

ICA0002: IT Infrastructure Services

Ansible Basics

Roman Kuchin
Juri Hudolejev
2024

Files from previous lab

```
roles/  
  init/  
    tasks/  
      main.yaml  
ansible.cfg  
hosts  
infra.yaml
```

Files from previous lab

roles/

Directory where all our infrastructure service roles are stored

Role defines a service type: web server, database server, mail server etc.

Each role is defined in its own subdirectory and should contain a file **tasks/main.yaml**

roles/init/

Custom role we have created to test Ansible setup

Files from previous lab

`roles/init/tasks/main.yaml`

YAML file, list of tasks for `init` role

For every server that has this role assigned,
every task from this role is executed

Task

Action to ensure that certain element (file, service etc.) is in desired state

On task execution Ansible will change the element state -- if needed

Ansible task example

```
name: Ansible ping module  
ansible.builtin.ping:
```

Where

name is free text task identifier for better readability

ansible.builtin.ping is Ansible module call

Note the colons! '**ping:**' means 'module **ping**, no parameters'.

Another Ansible task example

Ensures that Linux user with login 'elvis' exists; creates one if not:

```
name: Linux user elvis
```

```
ansible.builtin.user: ← Ansible module to manage OS users
```

```
  name: elvis ← Module parameter named 'name'  
           with value 'elvis'; note the indent!
```

All Ansible modules:

https://docs.ansible.com/ansible/latest/collections/index_module.html

Yet another Ansible task example

Ensures both APT packages `curl` and `wget` are installed; installs if not:

```
name: HTTP client packages
```

```
ansible.builtin.apt:
```

```
  name:
```

- `curl`
- `wget`

And another Ansible task example

Ensures local file is copied to a managed host; copies if not:

```
name: Really important file (no)
ansible.builtin.copy:
  src: foo.txt
  dest: /srv/foo.txt
```

Note: file `files/foo.txt` must exist in the role directory for this to work:

```
roles/my_role/files/foo.txt
roles/my_role/tasks/main.yml
^^^^^  ^^^^^^^^
```


And another Ansible task example

Ensures local file is copied to a managed host; copies if not:

```
name: Really important file (no)
ansible.builtin.copy:
  src: foo.txt
  dest: /srv/foo.txt
```

Always use full paths for remote files (**dest**: /...)!

Never use full paths for local files (**src**: ...)!

Files from previous lab

`infra.yaml`

Playbook, YAML file, list of plays

Ansible execution entry point

Play

Maps a group of roles to a group of servers (hosts)

Ansible plays and playbooks:

https://docs.ansible.com/ansible/latest/user_guide/playbooks_intro.html

Ansible play example

Assigns role `init` to all hosts:

```
name: Initial setup
hosts: all
roles:
  - init
```

Another Ansible play example

Assigns multiple roles to every defined host named 'web*' -- **web-1**, **web-2** etc.:

```
name: Public web servers
hosts: web*
roles:
  - firewall
  - nginx
```

Note that **web*** pattern will also match host groups like **web_servers**.

Ansible playbook example

- name: Database servers
hosts: db_servers
roles:
 - mysql
- name: Web servers
hosts: web_servers
roles:
 - nginx

Playbook is just a list of several plays -- may contain 0, 1 or more plays

YAML

Human readable data serialization language with minimal syntax

```
key: value           # map keys are followed by ':' and optional value
foo:                 # maps can include other maps
  bar: 0.42          # numbers are written as is
list:                # comments start with '#'
  - one element      # list elements start with '-'
  - another element  # indents are important!
```

More info: <https://yaml.org>

Not all Ansible files are YAML files!

Files from previous lab

hosts

Inventory file, high-level overview of your entire infrastructure:

- List of servers
- Server groups
- Sometimes server variables: connection parameters etc.

Format -- flavor of INI (other formats also supported):

```
[my_group]
```

```
server-name  param=value  another_param=another_value
```


Ansible inventory file example

```
[db_servers]
```

```
db-1    ansible_host=10.10.10.37
```

```
[web_servers]
```

```
web-1   ansible_host=10.10.10.13  ansible_port=9522
```

```
web-2   ansible_host=10.10.10.14  ansible_port=9622
```

```
[all:vars]
```

```
ansible_user=ansible
```

Ansible inventory file location

In this course: file named **hosts** in your repository

Default location: **/etc/ansible/hosts** ← do not use!

More about Ansible inventory:

https://docs.ansible.com/ansible/latest/user_guide/intro_inventory.html

Files from previous lab

`ansible.cfg`

Ansible configuration file -- changes some aspects of Ansible behavior

So far (lab 1) our Ansible configuration file only instructs Ansible to fetch the inventory from the same directory:

```
[defaults]  
inventory = ./hosts
```

We will improve Ansible configuration file later during this course

Ansible configuration file format

Flavor of INI format:

```
[section_name]
key_name = value
another_key_name = another_value

[another_section_name]
...
```

More about Ansible configuration:

https://docs.ansible.com/ansible/latest/reference_appendices/config.html

Ansible configuration file locations

ansible.cfg (root of your repository)

Project specific configuration

~/.ansible.cfg (your home directory)

User specific configuration for all projects on this machine

/etc/ansible/ansible.cfg

Global configuration for all users on this machine

Ansible configuration file locations

ansible.cfg

← go for it

Project specific configuration

~/.ansible.cfg

← do not use!

User specific configuration for all projects on this machine

/etc/ansible/ansible.cfg

← do not use!

Global configuration for all users on this machine

Summary

roles/	
init/	
tasks/main.yaml	← will add more tasks
{+++}/	← will add more roles
tasks/main.yaml	
{+++}/{+++}	← will add more resources to roles (files etc.)
ansible.cfg	← will add a few lines
hosts	← will change daily
infra.yaml	← will add more plays

Questions?