

# Capitolo 6 – Primo Playbook Reale: Setup Base Server

Dopo aver visto i comandi ad-hoc, è il momento di passare ai **playbook**, il cuore dell'automazione con Ansible.

Un playbook permette di definire lo **stato desiderato** dei server in modo dichiarativo e ripetibile.

---

## 6.1 Struttura di un Playbook

Un playbook è un file YAML composto da:

- `hosts` → gruppo di host target
- `become` → se abilitare l'escalation privilegi
- `tasks` → elenco di attività da eseguire

Esempio concettuale:

```
- name: Configurazione base server
  hosts: all
  become: true
  tasks:
    - name: Aggiornare cache pacchetti
      apt:
        update_cache: yes
```

---

## 6.2 Playbook Base

File: `playbooks/base.yml`

```
- name: Configurazione base server
hosts: all
become: true

tasks:
  - name: Aggiornamento cache apt
    apt:
      update_cache: yes
    when: ansible_os_family == "Debian"

  - name: Installazione pacchetti base
    package:
      name:
        - vim
        - curl
        - htop
        - git
      state: present

  - name: Creazione utente deploy
    user:
      name: deploy
      groups: sudo
      append: yes
      shell: /bin/bash
```

## Spiegazione

- `when` → condizione per eseguire il task
- `package` → modulo cross-distribution per installare pacchetti
- `user` → modulo per creare utenti con gruppo e shell

## 6.3 Esecuzione del Playbook

Esecuzione base:

```
ansible-playbook playbooks/base.yml
```

Modalità di **check** (simulazione senza modifiche):

```
ansible-playbook playbooks/base.yml --check
```

Modalità **verbose**:

```
ansible-playbook playbooks/base.yml -vvv
```

---

## 6.4 Best Practice per Playbook

- Non usare shell se esiste modulo nativo
- Testare con `--check` prima di eseguire in produzione
- Versionare i playbook con Git
- Separare ambienti (production, staging, dev)
- Usare variabili per parametri ripetuti
- Creare ruoli per componenti specifici (nginx, mysql, common)

---

## 6.5 Prossimi Passi

Dopo aver creato e testato il primo playbook:

- Strutturare progetti con ruoli
- Gestire variabili e segreti con `group_vars`, `host_vars` e `ansible-vault`
- Configurare playbook per deploy applicativi reali
- Implementare hardening base dei server
- Creare playbook modulari e riutilizzabili

---

## Conclusione

Il primo playbook rappresenta il primo passo concreto verso l'automazione.

Da questo momento è possibile standardizzare setup, gestire pacchetti, utenti e configurazioni in modo ripetibile e sicuro.

Nei capitoli successivi approfondiremo la **gestione di ruoli, variabili avanzate e segreti**, e la **strutturazione professionale di progetti Ansible**.

---

Revision #1

Created 2026-02-26 14:01:19 UTC by Pe

Updated 2026-02-26 14:01:43 UTC by Pe