

# Exercise 05 — Operating the Cluster

## Ansible for Proxmox VE · Module 05

credativ GmbH

~20 min · in ~/ansible-proxmox · needs a working cluster from exercise 03

**Objective:** give the cluster storage, a service account, and a guest that installs itself — all through the API, from one playbook.

### i Where to look

Everything here was on the slides. The handout has each task in full, with the curl equivalent and a link into the API viewer of your own cluster (<https://192.168.0.1:8006/pve-docs/api-viewer/>). proxmox\_storage · proxmox\_user · proxmox\_kvm · proxmox\_backup · blockinfile

playbooks/30-operations.yml is a skeleton again: the play structure, the module\_defaults block and every task name are there, each body a TODO. inventory/group\_vars/all.yml already holds the values.

## 1 — Check the ground before you build on it

The variables assume things about your cluster. Verify the one that actually varies:

```
ansible pve01 -m ansible.builtin.command -a 'pvesm status'
```

**Done when:** the Name column contains the storage named in pve\_vm\_storage. On this lab that is local-lvm; on a ZFS install it would be local-zfs.

## 2 — Storage, in two halves

Two tasks, two different worlds — that is the lesson of this one:

- the backup **directory** on the nodes: file, over SSH, on every host
- the same directory **registered as cluster storage**: proxmox\_storage, type: dir, one API call for the whole cluster
- and the shared **NFS library**: proxmox\_storage, type: nfs, content: [import, iso, vztmp]

```
ansible-doc community.proxmox.proxmox_storage
```

**Done when:** the three task bodies no longer say TODO.

### 3 — A service account with a token

Three tasks: the group, the user with a token, and the permission on /.

#### ! Important

`privsep: false` gives the token its user's permissions. With `privsep: true` the token additionally needs its own ACL, and the effective rights are the **intersection** of both — a token that works perfectly for one call and returns an empty list for the next is almost always this.

The API returns a token secret exactly once, at creation. The playbook writes it to `~/pve-token.json`; module 06 needs it.

**Done when:** those three bodies no longer say TODO.

### 4 — A guest that installs itself

Four tasks: activate the NFS library and find the image, create the VM from it, start it, wait for SSH.

```
ansible-doc community.proxmox.proxmox_kvm    # scsi, ide, net, boot, sshkeys,
timeout
```

Read the comment above the first of them before you write it. Proxmox VE mounts an NFS storage on **first activation**, not when you attach it, so importing in the same run fails with `can't find file` — and waiting for the path does not help, because nothing has asked the node to mount it yet. Listing the contents activates the storage and confirms the image in one go.

Then add the backup task after it.

**Done when:** the bodies are filled in. Do not run it yet.

### 5 — The direct route, then run it

The last play edits `/etc/network/interfaces` on the node with `blockinfile` and applies it with `ifreload -a`. No module owns this file.

#### ⚠ Warning

`ifreload -a` applies the change **immediately**. Get the block wrong and you can cut the node off the network. That is why the playbook keeps a pristine copy first, and why the marker comment matters: `blockinfile` uses it to find its own block again instead of appending a second one on every run.

Now run the whole thing, twice:

```
ansible-playbook playbooks/30-operations.yml
ansible-playbook playbooks/30-operations.yml
```

**Done when:** the first run looks like this, and the second is quiet apart from one task:

```
# first run
localhost : ok=11  changed=9  unreachable=0  failed=0
pve01     : ok=1   changed=1  unreachable=0  failed=0
pve02     : ok=1   changed=1  unreachable=0  failed=0
pve03     : ok=4   changed=4  unreachable=0  failed=0

# second run
localhost : ok=10  changed=1  unreachable=0  failed=0  skipped=1
```

That changed=1 is **Back up the VM**, and it will never be anything else: a backup is an action, not a state, so there is nothing for the module to compare against. Which task is skipped on the second run, and why? The answer is in the playbook.

web01 is running in the GUI, and `ssh student@192.168.0.120` gets you in without a password.

## If you finish early

- Look at `~/pve-token.json`. Why can the playbook only write it once?
- Open the API viewer and find `POST /nodes/{node}/qemu`. Compare its parameter names with the options you passed to `proxmox_kvm`.
- Create a container instead: `community.proxmox.proxmox` with `pve_ct_template`.
- Add a second guest by changing `pve_vm_id` and `pve_vm_name` on the command line with `--extra-vars`.
- Scheduled backup jobs have no module. Find the API path that would create one.

## Solutions

*Solutions are discussed live with the trainer.*