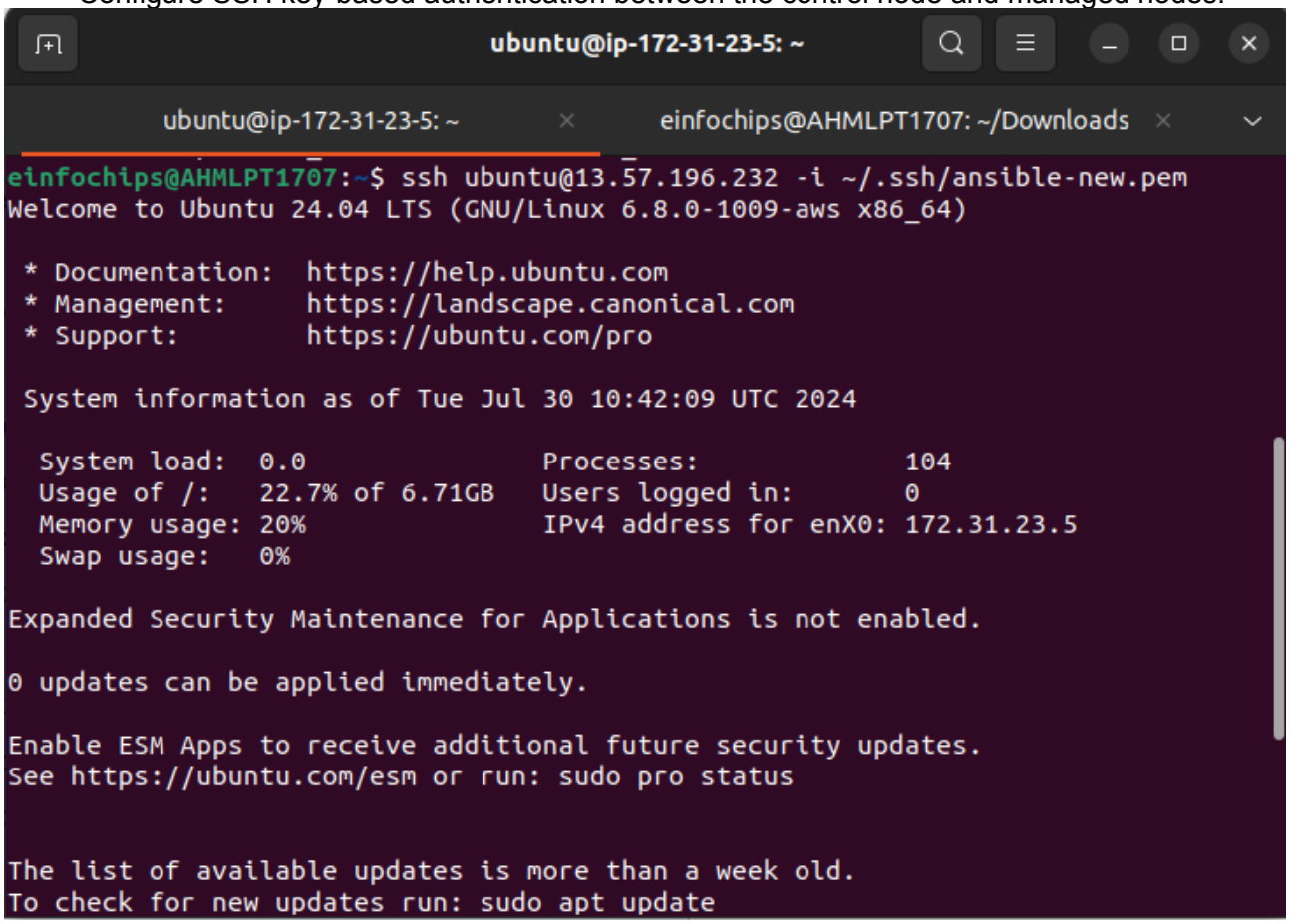
Day 16

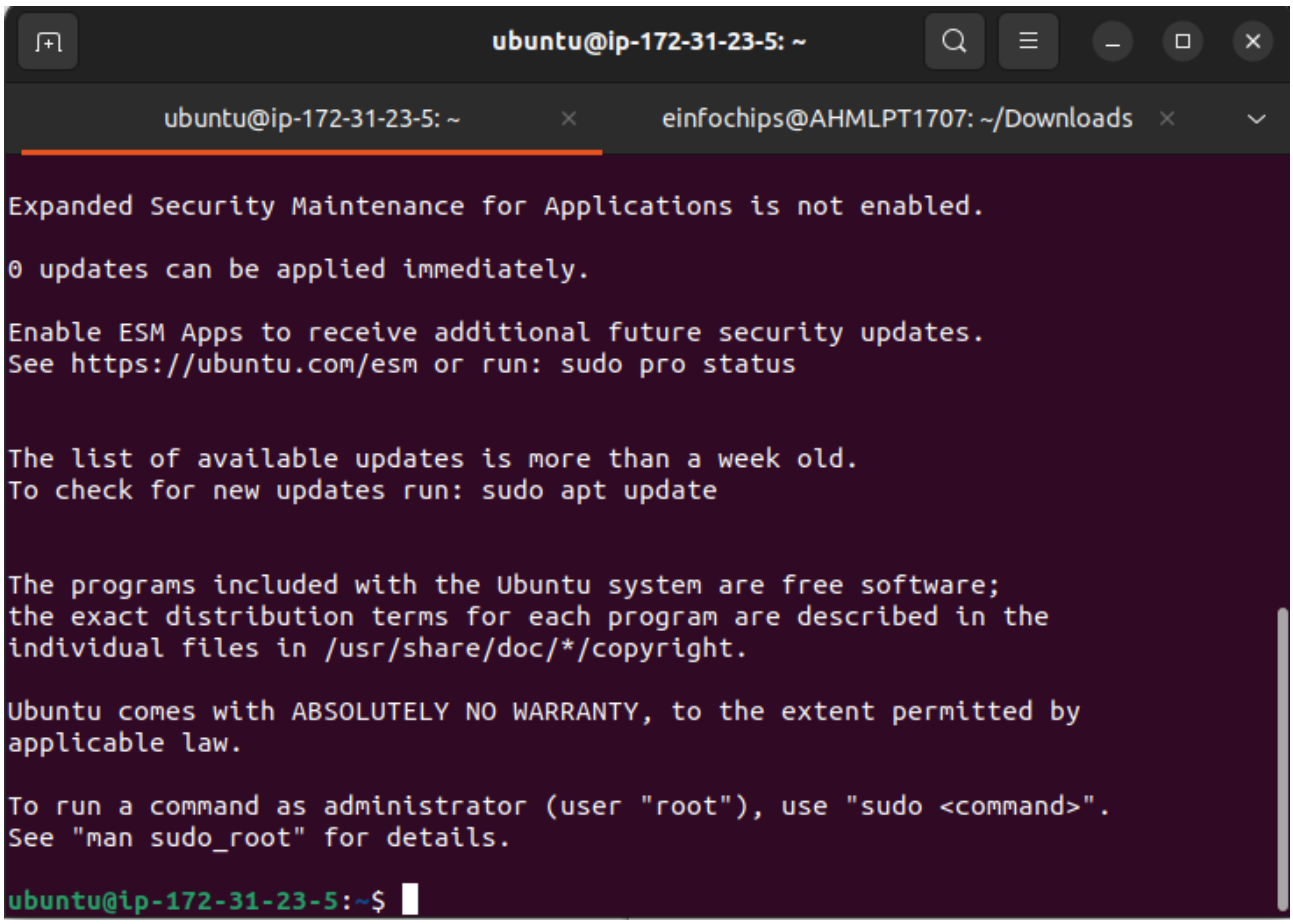
Control Node Setup:

- Install Ansible on the control node.

sudo apt update
sudo apt install software-properties-common
sudo add-apt-repository --yes --update ppa:ansible/ansible
sudo apt install ansible

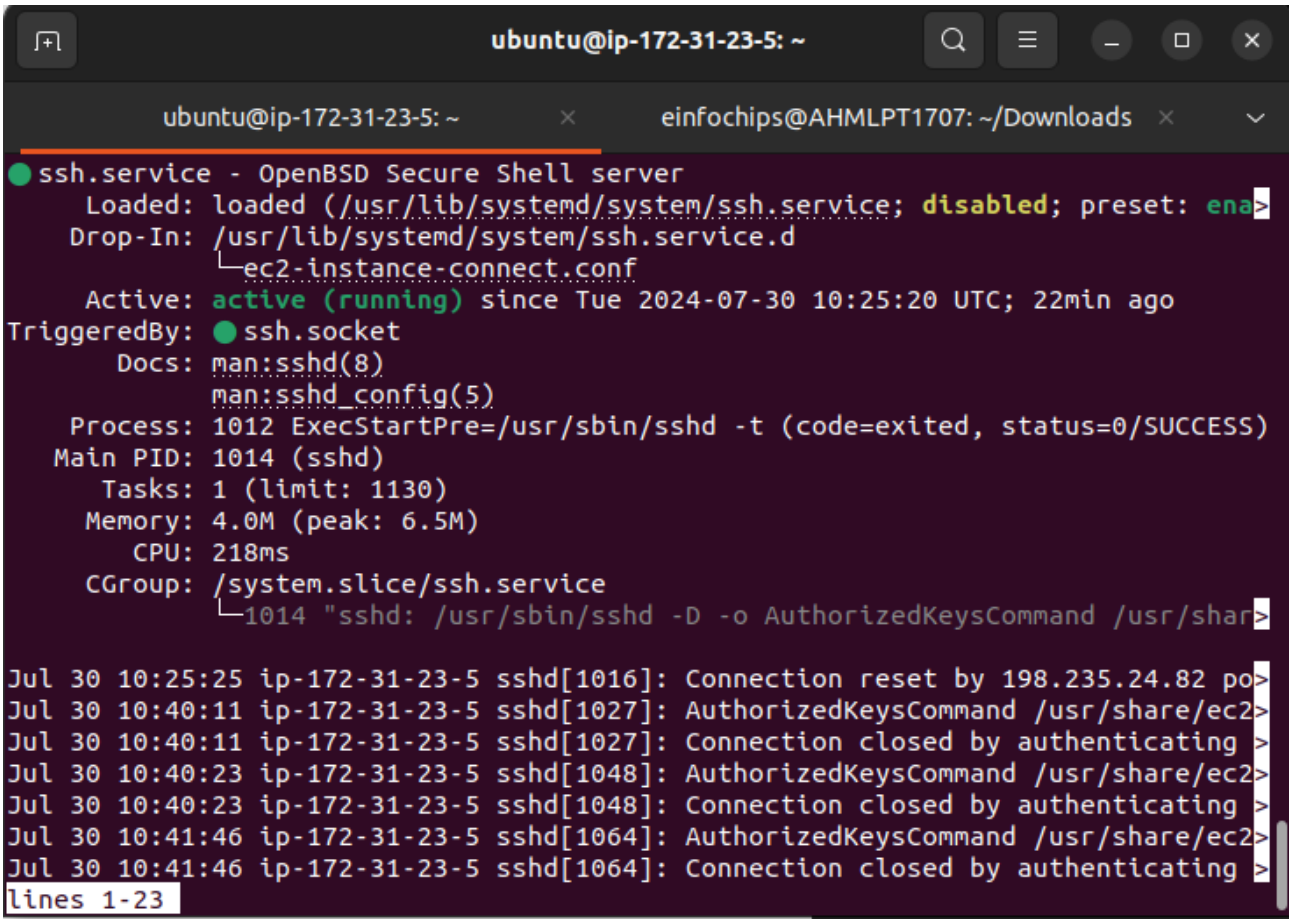
Configure SSH key-based authentication between the control node and managed nodes.





Managed Nodes Configuration:

- Ensure all managed nodes are properly configured to be controlled by Ansible.



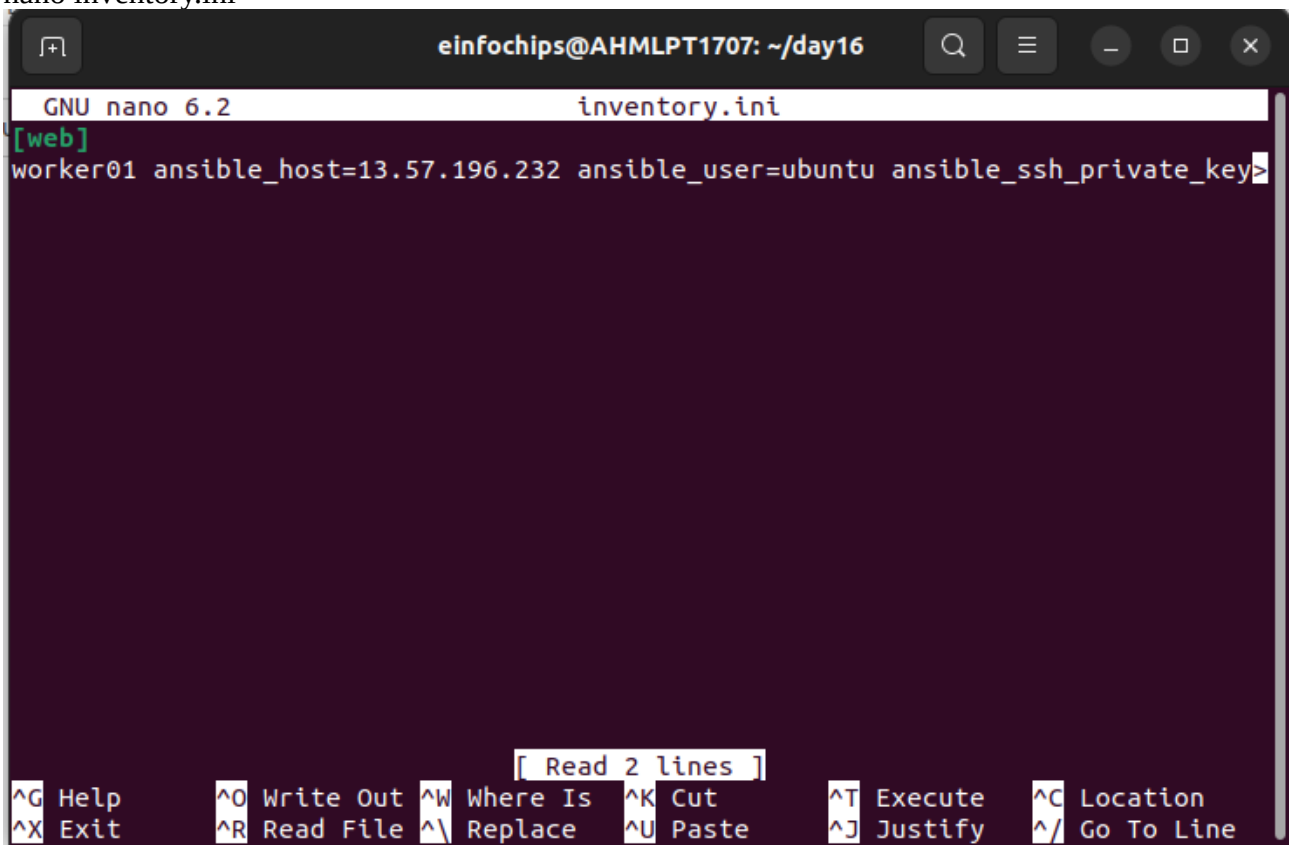
Configure Ansible on the Control Node

3.1. Set Up the Ansible Inventory File

The inventory file lists all managed nodes and their groups. By default, it is located at `/etc/ansible/hosts`. You can edit it or create a new inventory file in your project directory.

Creating a new inventory file

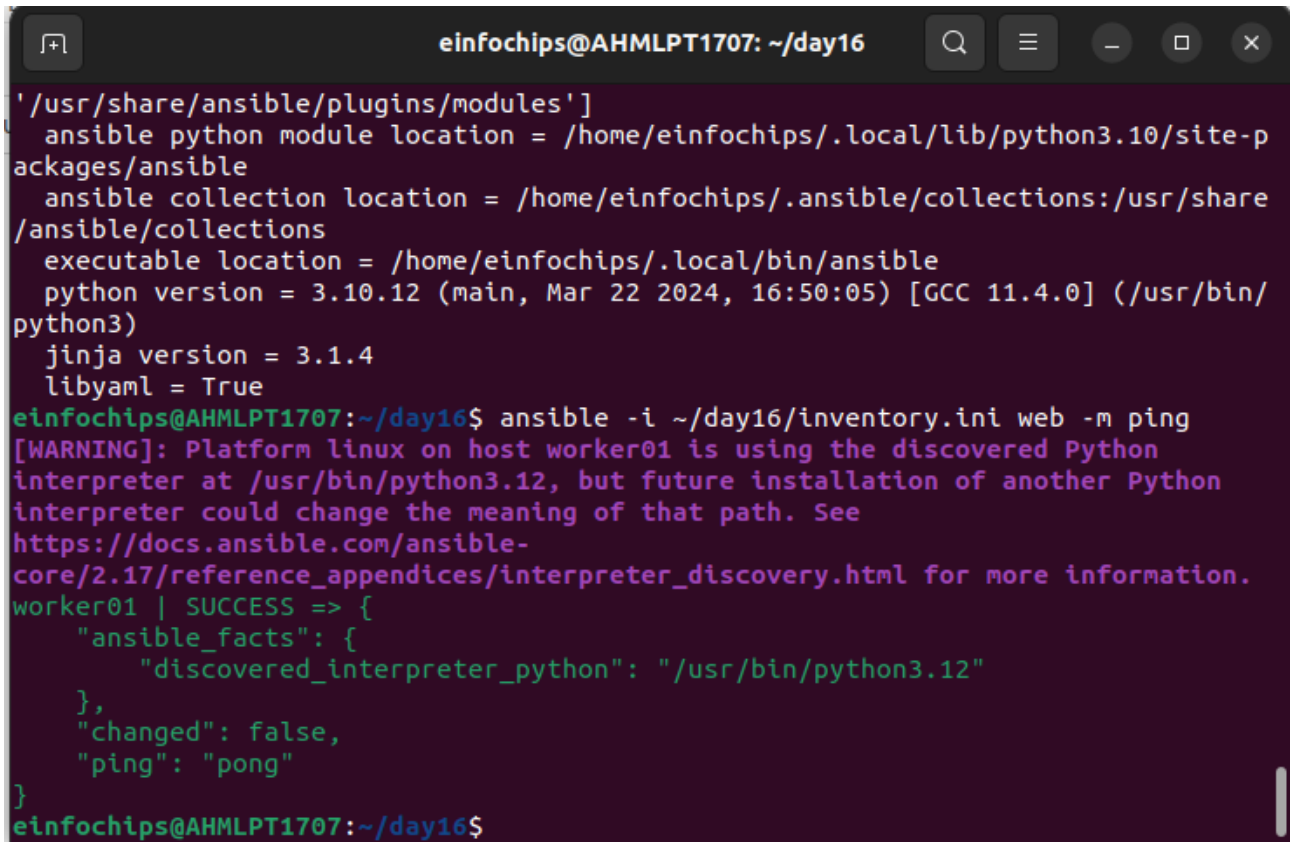
nano inventory.ini



Verify connectivity and proper setup between the control node and managed nodes.

ansible -i ~/day16/inventory.ini web -m ping

This command should return `pong` from each managed node if everything is set up correctly.

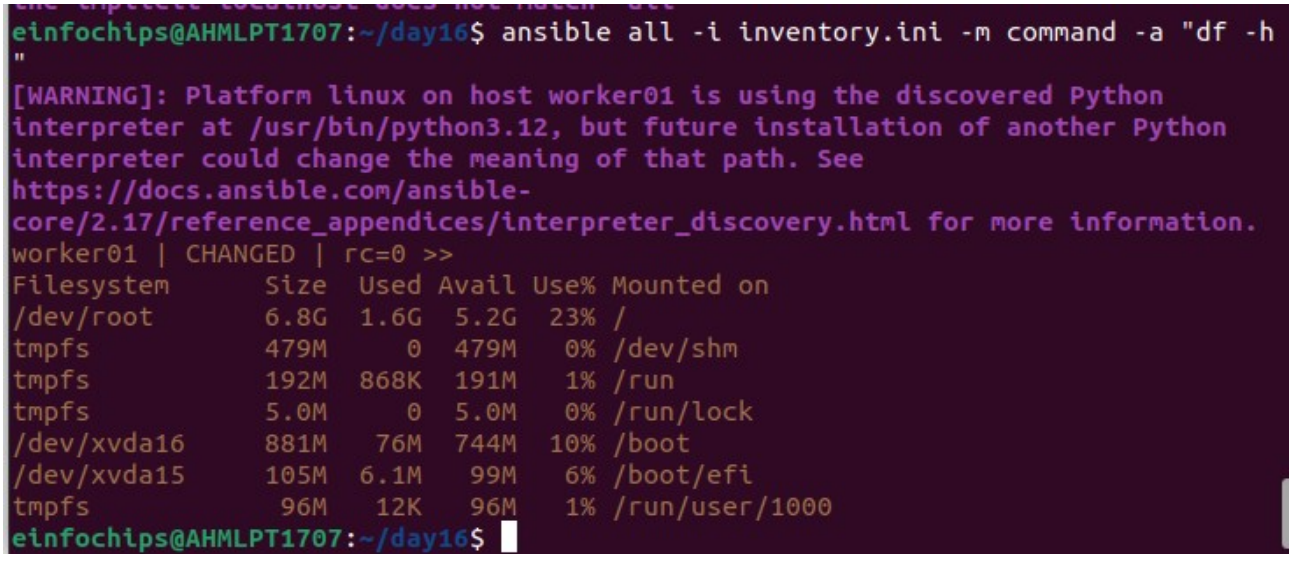


Ad-Hoc Ansible Commands

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.

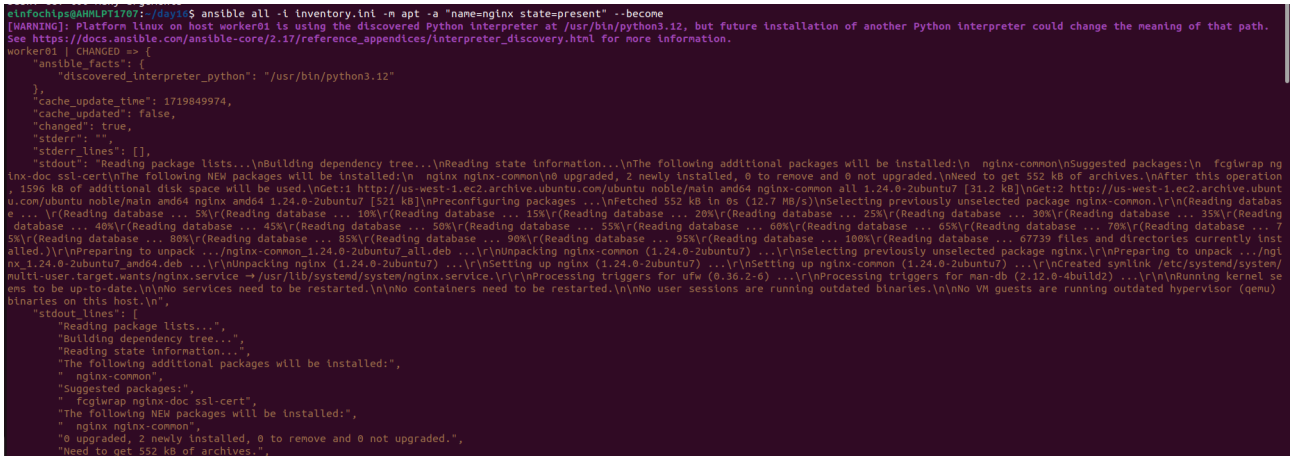
Execute commands to check disk usage across all managed nodes.

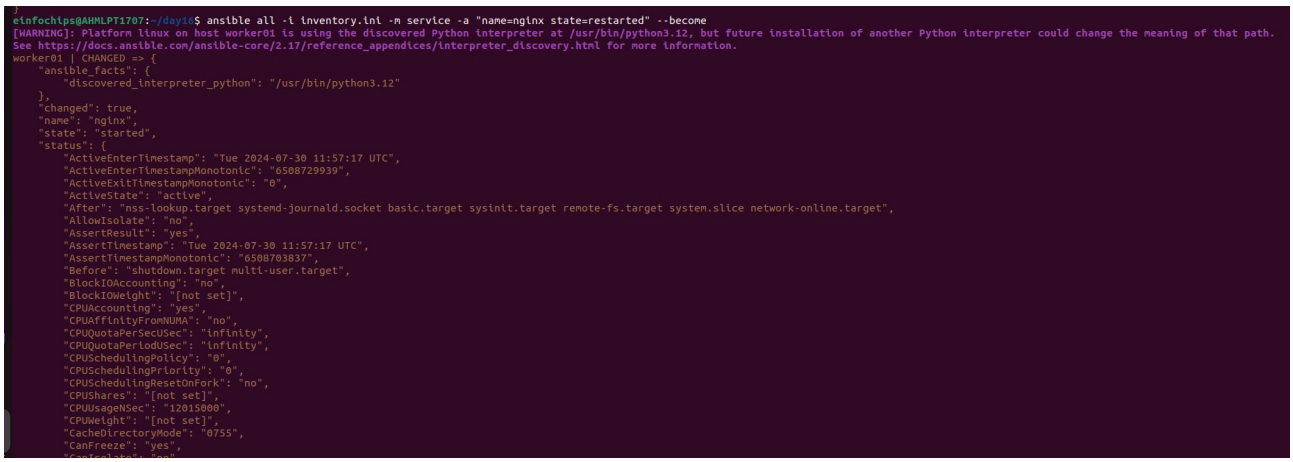
ansible all -i /path/to/your/inventory.ini -m command -a "df -h"



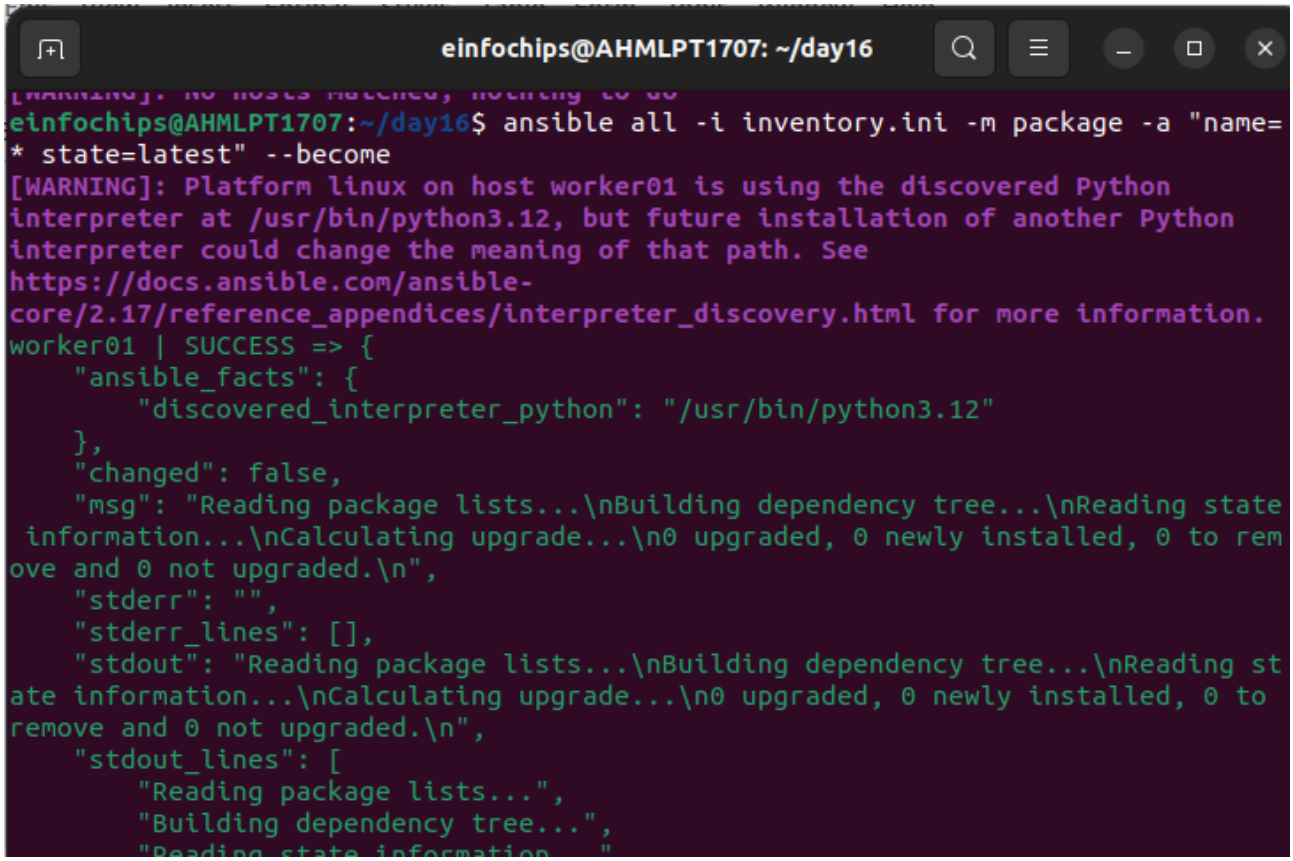
Restart a specific service on all managed nodes.

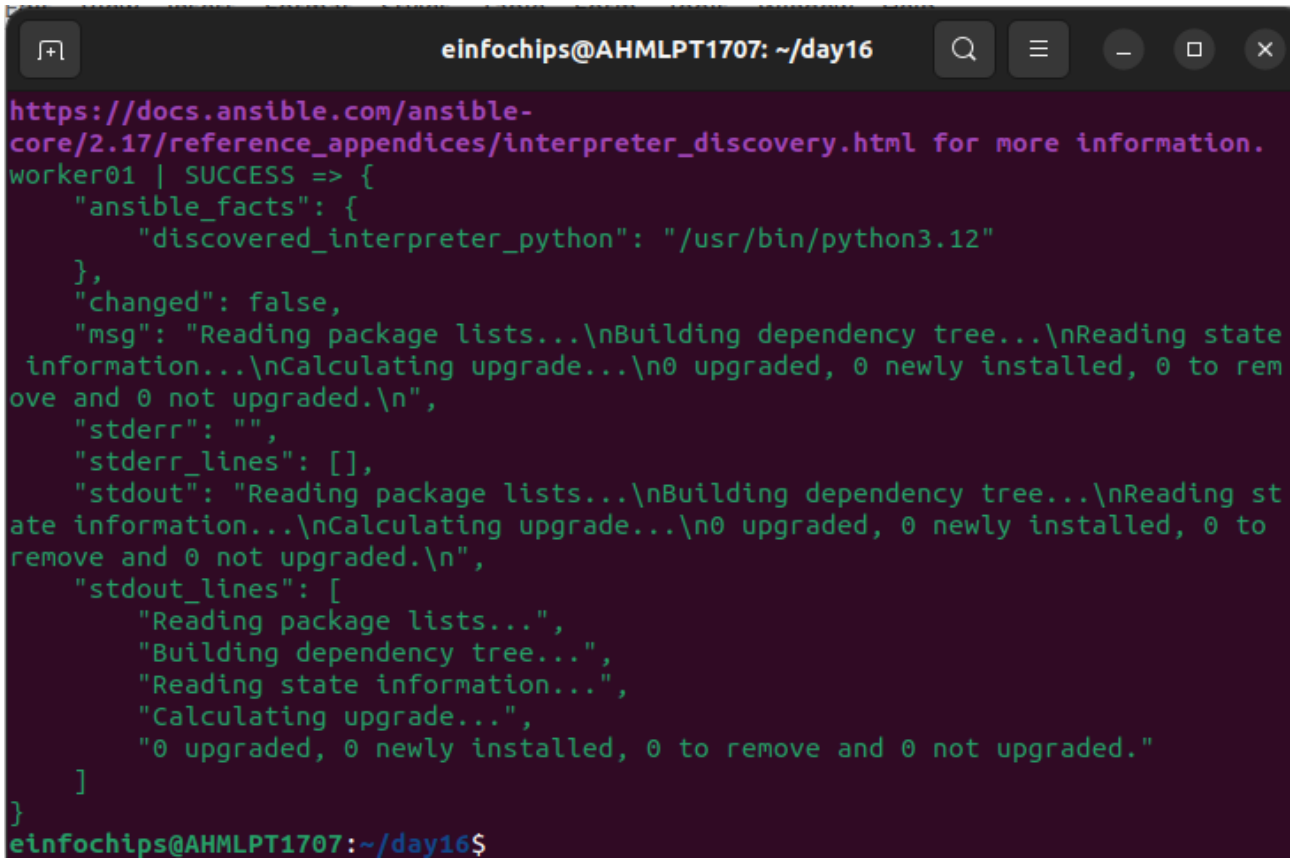
ansible all -i inventory.ini -m apt -a "name=nginx state=present" –become





Update all packages on a subset of managed nodes.

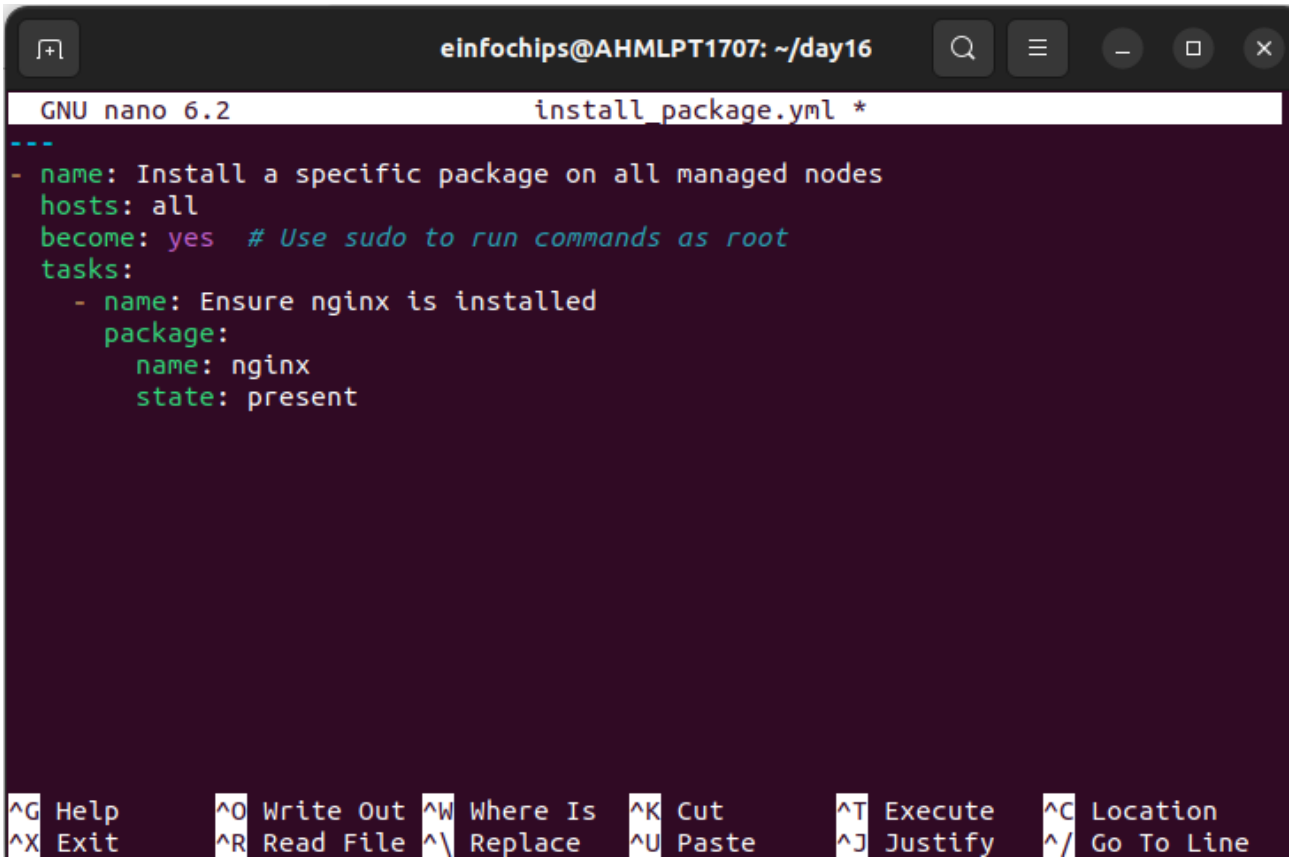

einfochips@AHMLPT1707: ~/day16
einfochips@AHMLPT1707:~/day16$ ansible all -i inventory.ini -m package -a "name=* state=latest" --become
[WARNING]: Platform linux on host worker01 is using the discovered Python interpreter at /usr/bin/python3.12, but future installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more information.
worker01 | SUCCESS => {
"ansible_facts": {
"discovered_interpreter_python": "/usr/bin/python3.12"
},
"changed": false,
"msg": "Reading package lists...\nBuilding dependency tree...\nReading state information...\nCalculating upgrade...\n0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.\n",
"stderr": "",
"stderr_lines": [],
"stdout": "Reading package lists...\nBuilding dependency tree...\nReading state information...\nCalculating upgrade...\n0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.\n",
"stdout_lines": [
"Reading package lists...",
"Building dependency tree...",

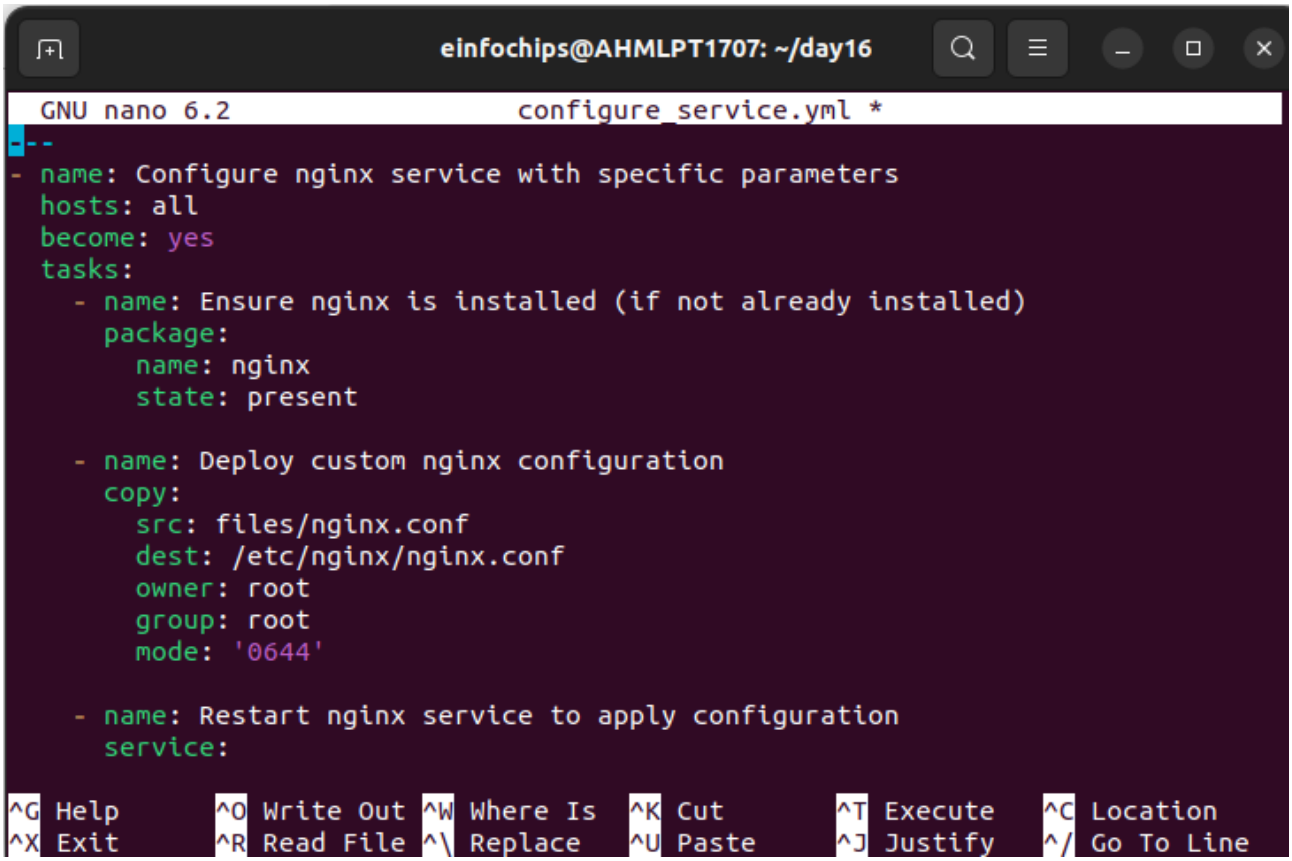

einfochips@AHMLPT1707: ~/day16
https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more information.
worker01 | SUCCESS => {
"ansible_facts": {
"discovered_interpreter_python": "/usr/bin/python3.12"
},
"changed": false,
"msg": "Reading package lists...\nBuilding dependency tree...\nReading state information...\nCalculating upgrade...\n0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.\n",
"stderr": "",
"stderr_lines": [],
"stdout": "Reading package lists...\nBuilding dependency tree...\nReading state information...\nCalculating upgrade...\n0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.\n",
"stdout_lines": [
"Reading package lists...",
"Building dependency tree...",
"Reading state information...",
"Calculating upgrade...",
"0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded."
]
}
einfochips@AHMLPT1707:~/day16$

Playbook Creation:

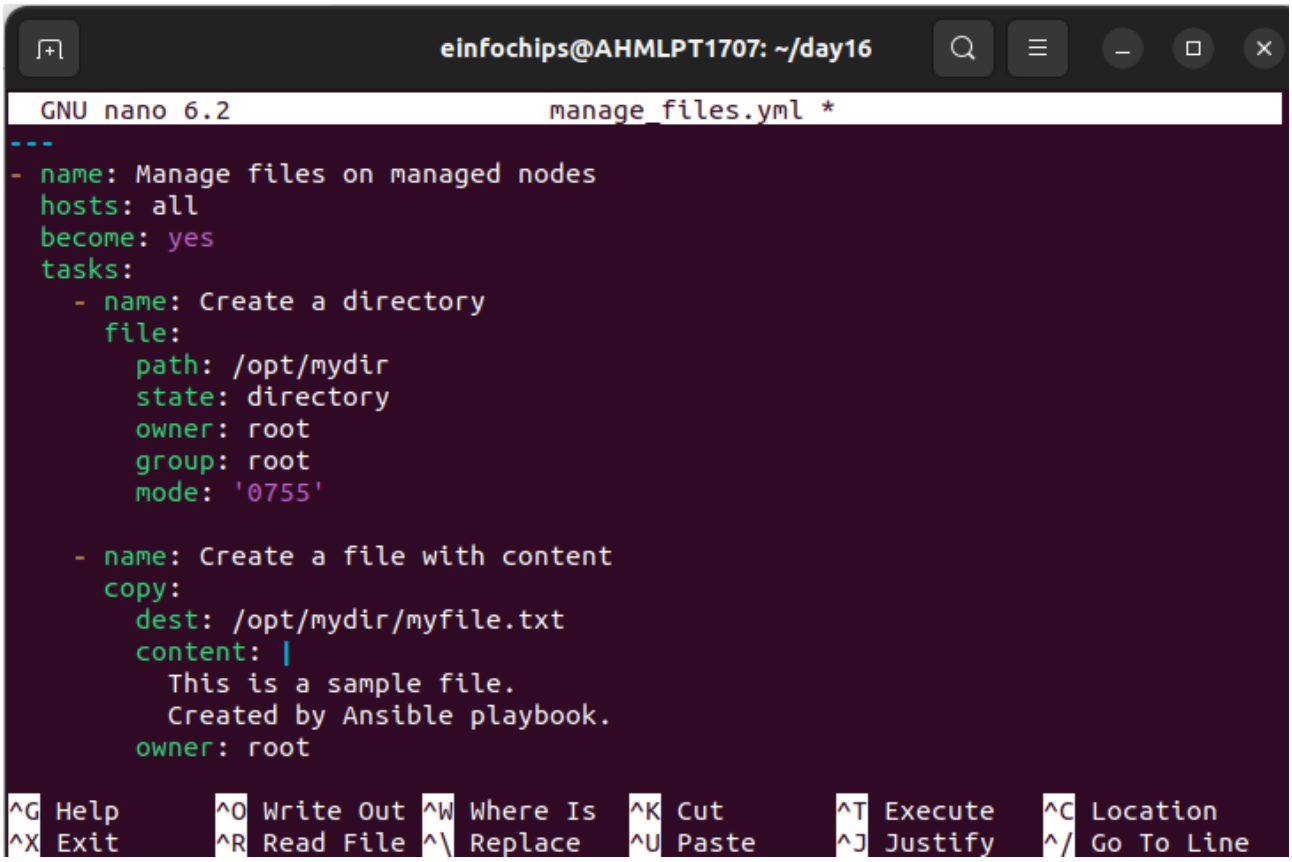
- Write a playbook to install a specific package on all managed nodes.
- Create a playbook to configure a service with specific parameters.
- Develop a playbook to manage files, such as creating, deleting, and modifying files on managed nodes.

2. **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.

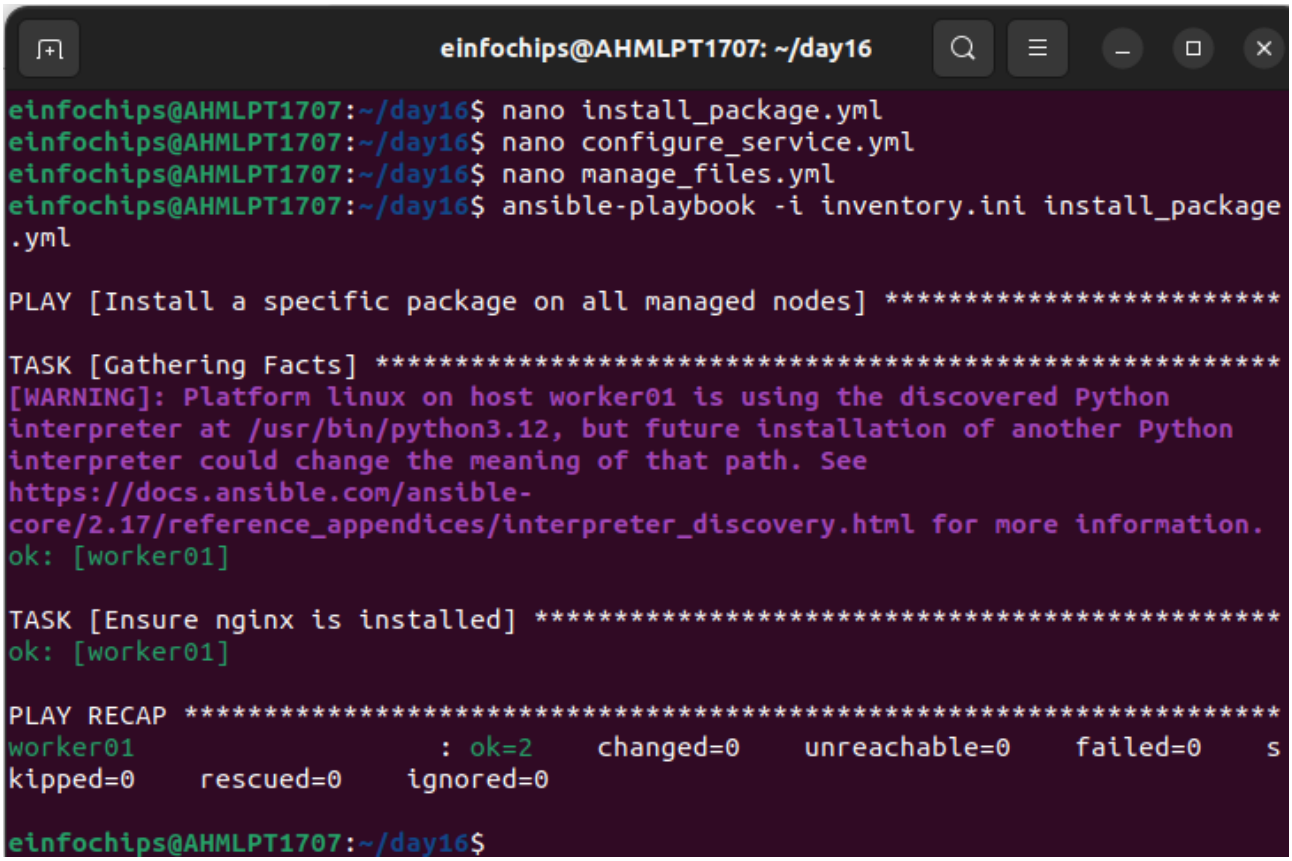


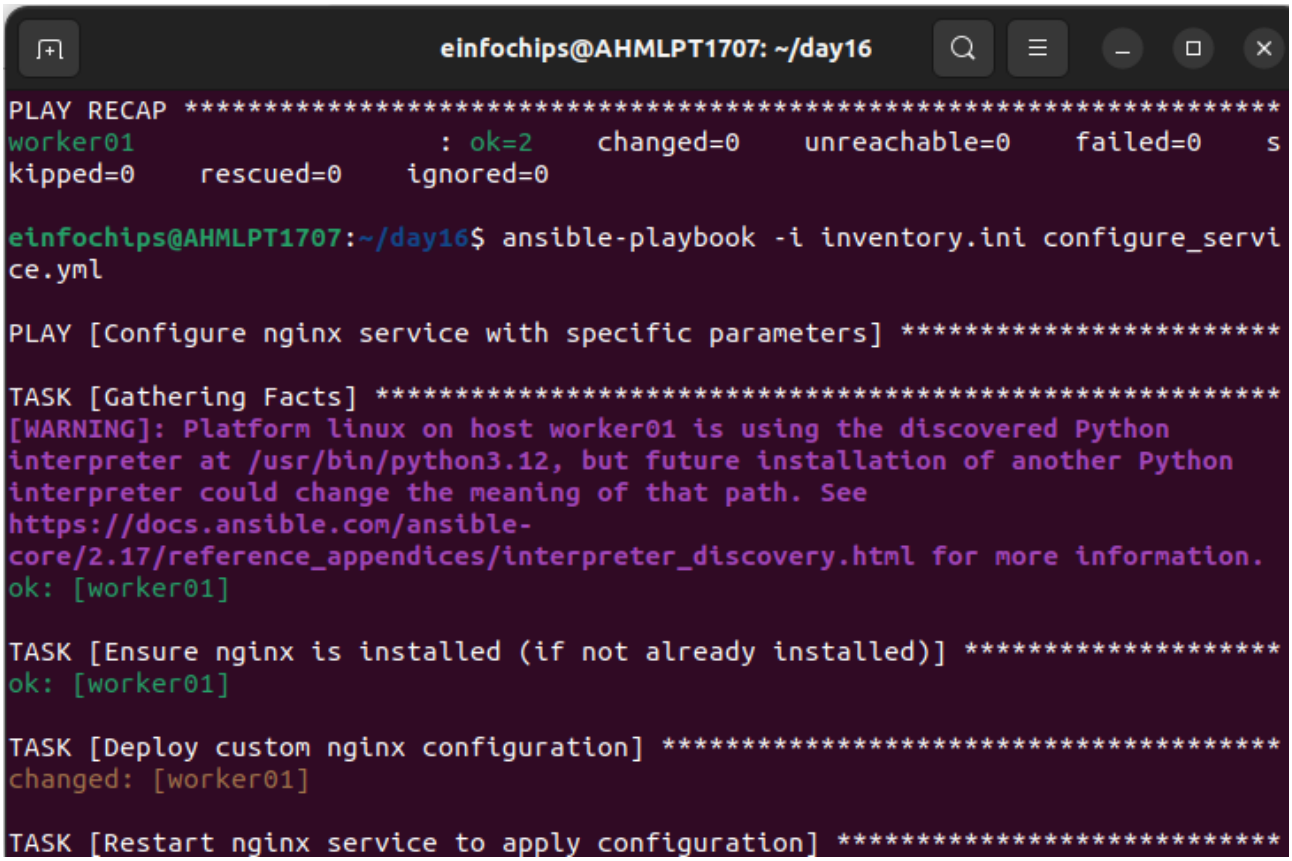

einfochips@AHMLPT1707: ~/day16
GNU nano 6.2 configure_service.yml *

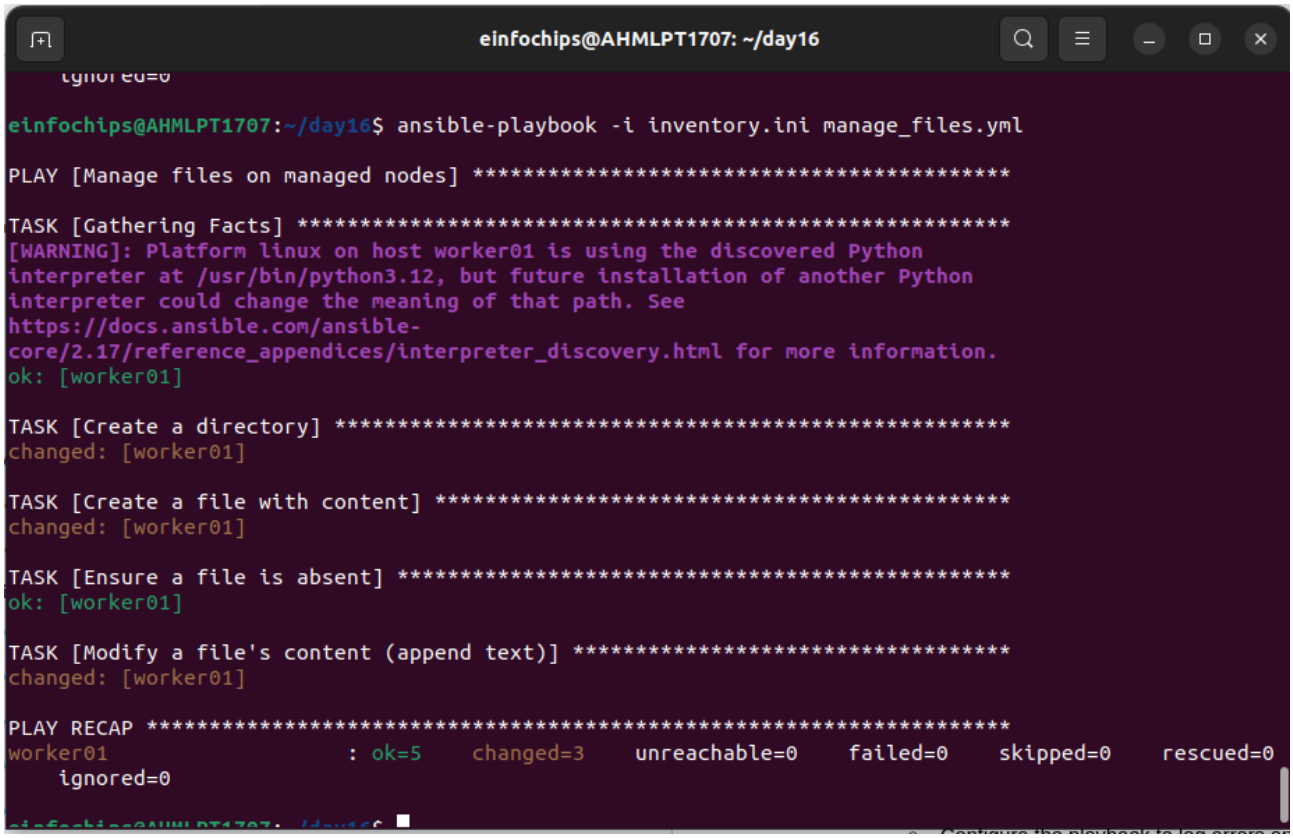
- name: Configure nginx service with specific parameters
hosts: all
become: yes
tasks:
- name: Ensure nginx is installed (if not already installed)
package:
name: nginx
state: present
- name: Deploy custom nginx configuration
copy:
src: files/nginx.conf
dest: /etc/nginx/nginx.conf
owner: root
group: root
mode: '0644'
- name: Restart nginx service to apply configuration
service:
^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line


einfochips@AHMLPT1707: ~/day16
GNU nano 6.2 manage_files.yml *

- name: Manage files on managed nodes
hosts: all
become: yes
tasks:
- name: Create a directory
file:
path: /opt/mydir
state: directory
owner: root
group: root
mode: '0755'
- name: Create a file with content
copy:
dest: /opt/mydir/myfile.txt
content: |
This is a sample file.
Created by Ansible playbook.
owner: root
^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line


einfochips@AHMLPT1707: ~/day16
einfochips@AHMLPT1707:~/day16$ nano install_package.yml
einfochips@AHMLPT1707:~/day16$ nano configure_service.yml
einfochips@AHMLPT1707:~/day16$ nano manage_files.yml
einfochips@AHMLPT1707:~/day16$ ansible-playbook -i inventory.ini install_package
.yml
PLAY [Install a specific package on all managed nodes] ************************
TASK [Gathering Facts] **
[WARNING]: Platform linux on host worker01 is using the discovered Python
interpreter at /usr/bin/python3.12, but future installation of another Python
interpreter could change the meaning of that path. See
https://docs.ansible.com/ansible-
core/2.17/reference_appendices/interpreter_discovery.html for more information.
ok: [worker01]
TASK [Ensure nginx is installed] **
ok: [worker01]
PLAY RECAP **
worker01 : ok=2 changed=0 unreachable=0 failed=0 s
kipped=0 rescued=0 ignored=0
einfochips@AHMLPT1707:~/day16$

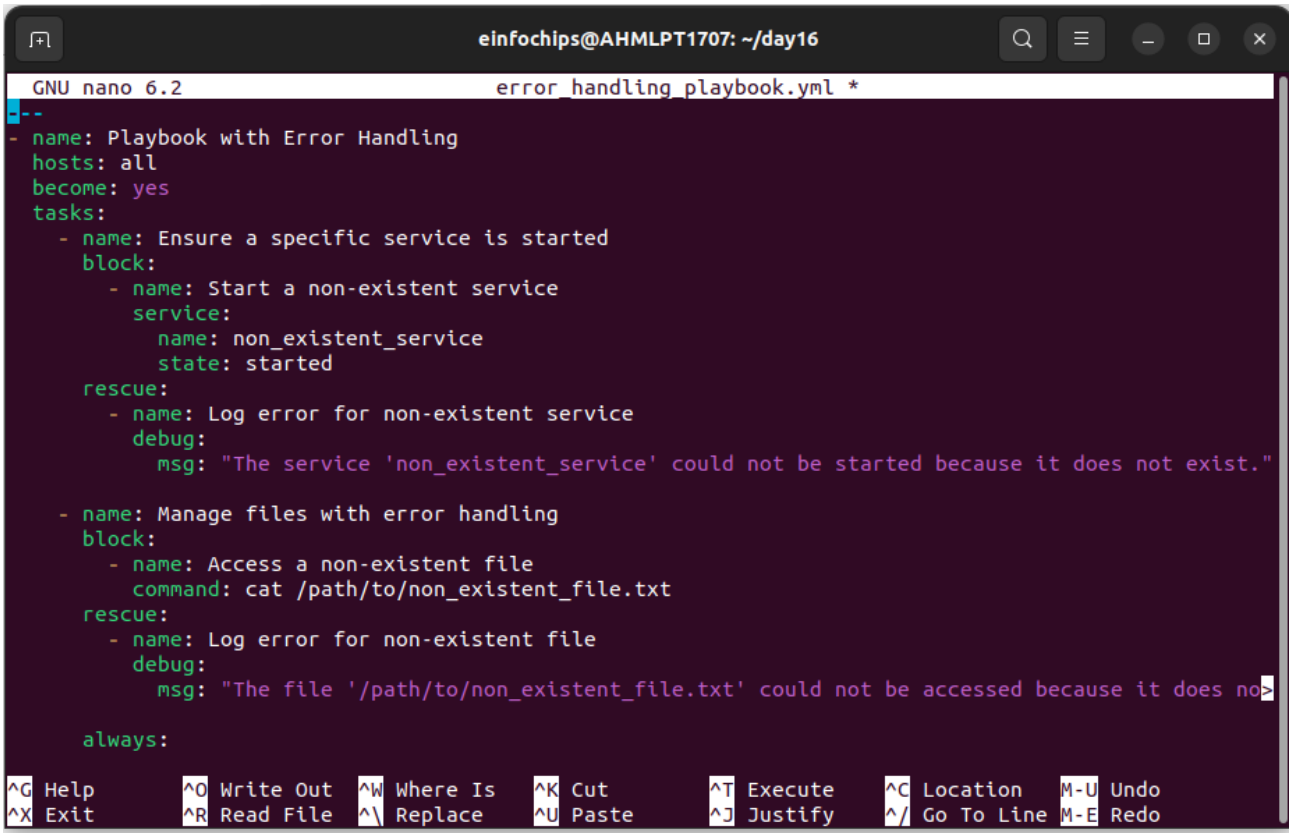

einfochips@AHMLPT1707: ~/day16
PLAY RECAP **
worker01 : ok=2 changed=0 unreachable=0 failed=0 s
kipped=0 rescued=0 ignored=0
einfochips@AHMLPT1707:~/day16$ ansible-playbook -i inventory.ini configure_servi
ce.yml
PLAY [Configure nginx service with specific parameters] ***********************
TASK [Gathering Facts] **
[WARNING]: Platform linux on host worker01 is using the discovered Python
interpreter at /usr/bin/python3.12, but future installation of another Python
interpreter could change the meaning of that path. See
https://docs.ansible.com/ansible-
core/2.17/reference_appendices/interpreter_discovery.html for more information.
ok: [worker01]
TASK [Ensure nginx is installed (if not already installed)] *******************
ok: [worker01]
TASK [Deploy custom nginx configuration] **************************************
changed: [worker01]
TASK [Restart nginx service to apply configuration] ***************************



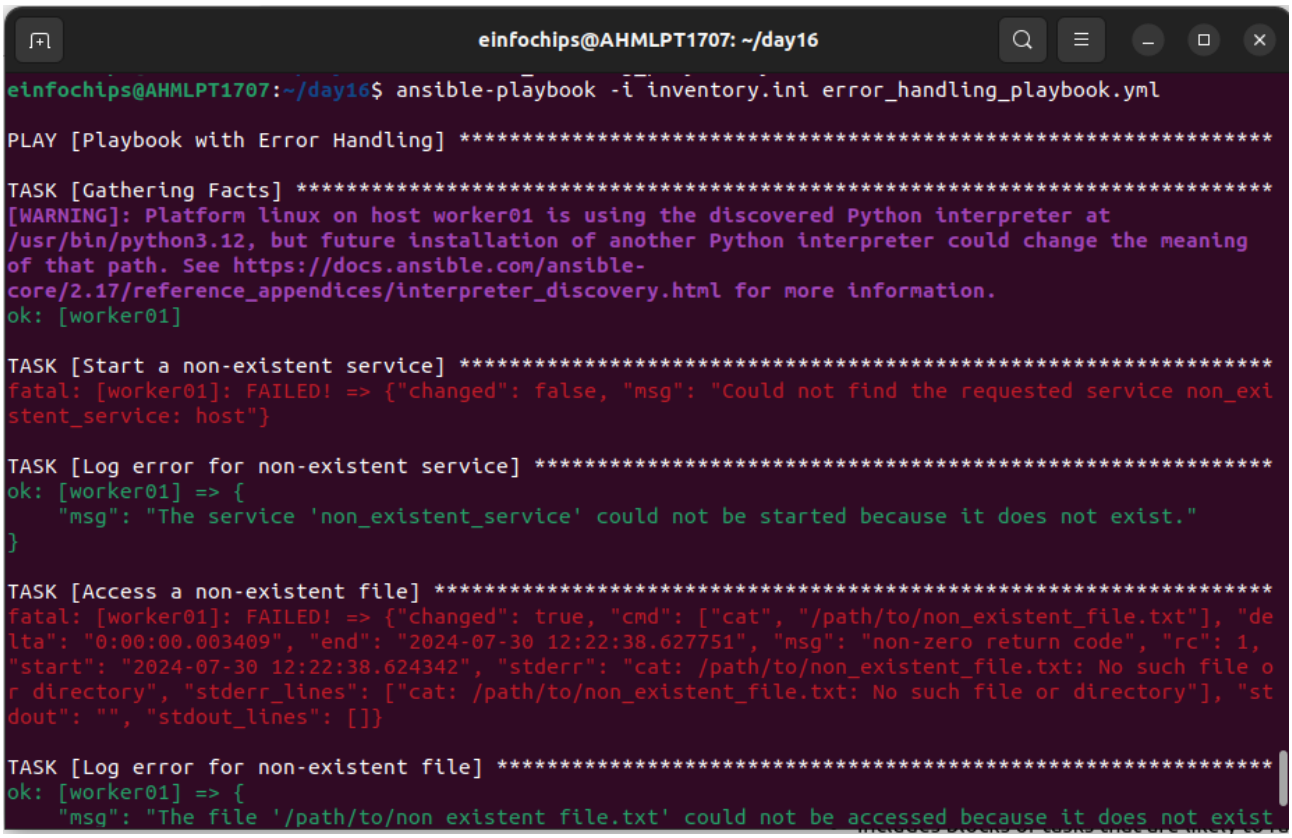
Playbook with Error Handling:

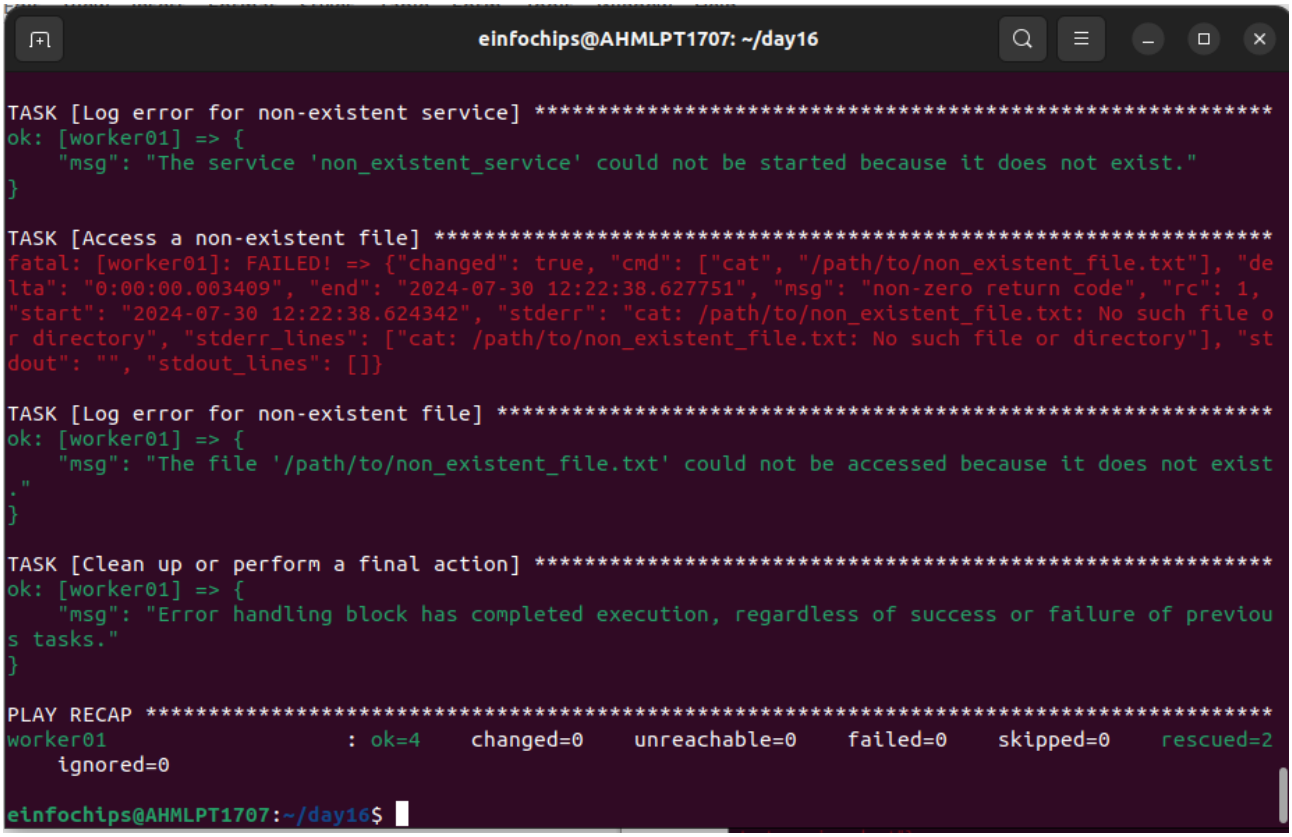
- Write a playbook that includes tasks likely to fail, such as starting a non-existent service or accessing a non-existent file.

Implement error handling strategies using modules like block, rescue, and always.


einfochips@AHMLPT1707: ~/day16
GNU nano 6.2 error_handling_playbook.yml *

- name: Playbook with Error Handling
hosts: all
become: yes
tasks:
- name: Ensure a specific service is started
block:
- name: Start a non-existent service
service:
name: non_existent_service
state: started
rescue:
- name: Log error for non-existent service
debug:
msg: "The service 'non_existent_service' could not be started because it does not exist."
- name: Manage files with error handling
block:
- name: Access a non-existent file
command: cat /path/to/non_existent_file.txt
rescue:
- name: Log error for non-existent file
debug:
msg: "The file '/path/to/non_existent_file.txt' could not be accessed because it does no
always:
^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line M-E Redo


einfochips@AHMLPT1707: ~/day16
einfochips@AHMLPT1707:~/day16$ ansible-playbook -i inventory.ini error_handling_playbook.yml
PLAY [Playbook with Error Handling] **
TASK [Gathering Facts] ***
[WARNING]: Platform linux on host worker01 is using the discovered Python interpreter at
/usr/bin/python3.12, but future installation of another Python interpreter could change the meaning
of that path. See https://docs.ansible.com/ansible-
core/2.17/reference_appendices/interpreter_discovery.html for more information.
ok: [worker01]
TASK [Start a non-existent service] **
fatal: [worker01]: FAILED! => {"changed": false, "msg": "Could not find the requested service non_exi
stent_service: host"}
TASK [Log error for non-existent service] **
ok: [worker01] => {
"msg": "The service 'non_existent_service' could not be started because it does not exist."
}
TASK [Access a non-existent file] **
fatal: [worker01]: FAILED! => {"changed": true, "cmd": ["cat", "/path/to/non_existent_file.txt"], "de
lta": "0:00:00.003409", "end": "2024-07-30 12:22:38.627751", "msg": "non-zero return code", "rc": 1,
"start": "2024-07-30 12:22:38.624342", "stderr": "cat: /path/to/non_existent_file.txt: No such file o
r directory", "stderr_lines": ["cat: /path/to/non_existent_file.txt: No such file or directory"], "st
dout": "", "stdout_lines": []}
TASK [Log error for non-existent file] ***
ok: [worker01] => {
"msg": "The file '/path/to/non existent file.txt' could not be accessed because it does not exist


einfochips@AHMLPT1707: ~/day16
TASK [Log error for non-existent service] ***
ok: [worker01] => {
"msg": "The service 'non_existent_service' could not be started because it does not exist."
}
TASK [Access a non-existent file] ***
fatal: [worker01]: FAILED! => {"changed": true, "cmd": ["cat", "/path/to/non_existent_file.txt"], "de
lta": "0:00:00.003409", "end": "2024-07-30 12:22:38.627751", "msg": "non-zero return code", "rc": 1,
"start": "2024-07-30 12:22:38.624342", "stderr": "cat: /path/to/non_existent_file.txt: No such file o
r directory", "stderr_lines": ["cat: /path/to/non_existent_file.txt: No such file or directory"], "st
dout": "", "stdout_lines": []}
TASK [Log error for non-existent file] **
ok: [worker01] => {
"msg": "The file '/path/to/non_existent_file.txt' could not be accessed because it does not exist
."
}
TASK [Clean up or perform a final action] ***
ok: [worker01] => {
"msg": "Error handling block has completed execution, regardless of success or failure of previou
s tasks."
}
PLAY RECAP ***
worker01 : ok=4 changed=0 unreachable=0 failed=0 skipped=0 rescued=2
ignored=0
einfochips@AHMLPT1707:~/day16$