

# Základy Ansible | Workshop | Seznam.cz

Mnoho strojů, málo času a jeden automatizační nástroj

---



Petr Krch @v1dlak

Radim Lipovčan @RadimLipovcan

Infrastructure SCIF engineers



1. Příprava prostředí
2. Práce s inventory
3. Ping modul
4. Shell modul
5. Gather facts
6. Ansible-playbook



Příprava prostředí

---

- `cd ansible-workshop/part_1/`
  - `./prepare.sh`
    - Připraví Docker prostředí pro naše testování:
    - `ansible-playbook -i inventory/workshop.ini prepare.yml`
    - Pokud se v playbooku vyskytly chyby, tak nám to ihned nahlašte.
    - V ideálním případě máte několik kontejnerů se systemd v současné době ready (využijeme image od **ownercz/linuxdays-ubuntu-systemd**)



Práce s inventory

---

- V současné době pouze jeden záznam:

```
[prg]  
adam ansible_connection=docker
```

- Standartní inventory může vypadat takto:

```
[workshop:children]  
testing  
staging  
[workshop:vars]  
ansible_user=root  
[staging]  
proxmox ansible_host=192.168.1.100 ansible_user=root  
[testing]  
test.seznam.cz
```

- Doplňte další 4 stroje: `[mona, eva, merlin, karel]`
  - inventory bude obsahovat skupiny: `[prg, brn]`
    - `prg: [adam, mona]`
    - `brn: [eva, merlin, karel]`
- Přidejte proměnnou `ansible_connection=docker` ke všem
- Budeme editovat soubor:  
`ansible-workshop/part-1/inventory/workshop.ini`
- Dokumentace



## Zadání:

- Doplňte další 4 stroje: `mona`, `eva`, `merlin`, `karel`
- Přidejte proměnnou `ansible_connection=docker` ke všem
- Inventory bude obsahovat skupiny: `prg`, `brn`
- `prg`: `adam`, `mona`
- `brn`: `eva`, `merlin`, `karel`

## Řešení:

```
[all:vars]
ansible_connection=docker

[prg]
adam
mona

[brn]
eva
merlin
karel
```







Ping modul

---

- Zkuste si následující příkaz:
  - `ansible prg -i inventory/workshop.ini -m ansible.builtin.ping`
- Problém?

```
adam | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
mona | UNREACHABLE! => {
  "changed": false,
  "msg": "Host unreachable"
}
```

- Měnili jsme inventory, ale nepromítli změny do našeho dockeru
- `./prepare.sh`
- `ansible -i inventory/workshop.ini prg -m ansible.builtin.ping`
- Bonus: Zkuste ping olimitovat podle jmen instancí/skupin `-l`

- Zkuste si následující příkaz:
  - `ansible prg -i inventory/workshop.ini -m ansible.builtin.ping`
- Problém?

```
adam | SUCCESS => {  
  "changed": false,  
  "ping": "pong"  
}  
mona | UNREACHABLE! => {  
  "changed": false,  
  "msg": "Host unreachable"  
}
```

- Měnili jsme inventory, ale nepromítli změny do našeho dockeru
- `./prepare.sh`
- `ansible -i inventory/workshop.ini prg -m ansible.builtin.ping`
- Bonus: Zkuste ping olimitovat podle jmen instancí/skupin `-l`

- Zkuste si následující příkaz:
  - `ansible prg -i inventory/workshop.ini -m ansible.builtin.ping`
- Problém?

```
adam | SUCCESS => {  
  "changed": false,  
  "ping": "pong"  
}  
mona | UNREACHABLE! => {  
  "changed": false,  
  "msg": "Host unreachable"  
}
```

- Měnili jsme inventory, ale nepromítli změny do našeho dockeru
- `./prepare.sh`
- `ansible -i inventory/workshop.ini prg -m ansible.builtin.ping`
- Bonus: Zkuste ping olimitovat podle jmen instancí/skupin `-l`



Shell modul

---

- Přímé spuštění příkazu na instancích
- Zkuste si následující:
  - Použijeme modul `ansible.builtin.shell` pro `adam` target z naší inventory.
  - `ansible adam -i inventory/workshop.ini -m ansible.builtin.shell -a 'cat /etc/issue'`

- Požadovaný výstup:

```
adam | CHANGED | rc=0 >>  
Ubuntu 24.04.3 LTS
```

- Task:
  - Zjistěte výstup příkazu `uptime` ve všech kontejnerech pomocí Ansible.
  - Jaké hostname mají jednotlivé konejnery?



- Přímé spuštění příkazu na instancích
- Zkuste si následující:
  - Použijeme modul `ansible.builtin.shell` pro `adam` target z naší inventory.
  - `ansible adam -i inventory/workshop.ini -m ansible.builtin.shell -a 'cat /etc/issue'`
- Požadovaný výstup:

```
adam | CHANGED | rc=0 >>  
Ubuntu 24.04.3 LTS
```

- Task:
  - Zjistěte výstup příkazu `uptime` ve všech kontejnerech pomocí Ansible.
  - Jaké hostname mají jednotlivé konejnery?



- Přímé spuštění příkazu na instancích
- Zkuste si následující:
  - Použijeme modul `ansible.builtin.shell` pro `adam` target z naší inventory.
  - `ansible adam -i inventory/workshop.ini -m ansible.builtin.shell -a 'cat /etc/issue'`

- Požadovaný výstup:

```
adam | CHANGED | rc=0 >>  
Ubuntu 24.04.3 LTS
```

- Task:
  - Zjistěte výstup příkazu `uptime` ve všech kontejnerech pomocí Ansible.
  - Jaké hostname mají jednotlivé konejner?





**Zadání:**

- Zjistěte výstup příkazu uptime ve všech kontejnerech pomocí Ansible.
- Jaké hostname mají jednotlivé kontejnery?

**Řešení:**

```
ansible all -i inventory/workshop.ini -m ansible.builtin.shell -a 'uptime'  
ansible all -i inventory/workshop.ini -m ansible.builtin.shell -a 'hostname'
```





Gather facts

---

- Ansible umí zjistit o vašem hostu spoustu informací [[dokumentace](#)]
- K tomu slouží modul `ansible.builtin.setup`

- Spustíte si příkaz:

- `ansible adam -i inventory/workshop.ini -m ansible.builtin.setup | less`

- Nebo pokud chcete konkrétní informaci, tak například:

- `ansible adam -i inventory/workshop.ini -m ansible.builtin.setup -a 'filter=ansible_distribution'`

- Ansible umí zjistit o vašem hostu spoustu informací [[dokumentace](#)]
- K tomu slouží modul `ansible.builtin.setup`
- Spusťte si příkaz:
  - `ansible adam -i inventory/workshop.ini -m ansible.builtin.setup | less`
- Nebo pokud chcete konkrétní informaci, tak například:
  - `ansible adam -i inventory/workshop.ini -m ansible.builtin.setup -a 'filter=ansible_distribution'`

- Ansible umí zjistit o vašem hostu spoustu informací [[dokumentace](#)]
- K tomu slouží modul `ansible.builtin.setup`
- Spustíte si příkaz:
  - `ansible adam -i inventory/workshop.ini -m ansible.builtin.setup | less`
- Nebo pokud chcete konkrétní informaci, tak například:
  - `ansible adam -i inventory/workshop.ini -m ansible.builtin.setup -a 'filter=ansible_distribution'`



Ansible-playbook

---

- Working dir: `ansible-workshop/part-2`
- `./prepare.sh`
- Přichystali jsme pro vás ukázkový Ansible playbook `linuxdays2025.yml`
- V playbooku je task na nakopírování souboru `LD2025` na všechny instance do `/opt/`
- Task ale obsahuje chyby, pojďme je společně opravit.
- Spustěte
  - `ansible-playbook -i inventory/workshop.ini linuxdays2025.yml -D -v`

- Bonusové úkoly:

- Zkuste změnit soubor v `files/` a znovu spouštět playbook.
- Zkuste na některé instanci soubor smazat a znovu spustit playbook.
  - `docker exec -it adam rm /opt/LD2025`



- Working dir: `ansible-workshop/part-2`
- `./prepare.sh`
- Přichystali jsme pro vás ukázkový Ansible playbook `linuxdays2025.yml`
- V playbooku je task na nakopírování souboru `LD2025` na všechny instance do `/opt/`
- Task ale obsahuje chyby, pojďme je společně opravit.
- Spustěte
  - `ansible-playbook -i inventory/workshop.ini linuxdays2025.yml -D -v`
- Bonusové úkoly:
  - Zkuste změnit soubor v `files/` a znovu spouštět playbook.
  - Zkuste na některé instanci soubor smazat a znovu spustit playbook.
    - `docker exec -it adam rm /opt/LD2025`





## Řešení:

```
- hosts: all
  handlers:
    - name: Echo change to log
      ansible.builtin.debug:
        msg: "linuxdays2025 file has been changed"
      listen: linuxdays2025-handler
  tasks:
    # Task obsahuje tři chyby, které je potřeba opravit.
    - name: Copy LD2025 file to /opt directory
      ansible.builtin.copy:
        src: LD2025
        dest: /opt/LD2025
        owner: root
        group: root
        mode: '0644'
      notify: linuxdays2025-handler
      when: inventory_hostname != "eva"
```



- Working dir: `ansible-workshop/part-3`
- `./prepare.sh`
- Vytvořte si vlastní playbook `all.yml` [dokumentace]
- Playbook provede na všech instancích instalaci libovolného editoru (vi, nano, vim, emacs, ...)
- Použijte modul `ansible.builtin apt` [dokumentace]
- Spusťte `ansible-playbook -i inventory/workshop.ini all.yml -D -v`



## Zadání:

- Working dir:  
`ansible-workshop/part-3`
- `./prepare.sh`
- Vytvořte si vlastní playbook `all.yml`
- Playbook provede na všech instancích instalaci libovolného editoru (vi, nano, vim, emacs, ...) pomocí modulu `apt`
- Spusťte `ansible-playbook -i inventory/workshop.ini all.yml -D -v`

## Řešení:

```
- name: Install editor
hosts: all
tasks:
  - name: Install vim
    ansible.builtin.apt:
      name: vim
      state: present
      update_cache: true
```



- Jak bychom mohli řešit situaci:
  - chceme instalovat různé editory na různé instance
    - chceme instalovat vim na dvě instance z různých skupin (adam, eva)
    - chceme instalovat
      - vim na karel a mona
      - nano na karel a eva
  - ukažme si možnosti:
    - použití groups
    - použití when v tasku
    - použití více playů v jednom playbooku
    - použití host\_vars a/nebo group\_vars
    - použití with\_items



- Jak bychom mohli řešit situaci:
  - chceme instalovat různé editory na různé instance
  - chceme instalovat `vim` na dvě instance z různých skupin (`adam`, `eva`)
  - chceme instalovat
    - `vim` na karel a mona
    - `nano` na karel a eva
  - ukažme si možnosti:
    - použití `groups`
    - použití `when` v tasku
    - použití více playů v jednom playbooku
    - použití `host_vars` a/nebo `group_vars`
    - použití `with_items`



- Jak bychom mohli řešit situaci:
  - chceme instalovat různé editory na různé instance
  - chceme instalovat `vim` na dvě instance z různých skupin (`adam`, `eva`)
  - chceme instalovat
    - `vim` na `karel` a `mona`
    - `nano` na `karel` a `eva`
- ukažme si možnosti:
  - použití `groups`
  - použití `when` v tasku
  - použití více `playů` v jednom `playbooku`
  - použití `host_vars` a/nebo `group_vars`
  - použití `with_items`



- Jak bychom mohli řešit situaci:
  - chceme instalovat různé editory na různé instance
  - chceme instalovat `vim` na dvě instance z různých skupin (`adam`, `eva`)
  - chceme instalovat
    - `vim` na `karel` a `mona`
    - `nano` na `karel` a `eva`
  - ukažme si možnosti:
    - použití `groups`
    - použití `when` v tasku
    - použití více `playů` v jednom playbooku
    - použití `host_vars` a/nebo `group_vars`
    - použití `with_items`



- Working dir: `ansible-workshop/part-4`
- `./prepare.sh`
- V cestě `roles/mysql/` najdete základní strukturu pro Ansible roli
  1. Do `roles/mysql/tasks/main.yml` přidejte task na instalaci MySQL serveru (balík `mariadb-server`)
  2. Přidejte task na nakopírování souboru `my.cnf` z `roles/mysql/files/` do `/etc/mysql/my.cnf`
  3. Nastartujte MySQL službu (`mariadb`)
  4. Přidejte do `roles/mysql/handlers/main.yml` handler na restart MySQL služby
  5. Doplněte volání handleru při změně `my.cnf`





- Working dir: `ansible-workshop/part-4`
- `./prepare.sh`
- V cestě `roles/mysql/` najdete základní strukturu pro Ansible roli
  1. Do `roles/mysql/tasks/main.yml` přidejte task na instalaci MySQL serveru (balík `mariadb-server`)
  2. Přidejte task na nakopírování souboru `my.cnf` z `roles/mysql/files/` do `/etc/mysql/my.cnf`
  3. Nastartujte MySQL službu (`mariadb`)
  4. Přidejte do `roles/mysql/handlers/main.yml` handler na restart MySQL služby
  5. Doplněte volání handleru při změně `my.cnf`



- Working dir: `ansible-workshop/part-4`
- `./prepare.sh`
- V cestě `roles/mysql/` najdete základní strukturu pro Ansible roli
  1. Do `roles/mysql/tasks/main.yml` přidejte task na instalaci MySQL serveru (balík `mariadb-server`)
  2. Přidejte task na nakopírování souboru `my.cnf` z `roles/mysql/files/` do `/etc/mysql/my.cnf`
  3. Nastartujte MySQL službu (`mariadb`)
  4. Přidejte do `roles/mysql/handlers/main.yml` handler na restart MySQL služby
  5. Doplněte volání handleru při změně `my.cnf`



- Working dir: `ansible-workshop/part-4`
- `./prepare.sh`
- V cestě `roles/mysql/` najdete základní strukturu pro Ansible roli
  1. Do `roles/mysql/tasks/main.yml` přidejte task na instalaci MySQL serveru (balík `mariadb-server`)
  2. Přidejte task na nakopírování souboru `my.cnf` z `roles/mysql/files/` do `/etc/mysql/my.cnf`
  3. Nastartujte MySQL službu (`mariadb`)
  4. Přidejte do `roles/mysql/handlers/main.yml` handler na restart MySQL služby
  5. Doplňte volání handleru při změně `my.cnf`



- Working dir: `ansible-workshop/part-4`
- `./prepare.sh`
- V cestě `roles/mysql/` najdete základní strukturu pro Ansible roli
  1. Do `roles/mysql/tasks/main.yml` přidejte task na instalaci MySQL serveru (balík `mariadb-server`)
  2. Přidejte task na nakopírování souboru `my.cnf` z `roles/mysql/files/` do `/etc/mysql/my.cnf`
  3. Nastartujte MySQL službu (`mariadb`)
  4. Přidejte do `roles/mysql/handlers/main.yml` handler na restart MySQL služby
  5. Doplněte volání handleru při změně `my.cnf`



- V cestě `roles/mysql/` najdete základní strukturu pro Ansible roli
  1. Do `roles/mysql/tasks/main.yml` přidejte task na instalaci MySQL serveru (balík `mariadb-server`)
  2. Přidejte task na nakopírování souboru `my.cnf` z `roles/mysql/files/` do `/etc/mysql/my.cnf`
  3. Nastartujte MySQL službu (`mariadb`)
  4. Přidejte do `roles/mysql/handlers/main.yml` handler na restart MySQL služby
  5. Doplněte volání handleru při změně `my.cnf`
- Bonusové úkoly:
  - Nastavte hodnotu `max_connections` v `roles/mysql/defaults/main.yml`
  - Změňte kopírování `my.cnf` na šablonu `my.cnf.j2` z `roles/mysql/templates/`
  - Nahraďte hodnotu `max_connections` v šabloně za proměnnou z `defaults`
  - Změňte hodnotu na jednom z hostů
  - Vyzkoušejte si práci s některým z modulů `community.mysql` [[dokumentace](#)]
  - Vytvořte uživatele a databázi
  - Vytvořte vault pro uložení hesla a následně použijte tento vault v template



Díky za pozornost a užijte si LinuxDays!

