

Bonus Exercises

Ansible for Proxmox VE

credativ GmbH

Table of contents

B1 — Make the backup idempotent	1
B2 — Scope the token to a pool	1
B3 — A scheduled backup job without a module	2
B4 — Template and clone	2
B5 — Let tags drive a play	2
B6 — A health check playbook	2
Solutions	2

For participants who finish the exercises early. Each one stands on its own and assumes the state after exercise 05: a quorate cluster, the storage pve-backup, the service account automation@pve with its token, and VM 9001 web01 on pve02.

They are ordered roughly by effort, not by importance.

Documentation

community.proxmox **collection** · ansible.builtin.uri · **Conditionals** · **API browser** · offline: ansible-doc -l community.proxmox

B1 — Make the backup idempotent

Objective: the backup task in playbooks/30-operations.yml reports changed on every run, because a backup is an action rather than a state. Guard it so that a second run within the same day does nothing.

Read the existing backups with community.proxmox.proxmox_backup_info and skip the backup task when one already exists for this VMID.

Check: run the playbook twice. The second run reports changed=0 for the whole play.

B2 — Scope the token to a pool

Objective: automation@pve currently holds PVEVMAAdmin on /, which is every guest in the cluster. Narrow it to one resource pool, and make the token itself the subject of the permission.

1. Create a pool and put VM 9001 in it (proxmox_pool, proxmox_pool_member).
2. Give the **token** PVEVMAAdmin on /pool/<id> (proxmox_access_acl with type: token).
3. Switch the token to privsep: true and remove the datacenter-wide grant.

Check: the token can still stop and start 9001, and can no longer create a guest outside the pool.

B3 — A scheduled backup job without a module

Objective: no module creates a scheduled backup job. Build one with `ansible.builtin.uri` against `/cluster/backup`, and give it the idempotency the module would have provided.

Query the existing jobs first, and create yours only when no job with your comment exists.

Check: run it twice; the second run creates nothing. The job appears under *Datacenter* → *Backup*.

B4 — Template and clone

Objective: turn `web01` into a template and deploy two guests from it.

`proxmox_kvm` converts an existing VM with `template: true`, and clones with `clone:` plus `newid:.` A template cannot be started, and it cannot be converted back.

Check: `qm list` on `pve02` shows the template and two clones. Running the playbook again creates nothing.

B5 — Let tags drive a play

Objective: close the loop between the dynamic inventory and a playbook. Tag a guest in the GUI, and have a play act on the group that the tag produces.

Write a play against the group `webserver`s from exercise 06 that prints the VMID and node of every member. Then add the tag to a second guest and re-run without touching any file.

Check: the second guest appears in the output, and you changed nothing but a tag.

B6 — A health check playbook

Objective: write `playbooks/90-check.yml` that only reads, changes nothing, and fails loudly when the cluster is not healthy.

Assert that the cluster is quorate, that every node in the inventory is a member and online, and that `pve-backup` is active. Use the `*_info` modules, not `command`.

Check: the playbook reports `changed=0` and passes. Stop `corosync` on one node and it fails with a message that says what is wrong.

Solutions

Solutions are discussed with the trainer.