

**LAPORAN PROJECT AKHIR  
NETWORK PROGRAMMABILITY**

**AUTOMATION ROUTER CONFIGURATION USING ANSIBLE AND  
BASH SCRIPT**

Oleh :  
**ELSAN PENTANUGRAHA**

**TALENT SCOUTING ACADEMY  
2022**

## 1. Latar Belakang

Untuk membangun jaringan sehingga dapat saling terhubung memerlukan sebuah perangkat jaringan salah satunya yaitu router. Pada router ini perlu dilakukan konfigurasi agar setiap host pada sebuah jaringan dapat saling berkomunikasi, sehingga Ketika menyelesaikan konfigurasi perangkat-perangkat jaringan ini membutuhkan waktu yang lebih lama. Sehingga perlu diterapkannya *network automation* ketika melakukan konfigurasi jaringan.

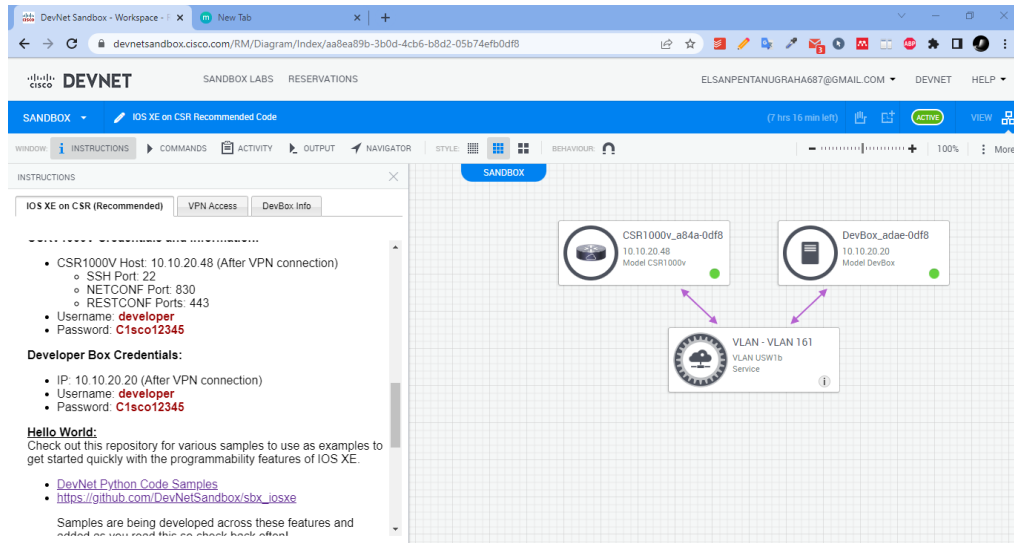
*Network automations* adalah sebuah proses konfigurasi, mengelola, serta menguji perangkat-perangkat jaringan secara otomatis. Salah satu tools dalam otomatisasi jaringan ini adalah *Ansible*. *Ansible* merupakan salah satu *automation tool* dan *infrastructure as a code* yang tersedia secara *open source*. Dengan *Ansible* teknisi dapat melakukan membangun sebuah server, manajemen konfigurasi perangkat, dan melakukan *deployment*. *Network automation* ini akan mempermudah teknisi dalam melakukan konfigurasi pada perangkat jaringan. Dengan menggunakan *Ansible*, teknisi jaringan tidak perlu mengkonfigurasi setiap perangkat yang terpasang satu per satu. Teknisi hanya perlu membuat skrip modul otomatisasi dan menjalankannya melalui *Ansible playbook* maka konfigurasi perangkat dapat berjalan secara otomatis.

## 2. Rancangan Umum Project

Konfigurasi router dilakukan secara otomatis dengan menjalankan *ansible-playbook* melalui *bash script*. Router yang dikonfigurasi yaitu *csr1000v* pada *devnet sandbox*, untuk mengakses *csr1000v* ini menggunakan *virtual machine devasc* yang digunakan pada *course devnet associate*.

Untuk melakukan otomatisasi konfigurasi *csr1000v* ini dibuat sebuah *playbook ansible* berisi task-task yang akan dikonfigurasi dalam bahasa *YAML*. Setelah dilakukan konfigurasi, maka konfigurasi dari *csr1000v* ini *dibackup* dan disimpan kedalam *repository github*.

### 3. Penjelasan Project



Gambar 3. 1 devnet sandbox

Router yang akan dikonfigurasi yaitu *csr1000v* dari *devnet sandbox* maka terlebih dahulu dilakukan reservasi, setelah itu *csr1kv* dengan *virtual machine devasc* dihubungkan dengan mengkoneksikan *virtual machine devasc* menggunakan *anyconnect vpn*. Setelah koneksi terhubung barulah *csr1kv* dapat diakses melalui *devasc*.

```
devasc@labvm:/etc/ansible
File Edit View Search Terminal Help
GNU nano 4.8 hosts
#db01.intranet.mydomain.net
#db02.intranet.mydomain.net
#10.25.1.56
#10.25.1.57

# Here's another example of host ranges, this time there are no
# leading 0s:

#db-[99:101]-node.example.com

[routers]
10.10.20.48

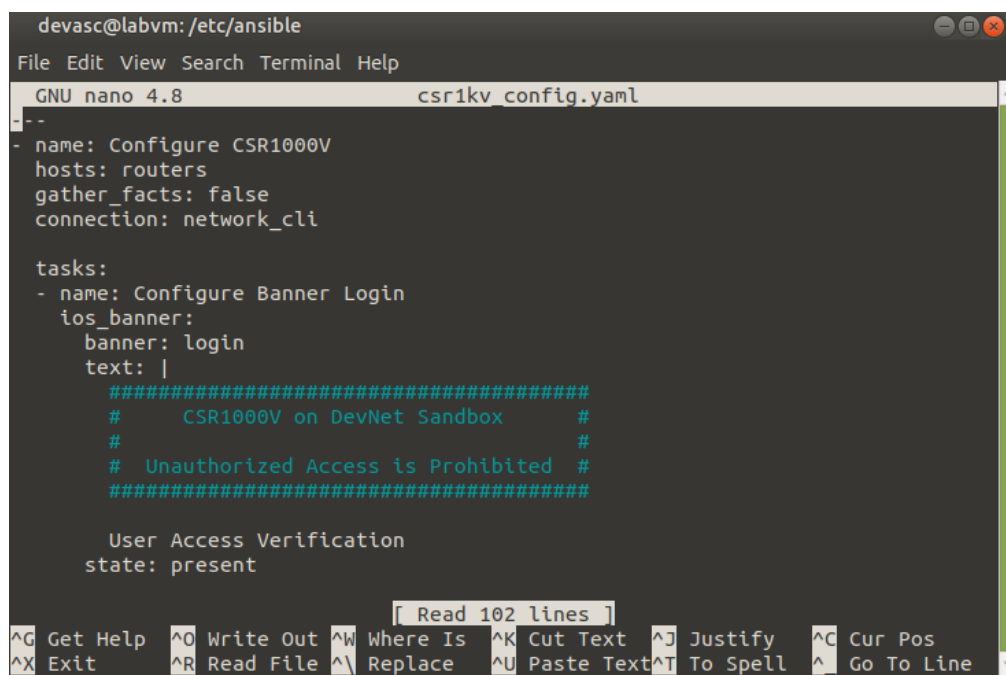
[routers:vars]
ansible_user=developer
ansible_password=Cisco12345
ansible_port=22
ansible_network_os=ios
ansible_connection=network_cli

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Paste Text ^T To Spell ^_ Go To Line
```

Gambar 3. 2 hosts configurations

Kemudian pada *devasc vm*, inisialisasi *hosts csr1000v* pada file *hosts* di direktori */etc/ansible* seperti gambar diatas. *[routers]* adalah inisialisasi grup host yang digunakan kemudian pada host, *ip address 10.10.20.48* merupakan alamat dari *csr1000v* dan juga di inisialisasikan parameternya yang menyimpan *username*, *password*, *port SSH*, dan *network os* dari *csr1kv*.

Pada file *ansible.cfg* dihilangkan “#” pada *host\_key\_checking = false* karena *local environment development* tidak memiliki kunci SSH yang disiapkan pada *local*.



```
devasc@labvm: /etc/ansible
File Edit View Search Terminal Help
GNU nano 4.8 csr1kv_config.yaml
- -
- name: Configure CSR1000V
  hosts: routers
  gather_facts: false
  connection: network_cli

  tasks:
  - name: Configure Banner Login
    ios_banner:
      banner: login
      text: |
        #####
        #   CSR1000V on DevNet Sandbox   #
        #   Unauthorized Access is Prohibited   #
        #####

        User Access Verification
        state: present

[ Read 102 lines ]
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^_ Replace ^U Paste Text ^T To Spell ^_ Go To Line
```

Gambar 3. 3 ansible playbook

*Csr1kv\_config* dibuat sebagai *ansible-playbook* dengan bahasa *YAML* untuk menyimpan task-task yang akan digunakan untuk mengkonfigurasi router secara otomatis melalui playbook ini.

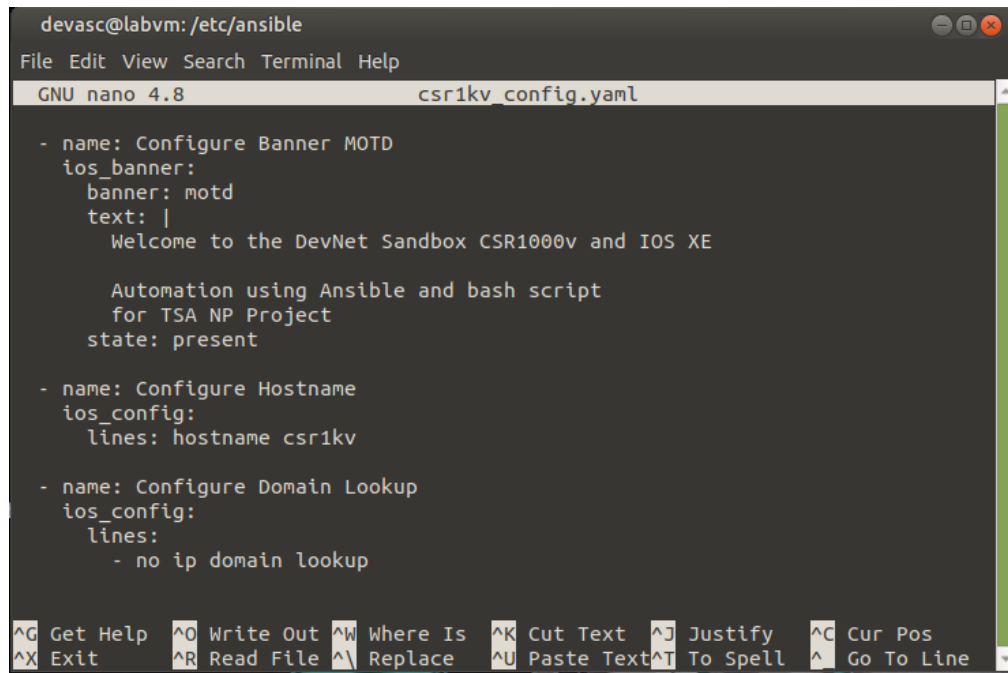
*Name: Configure CSR1000v* merupakan nama dari playbook ini

*Hosts: routers* merupakan nama dari grup atau host

*Connection: network\_cli* menunjukkan bahwa koneksi menggunakan SSH

*Tasks:* merupakan kata kunci yang menunjukkan tugas yang harus dilakukan

Pada task ini tugas yang dilakukan yaitu menambahkan banner login pada *csr100v*, selain itu juga tugas menambahkan *banner MOTD* seperti gambar dibawah.



```
devasc@labvm:/etc/ansible
File Edit View Search Terminal Help
GNU nano 4.8 cs1kv_config.yaml

- name: Configure Banner MOTD
  ios_banner:
    banner: motd
    text: |
      Welcome to the DevNet Sandbox CSR1000v and IOS XE

      Automation using Ansible and bash script
      for TSA NP Project
    state: present

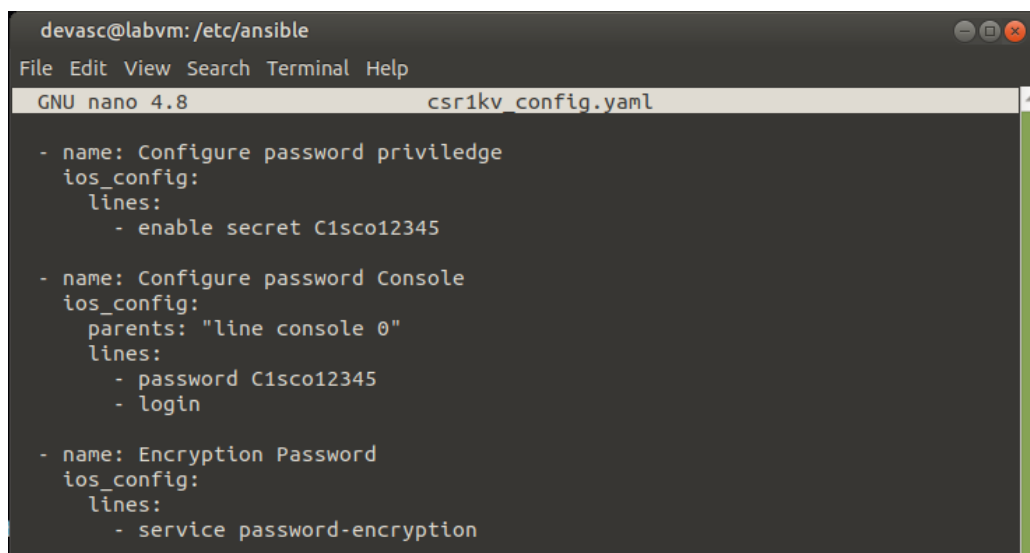
- name: Configure Hostname
  ios_config:
    lines: hostname cs1kv

- name: Configure Domain Lookup
  ios_config:
    lines:
      - no ip domain lookup

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Paste Text ^T To Spell ^_ Go To Line
```

Gambar 3. 4 ansible playbook

dan *name:* yang menyatakan nama dari task tersebut, *ios\_banner:* menyatakan bahwa task ini akan mengkonfigurasi banner, dan *lines:* digunakan untuk menyatakan command yang digunakan. Kemudian juga menambahkan task untuk melakukan konfigurasi *hostname* router, *privilege* dan *console password* serta enkripsi.



```
devasc@labvm:/etc/ansible
File Edit View Search Terminal Help
GNU nano 4.8 cs1kv_config.yaml

- name: Configure password privilege
  ios_config:
    lines:
      - enable secret C1sco12345

- name: Configure password Console
  ios_config:
    parents: "line console 0"
    lines:
      - password C1sco12345
      - login

- name: Encryption Password
  ios_config:
    lines:
      - service password-encryption
```

Gambar 3. 5 ansible playbook

```
devasc@labvm: /etc/ansible
File Edit View Search Terminal Help
GNU nano 4.8 cs1kv config.yaml

- name: Configure Interface Gig2 IP Addresses
  ios_config:
    parents: "interface GigabitEthernet2"
    lines:
      - description IPv4 ADDRESS
      - ip address 192.168.2.1 255.255.255.224
      - no shutdown

- name: Configure Interface Gig3 IP Addresses
  ios_config:
    parents: "interface GigabitEthernet3"
    lines:
      - description IPv4 ADDRESS
      - ip address 10.10.30.1 255.255.255.0
      - no shutdown

- name: Configure IPv6 Address
  ios_config:
    parents: "interface GigabitEthernet3"

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Paste Text ^T To Spell ^_ Go To Line
```

Gambar 3. 6 ansible playbook

Pada task yang digunakan untuk melakukan konfigurasi *ip address*, parameter *parents: "interface GigabitEthernet"* menunjukkan bahwa konfigurasi dilakukan pada mode *interface*.

```
devasc@labvm: /etc/ansible
File Edit View Search Terminal Help
GNU nano 4.8 cs1kv config.yaml

- name: Configure Domain Name
  ios_config:
    lines:
      - ip domain name devnet-project.com
      - username developer secret Cisco12345

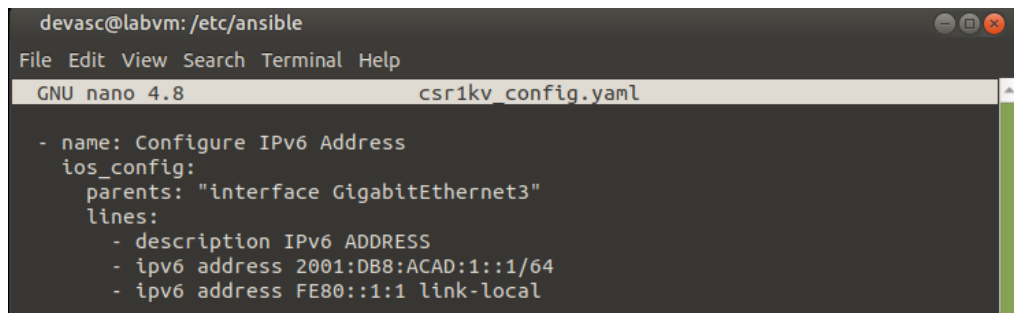
- name: Configure Remote Password
  ios_config:
    parents: "line vty 0 15"
    lines:
      - login local
      - transport input ssh

- name: Configure Remote SSH
  ios_config:
    lines:
      - crypto key generate rsa general-keys modulus 1024
      - ip ssh version 2

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Paste Text ^T To Spell ^_ Go To Line
```

Gambar 3. 7 ansible playbook

Kemudian pada *task* untuk mengkonfigurasi *remote SSH*, parameter *parents*: “*line vty*” menunjukkan bahwa konfigurasi dilakukan pada mode *line*.

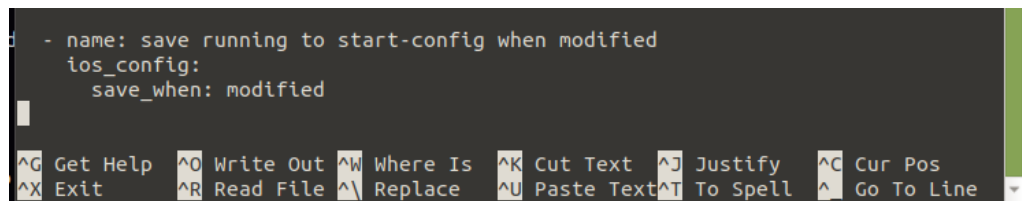


```
devasc@labvm:/etc/ansible
File Edit View Search Terminal Help
GNU nano 4.8 csr1kv_config.yaml

- name: Configure IPv6 Address
  ios_config:
    parents: "interface GigabitEthernet3"
    lines:
      - description IPv6 ADDRESS
      - ipv6 address 2001:DB8:ACAD:1::1/64
      - ipv6 address FE80::1:1 link-local
```

Gambar 3. 8 ansible playbook

Task ini adalah tugas yang akan melakukan konfigurasi IPv6 pada *interface GigabitEthernet3*.

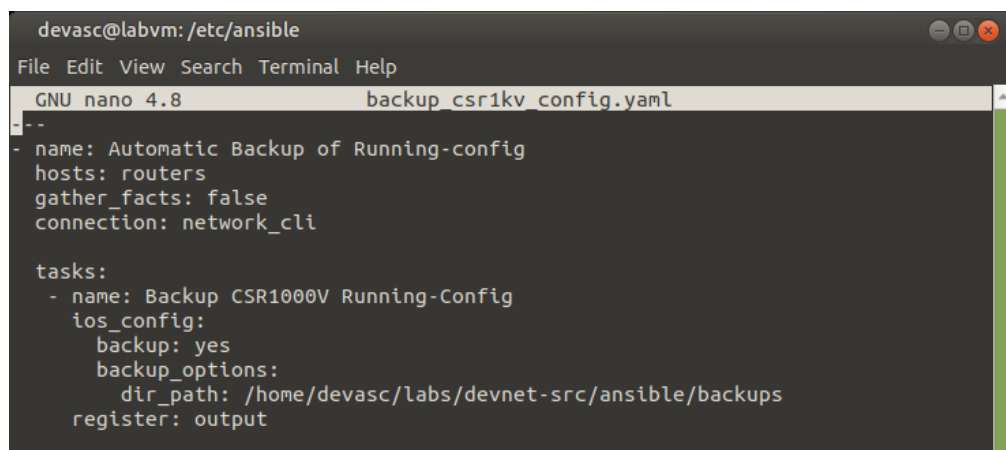


```
- name: save running to start-config when modified
  ios_config:
    save_when: modified
```

Gambar 3. 9 ansible playbook

Task ini akan menyimpan *running config* ke *start-config* ketika telah dilakukan modifikasi.

Kemudian dibuat *playbook* yang akan melakukan tugas melakukan *backup* konfigurasi terhadap *csr1000v* yang dilakukan dan menyimpannya ke direktori */home/devasc/labs/devnet-src/ansible/backups*.



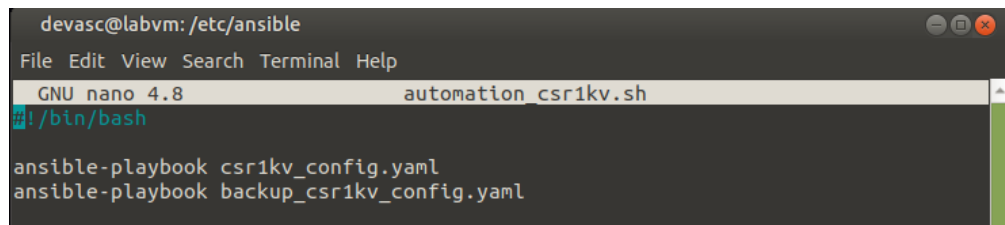
```
devasc@labvm:/etc/ansible
File Edit View Search Terminal Help
GNU nano 4.8 backup_csr1kv_config.yaml

- -
- name: Automatic Backup of Running-config
  hosts: routers
  gather_facts: false
  connection: network_cli

  tasks:
    - name: Backup CSR1000V Running-Config
      ios_config:
        backup: yes
        backup_options:
          dir_path: /home/devasc/labs/devnet-src/ansible/backups
        register: output
```

Gambar 3. 10 ansible playbook to backup config

Kedua *playbook* ini akan dijalankan melalui *bash script*. Pada file *bash* ini dimasukkan perintah untuk menjalankan kedua *playbook* diatas.

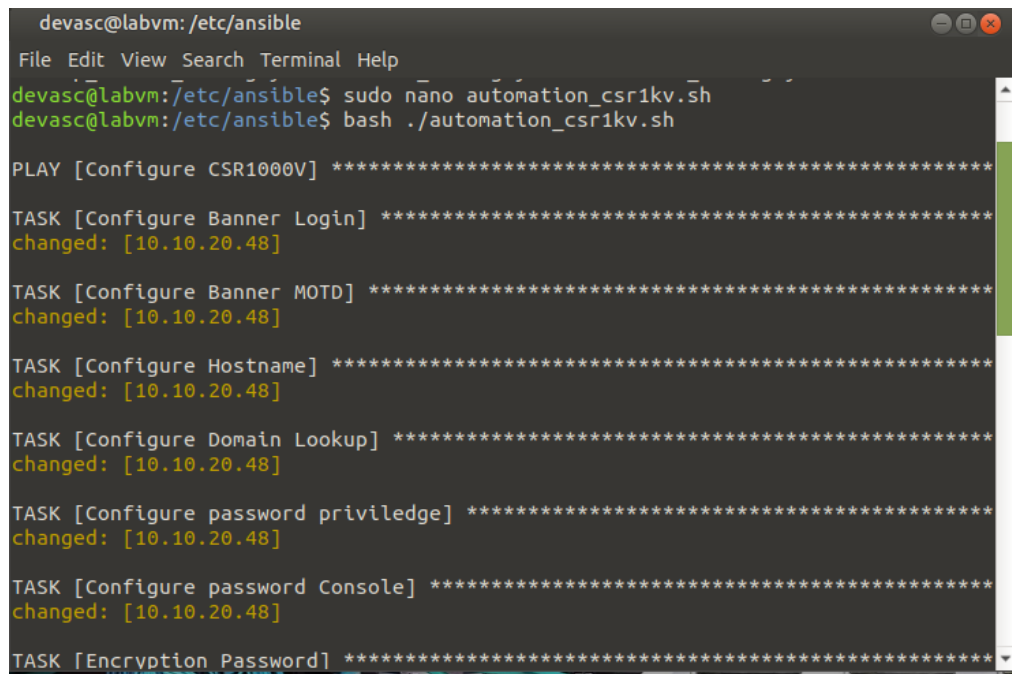


```
devasc@labvm: /etc/ansible
File Edit View Search Terminal Help
GNU nano 4.8 automation_cs1kv.sh
#!/bin/bash

ansible-playbook cs1kv_config.yaml
ansible-playbook backup_cs1kv_config.yaml
```

Gambar 3. 11 bash script

Setelah *automation\_cs1kv.sh* dijalankan maka secara otomatis akan melakukan konfigurasi pada router dan melakukan backup seperti gambar dibawah.



```
devasc@labvm: /etc/ansible
File Edit View Search Terminal Help
devasc@labvm: /etc/ansible$ sudo nano automation_cs1kv.sh
devasc@labvm: /etc/ansible$ bash ./automation_cs1kv.sh

PLAY [Configure CSR1000V] *****

TASK [Configure Banner Login] *****
changed: [10.10.20.48]

TASK [Configure Banner MOTD] *****
changed: [10.10.20.48]

TASK [Configure Hostname] *****
changed: [10.10.20.48]

TASK [Configure Domain Lookup] *****
changed: [10.10.20.48]

TASK [Configure password privilege] *****
changed: [10.10.20.48]

TASK [Configure password Console] *****
changed: [10.10.20.48]

TASK [Encryption Password] *****
```

Gambar 3. 12 running task in ansible-playbook



```
devasc@labvm: /etc/ansible
File Edit View Search Terminal Help
TASK [Configure Remote Password] *****
changed: [10.10.20.48]

TASK [Configure Remote SSH] *****
changed: [10.10.20.48]

TASK [save running to start-config when modified] *****
changed: [10.10.20.48]

PLAY RECAP *****
10.10.20.48 : ok=14 changed=14 unreachable=0 failed=0 s
kipped=0 rescued=0 ignored=0

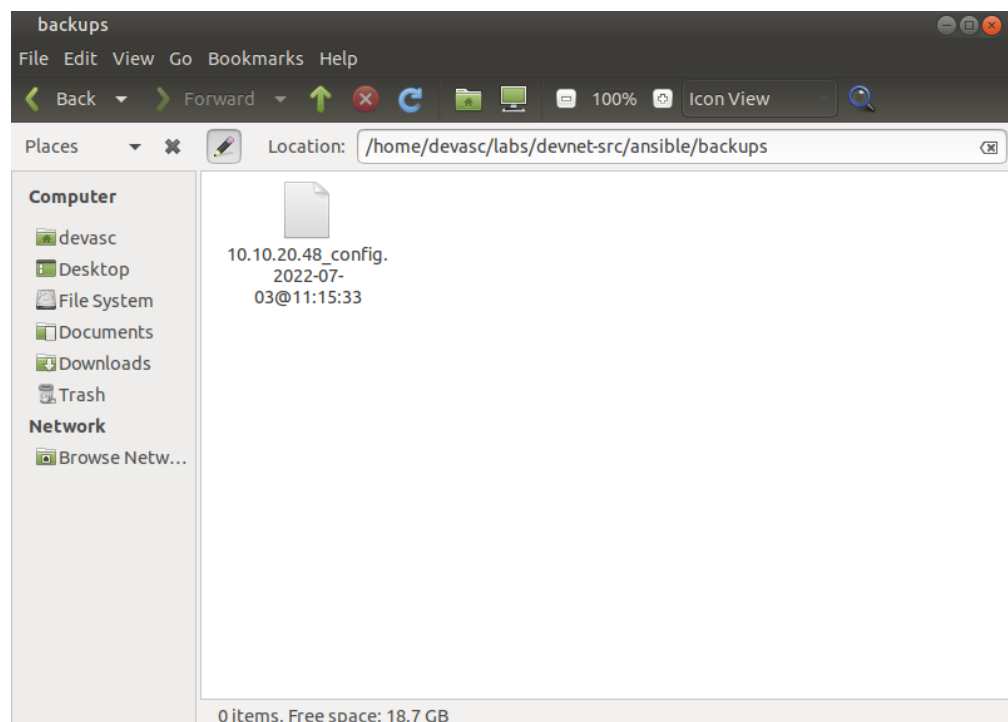
PLAY [Automatic Backup of Running-config] *****

TASK [Backup CSR1000V Running-Config] *****
changed: [10.10.20.48]

PLAY RECAP *****
10.10.20.48 : ok=1 changed=1 unreachable=0 failed=0 s
kipped=0 rescued=0 ignored=0

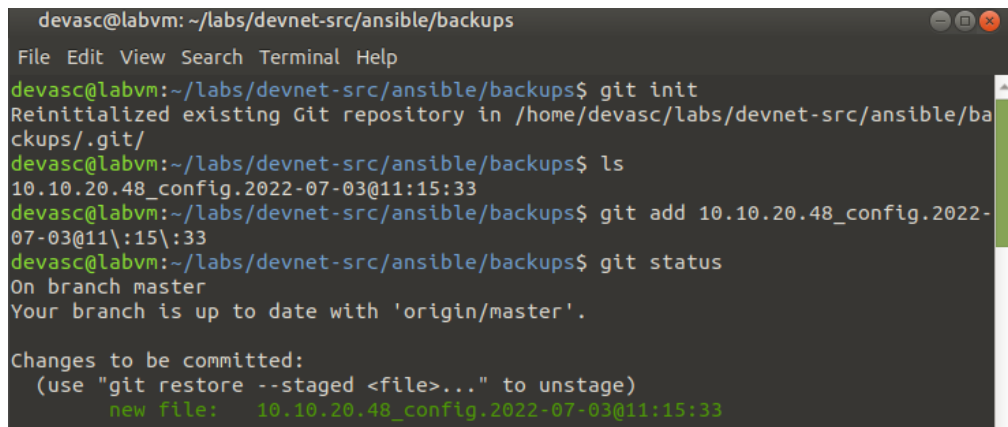
devasc@labvm: /etc/ansible$
```

Gambar 3. 13 running task in ansible-playbook



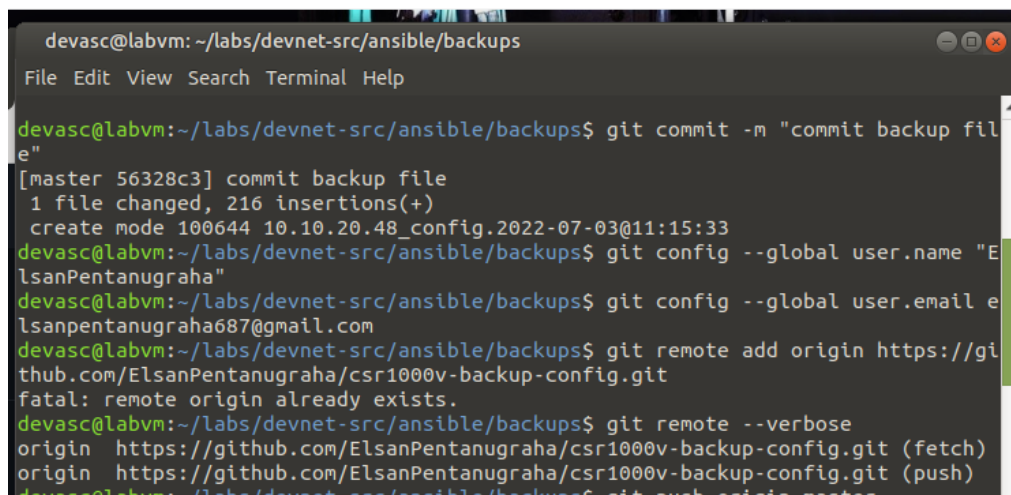
Gambar 3. 14 hasil backup config

Setelah dilakukan *backup*, maka file *backup* diunggah ke *repository github* menggunakan *software version control git*.

A terminal window titled 'devasc@labvm: ~/labs/devnet-src/ansible/backups'. The terminal shows the following commands and output:   
1. `git init`: Reinitialized existing Git repository in /home/devasc/labs/devnet-src/ansible/backups/.git/   
2. `ls`: Lists the file `10.10.20.48_config.2022-07-03@11:15:33`   
3. `git add 10.10.20.48_config.2022-07-03@11:15:33`: Stages the file   
4. `git status`: Shows 'On branch master' and 'Your branch is up to date with 'origin/master''. It also lists 'Changes to be committed: (use "git restore --staged <file>..." to unstage)' with 'new file: 10.10.20.48\_config.2022-07-03@11:15:33'.

Gambar 3. 15 git command

*Git init* merupakan perintah yang menginisiasi direktori *backups* sebagai repositori git. Kemudian *git add* “*nama file*” untuk melakukan *staging file backup*.

A terminal window titled 'devasc@labvm: ~/labs/devnet-src/ansible/backups'. The terminal shows the following commands and output:   
1. `git commit -m "commit backup file"`: Commits the file with message 'commit backup file'. Output shows '[master 56328c3] commit backup file', '1 file changed, 216 insertions(+)', and 'create mode 100644 10.10.20.48\_config.2022-07-03@11:15:33'   
2. `git config --global user.name "ElsanPentanugraha"`: Sets the global user name   
3. `git config --global user.email elsanpentanugraha687@gmail.com`: Sets the global user email   
4. `git remote add origin https://github.com/ElsanPentanugraha/csr1000v-backup-config.git`: Adds the origin remote. Output shows 'fatal: remote origin already exists.'   
5. `git remote --verbose`: Shows the origin remote as 'origin https://github.com/ElsanPentanugraha/csr1000v-backup-config.git (fetch)' and 'origin https://github.com/ElsanPentanugraha/csr1000v-backup-config.git (push)'   
6. `git push origin master`: Pushes the master branch to the origin remote.

Gambar 3. 16 git command

*Git commit* adalah perintah untuk melakukan committing file yang ditambahkan ke repositori git. *Git config --global user.name* dan *user.email* digunakan untuk mencocokkan dengan Github credensial. *Git remote add* digunakan untuk menambahkan url github sebagai remote. *Git remote --verbose* digunakan untuk memverifikasi git terhubung ke *github.com*.

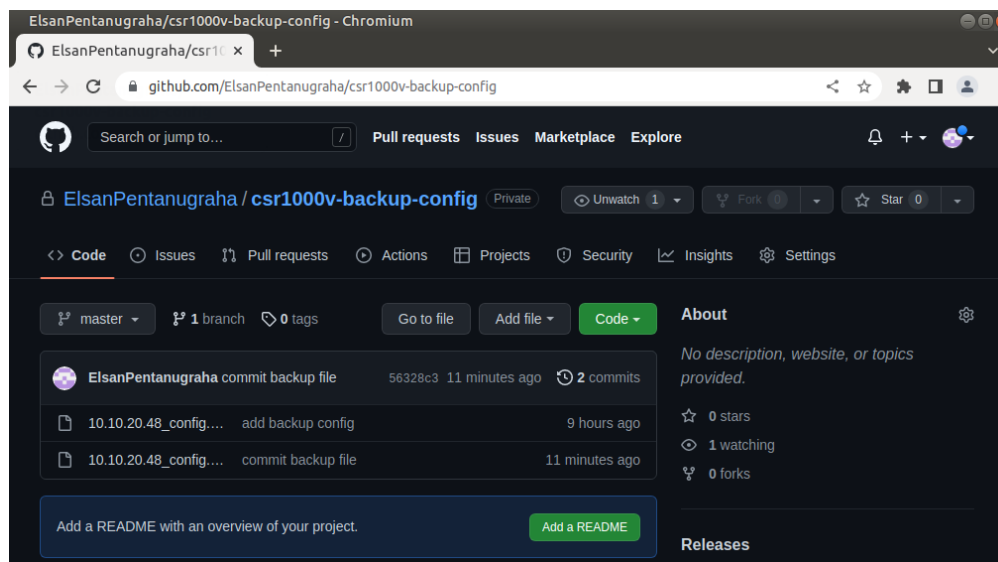
```

devasc@labvm:~/labs/devnet-src/ansible/backups$ git push origin master
Username for 'https://github.com': ElsanPentanugraha
Password for 'https://ElsanPentanugraha@github.com':
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 2 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 2.98 KiB | 611.00 KiB/s, done.
Total 6 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), done.
To https://github.com:ElsanPentanugraha/csr1000v-backup-config.git
 * [new branch]      master -> master
devasc@labvm:~/labs/devnet-src/ansible/backups$

```

Gambar 3. 17 git command

*Git push origin master* adalah perintah yang digunakan untuk mengirimkan (*push*) file ke repositori *github*. Pada perintah ini diperlukan *github username* dan *password* yang dimasukan menggunakan *personal access token*. Berikut adalah hasil tangkapan layar repositori *github* yang telah dikirimkan file backup dari *csr1000v*.



Gambar 3. 18 Github Repository