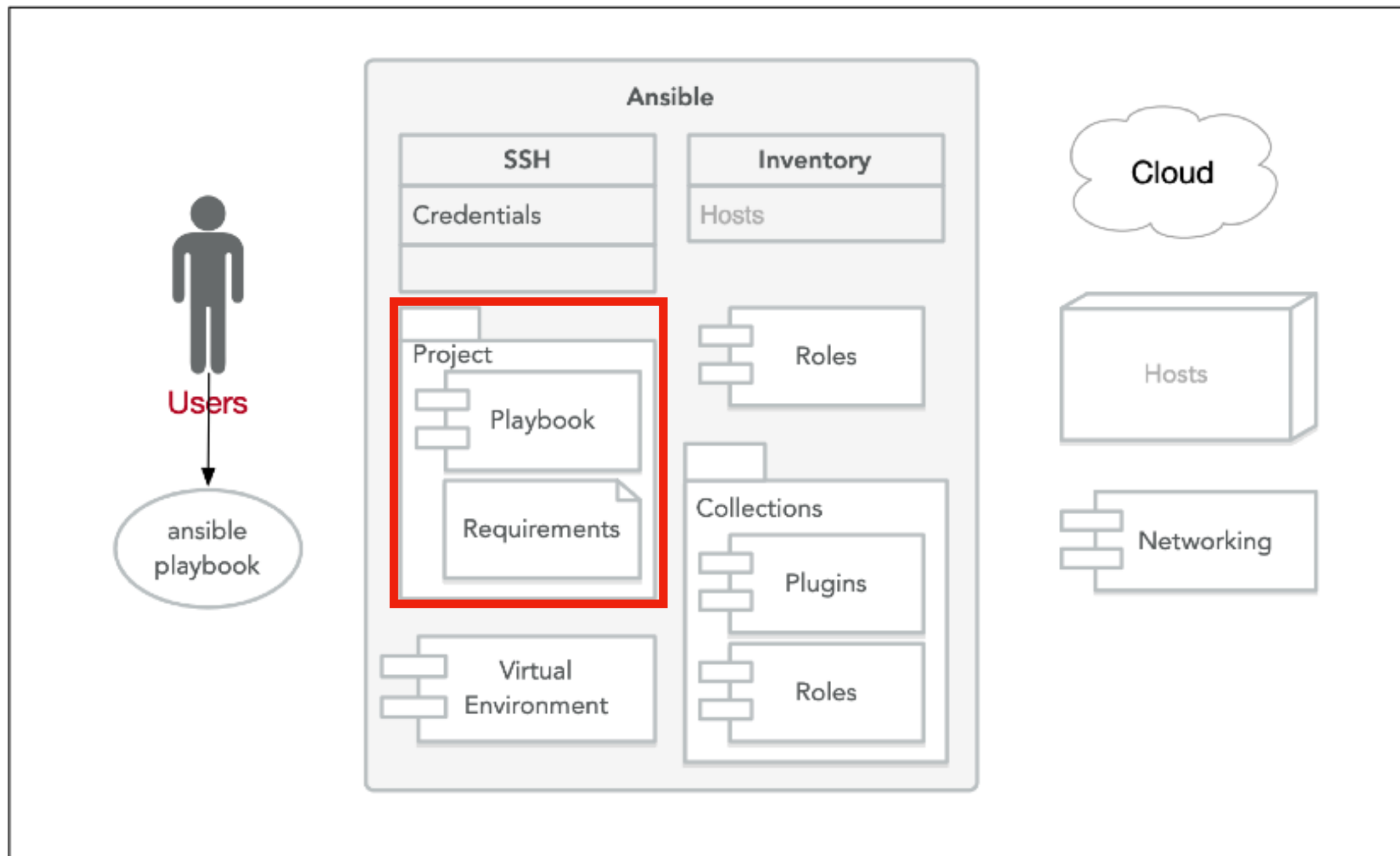


# How to work with **Projects**?



A **Project** can be used to organize content

A Project lists its dependencies as Requirements

Specific behaviour is de-coupled in **Roles** or **Collections**

A Project is stored in a Git repository

# Project layout

- files
- templates
- collections
- roles
- galaxy

```
ansible.cfg           # parameters that affect running
playbook.yml          # main playbook
collections/
    requirements.yml   # ansible-collection dependencies
files/
    static.txt         # 'files' is used for relative paths
    script.py          # file for use with the copy action
galaxy/
group_vars/
    db/                # script to use with the script action
    secrets            # roles imported from galaxy
library/
    requirements.txt    # project variables can be encrypted
roles/
    myrole/            # db corresponds to an inventory group
        defaults/      # ansible-vault encrypted file
            main.yml    # python libraries
        tasks/         # python library dependencies
            main.yml    # in-project roles
        handlers/      # this hierarchy represents a "role"
            main.yml    # defaults can be overridden
        templates/     # contains values for variables
            hosts.j2    # actions that drive modules
            requirements.yml  # could include other files
templates/          # can be notified by tasks on change
                    # defines state after change
                    # for use with the template module
                    # Jinja2 can be used
                    # list of ansible-role dependencies
                    # template file overrides
```