

Roles

Roles and their configuration

Role directories

A role is a set of instruction which describe how to install or update a fonctionnality. Roles are under the **roles** directory (no joke ;-)). Each role have several sub directories:

- **defaults** : contains default values for the role. These are the available variables you'll be able to set in your `host_vars/` or `group_vars/` to configure the service
- **tasks** : contains the action to run to configure the role
- **templates**: contains `jinja2` templates which will be deployed on the host
- **files**: contains files which will be deployed (as-is, no template processing)
- **vars**: contains variables used by the role. Usualy, variables which you should change are defined in defaults. In vars are defined variables used by the role which you shouldn't have to change
- **handlers** : containers handlers (eg, how to restart services when a configuration file changed)

defaults & variables

defaults is really the most important part of a role. Check the file `defaults/main.yml` of a role to see which variables you can tune. For example, for the role `docker` (which can install docker daemon on a host)

```
docker_data_dir: /opt/docker
docker_log_driver: journald

docker_base_conf:
  data-root: /opt/docker
  log-driver: journald
  storage-driver: overlay2
  storage-opts:
    - 'overlay2.override_kernel_check=true'
docker_extra_conf: {}
```

```
# docker_extra_conf:
#   log-opts:
#     max-size: 100m
#     max-file: 5

docker_conf: "{{ docker_base_conf | combine(docker_extra_conf, recursive=True) }}"
```

This is all the variable you can set to modify how Docker will be configured. You do not have to configure everything, just set the variables for which the default value doesn't fit your need.

hosts variables

For example, if you deploy docker on the host `docker.fws.fr`, just create **host_vars/docker.fws.fr/vars.yml**

```
docker_extra_conf:
  data-root: '/data'
  log-driver: 'json-file'
  log-opts:
    max-size: '100m'
    max-file: '5'
  iptables: False
  group: dockeradmins
  users-remap: default
  live-restore: True
  dns:
    - 10.118.1.1
```

groups variables

For some settings, you'll want to share them with a group of hosts (eg, the AD domain to join, or the Docker settings above, if you deploy several Docker hosts). In this case, you can create a group of host in your inventory file, for example :

```
[fws]
proxyin.fws.fr
docker1.fws.fr
docker2.fws.fr
```

```
[ fws_docker: vars ]
ansible_group_priority=2

[ fws_docker ]
docker1.fws.fr
docker2.fws.fr
```

Please, read ansible documentation if you need more detailed information on this
Now, you can create the files

- **group_vars/fws/vars.yml** : all the variables defined here will be inherited by **proxyin.fws.fr**, **docker1.fws.fr** and **docker2.fws.fr**
- **group_vars/fws_docker/vars.yml** : all the variables defined here will be inherited by **docker1.fws.fr** and **docker2.fws.fr**

With the above **ansible_group_priority**, if a variable is defined in both **fws** and **fws_docker**, the one from **fws_docker** will be used for **docker1.fws.fr** and **docker2.fws.fr**.

encrypted variables

You might need to set secret values in variables, like passwords. In this case, you do not want to store them as cleartext. Then, just use the

https://docs.ansible.com/ansible/latest/user_guide/vault.html **ansible-vault** utility.

```
ansible-vault create group_vars/fws/vault.yml
```

You'll be prompted for a password to encrypt the file. The syntaxe is the same as a normal file. If you want to edit an existing vault, use instead :

```
ansible-vault edit group_vars/fws/vault.yml
```

When you run the ansible playbook, if a host requires access to variables in a vault, you'll be prompted to enter the vault password

Révision #1

Créé 17 mars 2021 18:59:20 par Daniel Berteaud

Mis à jour 17 mars 2021 23:35:36 par Daniel Berteaud