Playbook

```
---
- name: Create FW rule
    hosts: all
    gat her _facts: no
    roles:
   - role: robert deb ock.mysql
        become: yes
        vars:
            mys ql_ use r_name: tony
    tasks:
   - includ e-t asks: tasks/ use ful.yml
   - firewalld:
          var: ansibl e_f acts`
           port: "{{ http_port | defaul t('88') }}/tcp "
           ser vice: http
           source: 192.0.0.0/24
           zone: public
           state: enabled
           per manent: yes #(persist reboot)
           imm ediate: yes #(enable now)
   - name: Generate password
        shell: pwgen -N 1 -s 30
        reg ister: myPass
   - name: Print the generated password
        debug:
           msg: "The password is {{ myPass }}";
```

- When the option expects a variable as parameter, you don't need double curly breaths around variables.
- When line start with a variable like on the `port` line above, you **must** use double quote " around the variable.
- Boolean value accept: ***yes, true, True, TRUE*** and ***no, NO, false, FALSE***

Block

```
- hosts: server1
    tasks:
   - name: Install git and restart nginx
        block:
       - name: install Git
           yum: name=git state= present
       - name: Restart nginx
            ser vice: name=nginx state= res tarted
        become: yes
        when: ansibl e_f act s[' dis tri but ion ']= ='C entOS'
        rescue: # Do this if the task fail
         - mail:
```

Block (cont)

```
>         to: admin@domain.com
          subject: Houston, We've Got a Problem
          body: task {{ ansible_failed_task.name }}
    always: # Always do this
    - mail:
        to: user@domain.com
        subject: "{{ ansible_play_name }} status"
        body: "{{ ansible_failed_result }}"
```

By **Gael.langlais**
cheatography.com/gael-langlais/

Published 26th January, 2023.
Last updated 18th September, 2025.

Variables precedences

Link to Ansible doc: https://shorturl.at/S62Sk

Magic Variables

hostvars['srv1'].ip_address	Get vars value defined for another host
hostvars['srv1'].ansible_facts.mounts	Get facts value from another host
hostvars['srv1'][ansible_facts][mounts]	Same as above writen diffrently
group_names	Get all groups the current host is part of
inventory_hostname	Get the current host name define in inventory not the FQDN

Performances tuning

gather_facts: False	Disable facts gathering
forks=50 (in ansible.cfg)	or **--forks 50** or **-f 50** on CLI (default is **5**)

[ssh_connection] (in ansible.cfg)
ssh_args = -o ControlMaster=auto -o ControlPersist=60s

pipelining = True (in ansible.cfg)	Reduce the number of SSH connections
strategy: free	Execute tasks without waiting for other hosts to finish their tasks

Use mitogen if you can:
https://mitogen.networkgenomics.com/ansible_detailed.html

Inventory

```
all:
    hosts:
        mai l.e xam ple.com:
            ans ibl e_host: 192.16 8.1.230
            ans ibl e_user: mailmin
    chi ldren:
        web ser vers:
            hosts:
                frc llw eb001:
                    ans ibl e_host: 192.16 8.1.231
```

ansible.cfg precedence

1. **$ANSIBLE_CONFIG**
2. **ansible.cfg in playbook dir**
3. **~/.ansible.cfg**
4. **/etc/ansible/ansible.cfg**

Error Handling

any_errors_fatal: true	Will stop & exit playbook if any error
ignore_errors: true	Execute next tasks on failed hosts
failed_when: <condition>	Task fail when condition is met

Avoid using **shell** or **command** modules, they will simply execute the command without any validations, use the appropriate module.

Ansible tips & triks

Make your playbook executable
Add the following sheebang at the top of your palybook:
`#!/usr /bi n/a nsible`
and add the executable bit to your playbook file
chmod +x <my_playbook.yml>
Now you can execute your playbook like a regular script **./my_playbook.yml**