Ansible User Management and Package Automation

This project demonstrates the use of Ansible to perform consistent, repeatable system administration tasks across a managed Linux host. The playbook automates local user management, enforces security best practices, and ensures required system packages are installed using a single, idempotent workflow.

The playbook targets a defined host from inventory and executes with elevated privileges to perform system-level changes. During execution, Ansible gathers system facts, validates the desired state of the system, and applies changes only where necessary, providing clear visibility into what was modified and what was already compliant.

Automated Tasks

The automation performs the following actions:

Creates and manages a local Linux user account with a defined login shell and home directory

Forces a password change on first login by expiring the user’s password

Ensures the tmux package is installed and available on the system

Executes safely using privilege escalation while maintaining idempotency

An Ansible playbook is open in vim on a Linux system, written in YAML and defining a play named “multiple tasks.” The play targets the host dev-app-eg3.procore.prod1 and runs with elevated privileges using become: true.

The tasks section contains three configuration steps:

Ensure local user tfleming exists

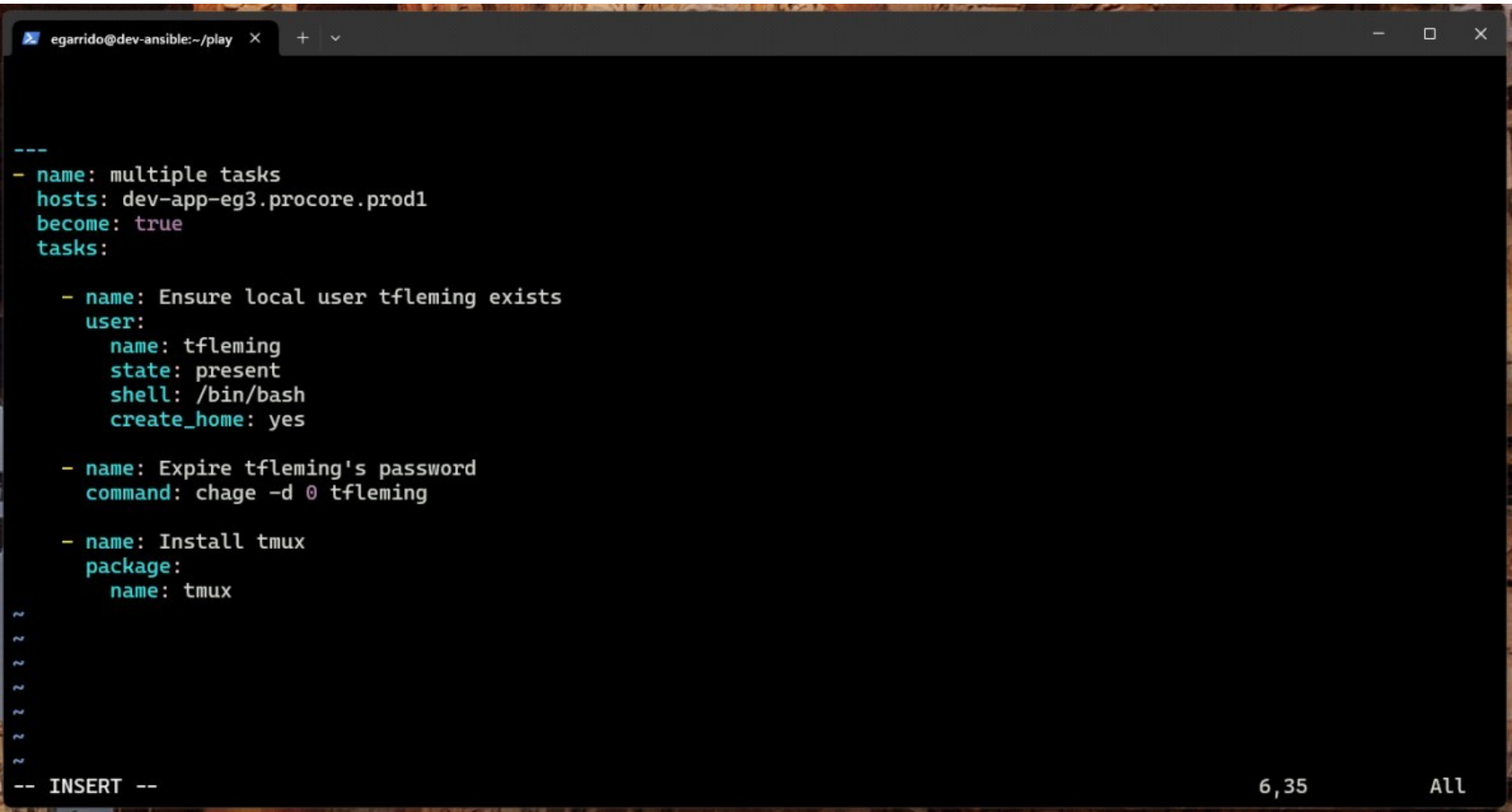
Uses the user module to create the local account tfleming, assign /bin/bash as the login shell, and create a home directory if one does not already exist.

Expire tfleming’s password

Executes the chage -d 0 tfleming command to force a password change on the user’s next login.

Install tmux

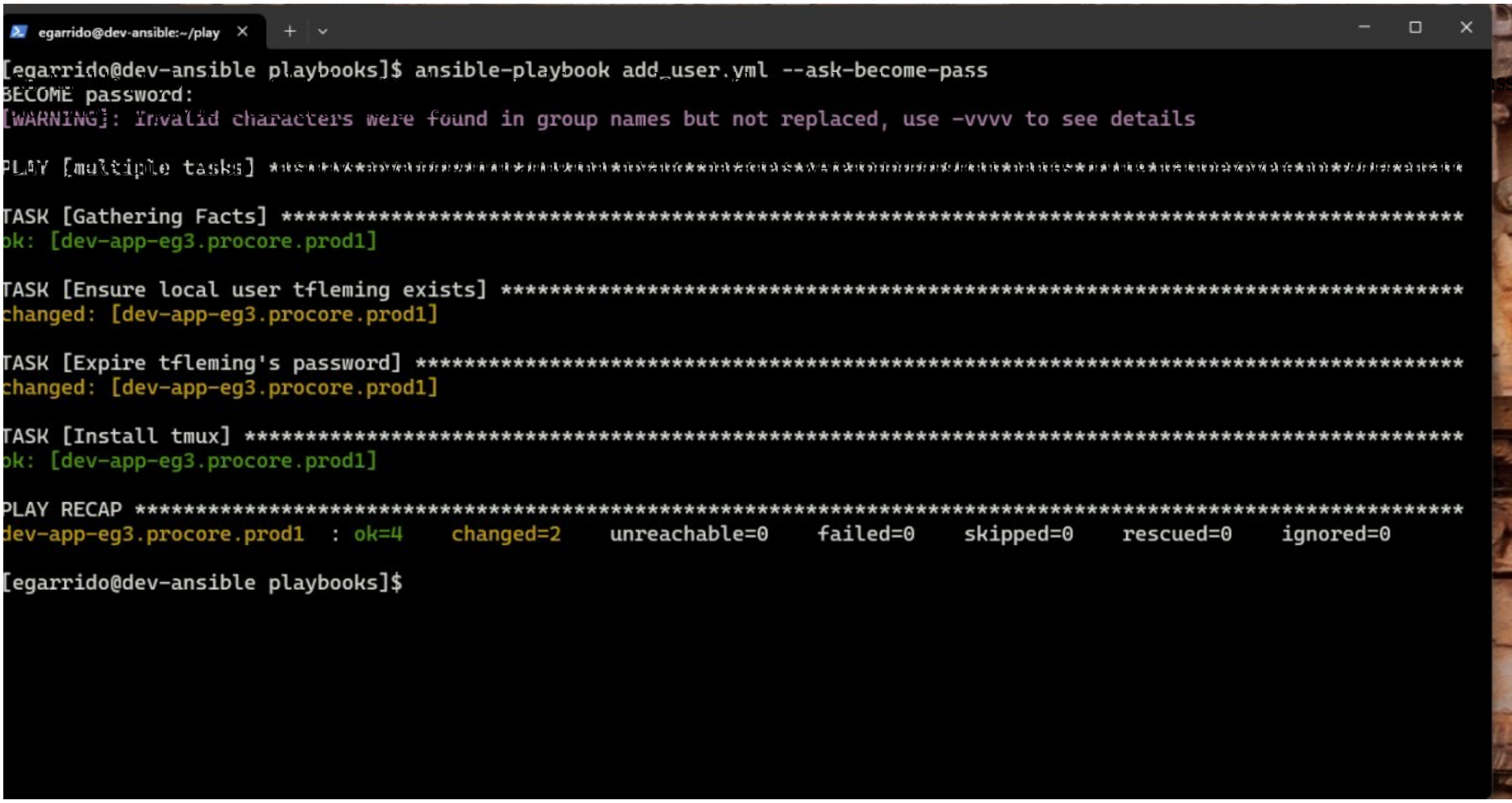
Uses the package module to install the tmux utility on the target system.



```
---
- name: multiple tasks
  hosts: dev-app-eg3.procore.prod1
  become: true
  tasks:

    - name: Ensure local user tfleming exists
      user:
        name: tfleming
        state: present
        shell: /bin/bash
        create_home: yes

    - name: Expire tfleming's password
      command: chage -d 0 tfleming

    - name: Install tmux
      package:
        name: tmux
~
~
~
~
~
~
~
-- INSERT --                                                                                              6,35          All
```

An Ansible playbook named add_user.yml is executed from the dev-ansible control node using the ansible-playbook command with --ask-become-pass, prompting for privilege escalation credentials.

During execution, Ansible displays a warning indicating that invalid characters were found in group names, noting that they were not replaced and suggesting -vvvv for additional debugging details. Despite the warning, the playbook proceeds normally.



```
[egarrido@dev-ansible playbooks]$ ansible-playbook add_user.yml --ask-become-pass
BECOME password:
[WARNING]: Invalid characters were found in group names but not replaced, use -vvvv to see details

PLAY [multiple tasks] ***************************************************************************************************

TASK [Gathering Facts] **************************************************************************************************
ok: [dev-app-eg3.procore.prod1]

TASK [Ensure local user tfleming exists] ********************************************************************************
changed: [dev-app-eg3.procore.prod1]

TASK [Expire tfleming's password] ***************************************************************************************
changed: [dev-app-eg3.procore.prod1]

TASK [Install tmux] *****************************************************************************************************
ok: [dev-app-eg3.procore.prod1]

PLAY RECAP **************************************************************************************************************
dev-app-eg3.procore.prod1  : ok=4    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

[egarrido@dev-ansible playbooks]$
```

his project showcases an Ansible playbook used to automate common Linux system dministration tasks. The playbook manages local user creation, enforces password xpiration to require aa change at first login, and ensures required packages are nstalled on the target host. Execution results confirm successful task completion with lear reporting of applied changes, demonstrating idempotent, repeatable automation uitable for enterprise environments.