

Exercise 02 — Preparing the Nodes

Ansible for Proxmox VE · Module 02

credativ GmbH

~15 min · in ~/ansible-proxmox · after exercise 01

Objective: bring three freshly installed nodes to the state a cluster needs, and prove that running it a second time changes nothing.

i Where to look

Everything here was on the slides; the handout for this module is the reference. **Handlers** · apt · deb822_repository · file · **Proxmox package repositories**

1 — Fill in the playbook

playbooks/10-node-prep.yml is a skeleton: the task names, the order and the handler are already there, and every body is an `ansible.builtin.debug` with a `TODO`. Replace the four that say `TODO`:

Task	Module
Update apt cache (the handler)	apt, update_cache: true
Disable the enterprise repositories	file, state: absent, over a loop
Enable the no-subscription repository	deb822_repository
Install the admin tools	apt

Uncomment the two `notify:` lines while you are there, and look the modules up rather than guessing:

```
ansible-doc ansible.builtin.deb822_repository  # uris, suites, components,
signed_by
ansible-doc -s ansible.builtin.apt             # every option as a snippet
```

`suites` should be the release the node reports, not the string `trixie`. The fact is `ansible_distribution_release`.

Done when: task 2 gets past all four without an error.

2 — See what it would do

```
ansible-playbook playbooks/10-node-prep.yml --check --diff
```

Done when: tasks report changed, nothing fails, and the diff shows you the repository file before it is written anywhere.

3 — Run it, then run it again

```
ansible-playbook playbooks/10-node-prep.yml
ansible-playbook playbooks/10-node-prep.yml
```

Done when:

```
# first run
pve01 : ok=8  changed=4  unreachable=0  failed=0
# second run
pve01 : ok=7  changed=0  unreachable=0  failed=0
```

changed=0 is the point of the exercise. One task fewer on the second run is worth a second of thought: which one, and why?

4 — Two tasks that do not change anything

The playbook ends with a resolution check and a time check. Neither configures anything. Read those two tasks and answer for yourself why `changed_when: false` is on one of them, and what `--check` would do without `check_mode: false` on the other.

Done when: you can say what would break if both lines were removed.

If you finish early

- Add `http` to `pve_packages` in `inventory/group_vars/pve.yml` and re-run. How many tasks report changed?
- Tag the two repository tasks with `tags: repos`, then run `--tags repos`.
- `ansible pve01 -m ansible.builtin.command -a 'pvecm status'` still fails. That is module 03.
- Break it on purpose: `ansible pve03 -m ansible.builtin.command -a 'hostname bogus'`, then re-run with `--limit pve03`. Which task notices, and which does not? Set the hostname back afterwards.

Solutions

Solution — Task 1

The finished playbook is `courses/ansible-proxmox/code/playbooks/10-node-prep.yml`. The three parts participants most often get wrong:

1. **notify must match the handler name exactly**, capitalisation included. A typo does not error; the handler simply never runs.

2. **meta: flush_handlers must sit before the package install.** Without it the cache is refreshed at the end of the play, after apt already needed it.
3. **suites is a fact, not a literal.** `ansible_distribution_release` keeps the playbook working on the next Debian.

`pve_packages` is already in `inventory/group_vars/pve.yml` in the starter, so nobody has to type the list. If someone gets `VARIABLE IS NOT DEFINED`, their `group_vars` sits in the wrong place: it belongs next to the inventory file, in `inventory/group_vars/`.

Solution — Task 2

```
ansible-playbook playbooks/10-node-prep.yml --check --diff
```

Expected: several changed tasks, nothing failed. The repository tasks show what they would remove and add.

Worth pointing out: `--check` normally skips command tasks, which would leave `ntp_state` undefined and make the following assert fail. That is why the playbook carries `check_mode: false` on the `timedatectl` task. It only reads, so it is safe in a dry run. If a participant omitted that line, this is exactly the error they will see, and it is worth walking through rather than just fixing.

Solution — Task 3

Measured on the lab, on freshly installed nodes:

```
# first run
pve01 : ok=8  changed=4  unreachable=0  failed=0
# second run
pve01 : ok=7  changed=0  unreachable=0  failed=0
```

The missing task on the second run is the handler. It runs only when a task notifies it, so on an unchanged system it does not appear in the recap at all. Ask the room which task disappeared before you tell them.

If the second run is not `changed=0`, the culprit is almost always one of:

Symptom	Cause
The <code>timedatectl</code> task reports changed	<code>changed_when: false</code> missing
The resolution check reports changed	<code>changed_when: false</code> missing
The <code>apt</code> task reports changed	<code>state: latest</code> instead of <code>state: present</code>

Solution — Task 4

- `changed_when: false` on the two read-only command tasks. Without it command reports changed every run, and the idempotency proof from task 3 is gone.
- `check_mode: false` on the `timedatectl` task. Without it `--check` skips the task, `ntp_state` is never registered, and the assert fails on an undefined variable — a dry run that fails for a reason that has nothing to do with the target state.

The resolution check is a check, not a fix. If it fails, the answer is to repair DNS or the provisioning, not to add a task that writes /etc/hosts: Proxmox VE resolves its own node name through that file.

Solution — Optional tasks

```
ansible pve -m ansible.builtin.command -a 'apt-get update' # no enterprise  
repo error  
ansible pve01 -m ansible.builtin.command -a 'getent hosts pve02 pve03'  
ansible pve01 -m ansible.builtin.command -a 'pvecm status' # still fails -  
module 03
```

Adding htop changes one task: the apt install. Everything else stays ok.

Close the module here: the nodes are ready, the cluster is not.

Automated verification

```
./tests/run.sh 02
```