DevOps and Security Lab 2 - Software configuration management tools (default lab)

> Individual lab

The goal is to easily, reliably and quickly maintain different kinds of systems at once.

Task 1 - Prerequisites

Prepare at least two different guest systems (VMs), e.g. an Ubuntu Server (aka "prod env") and a Fedora (aka "dev env"). Then choose a Software configuration management (SCM) tool to automate system preparations on those guests instances. **Ansible** is the default choice and the most popular tool for SCM. We recommend you to use **Ansible**. By the way, there are some alternatives that you are able to choose:

- SaltStack
- Puppet
- Chef
- Another one...

Task 2 - SCM Theory

Briefly answer for the following questions what is and for what:

1. **ansible** directory:

- `ansible.cfg`
- `inventory` folder
- `roles` folder
  - `tasks`
  - `defaults/group_vars`
  - `handlers`
  - `templates`
  - `vars`
- `meta` folder
- `playbooks` folder

2. Research, list and explain the most important parameter from `ansible.cfg` in your opinion.
3. Learn and explain Ansible variable precedence.

`If you use an other SCM tool, replace the task goals according to your tool specifications.`

Task 3 - Play with SCM

1. Play with SCM tool to automate system preparations on those. You might create and provide your own ideas or follow to some suggestions:

Light:

- Ansible Vault (definitely good choice)
- IPtables (close all machine ports exclude ssh, http, https)
- Docker containers deployment
- set the default shell
- Linux users and groups management (to create, to change permissions...)
- install some applications on the target hosts (apt, rpm, pip...)
- ...

A kinda advanced:

- NTP
- Apache Web Server
- Kubernetes helm chart deployment
- Nginx & PHP-FPM & MariaDB & WordPress installation and configuration
- development environments organization (ssh keys management...)
- certificates creation (let's encrypt)
- Golang application with microservices
- Resize volume partitions on the target hosts
- Make a network configuration between several virtual hosts (vlans, IPs, routing, firewall ...)
- ...

Requirements:

- never forget to follow to the best practices
- use *templates/jinja2*
- separate your configuration files into **inventory/roles/playbooks** files
- do not expose sensitive date and secrets in plain text!
- play with *handlers* and *meta* (optional)
- try to avoid a raw shell command in Ansible tasks as much as you can
- follow to **Idempotence** principle!
- play with Ansible variables priorities (precedence) (not mandantory)
- do not hard code variables values in `tasks`
- **DO NOT COPY/PAST EXISTING ANSIBLE PLAYBOOKS!** If you use any code as a base and example, just mention it and list what you've changed and how you understand it
- **ideal: complete two "light" and one "advanced" SCM usage scenarios**

Bonus: use Ansible Molecula and Ansible Lint to test your role before to run the corresponding playbook.

`If you use an other SCM tool, replace the task goals according to your tool specifications.`

2. Provide the link to git repository with all your labs configurations.

Bonus 1 - Ansible collections

Learn and play on practice with Ansible Galaxy.

Bonus 2 - Ansible AWX

Deploy Ansible AWX and demonstrate a PoC.

Links

Comparison of open-source configuration management software