

Playbook

```
---
- name: Create FW rule
  hosts: all
  gather_facts: no
  roles:
    - role: robert de_ock.mysql
      become: yes
      vars:
        mysql_username: tony
      tasks:
    - include_tasks: tasks/ansible.yml
    - firewall:
        var: ansible_facts
        port: "{{ http_port | default('88') }}/tcp "
        service: http
        source: 192.0.0.0/24
        zone: public
        state: enabled
        permanent: yes #(persist reboot)
        immediate: yes #(enable now)
- name: Generate password
  shell: pwgen -N 1 -s 30
  register: 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
          service: name=nginx state= restarted
      become: yes
      when: ansible_facts['distribution'] = 'CentOS'
      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 }}"
```

C

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

Published 26th January, 2023.
Last updated 18th September, 2025.
Page 3 of 3.

Sponsored by **CrosswordCheats.com**
Learn to solve cryptic crosswords!
<http://crosswordcheats.com>

Variables precedences

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

Magic Variables

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

Performances tuning

<code>gather_facts: False</code>	Disable facts gathering
<code>forks=50</code> (in ansible.cfg) or <code>--forks 50</code> or <code>-f 50</code> on CLI (default is 5)	
<code>[ssh_connection]</code> (in ansible.cfg)	
<code>ssh_args = -o ControlMaster=auto -o ControlPersist=60s</code>	
<code>pipelining = True</code> (in ansible.cfg)	Reduce the number of SSH connections
<code>strategy: free</code>	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:
    mail.example.com:
      ansible_host: 192.168.1.230
      ansible_user: mailmin
    children:
      web servers:
        hosts:
          firewall01:
            ansible_host: 192.168.1.231
```

ansible.cfg precedence

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

Error Handling

<code>any_errors_fatal: true</code>	Will stop & exit playbook if any error
<code>ignore_errors: true</code>	Execute next tasks on failed hosts
<code>failed_when: <condition></code>	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 & tricks

Make your playbook executable

Add the following sheebang at the top of your palybook:

```
#!/usr/bin/env ansible
```

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`