

# Exercise 01 — Ansible in 30 Minutes

## Ansible for Proxmox VE · Module 01

credativ GmbH

~10 min · on desktop, in ~/ansible-proxmox

**Objective:** talk to all three nodes from a project you understand, and watch a second run do nothing.

### ! Where the sheets are

All exercise sheets, handouts, the slides and the cheat sheet are on the share that is mounted on your desktop. Open it once and keep the tab:

```
xdg-open ~/Desktop/workshop/index.html
```

Every module has a handout (the reference, with more detail than the slides) and an exercise sheet like this one. You do not need to print or copy anything.

### i Where to look

Everything here was on the slides. When you want more than that: **Inventory · Ad-hoc commands · ansible.cfg · file module** · offline: `ansible-doc <module>` and the handout for this module.

## 1 — Know what you are running

~/ansible-proxmox is ready. Read `ansible.cfg` and `inventory/hosts.yml` before you use them, then check that Ansible parses the inventory the way you just read it:

```
cd ~/ansible-proxmox
ansible-inventory --graph
```

**Done when:** pve has three hosts, `pve_primary` one, `pve_secondary` two — and you can say why `pve01` appears twice.

## 2 — Reach the nodes

```
ansible pve -m ping
```

**Done when:** three SUCCESS blocks come back. Nothing asked for a password: `deploy.sh` put your SSH key on the nodes. Now see the unprepared case, which is what you will meet everywhere else:

```
ansible pve -m ping --ask-pass          # password123
```

### 3 — Aim

Say the number out loud *before* you press enter, three times:

```
ansible pve_primary -m ping
ansible pve_secondary -m ping
ansible 'pve0*' -m ping
```

**Done when:** one host answered, then two, then three. Groups and patterns select hosts; they are labels, not folders.

Ansible patterns are not shell globs: `*` works, `pve0[23]` matches nothing at all — and matching nothing is not an error, it is an empty run.

### 4 — Run the same thing twice

Create the directory `/root/demo` on all three nodes with the `file` module, then run your command a second time, unchanged.

The command is not written out here, and `file` is a module you have not used yet. Looking it up is half the exercise:

```
ansible-doc ansible.builtin.file          # OPTIONS near the top, EXAMPLES
further down
ansible-doc -s ansible.builtin.file      # every option as a paste-ready snippet
```

`ansible-doc` is the same content as [the module page online](#), offline and for the exact version you are running. Two options are enough here: `path` and `state`.

**Done when:** the first run reports CHANGED, the second SUCCESS, and nothing else in the result differs. If you can say why that matters, you have the point of this module.

### If you finish early

- Do task 4 again with `ansible.builtin.command` and `mkdir -p /root/demo`. What does it report the second time, and why?
- `ansible pve01 -m ansible.builtin.setup` with no filter. How many facts are there? Find `ansible_memtotal_mb`.
- `ansible-doc ansible.builtin.file` — which other values does `state` accept?
- Remove `/root/demo` again with a single ad-hoc command.
- `ansible pve01 -m ansible.builtin.command -a 'pvecm status'`. Read the error. It is true, and it is what module 03 is about.
- Write `ansible.cfg` and the inventory from scratch in an empty directory.

## **Solutions**

*Solutions are discussed live with the trainer.*