

Exercise 06 — Dynamic Inventory

Ansible for Proxmox VE · Module 06

credativ GmbH

~8 min · in ~/ansible-proxmox · needs exercise 05: web01 and the API token

Objective: stop maintaining a list of guests. Ask the cluster instead.

i Where to look

Everything here was on the slides; the handout has every option of the plugin. **Inventory plugin · Dynamic inventory** · ansible-inventory · offline: ansible-doc -t inventory community.proxmox.proxmox

inventory/lab.proxmox.yml is already in the project, configured except for task 3.

1 — Hand the plugin the token

Module 05 wrote the secret to ~/pve-token.json, once, at creation:

```
export PVE_TOKEN_SECRET=$(jq -r .ansible ~/pve-token.json)
```

Done when: echo \${#PVE_TOKEN_SECRET} prints 36.

An environment variable is fine for a workshop shell. In a pipeline the secret belongs in Ansible Vault or your CI secret store.

2 — Ask the cluster

```
ansible-inventory -i inventory/lab.proxmox.yml --graph
ansible-inventory -i inventory/lab.proxmox.yml --host web01
```

Done when: the graph has proxmox_nodes with your three nodes and proxmox_all_qemu with web01, and the host output shows the guest's own configuration — proxmox_ciuser: student, proxmox_ipconfig0 with the address you set in module 05. Nobody typed any of that into an inventory file.

i Note

A 401 here means the token does not match the cluster — most often the secret was never exported, or module 05 ran against a different cluster. The file name matters too: it **must** end in `.proxmox.yml`, otherwise the plugin ignores it silently and you get an empty inventory with no error at all.

3 — Let a tag decide the group

In the GUI, open `web01` and add the tag `web` (*Options* → *Tags*). Then uncomment the last block of `inventory/lab.proxmox.yml` and run the graph again.

Done when: the groups `tagweb` and `webservers` exist and contain `web01`.

The person creating the VM decided which group it belongs to, in the GUI, and your playbooks follow.

4 — Both inventories at once

```
ansible-inventory -i inventory/hosts.yml -i inventory/lab.proxmox.yml --graph
```

Done when: your static groups (`pve`, `pve_primary`, `pve_secondary`) and the generated ones appear in the same tree. The nodes stay static — they have to exist before anything can be dynamic.

If you finish early

- Create a second VM in the GUI, re-run the graph, and watch it appear without you touching a file.
- Stop `web01` and watch it move from `proxmox_all_running` to `proxmox_all_stopped`.
- Put `inventory = inventory/hosts.yml, inventory/lab.proxmox.yml` in `ansible.cfg` so both sources are used by default.
- Restrict the token to `PVEAuditor` and check that listing still works while creating a VM no longer does.

Solutions

Solution — Task 1: the token

```
export PVE_TOKEN_SECRET="$(jq -r '.ansible' ~/pve-token.json)"
echo "${PVE_TOKEN_SECRET:0:8}..." # sanity check without printing the whole secret
```

The exact JSON key depends on what `proxmox_user` returned; `jq .` on the file shows it. If the participant lost the secret, recreate the token:

```
ansible pve01 -m ansible.builtin.command -a 'pveum user token remove
automation@pve ansible'
ansible-playbook playbooks/30-operations.yml
```

Solution — the plugin configuration

File as printed.

The one failure mode that costs the most time: a file name not ending in `.proxmox.yml`. The plugin declines it silently and the inventory is empty, with no error and no warning. Demonstrate the diagnosis once:

```
ansible-inventory -i inventory/lab.yml --graph -vvv
# Skipping due to inventory source not ending in "proxmox.yml" nor
"proxmox.yml"
```

Solution — Task 2: what you get

```
ansible-inventory -i inventory/lab.proxmox.yml --graph
```

```
@all:
  |--@proxmox_all_qemu:
  | |--web01
  |--@proxmox_all_running:
  | |--web01
  |--@proxmox_all_stopped:
  |--@proxmox_nodes:
  | |--pve01
  | |--pve02
  | |--pve03
  |--@proxmox_pve02_qemu:
  | |--web01
  |--@ungrouped:
```

```
ansible-inventory -i inventory/lab.proxmox.yml --host web01
# proxmox_vmid: 9001, proxmox_node: pve02, proxmox_status: running, ...
```

If the result is empty but there is no error: wrong file name (see task 2), or the token has no permissions. A token without privileges returns an empty result set rather than a permission error. That asymmetry surprises everyone once.

Solution — Task 3: the tag

Tag in the GUI: select web01, *Summary* → *Tags* → *Edit*, add web. Alternatively from the command line:

```
ansible pve02 -m ansible.builtin.command -a 'qm set 9001 --tags web'
```

Then the additions printed in the task. Expected:

```
| --@tagweb:
|   |--web01
|--@webservers:
|   |--web01
```

If the groups are missing: `want_facts: true` is not set. `proxmox_tags_parsed` is a fact, and without facts the expression evaluates against an undefined variable and silently produces no group.

Make the operational point here: the person who creates the VM decides its role, with a tag, in the interface they already use. Nobody has to edit an Ansible file. That is the argument that wins people over to dynamic inventories.

Solution — Task 4: both at once

```
ansible-inventory -i inventory/hosts.yml -i inventory/lab.proxmox.yml --graph
```

Both trees merge. Note that `pve01` now appears twice conceptually, once as a static host with `ansible_host` and once inside `proxmox_nodes`, and that host variables from the *later* source win. In practice you keep the nodes static (they must exist before anything dynamic can be queried) and let the guests come from the API.

Solution — Optional tasks

```
# new VM in the GUI, then:
ansible-inventory -i inventory/lab.proxmox.yml --graph          # it is
simply there

# module 05 left the guest running, so stop it to see the move
ansible pve02 -m ansible.builtin.command -a 'qm stop 9001'
ansible-inventory -i inventory/lab.proxmox.yml --graph          # now in
proxmox_all_stopped
ansible pve02 -m ansible.builtin.command -a 'qm start 9001'    # put it back
```

```
# ansible.cfg
[defaults]
inventory = inventory/hosts.yml,inventory/lab.proxmox.yml
```

```
# read-only token for inventory use
- community.proxmox.proxmox_access_acl:
    path: /
    type: group
    ugid: automation
    roleid: PVEAuditor
    propagate: 1
```

After removing the PVEVMAdmin entry, listing still works and VM creation fails with a permission error. That contrast shows least privilege better than any explanation. Use it if the room is interested and you have the time.

Automated verification

```
./tests/run.sh 06
```

Checks: the inventory file is named correctly and parses, the plugin returns at least the three nodes and VM 9001, the generated groups exist, and once a tag is set, the webservers group contains the tagged guest.