# Register in Ansible

it is used to store result/output of a task into a variable.
-- you can print/log that output with debug.
-- Perform conditional Logic
-- Pass values to other task in playbook

While accessing the value of that variable you can below properties.
stdout :: the actual output
stderr :: any error output
stdout_lines :: same as stdout but result will be in list of lines
changed :: Boolean value which is indicating the task changed.
rc :: Return Code (0 mean success)

Use jinja2 Template for generating report
Also loop and conditional based logics in ansible using register variable

```
---
- name: install git and curl packages
hosts: aws1
become: yes

vars:
packages:
- git
- curl
report_path: /tmp/package_audit.txt

tasks:
- name: Install required packages
apt:
name: "{{ item }}"
state: present
update_cache: true
loop: "{{ packages }}"
register: pkg_install_results
changed_when: false
```

```
- name: Capture package versions
command: "{{ item }} --version "
loop: "{{ packages }}"
register: version_cmd
changed_when: false

#make structured Template

- name: Assemble version dictionary
set_fact:
package_versions: >-
{{
dict(
version_cmd.results |
map(attribute='item') | list |
zip(
version_cmd.results |
map(attribute='stdout') | list
)
)
}}

- name: Does a previous report exist?
stat:
path: "{{ report_path }}"
register: report_stat

- name: Backup Old report if exist
command: mv "{{ report_path }}" "{{ report_path }}.bak"
when: report_stat.stat.exists
register: backup_rc
changed_when: backup_rc.rc == 0

- name: Generate report
template:
src: templates/report.j2
dest: "{{ report_path }}"
owner: "{{ ansible_user | default('root') }}"
mode: "0644"
```

Assemble Structure:

{{

```
 dict(
 version_cmd.results |
 map(attribute='item') | list |
 zip(
 version_cmd.results |
 map(attribute='stdout') | map('split') | map('last') | list
 )
 )
 }}
```

map(attribute='item')  gets the .command name (git , curl)
map(attribute='stdout') : take the std output : git version 2.34.1
map('split') ['git', 'version', '2.34.1']
map('last') : picks last item
zip (combine name + version)
convert it into dictionary object