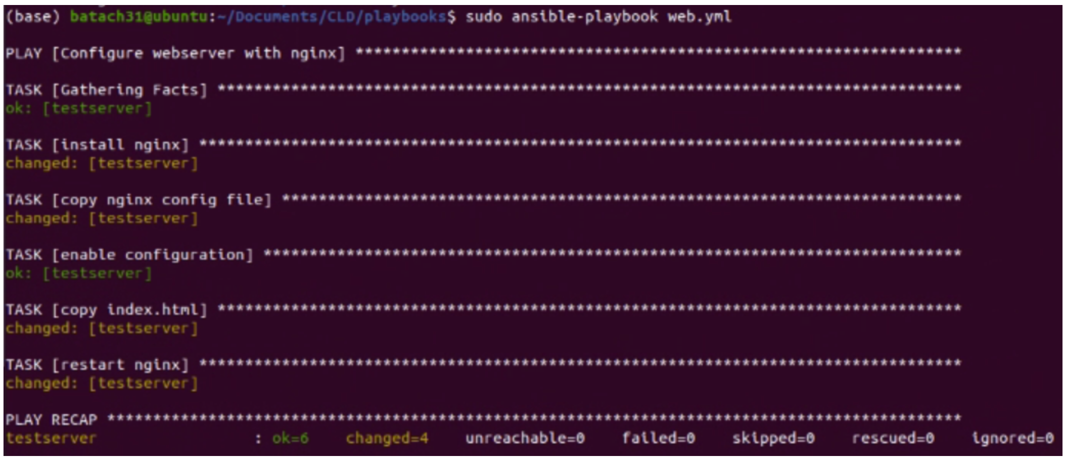
# CLD - Lab 07: Configuration management - Ansible

Authors : Baptiste Hardrick & David Jaquet

## Task 5: Test desired state configuration principles

1. Return to the output of running the *web.yml* playbook the first time. There is one additional task that was not in the playbook. Which one? Among the tasks that are in the playbook there is one task that Ansible marked as *ok*. Which one? Do you have a possible explanation?
   - The screenshot below shows the output of the first output of the command `ansible-playbook web.yml`. We can see the additional task `TASK [Gathering Facts]`.

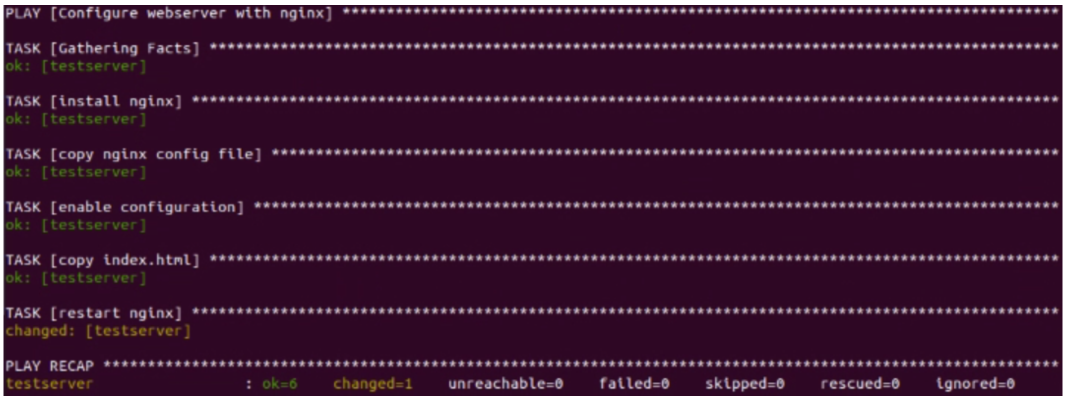
     

     ```
     (base) batach31@ubuntu:~/Documents/CLD/playbooks$ sudo ansible-playbook web.yml

     PLAY [Configure webserver with nginx] ************************************************************

     TASK [Gathering Facts] ***************************************************************************
     ok: [testserver]

     TASK [install nginx] *****************************************************************************
     changed: [testserver]

     TASK [copy nginx config file] ********************************************************************
     changed: [testserver]

     TASK [enable configuration] **********************************************************************
     ok: [testserver]

     TASK [copy index.html] ***************************************************************************
     changed: [testserver]

     TASK [restart nginx] *****************************************************************************
     changed: [testserver]

     PLAY RECAP ***************************************************************************************
     testserver                 : ok=6    changed=4    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
     ```

     Except the *Gathering Facts*, the other task marked as *ok* is the task `TASK [enable configuration]`. We do not specify a configuration for these tasks, so Ansible will use default value.

     Source : [medium.com]()

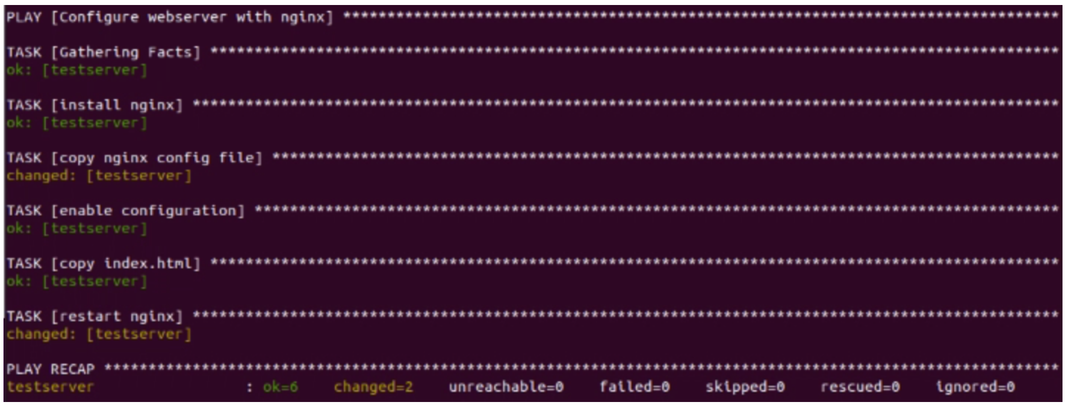
2. Re-run the *web.yml* playbook a second time. In principle nothing should have changed. Compare Ansible's output with the first run. Which tasks are marked as changed?
   - The screenshot below shows the output of the first output of the command `ansible-playbook web.yml`. We can see the task `TASK [restart nginx]` is marked as changed.

     ```
     PLAY [Configure webserver with nginx] ************************************************************

     TASK [Gathering Facts] ***************************************************************************
     ok: [testserver]

     TASK [install nginx] *****************************************************************************
     ok: [testserver]

     TASK [copy nginx config file] ********************************************************************
     ok: [testserver]

     TASK [enable configuration] **********************************************************************
     ok: [testserver]

     TASK [copy index.html] ***************************************************************************
     ok: [testserver]

     TASK [restart nginx] *****************************************************************************
     changed: [testserver]

     PLAY RECAP ***************************************************************************************
     testserver                 : ok=6    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
     ```

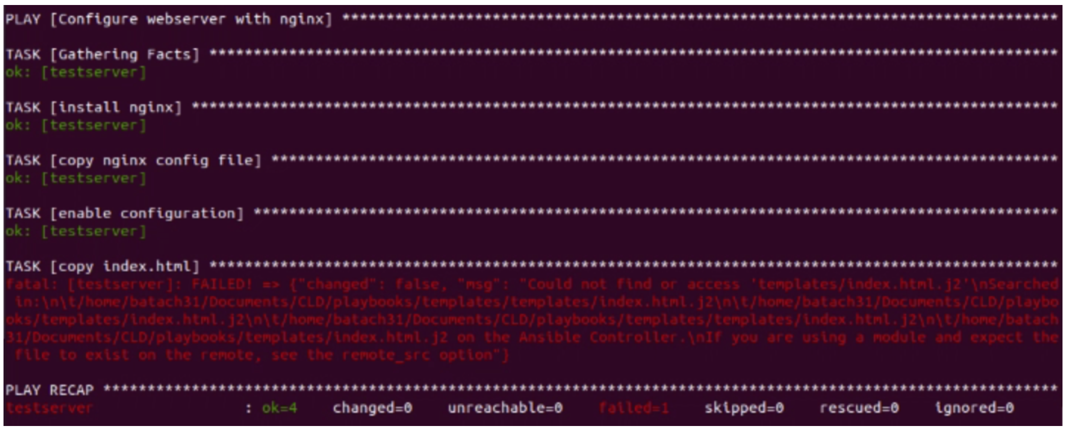
3. SSH into the managed server. Modify the nginx configuration file */etc/nginx/sites-available/default*, for example by adding a line with a comment. Re-run the playbook. What does Ansible do to the file and what does it show in its output?

- For this question, we add a comment in the nginx configuration file */etc/nginx/sites-available/default*. In the screenshot below, we can see Ansible re-copy the nginx configuration file as before the modification. So, the comment is no longer in the file.



```
PLAY [Configure webserver with nginx] ******************************************************************

TASK [Gathering Facts] *********************************************************************************
ok: [testserver]

TASK [install nginx] ***********************************************************************************
ok: [testserver]

TASK [copy nginx config file] **************************************************************************
changed: [testserver]

TASK [enable configuration] ****************************************************************************
ok: [testserver]

TASK [copy index.html] *********************************************************************************
ok: [testserver]

TASK [restart nginx] ***********************************************************************************
changed: [testserver]

PLAY RECAP *********************************************************************************************
testserver                 : ok=6    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

4. Do something more drastic like completely removing the homepage and repeat the previous question.
    - For this question we delete the file *index.html.j2* and re-run the command `ansible-playbook web.yml`. The screenshot below shows the output of this command. We can see a fatal error in the task `TASK [copy index.html]`. The process stops at the error and nginx was not restarted.



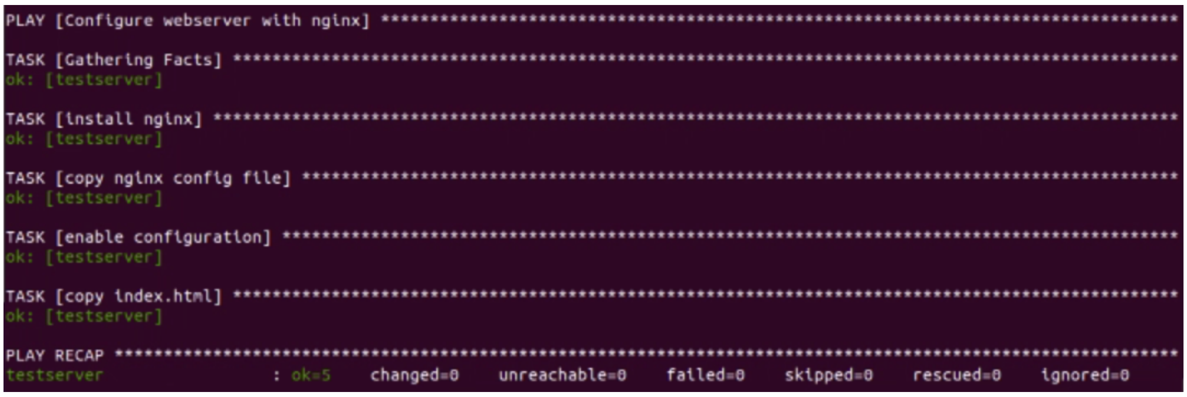
```
PLAY [Configure webserver with nginx] ******************************************************************

TASK [Gathering Facts] *********************************************************************************
ok: [testserver]

TASK [install nginx] ***********************************************************************************
ok: [testserver]

TASK [copy nginx config file] **************************************************************************
ok: [testserver]

TASK [enable configuration] ****************************************************************************
ok: [testserver]

TASK [copy index.html] *********************************************************************************
fatal: [testserver]: FAILED! => {"changed": false, "msg": "Could not find or access 'templates/index.html.j2'\nSearched
 in:\n\t/home/batach31/Documents/CLD/playbooks/templates/templates/index.html.j2\n\t/home/batach31/Documents/CLD/playbo
oks/templates/index.html.j2\n\t/home/batach31/Documents/CLD/playbooks/templates/templates/index.html.j2\n\t/home/batach
31/Documents/CLD/playbooks/templates/index.html.j2 on the Ansible Controller.\nIf you are using a module and expect the
 file to exist on the remote, see the remote_src option"}

PLAY RECAP *********************************************************************************************
testserver                 : ok=4    changed=0    unreachable=0    failed=1    skipped=0    rescued=0    ignored=0
```

## Task 6: Adding a handler for nginx restart

You can find below the new *web.yml* file.

```
- name: Configure webserver with nginx
  hosts: webservers
  become: True
  tasks:
    - name: install nginx
      apt: name=nginx update_cache=yes
    - name: copy nginx config file
      copy: src=files/nginx.conf dest=/etc/nginx/sites-available/default
      notify: restart nginx
    - name: enable configuration
      file: >
        dest=/etc/nginx/sites-enabled/default
        src=/etc/nginx/sites-available/default
        state=link
      notify: restart nginx
    - name: copy index.html
      template: src=templates/index.html.j2 dest=/usr/share/nginx/html/index.html mode=0644
  handlers:
    - name: restart nginx
      service:
        name: nginx
        state: restarted
```

After we modify the *web.yml* file, we run the command `ansible-playbook web.yml`. As shown in the screenshot below, we do not have the `restart nginx` task and all the tasks as marked as *ok*.



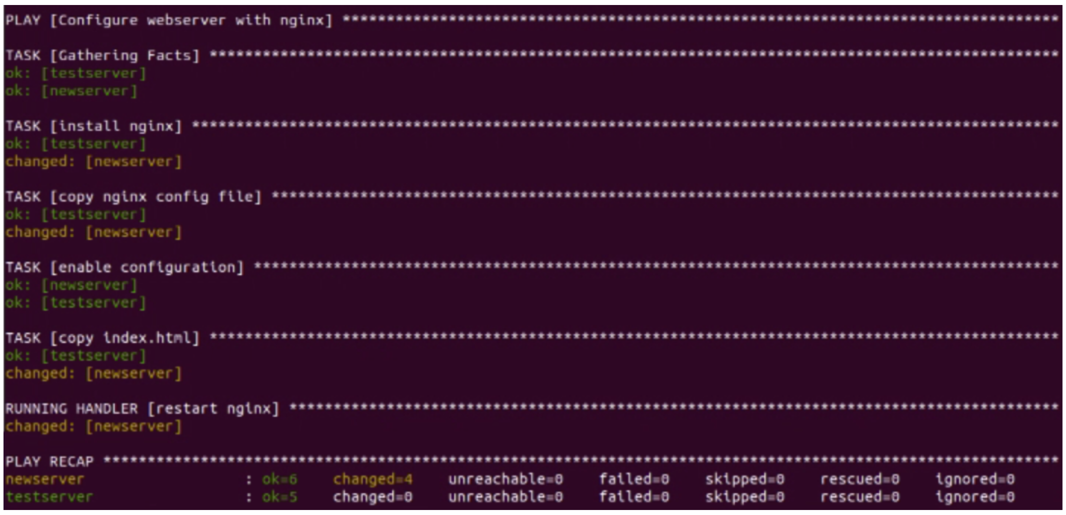
```
PLAY [Configure webserver with nginx] ******************************************************************

TASK [Gathering Facts] *********************************************************************************
ok: [testserver]

TASK [install nginx] ***********************************************************************************
ok: [testserver]

TASK [copy nginx config file] **************************************************************************
ok: [testserver]

TASK [enable configuration] ****************************************************************************
ok: [testserver]

TASK [copy index.html] *********************************************************************************
ok: [testserver]

PLAY RECAP *********************************************************************************************
testserver                 : ok=5    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

## Task 7: Add more managed servers

1. In AWS create another EC2 instance using the same parameters as before. Add the IP address of that instance to the *webservers* group in the *playbooks/hosts* inventory file.
   - For this question we add a line in the *hosts* file, the file has been edited as following :

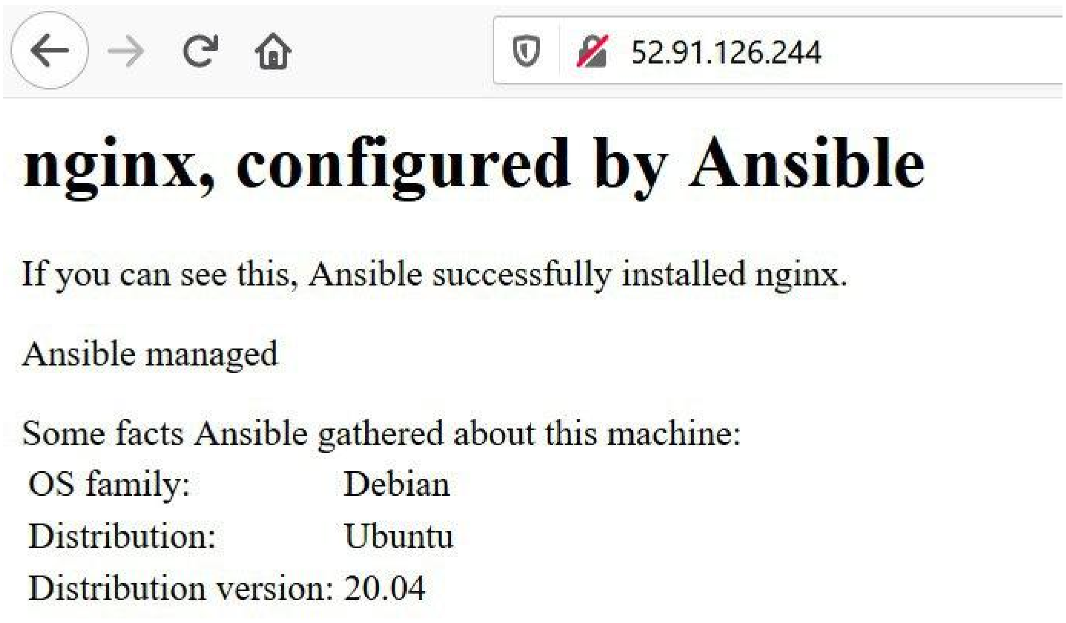
   ```
   [webservers]
   testserver ansible_ssh_host=100.25.117.198
   newserver ansible_ssh_host=52.91.126.244
   ```

2. Re-run the *web.yml* playbook. What do you observe in Ansible's output?
   - For this question, we run again the command `ansible-playbook web.yml`. The output is shown in the screenshot below. We can see a new line on each task. The new lines represent the new server. The *PLAY RECAP* part shows the summary about the two servers. Furthermore, we can see the output for the second server is the same as the one when we execute the command for first time with only the `testserver`.

   ```
   PLAY [Configure webserver with nginx] ******************************************************************

   TASK [Gathering Facts] *********************************************************************************
   ok: [testserver]
   ok: [newserver]

   TASK [install nginx] ***********************************************************************************
   ok: [testserver]
   changed: [newserver]

   TASK [copy nginx config file] **************************************************************************
   ok: [testserver]
   changed: [newserver]

   TASK [enable configuration] ****************************************************************************
   ok: [newserver]
   ok: [testserver]

   TASK [copy index.html] *********************************************************************************
   ok: [testserver]
   changed: [newserver]

   RUNNING HANDLER [restart nginx] ************************************************************************
   changed: [newserver]

   PLAY RECAP *********************************************************************************************
   newserver                  : ok=6    changed=4    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
   testserver                 : ok=5    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
   ```

3. Test the new server by pointing your web browser to it.
   - The screenshot below shows the new server seen by a web browser.



# nginx, configured by Ansible

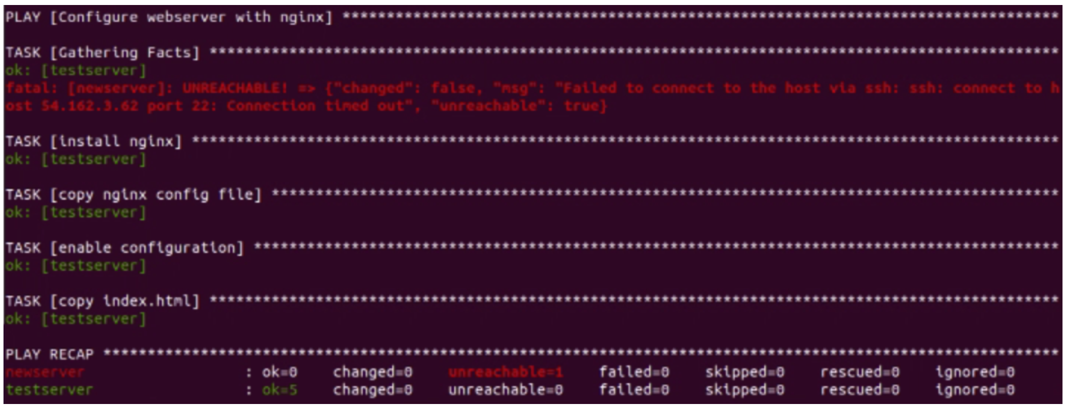
If you can see this, Ansible successfully installed nginx.

Ansible managed

Some facts Ansible gathered about this machine:

|  |  |
|---|---|
| OS family: | Debian |
| Distribution: | Ubuntu |
| Distribution version: | 20.04 |

4. What happens if a server is not reachable? Shut down the second instance and re-run the playbook.
   - You can find below the output of the `ansible-playbook web.yml` command. We can see that the new server is unreachable at the first task. For the following task, Ansible does not work with the second server.



```
PLAY [Configure webserver with nginx] **************************************************************

TASK [Gathering Facts] *****************************************************************************
ok: [testserver]
fatal: [newserver]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: ssh: connect to h
ost 54.162.3.62 port 22: Connection timed out", "unreachable": true}

TASK [install nginx] *******************************************************************************
ok: [testserver]

TASK [copy nginx config file] **********************************************************************
ok: [testserver]

TASK [enable configuration] ************************************************************************
ok: [testserver]

TASK [copy index.html] *****************************************************************************
ok: [testserver]

PLAY RECAP *****************************************************************************************
newserver                  : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0    ignored=0
testserver                 : ok=5    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

5. Suppose you now have 10 web servers in production that you have configured using Ansible. You are working in the IT department of a company and some of your system administrator colleagues who don't use Ansible have logged manually into some of the servers to fix certain things. You don't know what they did exactly. What do you need to do to bring all 10 servers again to the initial state? We'll exclude drastic changes by your colleagues for this question.
   - We can edit the *web.yml* by modifying the value of nginx service's state to have `absent`. Now, we have `restarted`. Change this value makes possible to do a rollback.
   - An another solution is to have a snaphot before the modifications. We can restore the snapshot at any time to have a correct configuration.