

Ansible – automatizace správy serverů

Eliška Jégrová

25. 11. 2025

Obsah

Úvod

Instalace

Inventář

Samostatná práce

Proč Ansible

- ▶ Dosud: ruční správa přes SSH nebo konzoli
- ▶ Pro jeden server to stačí, ale co budeme dělat v případě, kdy budeme mít serverů 1000?
- ▶ Nebo budeme potřebovat zreplicovat specifické nastavení na další server?
- ▶ Jak rychle dokážet nahradit nefunkční server?

Co je Ansible

- ▶ Ansible je nástroj pro **automatizaci**:
 - ▶ **konfigurační management** – udržování konzistentního nastavení systémů (balíčky, služby, konfiguráky, uživatelé)
 - ▶ **deploy aplikací** – nahrání nové verze, konfigurace, restart služeb,
 - ▶ **orchestrace** více serverů.
- ▶ **Bezagentový nástroj** (*agentless*):
 - ▶ na spravovaných serverech není potřeba žádný démon,
 - ▶ používá **SSH** a Python.
- ▶ **Typické použití**:
 - ▶ nastavení a správa webových serverů,
 - ▶ vytváření a správa uživatelů,
 - ▶ instalace balíčků,
 - ▶ úprava konfigurací a restart služeb.

Základní principy Ansible

- ▶ V Ansible konfiguraci popisujeme **deklarativně**:
 - ▶ **jaký má být výsledný stav**, ne jednotlivé kroky,
 - ▶ např. „balíček `httpd` je nainstalovaný“, „služba `httpd` běží a je povolená po startu“.
- ▶ **Idempotence**:
 - ▶ opakované spuštění vede ke stejnému stavu:
 - ▶ při prvním spuštění se něco změní,
 - ▶ při dalším spuštění už není co měnit.
 - ▶ Příklady idempotentního chování:
 - ▶ `package: state=present` nainstaluje balíček jen pokud chybí,
- ▶ Konfigurace je psaná v souborech ve formátu **YAML** (`.yaml`, `.yml`).

Instalace Ansible a první playbook

- ▶ Na řídicím serveru (naší virtuálce) nainstalujeme Ansible:

```
# dnf install ansible
```

- ▶ V pracovním adresáři vytvoříme soubor `setup.yaml`, např.:

```
- hosts: localhost
  connection: local
  become_user: root
  tasks:
    - name: Install htop
      become: yes
      dnf:
        state: latest
        name:
          - htop
```

Klíčová slova v setup.yaml

- ▶ `hosts: localhost`
 - ▶ na které hosty se playbook aplikuje (skupina `all` = všichni).
- ▶ `connection: local`
 - ▶ způsob jak se připojit
- ▶ `become_user: root`
 - ▶ jaký uživatel bude použit
- ▶ `tasks:`
 - ▶ seznam kroků, které se mají provést.
- ▶ `name: Install htop`
 - ▶ název konkrétní úlohy (tasku).
- ▶ `become: yes`
 - ▶ úlohy poběží s právy zadaného uživatele.

Spuštění playbooku setup.yaml

- ▶ Playbook spouštíme příkazem `ansible-playbook`.

```
$ ansible-playbook -K setup.yaml -i hosts.yaml
```

- ▶ `-i` – určuje soubor s inventářem (seznam hostů).
- ▶ `-K` – Ansible se zeptá na heslo pro sudo (become heslo), aby mohl spouštět úlohy s vyššími právy.
- ▶ Během běhu Ansible ukazuje pro každý task:
 - ▶ jestli proběhl v pořádku (ok),
 - ▶ jestli něco změnil (changed),
 - ▶ případné chyby (failed).

Inventář – seznam spravovaných serverů

- ▶ Inventář říká, **na které stroje** se mají playbooky používat.
- ▶ Každý řádek = jeden server (jméno nebo IP adresa).

hosts.yml - jednoduchý inventář v YAML

web:

hosts:

192.168.122.101:

192.168.122.102:

db:

hosts:

192.168.122.110:

- ▶ Jména web a db jsou **skupiny** hostů
- ▶ V setup.yml pak můžeme použít: hosts: web
a playbook se spustí na všech web serverech.

Samostatná práce – první kroky s Ansible

- ▶ Otevři soubor `setup.yaml`,
 1. Uprav task tak, aby instaloval víc „užitečných nástrojů“: `keepassxc` a `gnome-tweaks`.
 2. Spust' playbook:
 - ▶ `ansible-playbook -K setup.yaml`Sleduj, které balíčky mají stav `changed` a které jen `ok`.
 3. Do seznamu `tasks`: přidej úlohu, která:
 - ▶ v systému vytvoří novou uživatelku `zuzana`.
 - ▶ použij k tomu modul `user`
 4. Přidej další úlohu, která:
 - ▶ zajistí, že balíček `sl` **není** nainstalovaný
 5. Přidej úlohu, která:
 - ▶ zajistí, že služba `sshd` běží,
 - ▶ a je **povolena po startu systému** (`enabled`).
 - ▶ použij modul `systemd`