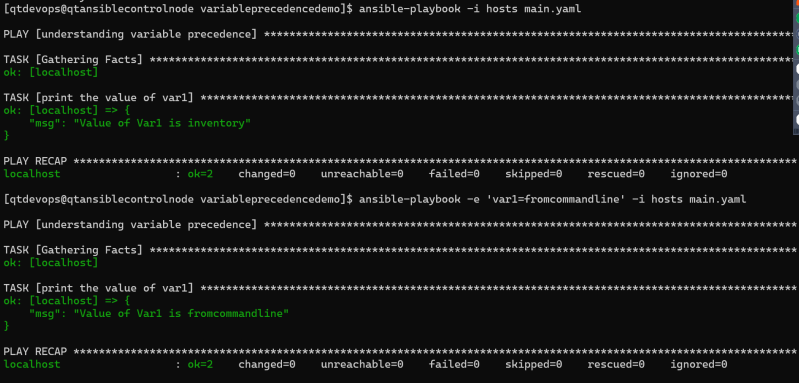
### Ansible Variable Precedence

* In Ansible we can define the variables at multiple places.
* If the variable is defined only once then that variable will be applied, But if the variable is defined at multiple places (inventory file, host\_vars, group\_vars, role\_defaults,…) then ansible has a precedence for variables
* <https://docs.ansible.com/ansible/latest/user_guide/playbooks_variables.html#variable-precedence-where-should-i-put-a-variable> for the official docs
* <https://github.com/asquarezone/AnsibleZone/commit/9f0d60d837ba2349f10b57f987017ed57d1f0b22> for the change set 
* [Refer Here](https://github.com/asquarezone/AnsibleZone/commit/9adae477d7554d2fd6c5c58b5d606d8c8168fcaf) for the next change set

### Ansible Forks and controlling playbook execution

* Fork: This represents how many parallel nodes can ansible execute. By default ansible is configured to run on 5 nodes
* Ansible forks can be set
  + by passing argument to ansible-playbook command (ansible-playbook -f 20 ….)
  + Change the ansible.cfg and add the following which applies to all the ansible playbook executions
* [defaults]
* forks = 25

### Debugging in Ansible

* In Ansible we can resolve errors in the debugger
* For that add debugger in the play (debugger: on\_failed)

- name: test playbook

hosts: all

debugger: on\_failed

tasks:

- name: Wrong variable

file:

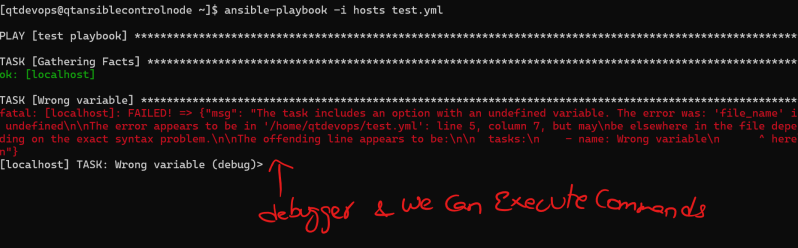
path: "{{ file\_name }}"

state: touch

- name: Learning

debug:

msg: "Learning debugging"

* now execute the playbook 
* Available debug commands

| **Command** | **Shortcut** | **Action** |
| --- | --- | --- |
| print | p | Prints information about the task |
| task.args[key] = value | no shortcut | Update module args |
| task\_vars[key] = value | no shortcut | Update task variables |
| update\_task | u | Recreate a task with updated task variables |
| redo | r | Run the task again |
| continue | c | Continue executing, starting with next task |
| quit | q | Quit the debugger |

