

Ansible Directory Management Automation

This repository demonstrates the use of Ansible automation to standardize directory creation and permissions across multiple Linux hosts in a development and staging environment. The playbook ensures that a user-specific directory is consistently created under /opt/scripts on all targeted systems.

The automation leverages Ansible variables to dynamically create directories based on the executing user, while enforcing correct ownership, group assignment, and permissions. Syntax validation was performed prior to execution to ensure proper YAML structure and reliable deployment.

The playbook was executed with privilege escalation and verified across multiple hosts, confirming successful and repeatable results. Post-execution validation shows that the desired directory structure exists uniformly across environments, demonstrating idempotent behavior and configuration consistency.

This project highlights practical skills in:

- Writing and validating Ansible playbooks

- Using privilege escalation securely (become)

- Managing file system state across multiple hosts

- Enforcing permissions, ownership, and group policies

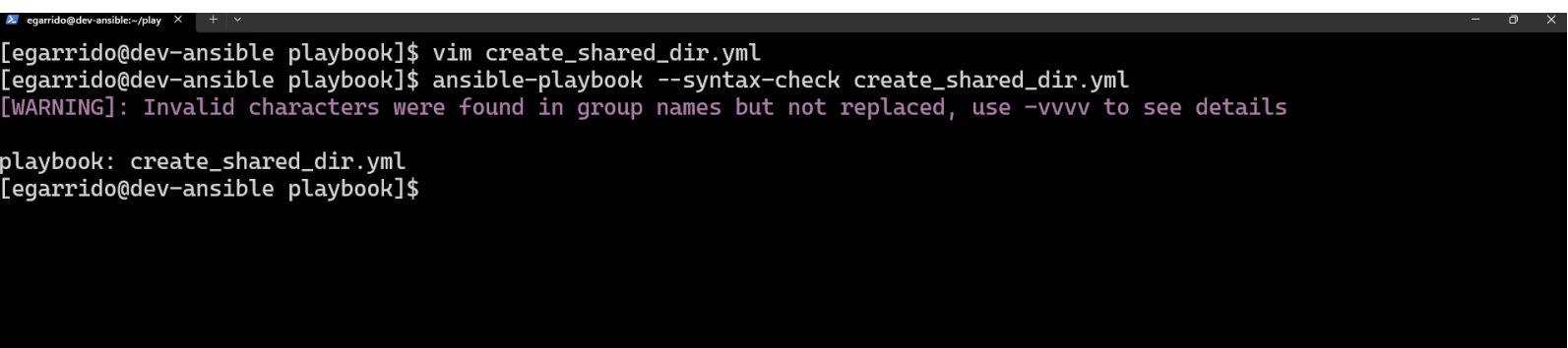
- Verifying results through manual and automated checks

Overall, this repository reflects real-world infrastructure automation practices commonly used in enterprise Linux and DevOps environments.

An Ansible playbook was created to ensure a standardized directory structure under /opt/scripts for the executing user across the dev-eg3 host group. The task enforces directory existence, correct ownership, group assignment, and permissions using Ansible variables, demonstrating idempotent configuration management and secure privilege escalation.

```
egarrido@dev-ansible: ~/play
+ ~
---
hosts: dev-eg3
become: true
tasks:
  - name: Ensure /opt/scripts/{{ ansible_user_id }}/ exists
    file:
      path: "/opt/scripts/{{ ansible_user_id }}/"
      state: directory
      owner: "{{ ansible_user_id }}"
      group: webmaster
      mode: '0775'
```

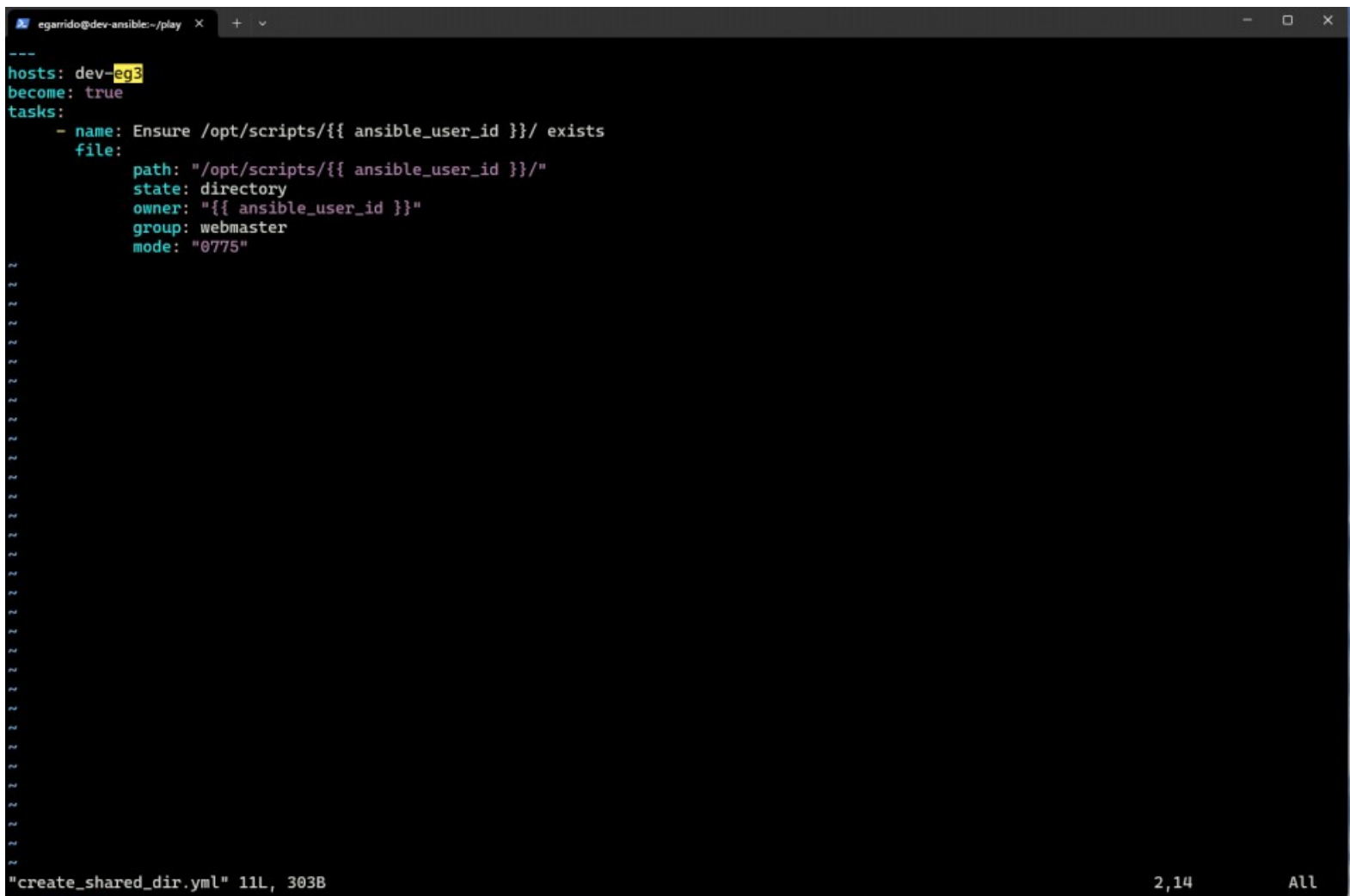
The Ansible playbook was validated using `ansible-playbook --syntax-check`, confirming correct YAML structure and task definitions. The syntax check completed successfully, indicating the playbook does not have a syntax error

A terminal window with a dark background and light text. The window title is 'egarrido@dev-ansible:~/play'. The terminal shows the following commands and output:

```
[egarrido@dev-ansible playbook]$ vim create_shared_dir.yml
[egarrido@dev-ansible playbook]$ ansible-playbook --syntax-check create_shared_dir.yml
[WARNING]: Invalid characters were found in group names but not replaced, use -vvvv to see details

playbook: create_shared_dir.yml
[egarrido@dev-ansible playbook]$
```

An Ansible playbook was defined to create a user-specific directory under /opt/scripts with enforced ownership, group assignment, and permissions using templated variables.



```
---
hosts: dev-eg3
become: true
tasks:
  - name: Ensure /opt/scripts/{{ ansible_user_id }}/ exists
    file:
      path: "/opt/scripts/{{ ansible_user_id }}/"
      state: directory
      owner: "{{ ansible_user_id }}"
      group: webmaster
      mode: "0775"
```

"create_shared_dir.yml" 11L, 303B 2,14 All

An Ansible playbook syntax issue was identified and resolved by validating the file with `ansible-playbook --syntax-check`. After correcting the playbook structure to use a proper list of plays, the syntax check completed successfully, confirming the playbook is valid and ready to run.

```
egarrido@dev-ansible:~/play x + -
egarrido@dev-app-eg3 ~]$ ssh egarrido@dev-ansible
egarrido@dev-ansible) Password:
ast login: Tue Sep 23 11:39:33 2025 from 10.1.31.124
egarrido@dev-ansible ~]$ cd playbooks
egarrido@dev-ansible playbooks]$ ll
total 4
-rw-r--r--. 1 egarrido egarrido 303 Sep 23 09:40 create_shared_dir.yml
egarrido@dev-ansible playbooks]$ ansible-play --syntax-check dev-eg3
bash: ansible-play: command not found
egarrido@dev-ansible playbooks]$ ansible -m ping dev-eg3
[WARNING]: Invalid characters were found in group names but not replaced, use -vvvv to see details
ev-app-eg3.procore.prod1 | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
ev-performance-eg3.procore.prod1 | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
egarrido@dev-ansible playbooks]$ ansible-play --syntax-check create_shared_dir.yml
bash: ansible-play: command not found
egarrido@dev-ansible playbooks]$ vim create_shared_dir.yml
egarrido@dev-ansible playbooks]$ ansible-play --syntax-check dev-eg3
bash: ansible-play: command not found
egarrido@dev-ansible playbooks]$ vim create_shared_dir.yml
egarrido@dev-ansible playbooks]$ ansible-check create_shared_dir.yml
[WARNING]: Invalid characters were found in group names but not replaced, use -vvvv to see details
ERROR! A playbook must be a list of plays, got a <class 'ansible.parsing.yaml.objects.AnsibleMapping'> instead: /home/egarrido/playbooks/
create_shared_dir.yml

The error appears to be in '/home/egarrido/playbooks/create_shared_dir.yml': line 2, column 1, but may
be elsewhere in the file depending on the exact syntax problem.

The offending line appears to be:

--
hosts: dev-eg3
  here
egarrido@dev-ansible playbooks]$ vim create_shared_dir.yml
egarrido@dev-ansible playbooks]$ ansible-playbook --syntax-check create_shared_dir.yml
[WARNING]: Invalid characters were found in group names but not replaced, use -vvvv to see details

playbook: create_shared_dir.yml
egarrido@dev-ansible playbooks]$
```

the Ansible^{3/33/14} playbook initially failed due to missing privilege escalation credentials. After rerunning the playbook with `--ask-become-pass`, it executed successfully and created the required `/opt/scripts` directory on all target hosts, confirming proper use of become and successful task execution.

```
egarrido@dev-ansible:~/playbooks$ ansible-playbook create_shared_dir.yml
[WARNING]: Invalid characters were found in group names but not replaced, use -vvvv to see details

PLAY [dev-eg3] *****

TASK [Gathering Facts] *****
fatal: [dev-app-eg3.procore.prod1]: FAILED! => {"msg": "Missing sudo password"}
fatal: [dev-performance-eg3.procore.prod1]: FAILED! => {"msg": "Missing sudo password"}

PLAY RECAP *****
dev-app-eg3.procore.prod1 : ok=0    changed=0    unreachable=0    failed=1    skipped=0    rescued=0    ignored=0
dev-performance-eg3.procore.prod1 : ok=0    changed=0    unreachable=0    failed=1    skipped=0    rescued=0    ignored=0

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

PLAY [dev-eg3] *****

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

TASK [Ensure /opt/scripts/root/ exists] *****
changed: [dev-app-eg3.procore.prod1]
changed: [dev-performance-eg3.procore.prod1]

PLAY RECAP *****
dev-app-eg3.procore.prod1 : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
dev-performance-eg3.procore.prod1 : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

egarrido@dev-ansible playbooks$
```

The Ansible playbook was executed with privilege escalation using `--become -K` and completed successfully across all target hosts. The required user-specific directory under `/opt/scripts` was created with correct permissions, confirming consistent execution across the environment.

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

PLAY [dev-eg3] *****

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

TASK [Ensure /opt/scripts/egarrido/ exists] *****
changed: [dev-app-eg3.procore.prod1]
changed: [dev-performance-eg3.procore.prod1]
changed: [stage-web-eg3.procore.prod1]

PLAY RECAP *****
dev-app-eg3.procore.prod1 : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored
=0
dev-performance-eg3.procore.prod1 : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
stage-web-eg3.procore.prod1 : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored
d=0

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


Post-execution validation confirmed that the /opt/scripts/egarrido directory was successfully created on the target host with the expected ownership, group assignment, and permissions, verifying correct playbook execution and desired state enforcement.

```
egarrido@dev-app-eg3: /opt/ + ▼
drwxrwxr-x. 3 root root  23 Jul  6 08:04 script
drwxr-xr-x. 6 root root 141 Sep 22 22:10 scripts
[egarrido@dev-ansible opt]$ cd scripts
[egarrido@dev-ansible scripts]$ ll
total 12
-rw-r--r--. 1 root      root      522 May 18 16:39 create_script_dir.yml
drwxrwxr-x. 2 dmckelvey webmasters  6 May 17 17:16 dmckelvey
drwxr-xr-x. 2 root      root        6 Jul 15 02:19 kgrant
drwxr-xr-x. 2 root      root        6 Jul 24 21:53 psekara
drwxr-xr-x. 2 root      root        6 Aug 12 23:37 rsuprina
-rw-r--r--. 1 root      root      582 Aug 12 13:21 shared_script_dir.yml
-rw-r--r--. 1 root      root      433 Sep  7 20:00 tacquaye
[egarrido@dev-ansible scripts]$ cd
[egarrido@dev-ansible ~]$ ssh egarrido@dev-app-eg3
(egarrido@dev-app-eg3) Password:
last login: Thu Sep 25 00:23:34 2025 from 10.1.30.41
[egarrido@dev-app-eg3 ~]$ ls -ld /opt/scripts/egarrido
ls: cannot access '-': No such file or directory
ls: cannot access 'ld': No such file or directory
/opt/scripts/egarrido:
[egarrido@dev-app-eg3 ~]$ hostname
dev-app-eg3.procure.prod1
[egarrido@dev-app-eg3 ~]$ cd /opt
[egarrido@dev-app-eg3 opt]$ cd scripts
[egarrido@dev-app-eg3 scripts]$ ll
total 0
drwxrwxr-x. 2 egarrido webmasters 6 Sep 25 00:23 egarrido
drwxrwxr-x. 2 root     webmasters 6 Sep 23 13:52 root
[egarrido@dev-app-eg3 scripts]$
```


Directory validation was performed across multiple hosts, confirming that /opt/scripts/egarrido exists consistently with the correct ownership, group assignment, and permissions. This verification demonstrates successful, repeatable Ansible execution and consistent state enforcement across the environment.

```
egarrido@dev-performance-4 X + v
[egarrido@dev-app-eg3 scripts]$ ll
total 0
drwxrwxr-x. 2 egarrido webmasters 6 Sep 25 00:23 egarrido
drwxrwxr-x. 2 root      webmasters 6 Sep 23 13:52 root
[egarrido@dev-app-eg3 scripts]$ ssh egarrido@dev-ansible
(egarrido@dev-ansible) Password:
Last login: Thu Sep 25 00:19:55 2025 from 10.1.31.136
[egarrido@dev-ansible ~]$ cd /opt
[egarrido@dev-ansible opt]$ cd scripts
[egarrido@dev-ansible scripts]$ ll
total 12
-rw-r--r--. 1 root      root      522 May 18 16:39 create_script_dir.yml
drwxrwxr-x. 2 dmckelvey webmasters 6 May 17 17:16 dmckelvey
drwxr-xr-x. 2 root      root      6 Jul 15 02:19 kgrant
drwxr-xr-x. 2 root      root      6 Jul 24 21:53 psekar
drwxr-xr-x. 2 root      root      6 Aug 12 23:37 rsuprina
-rw-r--r--. 1 root      root      582 Aug 12 13:21 shared_script_dir.yml
-rw-r--r--. 1 root      root      433 Sep 7 20:00 tacquaye
[egarrido@dev-ansible scripts]$ cd
[egarrido@dev-ansible ~]$ ssh egarrido@dev-performance-eg3
(egarrido@dev-performance-eg3) Password:
Last login: Thu Sep 25 00:23:34 2025 from 10.1.30.41
[egarrido@dev-performance-eg3 ~]$ cd /opt
[egarrido@dev-performance-eg3 opt]$ cd scripts
[egarrido@dev-performance-eg3 scripts]$ ll
total 0
drwxrwxr-x. 2 egarrido webmasters 6 Sep 25 00:23 egarrido
drwxrwxr-x. 2 root      webmasters 6 Sep 23 13:52 root
[egarrido@dev-performance-eg3 scripts]$
```

Final validation across all environments confirmed that /opt/scripts/egarrido exists on each host with consistent ownership, group membership, and permissions, verifying reliable and repeatable Ansible automation.

```
egarrido@stage-web-eg3/op X + v
egarrido@dev-ansible opt]$ cd scripts
egarrido@dev-ansible scripts]$ ll
total 12
-rw-r--r--. 1 root      root      522 May 18 16:39 create_script_dir.yml
drwxrwxr-x. 2 dmckelvey webmasters 6 May 17 17:16 dmckelvey
drwxr-xr-x. 2 root      root       6 Jul 15 02:19 kgrant
drwxr-xr-x. 2 root      root       6 Jul 24 21:53 psekara
drwxr-xr-x. 2 root      root       6 Aug 12 23:37 rsuprina
-rw-r--r--. 1 root      root     582 Aug 12 13:21 shared_script_dir.yml
-rw-r--r--. 1 root      root     433 Sep  7 20:00 tacquaye
egarrido@dev-ansible scripts]$ cd
egarrido@dev-ansible ~]$ ssh egarrido@dev-performance-eg3
egarrido@dev-performance-eg3) Password:
last login: Thu Sep 25 00:23:34 2025 from 10.1.30.41
egarrido@dev-performance-eg3 ~]$ cd /opt
egarrido@dev-performance-eg3 opt]$ cd scripts
egarrido@dev-performance-eg3 scripts]$ ll
total 0
drwxrwxr-x. 2 egarrido webmasters 6 Sep 25 00:23 egarrido
drwxrwxr-x. 2 root      webmasters 6 Sep 23 13:52 root
egarrido@dev-performance-eg3 scripts]$ cd
egarrido@dev-performance-eg3 ~]$ ssh egarrido@stage-web-eg3.procore.prod1
last login: Thu Sep 25 00:23:34 2025 from 10.1.30.41
egarrido@stage-web-eg3 ~]$ cd /opt
egarrido@stage-web-eg3 opt]$ cd scripts
egarrido@stage-web-eg3 scripts]$ ll
total 0
drwxrwxr-x. 2 egarrido webmasters 6 Sep 25 00:23 egarrido
egarrido@stage-web-eg3 scripts]$
```

Summary

An Ansible playbook was developed to create a user-specific directory under /opt/scripts with enforced ownership, group membership, and permissions. The playbook was validated using `ansible-playbook --syntax-check`, during which a structural syntax issue was identified and corrected. Initial execution failed due to missing privilege escalation credentials, which was resolved by rerunning the playbook with `--become -K`.

After successful execution, the directory was created consistently across all targeted development and staging hosts. Manual verification on each system confirmed the presence of /opt/scripts/egarrido with the correct permissions and ownership, demonstrating reliable, repeatable, and idempotent Ansible automation across multiple environments.