Principes des Systèmes d'exploitation

Groups, subgroups and variables

Guillerme Duvillié

1. Variables

2. Groups

Plan

1. Variables
2. Groups

First

Create a `server` playbook that:

- Update and upgrade all the hosts
- Install and launch `apache` and `php` on the Fedora and Debian 10 hosts,
- Install and launch `mariadb` on the Arch host,
- Install and launch `samba` on the Arch and Debian 11 hosts.

Use the `service` module:

Documentation

Host variables

- Variables can be defined for hosts in inventory files

```
[apt]
debian10 \
    first_var=value1 \
    second_var=value2
www.myserver.com
    first_var=value3 \
    second_var=value4
```

- Values can be accessed in playbooks with syntax: `{{ first_var }}`
- If a string is required then, variable access must be quoted: `"{{ first_var }}"`

Cleaning host file

- Host aliases may be defined using `ansible_host` behavioural variable:

```
1  deb_server1 \
2      ansible_host=debian10 \
3      first_var=value1
4  deb_server2 \
5      ansible_host=www.myserver.com \
6      first_var=value3
7
8  [apt]
9  deb_server1
10 deb_server2
```

- Host variables for host `example` may be defined in file `host_vars/example.yml`
- Variables can be declared for groups in `group_vars/group_name.yml`

Exercise

- Using `package` generic module and variables, clean up inventory file and the `server` play book.

Plan

1. Variables
2. Groups

Subgroups

- Subgroups can be defined in inventory files:

```
server[1:4]

[bruxelles]
server3
server4

[namur]
server1

[marche]
server2

[webservers:children]
namur
marche

[webservers:vars]
vpn_name=web.mydomain.com
```

Subgroups (2)

- Variables are not interpreted the same way regarding to their definition location:
  - host variables are pythonishly interpreted,
  - section declared variables are only strings.

Variable merge

- Defining variables at several point may lead to conflicts.
- Merge order (from lowest priority to highest):
  1. All hosts variables
  2. Parent group variables
  3. Child group variables
  4. Host variables

Inventory file: from INI…

```
green.ex.com ansible_host=10.0.101.100
192.168.100.1

[webservers]
beta.ex.org ansible_host=192.168.200.122
www[001:006].ex.com

[webservers:vars]
nginx_http_port=80
nginx_https_port=443

[dbservers]
db01.intranet.mydomain.net
db-[99:101]-node.ex.com
```

. . . To Yaml

```yaml
---
ungrouped:
  hosts:
    green.ex.com:
      ansible_host: 10.0.101.100
    192.168.100.1:

webservers:
  hosts:
    beta.ex.org:
      ansible_host: 192.168.200.122
    www[001:006].ex.com:
  vars:
    nginx_http_port: 80
    nginx_https_port: 443

dbservers:
  hosts:
    db01.intranet.mydomain.net:
    db-[99:101]-node.ex.com:
```

Your turn

Exercise

Convert your INI host file to a yaml one.

Hints

Use the `--list` option from **`ansible-playbook`** to compare results.