Ansible Environment

To use Ansible, generally you create a “playbooks” directory in your /home/user/, then get the following filesystem inside :

```
playbooks:-> <role_name>.yml  (Playbooks to run, often matches the name of a
single role you want to run, or
         |                     another when you run several roles)
         |-> inventory.ini    (List of all servers to run playbooks on, with
host specific variables)
         |-> isc_compute.ini  (Chacha Disco and Tango for example)
         |-> labs.ini         (Calypso lab nodes)
         |-> test.ini         (Custom list for testing)
         |-> users.yml        (The main source for users IDs and assignation
of groups, since we don't have access
         |                     to the HES LDAP to manage groups ourselves
there. Contains UID/GID, shell and pass
         |                     only for the default/rescue user)
         |-> ansible.cfg      (Ansible local configuration for your user :
python interpreter version for example)
         |
         L-> roles:-> <rolename>:
                               |
                               L-> files:       (Contains all files needed
for your playbook, ansible checks this
                               |        |        directory first when you
reference a file)
                               |        L-> authorized_keys/remi.key
                               |        L-> remi_home.tar.gz
                               |
                               L-> handlers:    (Small repetitive tasks to
be called in a playbook, like restart a
                               |           |     systemd service)
                               |           L->
restart_mysql_then_apache.yml
                               |
                               L-> tasks:       (Main dir for all tasks to
run)
                               |        L-> main.yml    (The first task
called when the role is run)
                               |        L-> isc3.yml    (Supplementary
tasks you can call in your main task)
                               |
                               L-> vars:
                                       L-> Specific variable file for the
role
```

Ansible is made to run all playbooks using this “playbooks” dir as your working directory. From there all relative paths works as intended.

You can use some pre-defined playbooks using the Galaxy store :

```
# List installed roles :
ansible-galaxy collection list

# Installs a role from prometheus space called prometheus too :
ansible-galaxy collection install prometheus.prometheus
```

You can run playbooks simply like this :

```
ansible-playbook -i labs.ini calypso-sys.yml
```

Or with more options :

```
# Run only on calypso0, import the users.yml variable file to have all users UID/GID, start the playbook at the step that runs the ISC2 task
ansible-playbook -i labs.ini calypso-sys.yml --limit calypso0 --extra-vars='@users.yml' --start-at-task="Check or Add ISC2 users"
```

Ansible playbooks

For now that we don't really have a server for this purpose, I'm using my laptop to deploy configurations using Ansible. (Remi)

```
playbooks:-> <role_name>.yml  (Playbooks to run)
          |-> inventory.ini    (List of all servers to run playbooks on)
          |-> isc_compute.ini  (Chacha and Disco for example)
          |-> labs.ini         (Calypso nodes)
          |-> test.ini         (Custom list)
          |-> users.yml        (The main source for users IDs and assignation of groups, since we don't have access
          |                     to the HES LDAP to manage groups ourselves there. Contains UID/GID, shell and pass
          |                     only for the default/rescue user)
          |
          L-> roles:-> calypso-sys:    (Configures all calypso worker nodes, with all
                   |                    users, software, and system configs)
                   |-> isc_compute:    (Configures Disco and Chacha users, software and system configs)
                   |-> julia:          (Installs Julia for a user, since there are no system-wide install)
                   |-> k8s:            (Configures Kubernetes on all Calypso nodes)
                   |-> learn           (TODO: Started a playbook to configure from scratch a webserver with all
                   |                    Moodle prerequisites, to rebuild
```

```
Hannibal in minutes)
                  |-> master-sys:     (Configures system configs specific on
calypsomaster only)
                  |-> munge:          (Needed to authenticate SLURM nodes on
Calypso and ISC Compute)
                  |-> nvidia-cuda:    (Needed to allow servers to use Nvidia
GPUs)
                  |-> prometheus:     (Empty, TODO : Installs prometheus
exporters and server, separate in 2
                  |                    playbooks)
                  |-> rumba-sys:      (Configures rumba master, with all
                  |                    users, software, and system configs)
                  |-> slurm_research_TODO: (Empty, TODO : redo all
installation part from compilation
                  |                    install, separate install from
config)
                  |-> slurm_calypso:  (Empty, TODO : redo all installation
part from compilation
                  |                    install, separate install from
config)
                  |-> slurm_calypso_old : (Old installation of SLURM using
ubuntu outdated packages)
                  L-> vps:             (Installs or checks base system
config / users for Hannibal/Hasdrubal)
```

Ansible roles

Calypso-sys

Manages all Calypso lab servers configurations.

```
calypso-sys/-> tasks/-> main.yml    (All tasks to configure timezone, system
umask, groups and users, default user,
           |        |               base packages, ssh keys deployment, NFS
homes configs, symlinks in each home,
           |        |               fastfetch install, systemd tweaks to
fasten boot time, set /etc/hosts entries,
           |        |               put sudoers files, install Apptainer,
then run isc[1,2,3].yml tasks )
           |        |-> root.yml    (Specific root user configuration)
           |        |-> iscx.yml    (Configures students accounts for
isc[1,2,3] class)
           |
           |-> files/-> authorized_keys/-> user.name  (SSH public keys to
deploy for each user)
                     |
                     |-> admin_user_home_config/      (Containing configs
like .zshrc, .config/ and .oh-my-zsh/ )
```

```
                    |
                    |-> sudoers.d/username            (Containing sudoers
specific files for a user, without
                                                          dot in the name)
```