

Exercise 04 — From Playbook to Role

Ansible for Proxmox VE · Module 04

credativ GmbH

~10 min · in ~/ansible-proxmox · needs exercise 02; exercise 03 is not required

Objective: turn `playbooks/10-node-prep.yml` into the role `pve_node` and prove that the behaviour did not change.

i Where to look

Everything here was on the slides; the handout has the finished role file by file. [Roles](#) · [Variable precedence](#) · [template](#) · [ansible-lint](#)

The directories, `site.yml` and the template are already in the starter. What is missing is the content.

1 — Give the role its defaults

`roles/pve_node/defaults/main.yml` has the values as comments. Uncomment them, then **delete `pve_packages` from `inventory/group_vars/pve.yml`**.

Leave `pve_cluster_name` and `pve_root_password` where they are: those describe your environment, not the role.

Done when: `pve_packages` appears exactly once in the project.

```
grep -rn pve_packages inventory/ roles/
```

2 — Move the tasks and the handler

From `playbooks/10-node-prep.yml` into `roles/pve_node/tasks/main.yml` and `handlers/main.yml`: the tasks only — no `hosts:`, no `tasks:` key, indentation shifted left. Replace the hard-coded repository values with the variables you just uncommented.

Leave the old playbook alone. It stays as the before-picture.

Done when: no `ansible.builtin.debug` placeholder is left in the role.

3 — Render the message of the day

`roles/pve_node/templates/motd.j2` is written for you — read it first, it is the whole point of the module. Add the task that renders it:

```
ansible-doc ansible.builtin.template    # src, dest, owner, group, mode
```

i Note

/etc/motd is a deliberate choice. A template owns the **whole** file, so pick one nothing else writes. /etc/hosts would be wrong: the provisioning owns it and Proxmox VE resolves its own node name through it.

Done when: the role has a template task with tags: [motd].

4 — Run it three times

```
ansible-playbook site.yml --check --diff
ansible-playbook site.yml
ansible-playbook site.yml
```

Done when:

```
# first real run
pve01 : ok=8  changed=1  unreachable=0  failed=0
# second run
pve01 : ok=8  changed=0  unreachable=0  failed=0
```

The single changed is /etc/motd, because the flat playbook never wrote it. Everything else was already in the state the role describes — that is the proof the refactoring changed the structure and not the outcome.

If you finish early

- Put pve_repo_component: from-group-vars into inventory/group_vars/pve.yml, then `ansible pve01 -m ansible.builtin.debug -a 'var=pve_repo_component'`. Which value wins over the role default, and why?
- Add `--extra-vars 'pve_repo_component=from-cli'`. Which wins now? Use debug only — do not run site.yml with a bogus repository component. Remove the override afterwards.
- `ansible-playbook site.yml --list-tags`, then run only `--tags motd`.
- `ansible-lint site.yml roles/ playbooks/`. Then read .ansible-lint in the project: which finding was fixed, which was recorded as a decision, and why?

Solutions

Solutions are discussed live with the trainer.