**Implement Little Ansible**

**Context**

In the context of automation, generalized orchestration is increasingly important. This project was created to help understand the fundamentals of Ansible. Your **"Little Ansible"** should be able to parse YAML files and execute many actions on a remote server.

### ****Objectives****

* Create a lightweight orchestrator
* Implement secure SSH connection
* Execute multiple types of modules (commands, package management, service configuration)
* Produce clean and logical Python code following the **PEP8** standard

### ****Developed Features****

#### 1. ****Loading Files****

* todos.yaml: contains the sequence of tasks to execute
* inventory.yaml: contains the list of hosts with connection information (IP, port, username, password or SSH key)
* default.conf.j2: contains the template to copy and replace variable with jinja

#### 2. ****SSH Connection****

* Authentication via **username + password**
* Simple authentication only
* Uses the **Paramiko** library to establish the SSH connection

#### 3. ****Executable Modules****

Each task is linked to a module and parameters. The supported modules are:

* command: executes shell commands
* apt: installs or removes Debian packages
* sysctl: configures kernel parameters
* service: starts or stops services
* copy: copies local/remote files
* template: renders a Jinja2 template into a configuration file

#### 4. ****Display & Logging****

* Uses the logging module to display processing steps
* Shows the result for each task with execution status (stdout)

#### 5. ****PEP8 Compliance****

* The code is formatted using the tool **Black** to comply with **PEP8** standards

#### 6. ****Script Execution****

The main script uses **Click** to run the tasks. An operator should use it from the terminal as follows:

### ****Expected Output Format****

At the end of execution, each task produces a clear summary, structured as follows:

2025-07-03 10:30:21 - main - INFO - Connecting to webserver (192.168.1.10:22)

[webserver] Package 'nginx-common' installed

[webserver] Service 'nginx' started

[webserver] Kernel parameter 'net.core.somaxconn' = 8192 (non-persistent)

[webserver] File /app/dummy.txt copied to ./dummy.txt

[webserver] Template nginx applied with variables: listen\_port=8000, server\_name=\_

[webserver] Custom command executed: echo "test" > /tmp/dummy-file

2025-07-03 10:30:30 - main - INFO - Processing completed for webserver

### ****Summary:****

These messages must be written in plain, human-friendly language (no unnecessary technical jargon), so that a user can easily understand the state of the system after execution.

Good luck! May the force be with you !