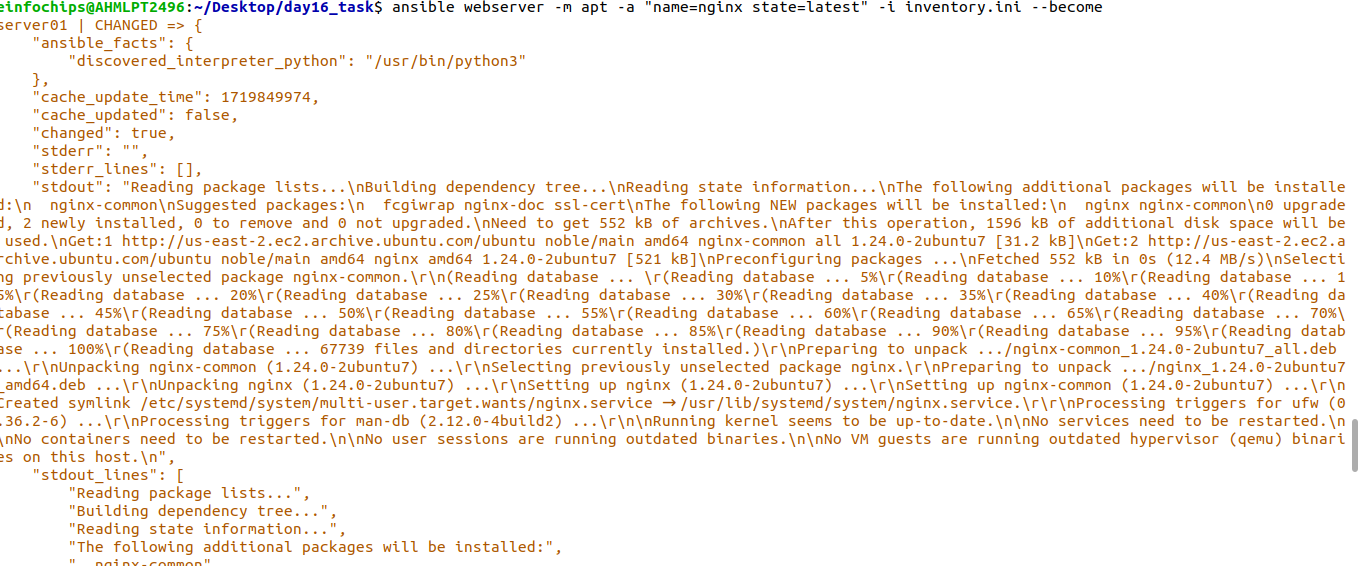
### **Project 3: Working with Ansible Inventories**

**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).

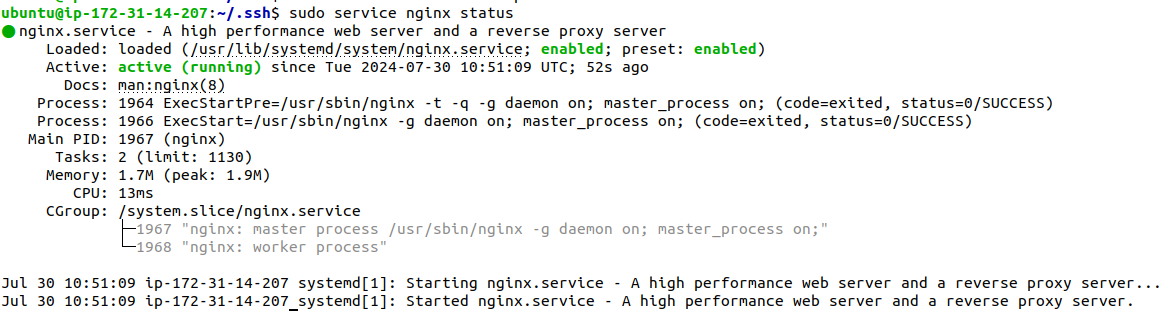
**Deliverables:**

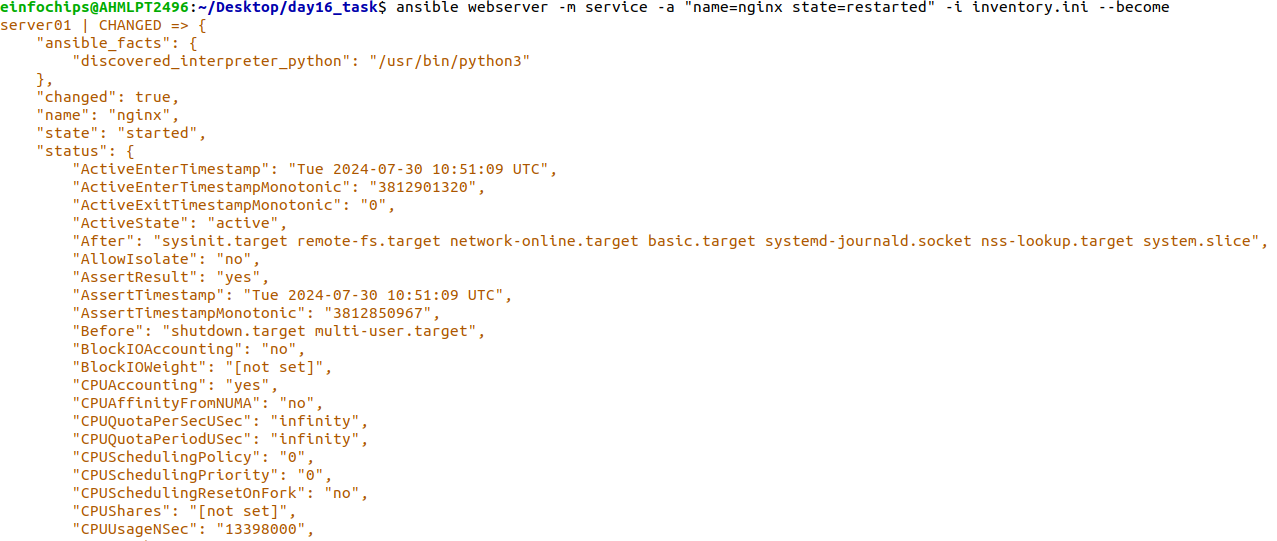
1. **Static Inventory:**
   * Create a static inventory file with different groups for various environments and roles.
   * Verify that the inventory is correctly structured and accessible by Ansible.
2. **Dynamic Inventory:**
   * Implement a dynamic inventory script or use a dynamic inventory plugin.
   * Configure the dynamic inventory to categorize servers automatically based on predefined criteria.
3. **Documentation:**
   * Instructions for setting up and using static and dynamic inventories.
   * Examples of playbooks utilizing both types of inventories.





Check weather nginx is installed or not.



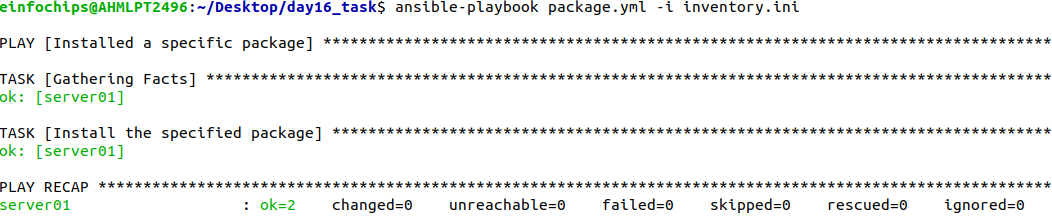


To check weather service is restart or not:  
 sudo service nginx status

If you restart the service it will show **active status** on nginx service

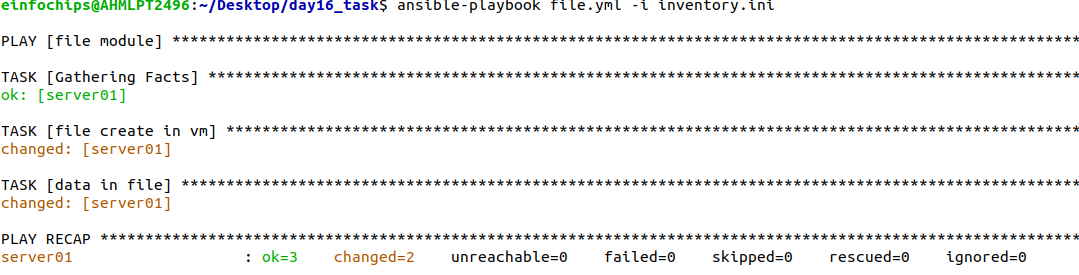






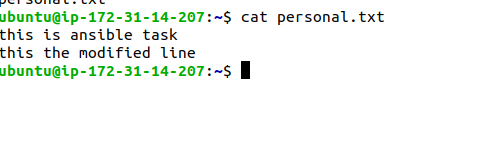
Develop a playbook to manage files, such as creating, deleting, and modifying files on managed nodes:

File created:

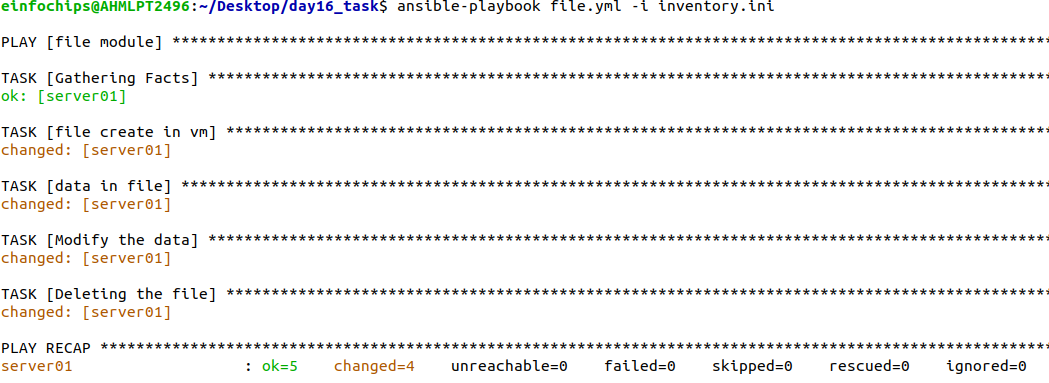


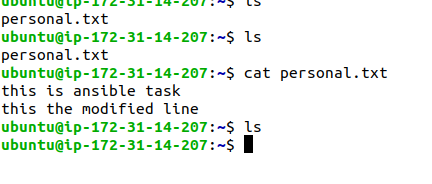


Modify the file:

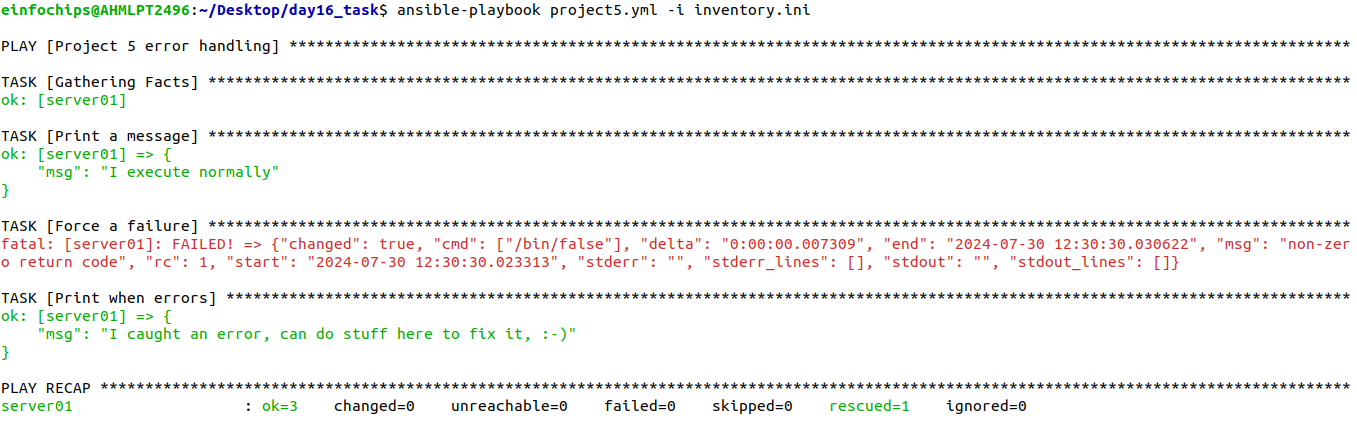


Delete the file:





Here the second task fail but the playbook executed successfully. By this method we can handle error in task.



Block with always section:

