

Exercise 03 — Forming the Cluster

Ansible for Proxmox VE · Module 03

credativ GmbH

~20 min · in ~/ansible-proxmox · after exercise 02

Objective: create the cluster on pve01, join the other two one at a time, and end with a playbook you can run again without it doing anything.

i Where to look

Everything here was on the slides; the handout has the same playbook with every line explained, and the curl equivalents. proxmox_cluster · proxmox_cluster_join_info · delegate_to · serial · [Proxmox VE Cluster Manager](#)

1 — Give the playbook the password

inventory/group_vars/pve.yml ships with a placeholder. The API modules authenticate with it, so replace it:

```
pve_root_password: "password123"
```

Done when:

```
ansible pve01 -m ansible.builtin.debug -a 'var=pve_root_password'
```

prints the password rather than VARIABLE IS NOT DEFINED.

The collection is already installed — deploy.sh put it in place offline. Check with ansible-doc -l community.proxmox | wc -l; it lists 62 modules.

2 — Fill in the playbook

playbooks/20-cluster.yml has the structure already: two plays, serial: 1 on the second, and a TODO in each of the five task bodies. The comments say which module and which options.

```
ansible-doc community.proxmox.proxmox_cluster
ansible-doc community.proxmox.proxmox_cluster_join_info
```

Three things the skeleton does not do for you:

- every API task needs delegate_to: localhost — the module talks HTTPS, it does not run on the node

- the two waiting tasks are `until / retries / delay` loops, not `wait_for`
- the fingerprint for the join sits in the registered join info, under the primary's entry in `cluster_join.nodelist`

Done when: the file has no `ansible.builtin.debug` left in it.

3 — Run it

```
ansible-playbook playbooks/20-cluster.yml
```

Open the GUI of pve01 while it runs. One to three minutes.

i Note

No --check here. A dry run creates no cluster, so the join play has nothing to read the join information from and stops with “*Node is not part of a cluster and does not have any join information*”. That is correct behaviour: check mode cannot rehearse a sequence whose second step depends on the first having really happened.

Done when:

```
pve01 : ok=2  changed=1  unreachable=0  failed=0
pve02 : ok=3  changed=1  unreachable=0  failed=0
pve03 : ok=3  changed=1  unreachable=0  failed=0
```

and all three nodes appear under *Datacenter* in the GUI.

4 — Prove it, twice

```
ansible pve01 -m ansible.builtin.command -a 'pvecm status'
ansible pve  -m ansible.builtin.command -a 'ls /etc/pve/nodes'
ansible-playbook playbooks/20-cluster.yml
```

Done when: Quorate: Yes with three nodes, the same three directories on every node — that is `pmxcfs` replicating — and the second playbook run reports `changed=0` everywhere.

If you finish early

- Look at `/etc/pve/corosync.conf` on any node. Find your `link0` addresses and the `config_version` counter.
- Write a file into `/etc/pve/` on pve02 and find it on pve03: `ansible pve02 -m ansible.builtin.copy -a 'content=hello dest=/etc/pve/hello.txt'`. Remove it again afterwards.
- Stop `corosync` on pve03 (`systemctl stop corosync`), then run `pvecm status` on pve01 and on pve03 and compare the two answers. Start it again.
- `ssh root@192.168.0.1` and look at `/root/.ssh/authorized_keys`. Where does it actually point, and what does that mean for a node that joins later?

Solutions

Solutions are discussed live with the trainer.