Capitolo 5 – Comandi Ad-Hoc e Operatività Quotidiana

I comandi ad-hoc permettono di eseguire operazioni rapide sui nodi gestiti senza scrivere un playbook.

Sono utili per:

- Verifiche veloci
- Interventi urgenti
- Troubleshooting
- Operazioni una tantum

Non sostituiscono i playbook in ambienti strutturati, ma sono uno strumento operativo fondamentale per un sysadmin.

5.1 Sintassi Base

Struttura generale:

```
ansible <gruppo_host> -m <modulo> -a "<argomenti>"
```

Esempio:

```
ansible all -m ping
```

Componenti:

- `<gruppo_host>` → gruppo definito nell'inventory
- `-m` → modulo da utilizzare
- `-a` → argomenti del modulo

5.2 Verifica Connettività

Test base su tutti i server:

```
ansible all -m ping
```

Se tutto è configurato correttamente, l'output sarà:

- SUCCESS
- Nessun errore SSH
- Nessun errore Python

5.3 Eseguire Comandi Remoti

Modulo command

Esegue un comando senza passare dalla shell.

```
ansible all -m command -a "uptime"
```

Caratteristiche:

- Non interpreta pipe
- Non interpreta redirect
- Non interpreta variabili shell

È più sicuro rispetto a `shell`.

Modulo shell

Usa la shell del sistema remoto.

```
ansible all -m shell -a "df -h | grep /dev/sda1"
```

Usare solo quando necessario.

Best practice: preferire sempre moduli nativi.

5.4 Gestione Pacchetti

Installare un pacchetto su gruppo web:

```
ansible web -m apt -a "name=nginx state=present update_cache=yes"
```

Rimuovere un pacchetto:

```
ansible web -m apt -a "name=nginx state=absent"
```

Versione generica (cross-distribution):

```
ansible all -m package -a "name=htop state=present"
```

5.5 Gestione Servizi

Avviare un servizio:

```
ansible web -m service -a "name=nginx state=started"
```

Riavviare:

```
ansible web -m service -a "name=nginx state=restarted"
```

Abilitare all’avvio:

```
ansible web -m service -a "name=nginx enabled=yes"
```

5.6 Gestione Utenti

Creare un utente:

```
ansible all -m user -a "name=deploy shell=/bin/bash groups=sudo append=yes"
```

Rimuovere un utente:

```
ansible all -m user -a "name=deploy state=absent remove=yes"
```

5.7 Copiare File

Copiare file locale su remoto:

```
ansible web -m copy -a "src=/tmp/test.txt dest=/tmp/test.txt owner=root group=root mode=0644"
```

5.8 Uso di Privilege Escalation

Se necessario eseguire come root:

```
ansible all -m apt -a "name=vim state=present" --become
```

Se `become` è già abilitato in ansible.cfg, non serve specificarlo.

5.9 Limitare l’Esecuzione

Eseguire solo su un host specifico:

```
ansible web01 -m command -a "hostname"
```

Limitare tramite opzione `--limit`:

```
ansible all -m ping --limit web
```

5.10 Esecuzione Parallela e Fork

Ansible esegue operazioni in parallelo.

Numero default: 5 fork.

Modificabile:

```
ansible all -m ping -f 20
```

Oppure in ansible.cfg:

```
forks = 20
```

5.11 Output e Debug

Aumentare verbosità:

```
ansible all -m ping -vvv
```

Livelli disponibili:

- -v
- -vv
- -vvv
- -vvvv

Utile per troubleshooting SSH o problemi di autenticazione.

5.12 Quando Usare i Comandi Ad-Hoc

Utilizzare quando:

- Serve un controllo rapido
- È richiesta un’azione urgente
- Non è necessario mantenere traccia strutturata

Non utilizzare quando:

- L’operazione deve essere ripetibile
- Serve versionamento
- È una configurazione strutturale

In questi casi è corretto scrivere un playbook.

Conclusione

I comandi ad-hoc sono uno strumento operativo quotidiano per il sysadmin.

Permettono interventi rapidi e controllati, ma non sostituiscono l’automazione strutturata.

Nel prossimo capitolo entreremo nel cuore di Ansible: i Playbook.

Revision #1
Created 2026-02-26 13:57:18 UTC by Pe
Updated 2026-02-26 13:57:47 UTC by Pe