Case Study. Configuring CISCO router using Ansible	
Name: Torrecampo, Juan Piolo	**Date:** April 20, 2023
Course/Section: CPE 234 - CPE32S3	**Instructor:** Engr. Taylar

Procedure:

Intended Learning Outcomes

- Create a Cisco Router VM on VirtualBox
- Automate the configuration of the router using Ansible

Scenario

Configuring a Cisco router virtual machine using Ansible involves several steps.

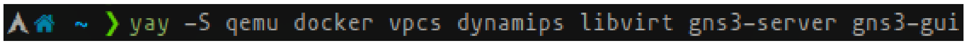
1. **Install Ansible**: The first step is to install Ansible on your system. You can follow the instructions provided on the official Ansible website to install it.
2. **Create an Ansible playbook**: The next step is to create an Ansible playbook that will be used to configure the Cisco router virtual machine. The playbook will contain a set of tasks that will be executed on the virtual machine to configure it.
3. **Define the inventory**: The inventory is a list of hosts that Ansible can manage. You will need to define the IP address or hostname of the virtual machine that you want to configure.
4. **Configure the Cisco router**: You can use the Ansible IOS module to configure the Cisco router. This module provides a set of tasks that can be used to configure various aspects of the router, such as interfaces, VLANs, routing protocols, and security features.
5. **Execute the playbook**: Once you have defined the playbook and the inventory, you can execute the playbook using the ansible-playbook command. Ansible will connect to the virtual machine and execute the tasks defined in the playbook.

Instructions:

- Download the GNS3 VM here. Software | GNS3
- Links to an external site.
- Install the GNS3 VM in Virtualbox
- Create a basic configuration for the Cisco router using the Ansible IOS module (check the reference for Cisco Router basic configuration)
- Show evidence of the playbook running successfully
- Verify the configuration done on the Cisco router (this should be seen in the managed node)
- Verify the configuration done on the Cisco router(manually check the configuration on the Cisco router VM)

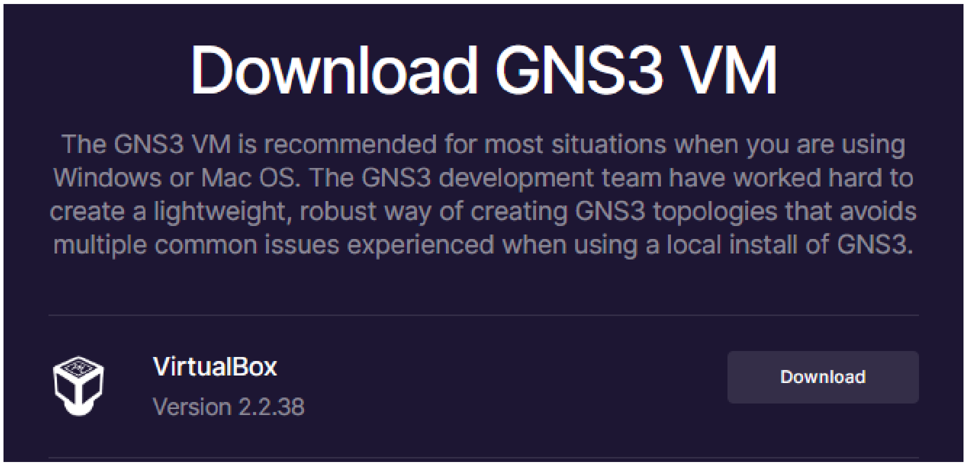
Output:

1. **Download GNS3 VM and GNS3 GUI.**



```
~ ❯ yay -S qemu docker vpcs dynamips libvirt gns3-server gns3-gui
```

Figure 1. Installing GNS3 and its dependencies.



Download GNS3 VM

The GNS3 VM is recommended for most situations when you are using Windows or Mac OS. The GNS3 development team have worked hard to create a lightweight, robust way of creating GNS3 topologies that avoids multiple common issues experienced when using a local install of GNS3.

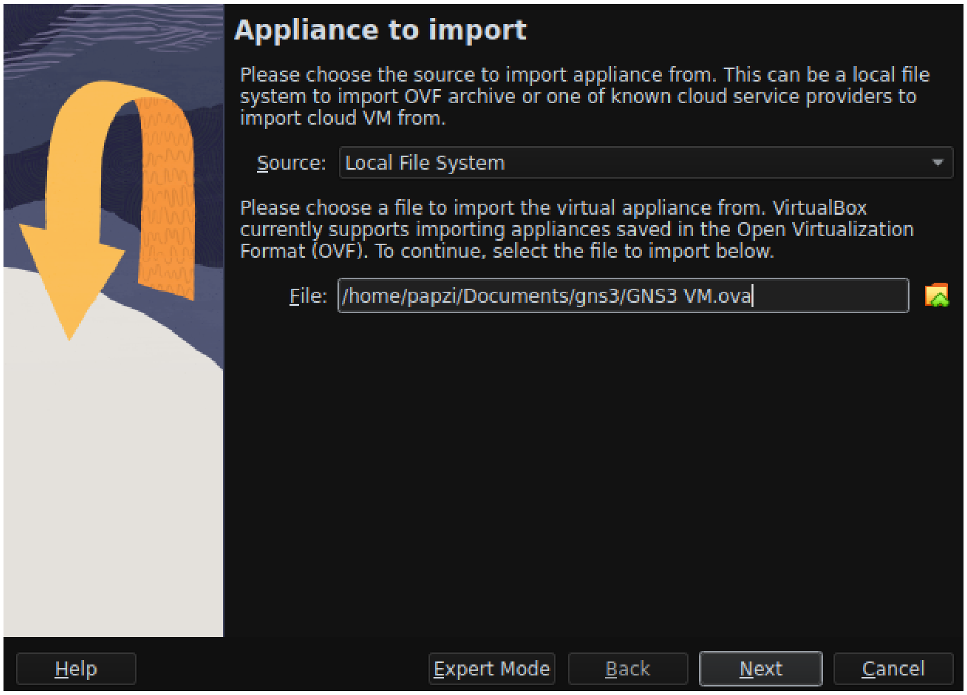
VirtualBox

Version 2.2.38

Figure 2. Downloading GNS3 VM for Virtualbox ova file.

Note. Unarchive the zipped file to reveal the .ova file.

2. **Install the GNS3 VM in Virtualbox by using "Import Appliances…" under the "File" menu.**



Appliance to import

Please choose the source to import appliance from. This can be a local file system to import OVF archive or one of known cloud service providers to import cloud VM from.

Source: Local File System

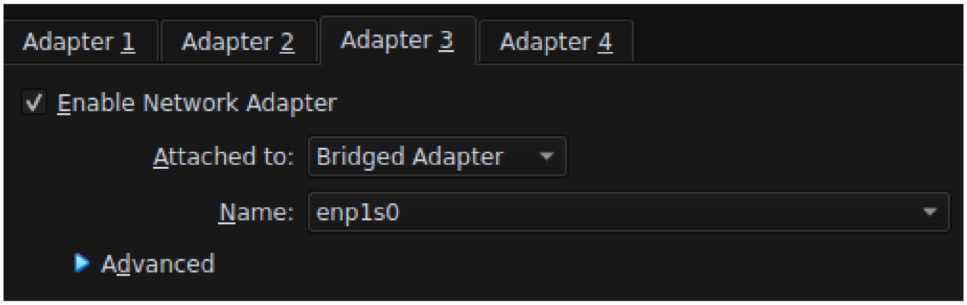
Please choose a file to import the virtual appliance from. VirtualBox currently supports importing appliances saved in the Open Virtualization Format (OVF). To continue, select the file to import below.

File: /home/papzi/Documents/gns3/GNS3 VM.ova

Figure 3. Importing the earlier downloaded .ova file.

Note. Make sure that you have allotted a good amount of RAM and CPU for the virtual machine. (Current Setup. 4GB RAM, 2 Cores CPU)

3. **Add a network interface card.**



☑ Enable Network Adapter

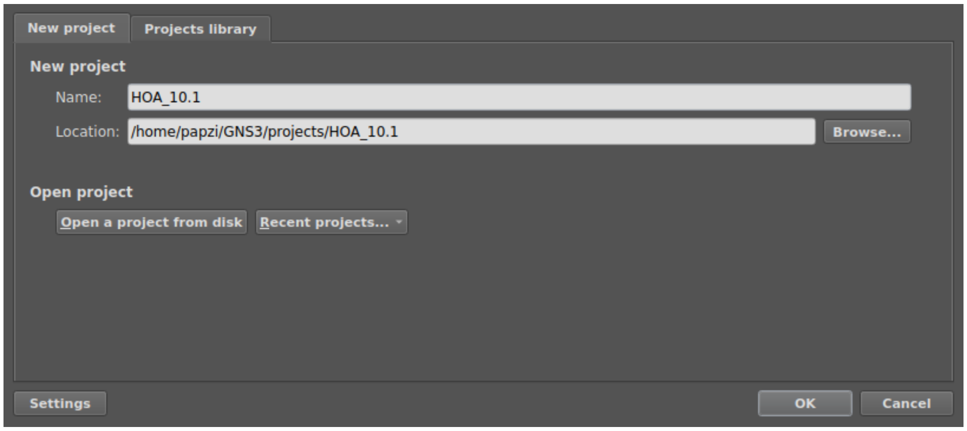
Attached to: Bridged Adapter

Name: enp1s0

Figure 4. Adding another adapter called Bridged Adapter.

Note. The bridge adapter is used to make sure that your VM is directly connected to your own LAN. It will be easier to connect the GNS3 network to my local machine using my own local network.

4. **Open GNS3. Create a Project.**



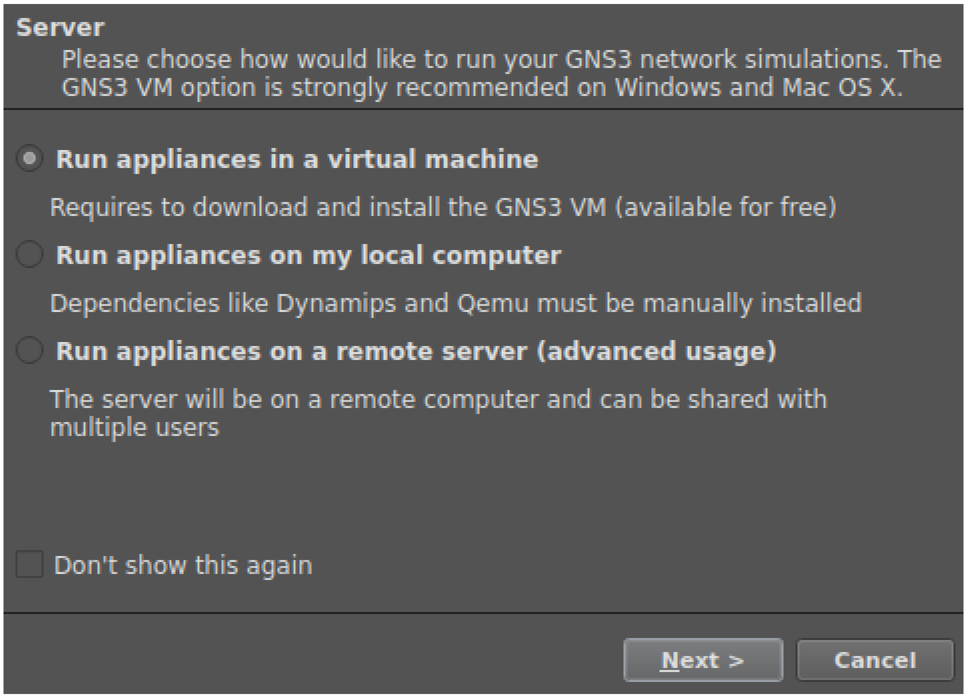
New project

Name: HOA_10.1

Location: /home/papzi/GNS3/projects/HOA_10.1

Figure 5. Creating a project.

5. **Go to Setup wizard. Help -> Setup Wizard.**



Server

Please choose how would like to run your GNS3 network simulations. The GNS3 VM option is strongly recommended on Windows and Mac OS X.

● **Run appliances in a virtual machine**

Requires to download and install the GNS3 VM (available for free)

○ **Run appliances on my local computer**

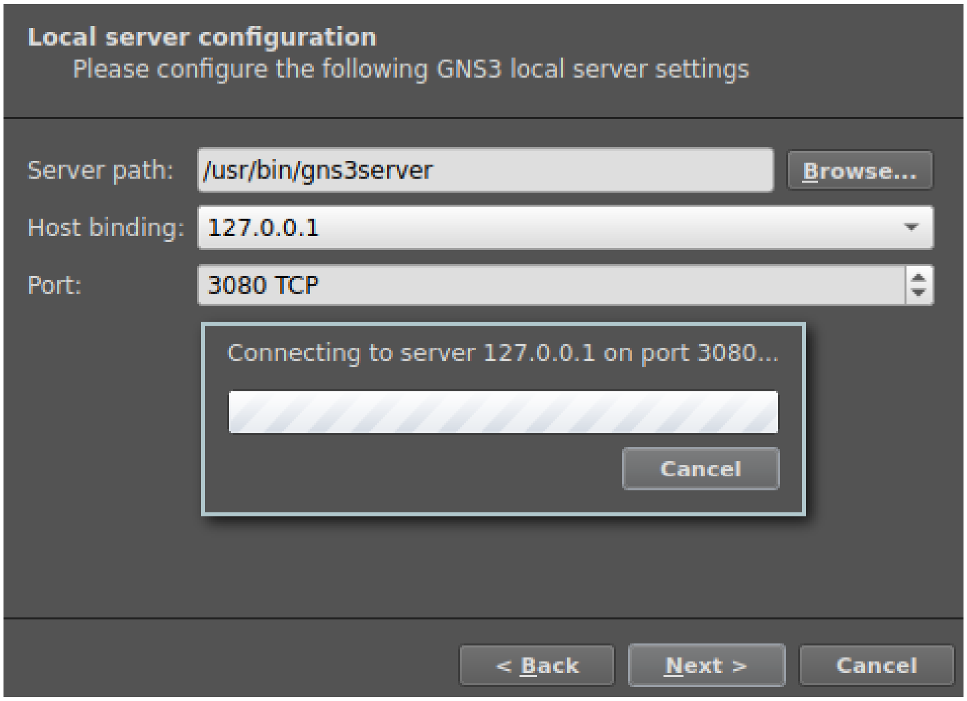
Dependencies like Dynamips and Qemu must be manually installed

○ **Run appliances on a remote server (advanced usage)**

The server will be on a remote computer and can be shared with multiple users

□ Don't show this again

Figure 6. Selecting "Run appliances in virtual machine".



Local server configuration

Please configure the following GNS3 local server settings

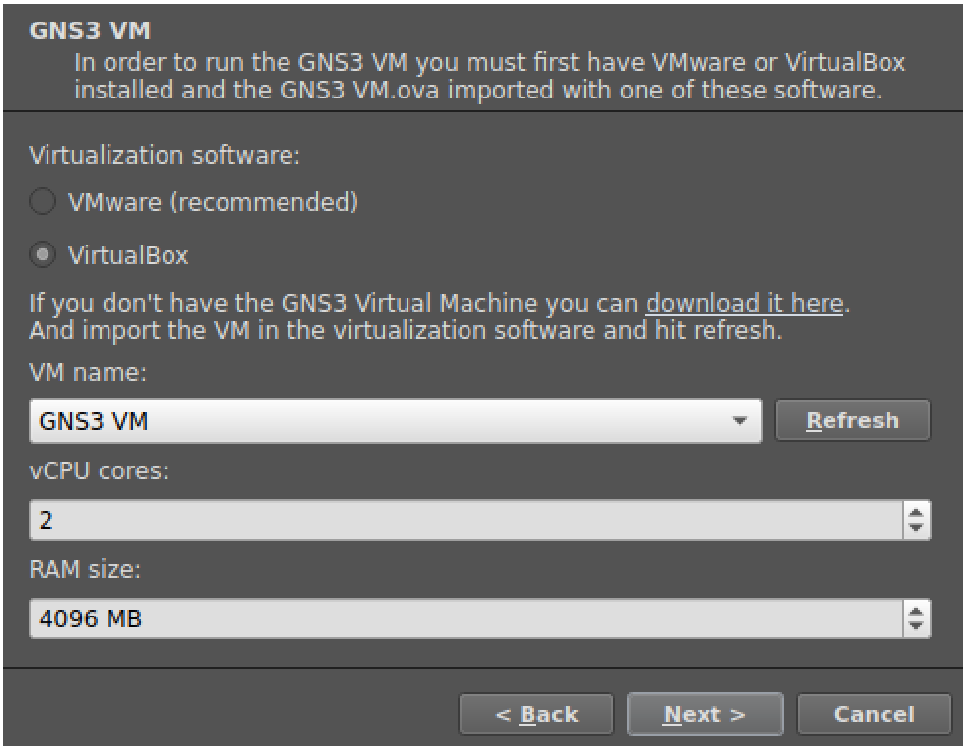
Server path: /usr/bin/gns3server

Host binding: 127.0.0.1

Port: 3080 TCP

Connecting to server 127.0.0.1 on port 3080...

Figure 7. Setting the local server configuration to default.



GNS3 VM

In order to run the GNS3 VM you must first have VMware or VirtualBox installed and the GNS3 VM.ova imported with one of these software.

Virtualization software:

○ VMware (recommended)

● VirtualBox

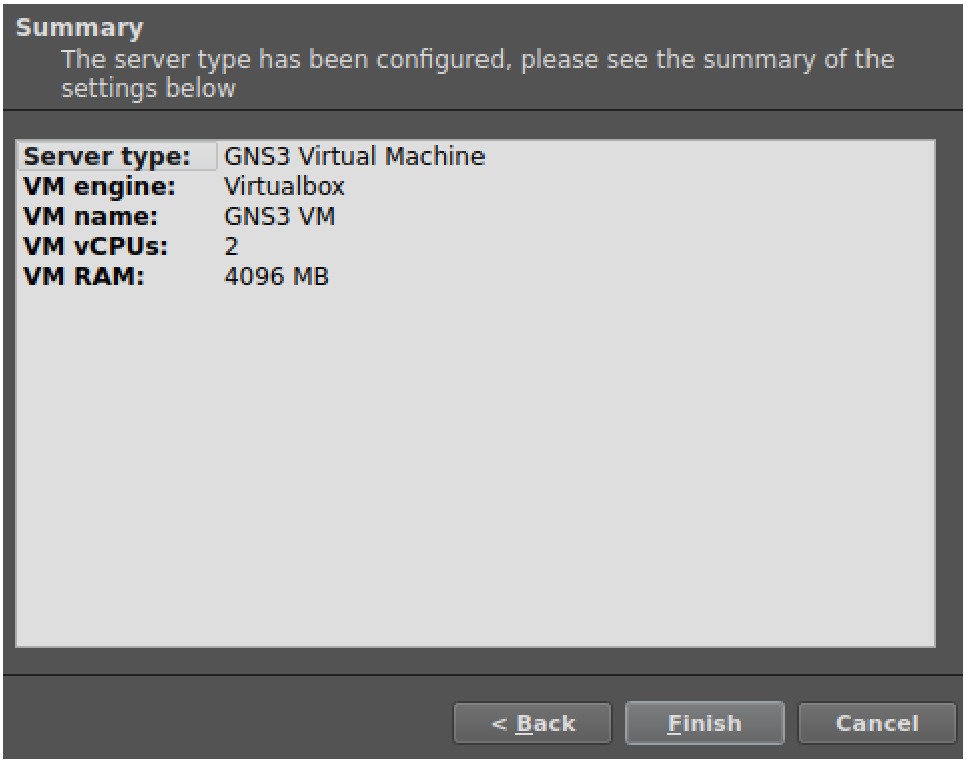
If you don't have the GNS3 Virtual Machine you can download it here. And import the VM in the virtualization software and hit refresh.

VM name: GNS3 VM

vCPU cores: 2

RAM size: 4096 MB

Figure 8. Selecting the VM that is named GNS3 VM which the recently installed step 2. Configuring the RAM of 4096 and setting it at 2 Cores allotted for the VM.

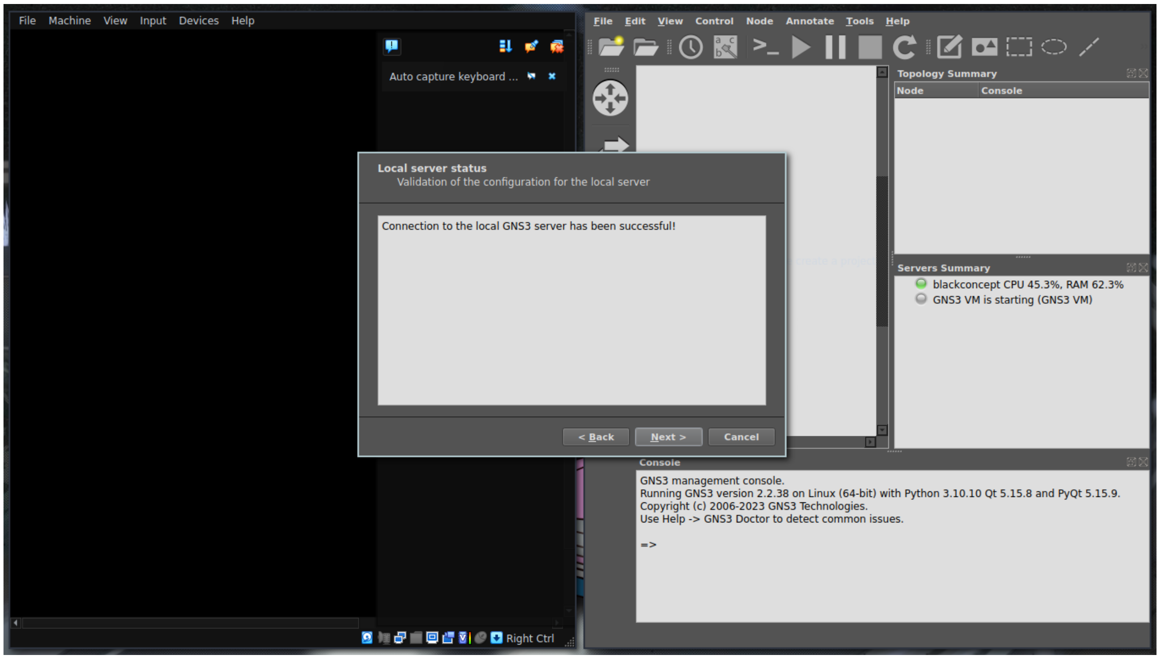


Summary

The server type has been configured, please see the summary of the settings below

Server type:	GNS3 Virtual Machine
VM engine:	Virtualbox
VM name:	GNS3 VM
VM vCPUs:	2
VM RAM:	4096 MB

Figure 9. Confirming the inputted configurations.


File Machine View Input Devices Help
Auto capture keyboard ...
Right Ctrl
File Edit View Control Node Annotate Tools Help
Topology Summary
Node
Console
< Back
Next >
Cancel

Local server status
Validation of the configuration for the local server

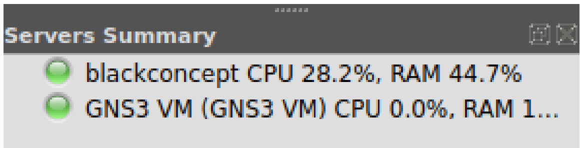
Connection to the local GNS3 server has been successful!

Servers Summary
blackconcept CPU 45.3%, RAM 62.3%
GNS3 VM is starting (GNS3 VM)

Console

```
GNS3 management console.
Running GNS3 version 2.2.38 on Linux (64-bit) with Python 3.10.10 Qt 5.15.8 and PyQt 5.15.9.
Copyright (c) 2006-2023 GNS3 Technologies.
Use Help -> GNS3 Doctor to detect common issues.

=>
```

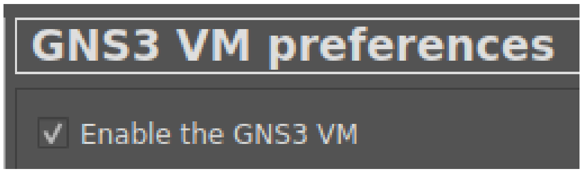
Figure 10. If successful the virtual machine will automatically launch as seen in the image above.



Servers Summary
blackconcept CPU 28.2%, RAM 44.7%
GNS3 VM (GNS3 VM) CPU 0.0%, RAM 1...

Figure 11. Make sure the GNS3 VM is in green.

6. **Go to Edit -> Preferences -> GNS3 VM.**



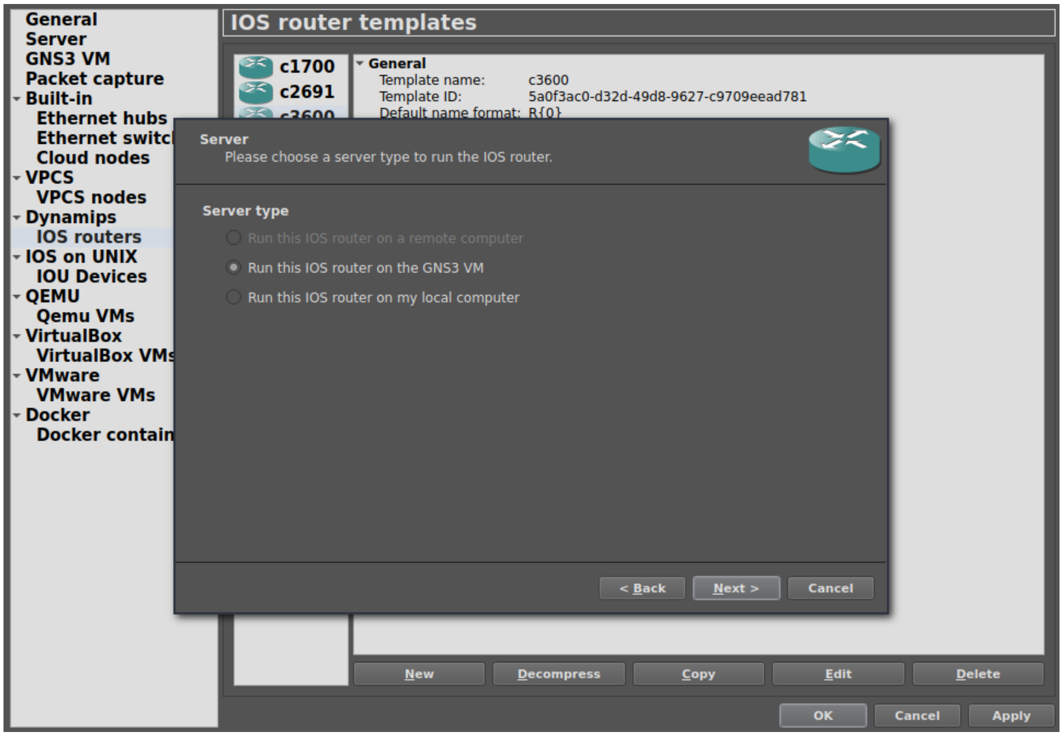
GNS3 VM preferences

☑ Enable the GNS3 VM

Figure 12. Enabling GNS3 VM.

7. **Download the [c7200-a3jk9s-mz.124-25g.bin](c7200-a3jk9s-mz.124-25g.bin).**

8. **To add an IOS router, go to Edit -> Preferences -> Dynamips -> IOS routers then click New.**

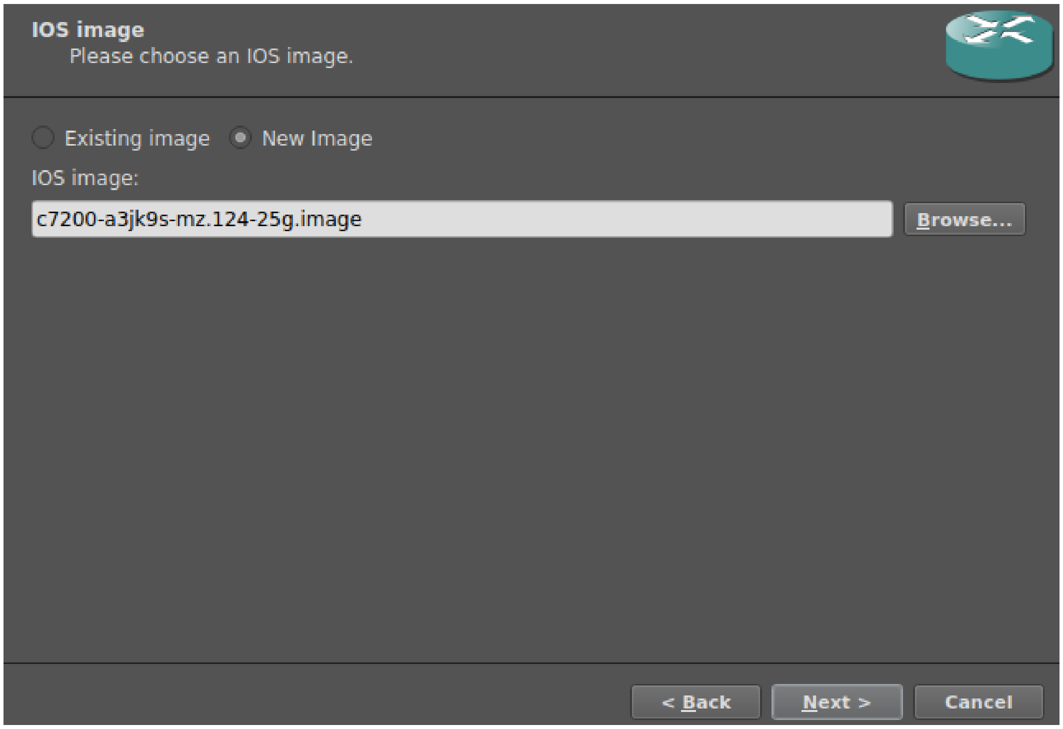


General
Template name: c3600
Template ID: 5a0f3ac0-d32d-49d8-9627-c9709eead781
Default name format: R{0}

Server
Please choose a server type to run the IOS router.

Server type
○ Run this IOS router on a remote computer
● Run this IOS router on the GNS3 VM
○ Run this IOS router on my local computer

Figure 13. Selecting the first available option because we are running the IOS under the VM.

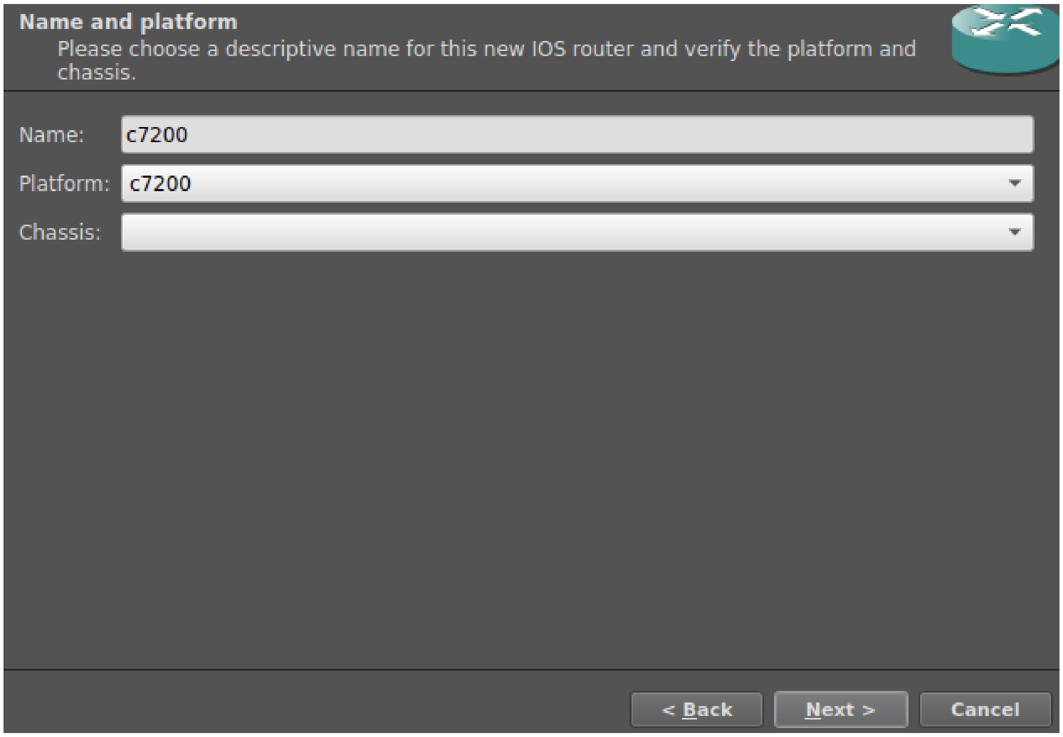


IOS image
Please choose an IOS image.

○ Existing image ● New Image

IOS image: c7200-a3jk9s-mz.124-25g.image

Figure 14. Browsing the downloaded image and hitting next.



Name and platform

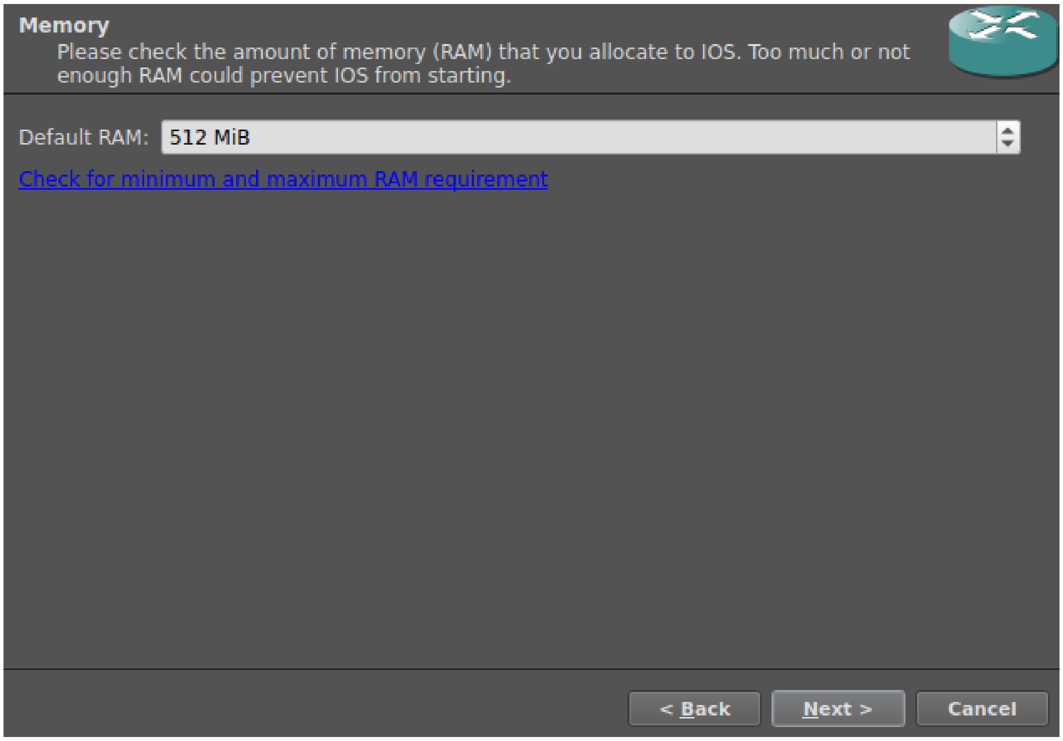
Please choose a descriptive name for this new IOS router and verify the platform and chassis.

Name: c7200

Platform: c7200

Chassis:

Figure 15. Inputting the name of the router.



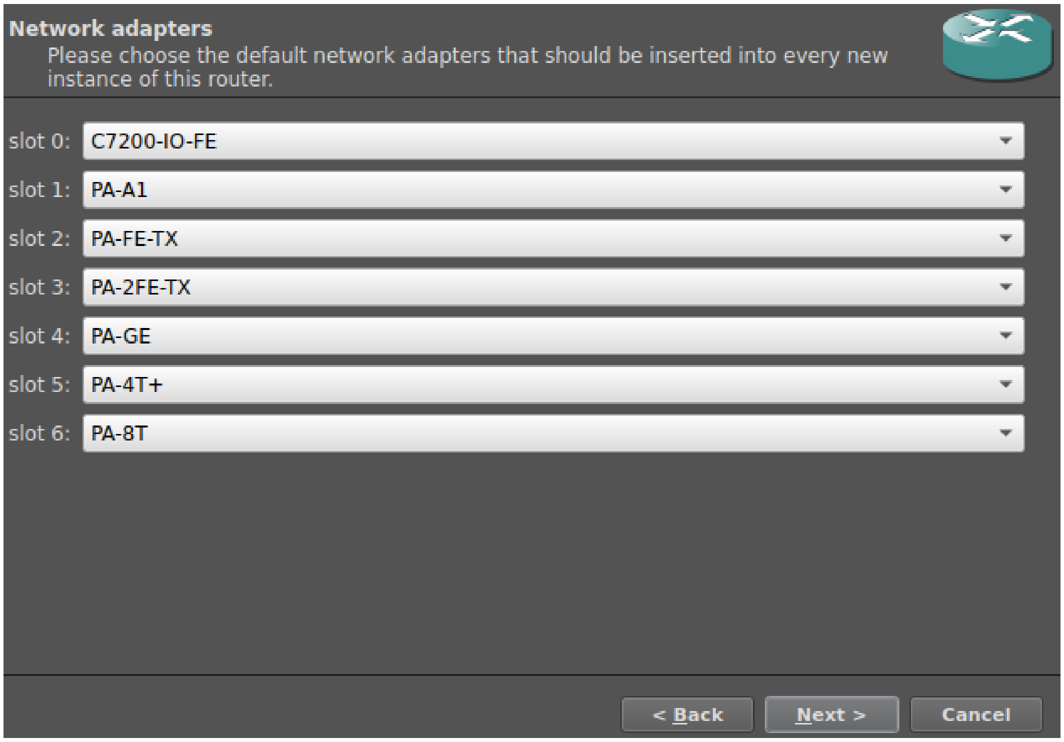
Memory

Please check the amount of memory (RAM) that you allocate to IOS. Too much or not enough RAM could prevent IOS from starting.

Default RAM: 512 MiB

Check for minimum and maximum RAM requirement

Figure 16. Setting the RAM as default.



Network adapters

Please choose the default network adapters that should be inserted into every new instance of this router.

slot 0: C7200-IO-FE

slot 1: PA-A1

slot 2: PA-FE-TX

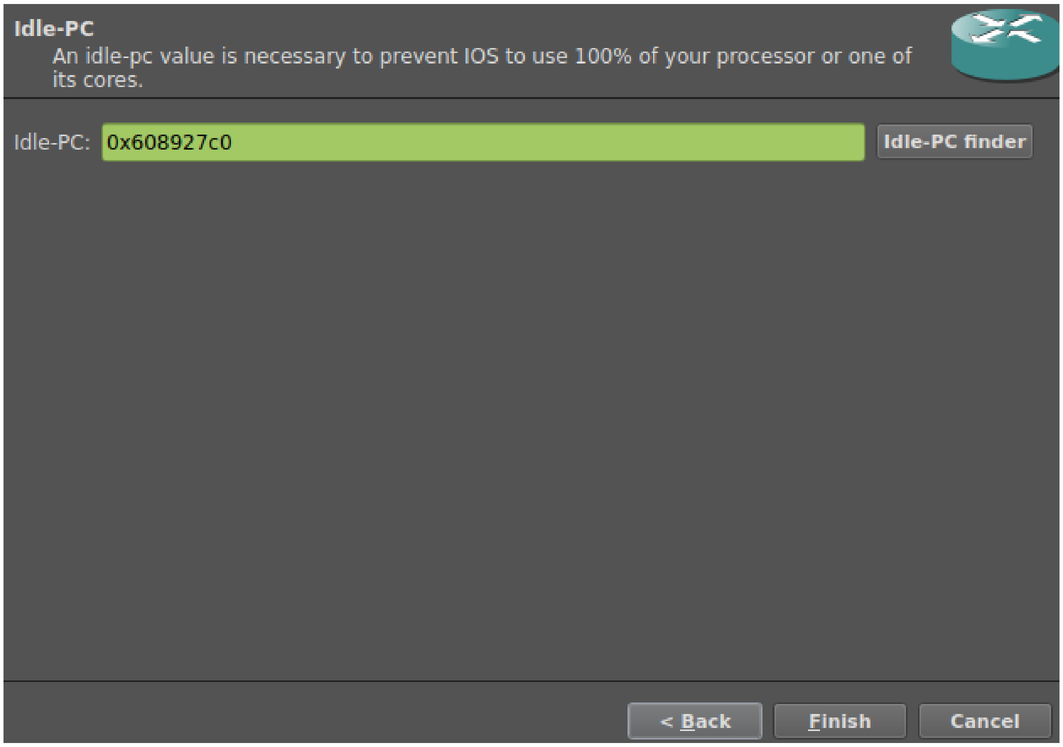
slot 3: PA-2FE-TX

slot 4: PA-GE

slot 5: PA-4T+

slot 6: PA-8T

Figure 17. Adding some network adapters.



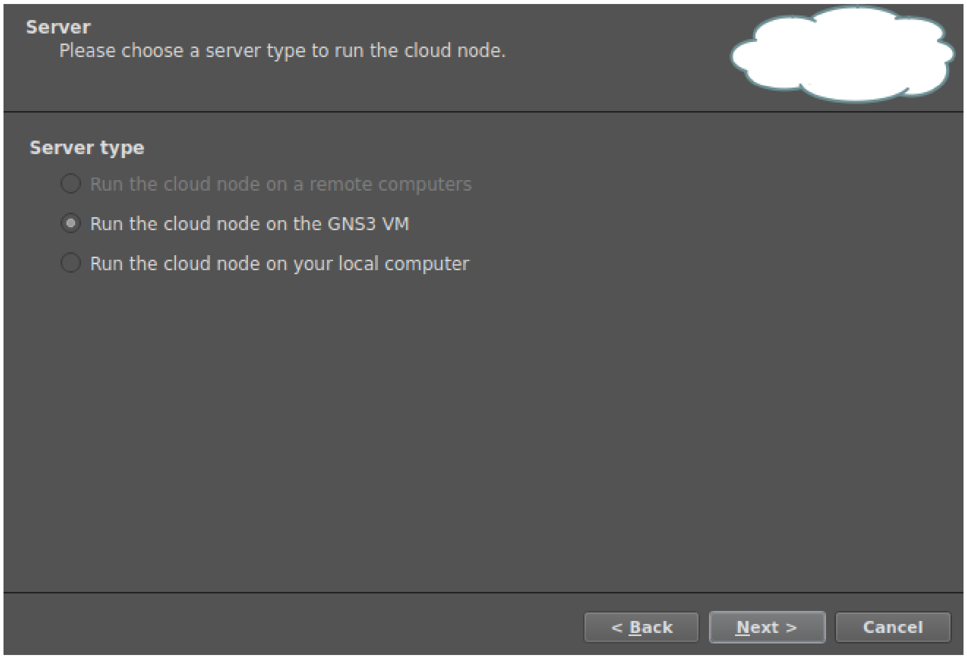
Idle-PC

An idle-pc value is necessary to prevent IOS to use 100% of your processor or one of its cores.

Idle-PC: 0x608927c0

Figure 18. Clicking the Idle-PC finder.

9. **Creating cloud notes. Go to Edit -> Preference -> Built-in -> Cloud nodes and select New. Click Apply then Ok if you are done configuring the cloud.**



Server

Please choose a server type to run the cloud node.

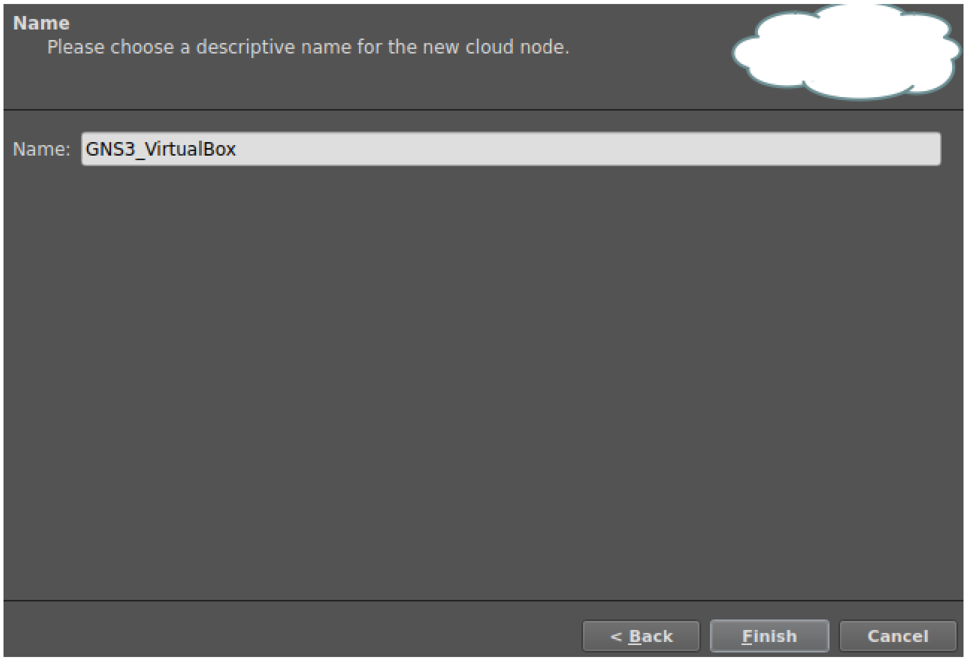
Server type

○ Run the cloud node on a remote computers

● Run the cloud node on the GNS3 VM

○ Run the cloud node on your local computer

Figure 19. Selecting the first available option because we are running the IOS under the VM.



Name

Please choose a descriptive name for the new cloud node.

Name: GNS3_VirtualBox

Figure 20. Configuring the name of the cloud.

10. **Creating the topology. Put the c7200 router to the canvas and the GNS3_VirtualBox cloud. Connect them using ethernet cable. Click the Play button to start the router.**

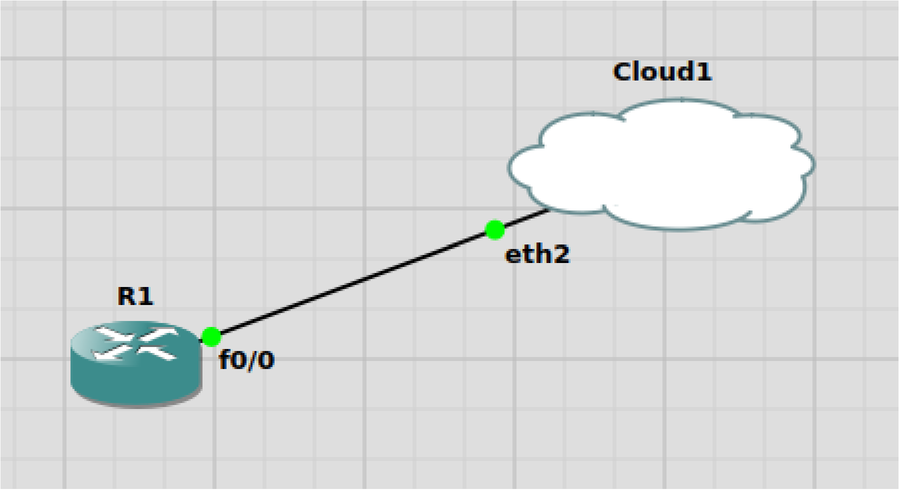
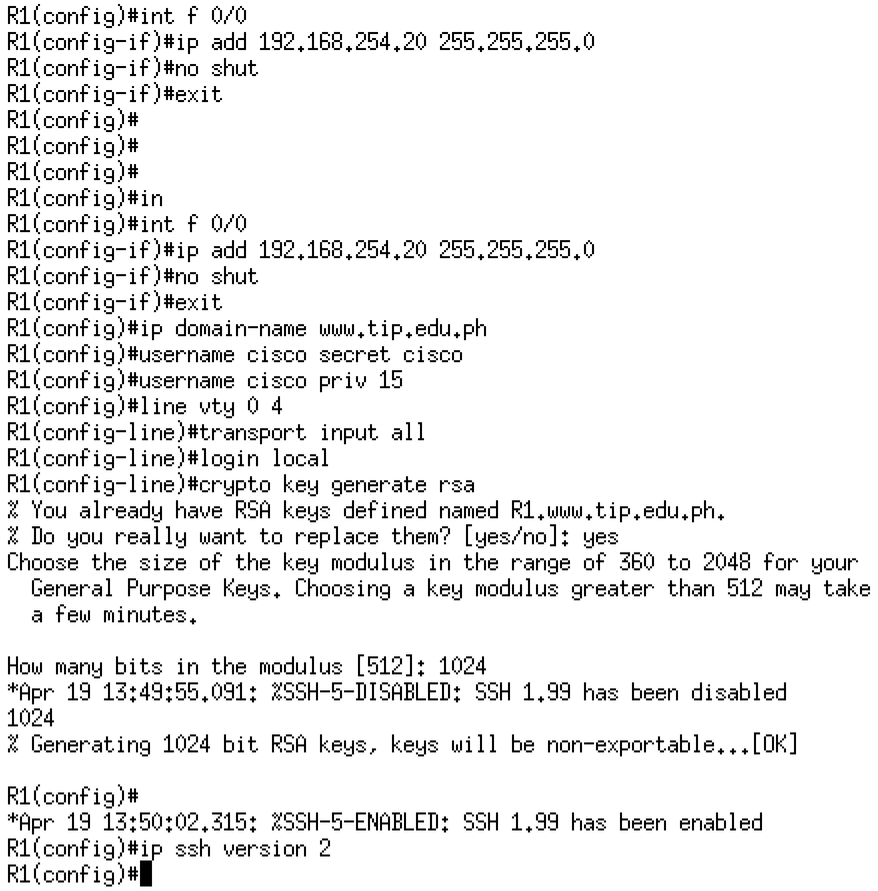


Figure 21. Connecting the cloud and the router.

Note. Make sure to connect the cable into the bridge adapter of the cloud. In my case my bridge adapter is eth2.

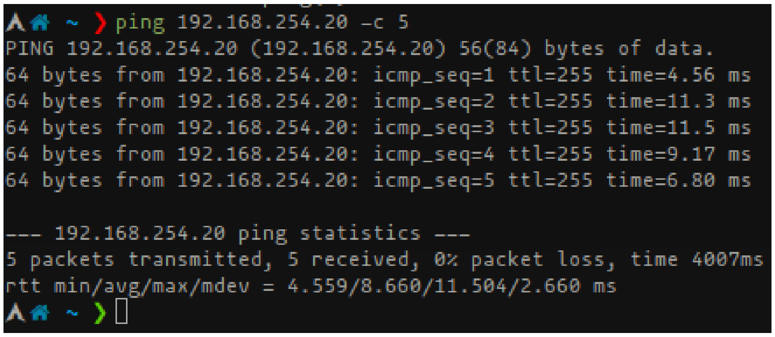
11. **Adding basic configuration to the router to make sure that we can use ssh between router and the workstation. Right click the router and select console then configure the router using the configuration below.**

```
enable
configure terminal
int f 0/0
ip address 192.168.254.20 255.255.255.0
no shut
ip domain-name www.tip.edu.ph
username cisco secret cisco
username cisco privilege 15
line vty 0 4
transport input all
login local
crypto key generate rsa
ip ssh version 2
```



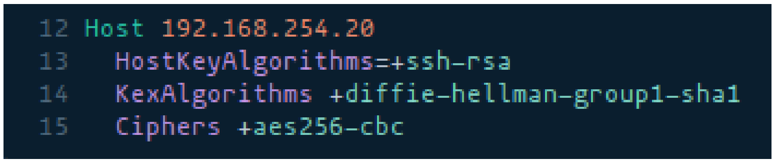
```
R1(config)#int f 0/0
R1(config-if)#ip add 192.168.254.20 255.255.255.0
R1(config-if)#no shut
R1(config-if)#exit
R1(config)#
R1(config)#
R1(config)#
R1(config)#in
R1(config)#int f 0/0
R1(config-if)#ip add 192.168.254.20 255.255.255.0
R1(config-if)#no shut
R1(config-if)#exit
R1(config)#ip domain-name www.tip.edu.ph
R1(config)#username cisco secret cisco
R1(config)#username cisco priv 15
R1(config)#line vty 0 4
R1(config-line)#transport input all
R1(config-line)#login local
R1(config-line)#crypto key generate rsa
% You already have RSA keys defined named R1.www.tip.edu.ph.
% Do you really want to replace them? [yes/no]: yes
Choose the size of the key modulus in the range of 360 to 2048 for your
  General Purpose Keys. Choosing a key modulus greater than 512 may take
  a few minutes.

How many bits in the modulus [512]: 1024
*Apr 19 13:49:55.091: %SSH-5-DISABLED: SSH 1.99 has been disabled
1024
% Generating 1024 bit RSA keys, keys will be non-exportable...[OK]

R1(config)#
*Apr 19 13:50:02.315: %SSH-5-ENABLED: SSH 1.99 has been enabled
R1(config)#ip ssh version 2
R1(config)#
```

Figure 22. Pre configuring the router to support ssh.



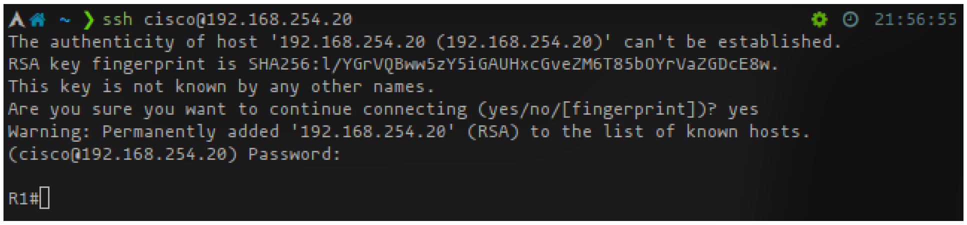
```
⋀ 🏠 ~ ❯ ping 192.168.254.20 -c 5
PING 192.168.254.20 (192.168.254.20) 56(84) bytes of data.
64 bytes from 192.168.254.20: icmp_seq=1 ttl=255 time=4.56 ms
64 bytes from 192.168.254.20: icmp_seq=2 ttl=255 time=11.3 ms
64 bytes from 192.168.254.20: icmp_seq=3 ttl=255 time=11.5 ms
64 bytes from 192.168.254.20: icmp_seq=4 ttl=255 time=9.17 ms
64 bytes from 192.168.254.20: icmp_seq=5 ttl=255 time=6.80 ms

--- 192.168.254.20 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4007ms
rtt min/avg/max/mdev = 4.559/8.660/11.504/2.660 ms
⋀ 🏠 ~ ❯
```

Figure 23. Pinging the router using the workstation.



```
Host 192.168.254.20
  HostKeyAlgorithms=+ssh-rsa
  KexAlgorithms +diffie-hellman-group1-sha1
  Ciphers +aes256-cbc
```

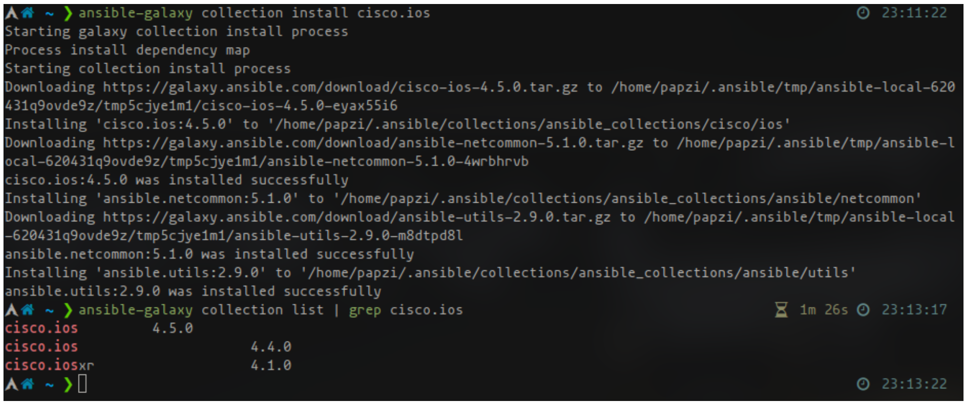
Figure 24. Add this lines under ~/.ssh/config



```
~ ❯ ssh cisco@192.168.254.20                                          21:56:55
The authenticity of host '192.168.254.20 (192.168.254.20)' can't be established.
RSA key fingerprint is SHA256:l/YGrVQBww5zY5iGAUHxcGveZM6T85bOYrVaZGDcE8w.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.254.20' (RSA) to the list of known hosts.
(cisco@192.168.254.20) Password:

R1#
```

Figure 25. Testing ssh to the router.

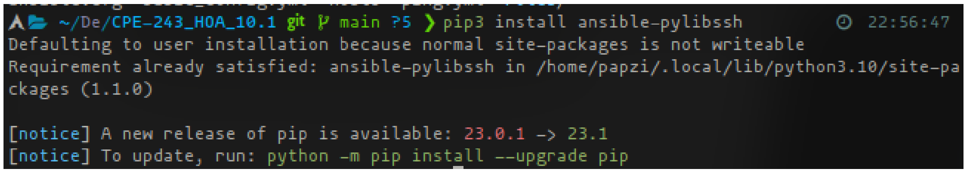
12. **Create a basic configuration for the Cisco router using the Ansible IOS module (check the reference for Cisco Router basic configuration)**



```
~ ❯ ansible-galaxy collection install cisco.ios                                              23:11:22
Starting galaxy collection install process
Process install dependency map
Starting collection install process
Downloading https://galaxy.ansible.com/download/cisco-ios-4.5.0.tar.gz to /home/papzi/.ansible/tmp/ansible-local-620
431q9ovde9z/tmp5cjye1m1/cisco-ios-4.5.0-eyax55i6
Installing 'cisco.ios:4.5.0' to '/home/papzi/.ansible/collections/ansible_collections/cisco/ios'
Downloading https://galaxy.ansible.com/download/ansible-netcommon-5.1.0.tar.gz to /home/papzi/.ansible/tmp/ansible-l
ocal-620431q9ovde9z/tmp5cjye1m1/ansible-netcommon-5.1.0-4wrbhrvb
cisco.ios:4.5.0 was installed successfully
Installing 'ansible.netcommon:5.1.0' to '/home/papzi/.ansible/collections/ansible_collections/ansible/netcommon'
Downloading https://galaxy.ansible.com/download/ansible-utils-2.9.0.tar.gz to /home/papzi/.ansible/tmp/ansible-local
-620431q9ovde9z/tmp5cjye1m1/ansible-utils-2.9.0-m8dtpd8l
ansible.netcommon:5.1.0 was installed successfully
Installing 'ansible.utils:2.9.0' to '/home/papzi/.ansible/collections/ansible_collections/ansible/utils'
ansible.utils:2.9.0 was installed successfully
~ ❯ ansible-galaxy collection list | grep cisco.ios                                 1m 26s   23:13:17
cisco.ios        4.5.0
cisco.ios                   4.4.0
cisco.iosxr                 4.1.0
~ ❯                                                                                          23:13:22
```

Figure 26. Downloading Cisco IOS module.

ansible-galaxy collection install cisco.ios

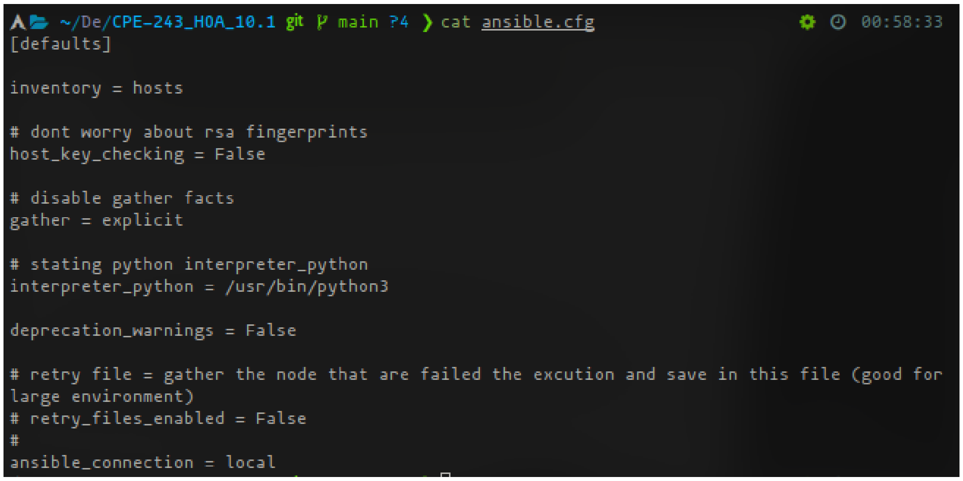


```
~/De/CPE-243_HOA_10.1 git  main ?5 ❯ pip3 install ansible-pylibssh          22:56:47
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: ansible-pylibssh in /home/papzi/.local/lib/python3.10/site-pa
ckages (1.1.0)

[notice] A new release of pip is available: 23.0.1 -> 23.1
[notice] To update, run: python -m pip install --upgrade pip
```

Figure 27. Installing “ansible-pylibssh”.

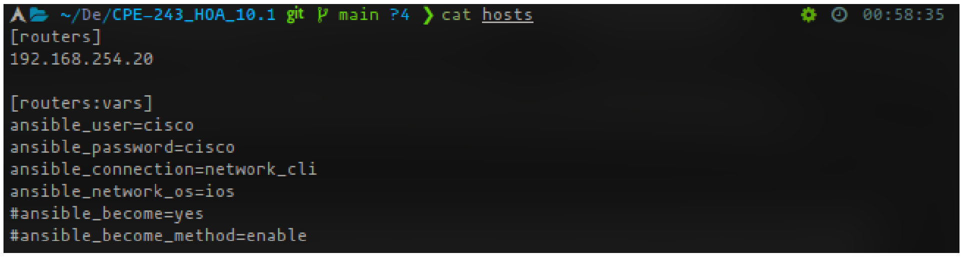
CODES	
Filename	**Screenshot**

ansible.cfg



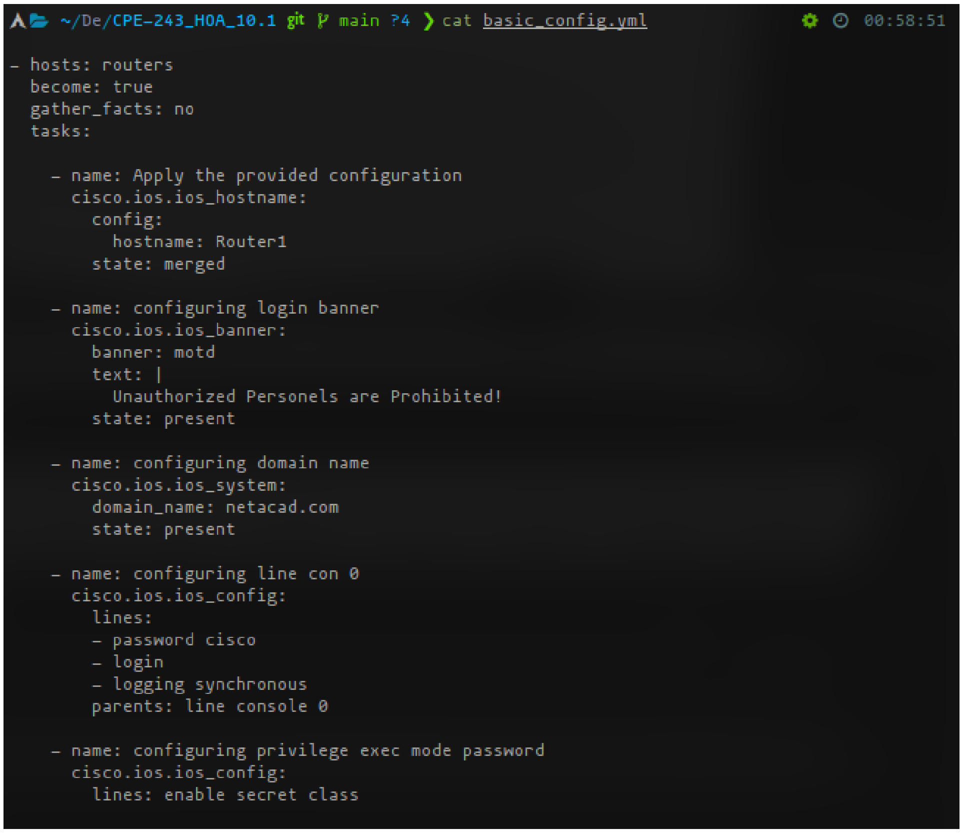
```
~/De/CPE-243_HOA_10.1 git main ?4 ❯ cat ansible.cfg                                    00:58:33
[defaults]

inventory = hosts

# dont worry about rsa fingerprints
host_key_checking = False

# disable gather facts
gather = explicit

# stating python interpreter_python
interpreter_python = /usr/bin/python3

deprecation_warnings = False

# retry file = gather the node that are failed the excution and save in this file (good for
large environment)
# retry_files_enabled = False
#
ansible_connection = local
```

hosts/invetory

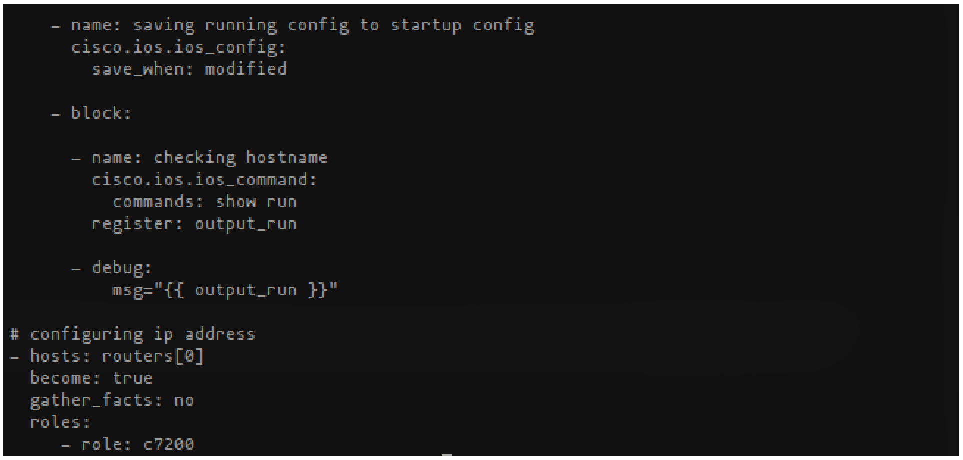


```
~/De/CPE-243_HOA_10.1 git main ?4 ❯ cat hosts                                          00:58:35
[routers]
192.168.254.20

[routers:vars]
ansible_user=cisco
ansible_password=cisco
ansible_connection=network_cli
ansible_network_os=ios
#ansible_become=yes
#ansible_become_method=enable
```

basig_config.yml

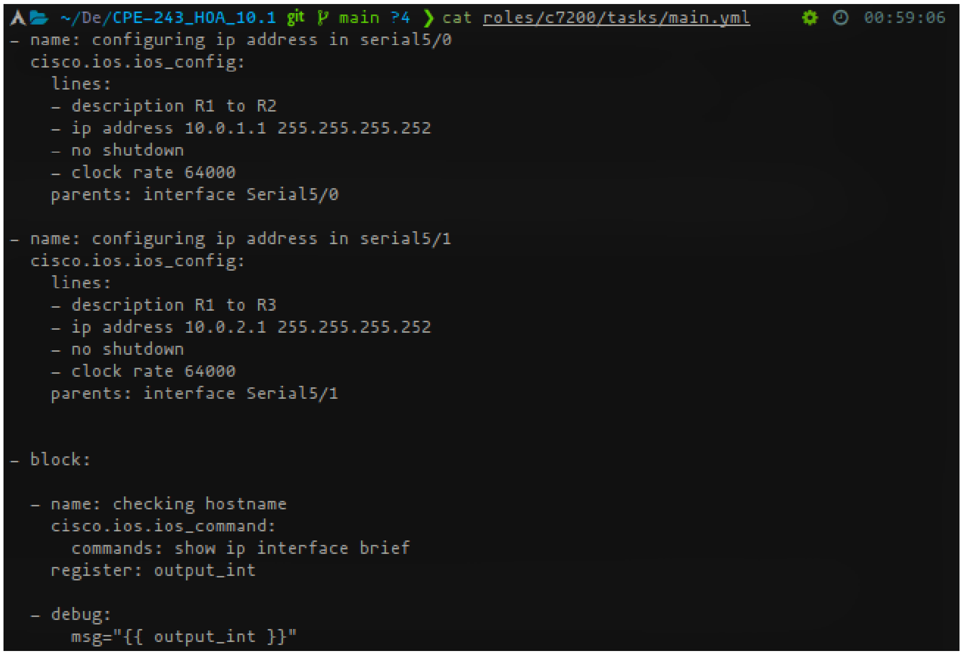


```
~/De/CPE-243_HOA_10.1 git main ?4 ❯ cat basic_config.yml                               00:58:51

- hosts: routers
  become: true
  gather_facts: no
  tasks:

    - name: Apply the provided configuration
      cisco.ios.ios_hostname:
        config:
          hostname: Router1
        state: merged

    - name: configuring login banner
      cisco.ios.ios_banner:
        banner: motd
        text: |
          Unauthorized Personels are Prohibited!
        state: present

    - name: configuring domain name
      cisco.ios.ios_system:
        domain_name: netacad.com
        state: present

    - name: configuring line con 0
      cisco.ios.ios_config:
        lines:
        - password cisco
        - login
        - logging synchronous
        parents: line console 0

    - name: configuring privilege exec mode password
      cisco.ios.ios_config:
        lines: enable secret class
```



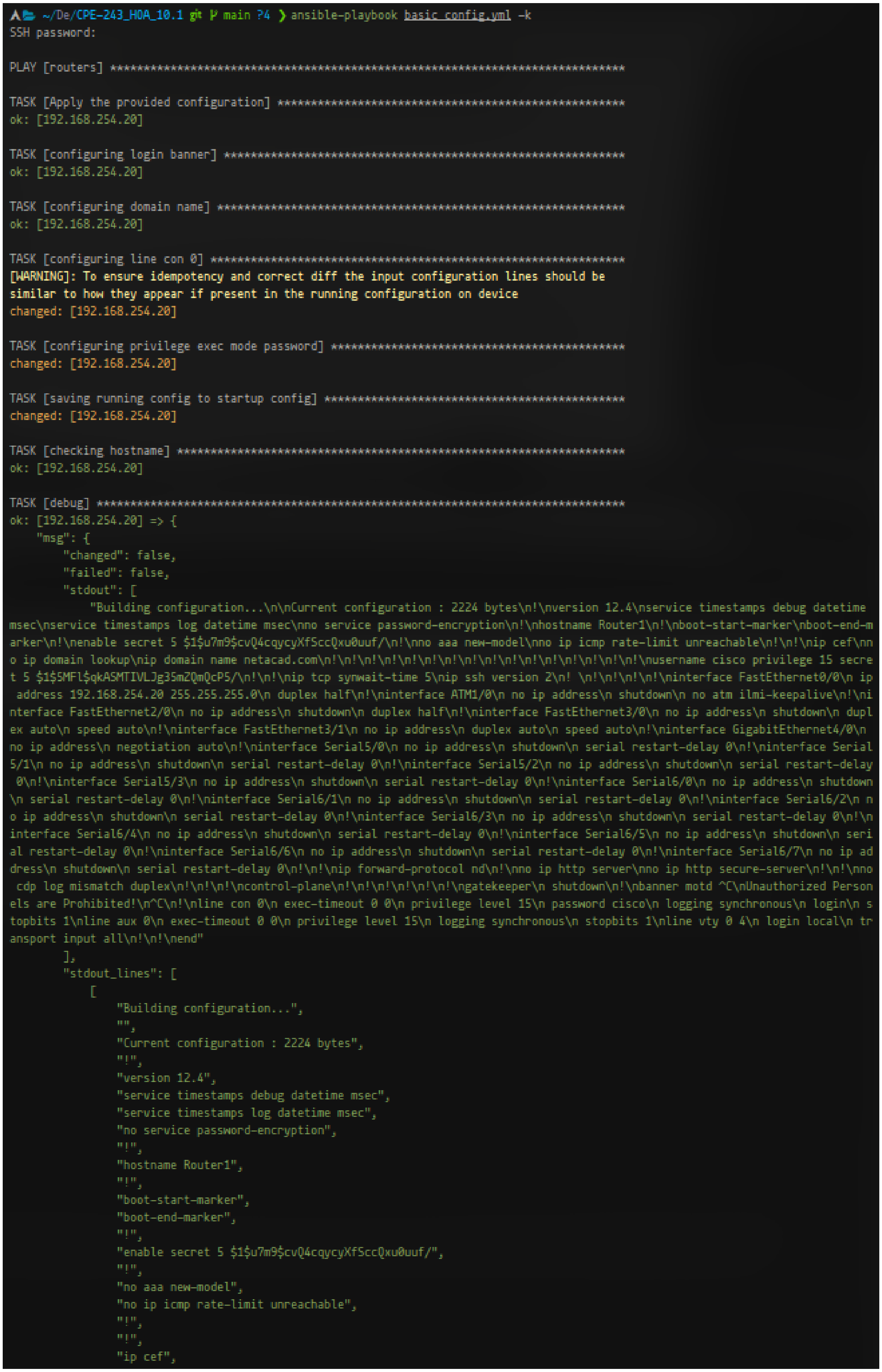
```
    - name: saving running config to startup config
      cisco.ios.ios_config:
        save_when: modified

    - block:

      - name: checking hostname
        cisco.ios.ios_command:
          commands: show run
        register: output_run

      - debug:
          msg="{{ output_run }}"

# configuring ip address
- hosts: routers[0]
  become: true
  gather_facts: no
  roles:
     - role: c7200
```

/roles/c7200/tasks/main.yml

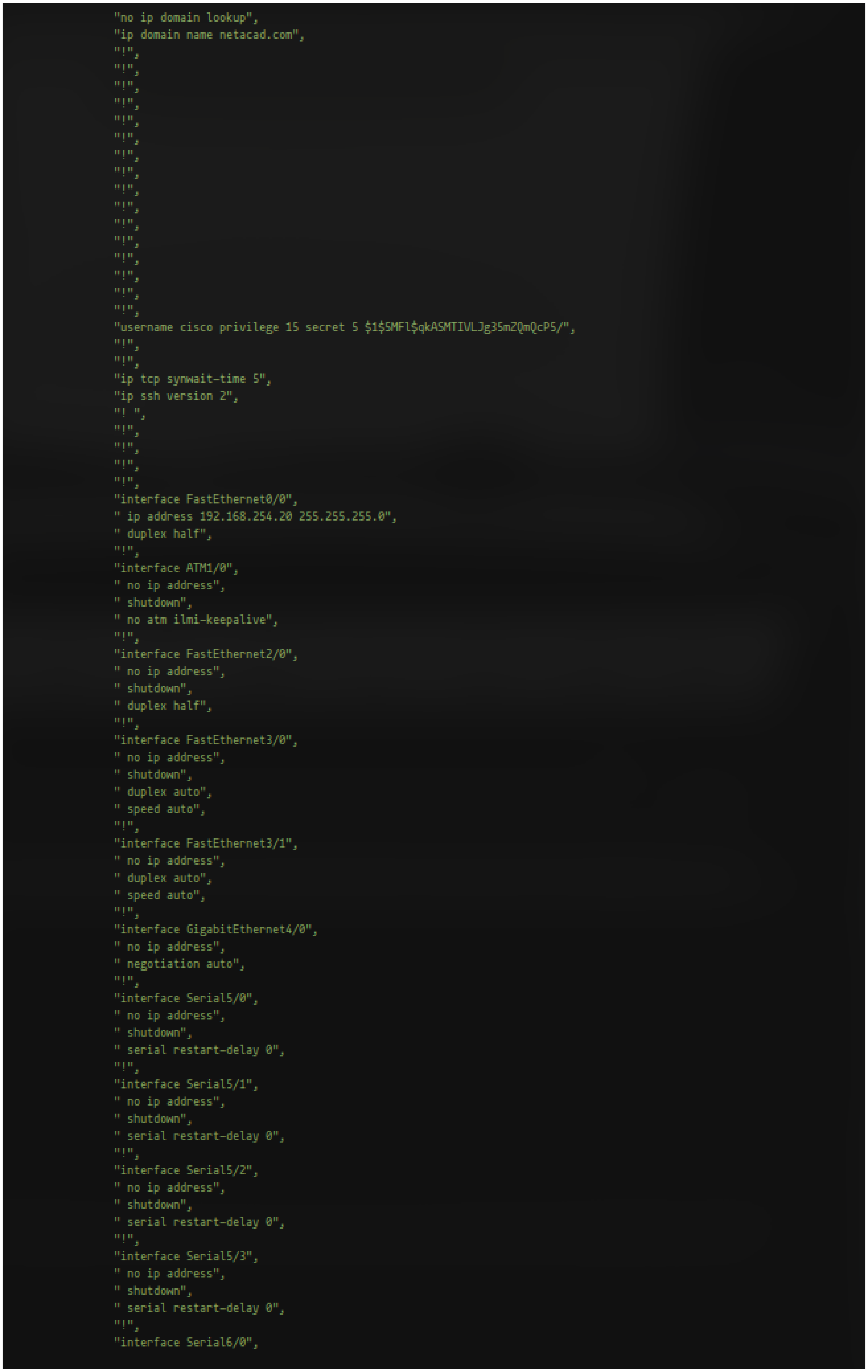


```
~/De/CPE-243_HOA_10.1 git main ?4 ❯ cat roles/c7200/tasks/main.yml      00:59:06
- name: configuring ip address in serial5/0
  cisco.ios.ios_config:
    lines:
    - description R1 to R2
    - ip address 10.0.1.1 255.255.255.252
    - no shutdown
    - clock rate 64000
    parents: interface Serial5/0

- name: configuring ip address in serial5/1
  cisco.ios.ios_config:
    lines:
    - description R1 to R3
    - ip address 10.0.2.1 255.255.255.252
    - no shutdown
    - clock rate 64000
    parents: interface Serial5/1


- block:

  - name: checking hostname
    cisco.ios.ios_command:
      commands: show ip interface brief
    register: output_int

  - debug:
      msg="{{ output_int }}"
```

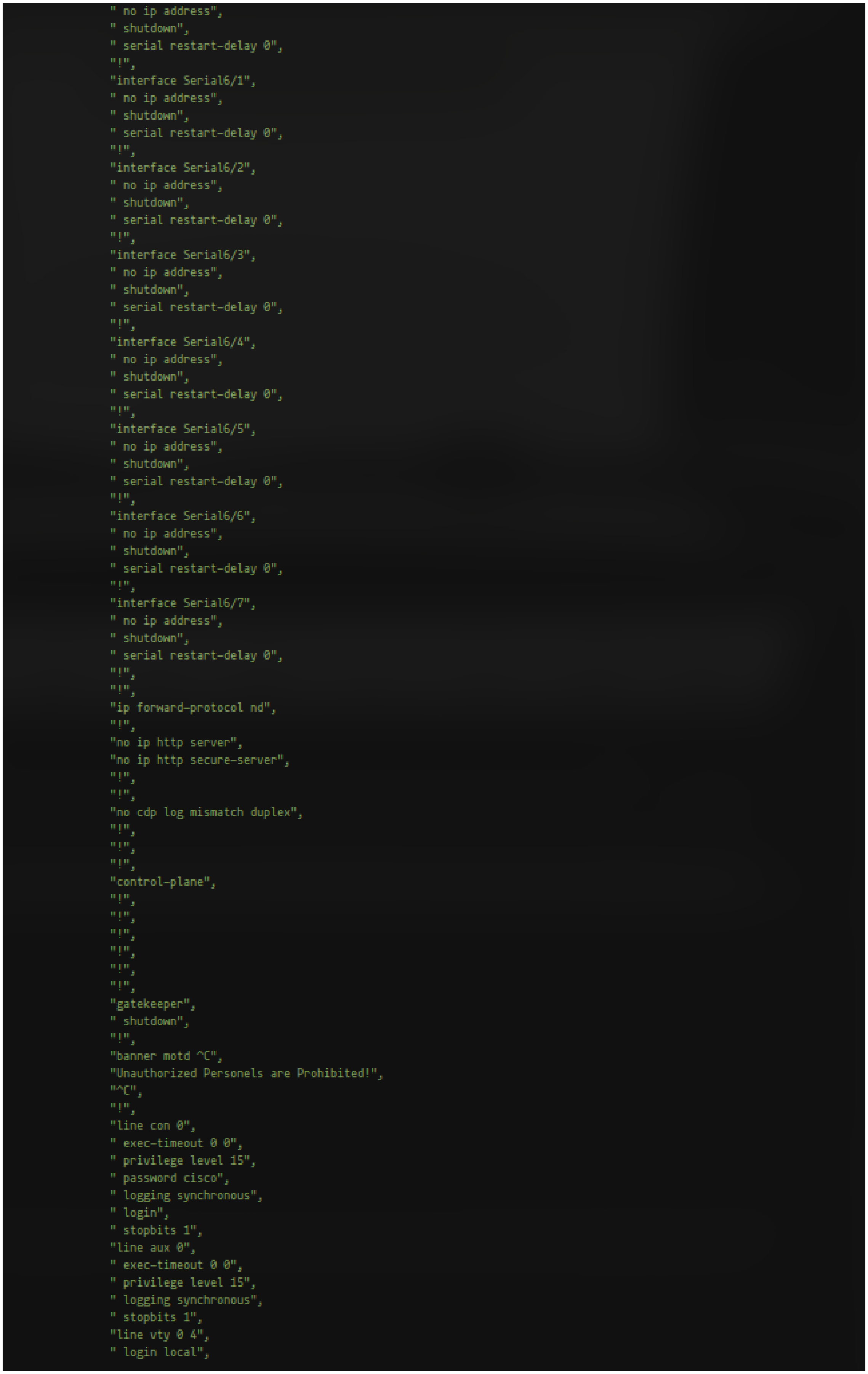
13. Show evidence of the playbook running successfully and verify the configuration done on the Cisco router (this should be seen in the managed node).



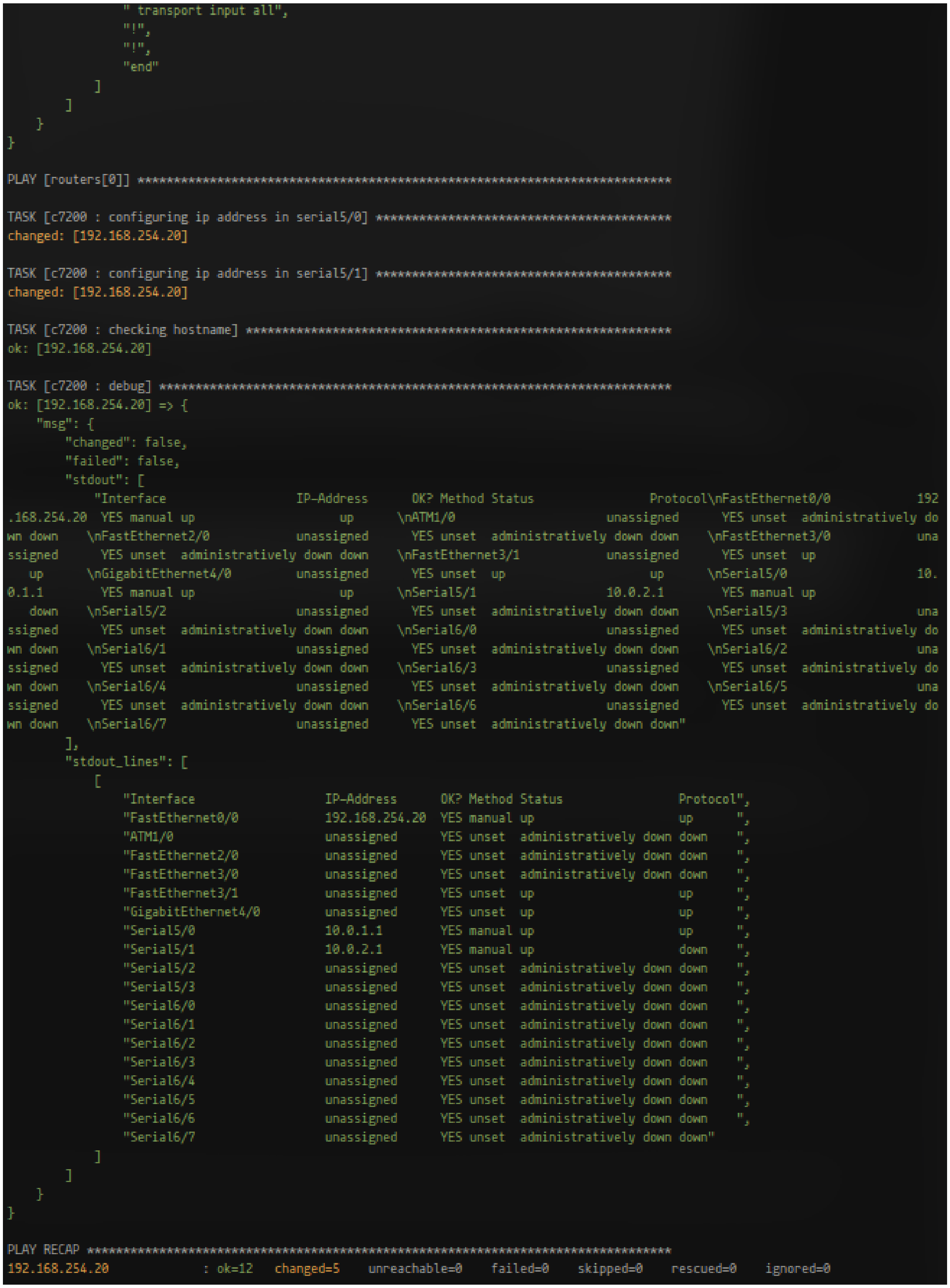
```
⅄ ~/De/CPE-243_HOA_10.1 git  main ?4 ❯ ansible-playbook basic_config.yml -k
SSH password:

PLAY [routers] ****************************************************************************

TASK [Apply the provided configuration] ***************************************************
ok: [192.168.254.20]

TASK [configuring login banner] ***********************************************************
ok: [192.168.254.20]

TASK [configuring domain name] ************************************************************
ok: [192.168.254.20]

TASK [configuring line con 0] *************************************************************
[WARNING]: To ensure idempotency and correct diff the input configuration lines should be
similar to how they appear if present in the running configuration on device
changed: [192.168.254.20]

TASK [configuring privilege exec mode password] *******************************************
changed: [192.168.254.20]

TASK [saving running config to startup config] ********************************************
changed: [192.168.254.20]

TASK [checking hostname] ******************************************************************
ok: [192.168.254.20]

TASK [debug] ******************************************************************************
ok: [192.168.254.20] => {
    "msg": {
        "changed": false,
        "failed": false,
        "stdout": [
            "Building configuration...\n\nCurrent configuration : 2224 bytes\n!\nversion 12.4\nservice timestamps debug datetime
msec\nservice timestamps log datetime msec\nno service password-encryption\n!\nhostname Router1\n!\nboot-start-marker\nboot-end-m
arker\n!\nenable secret 5 $1$u7m9$cvQ4cqycyXfSccQxu0uuf/\n!\nno aaa new-model\nno ip icmp rate-limit unreachable\n!\n!\nip cef\nn
o ip domain lookup\nip domain name netacad.com\n!\n!\n!\n!\n!\n!\n!\n!\n!\n!\n!\n!\n!\n!\n!\n!\nusername cisco privilege 15 secre
t 5 $1$5MFl$qkASMTIVLJg35mZQmQcP5/\n!\n!\nip tcp synwait-time 5\nip ssh version 2\n! \n!\n!\n!\n!\ninterface FastEthernet0/0\n ip
 address 192.168.254.20 255.255.255.0\n duplex half\n!\ninterface ATM1/0\n no ip address\n shutdown\n no atm ilmi-keepalive\n!\ni
nterface FastEthernet2/0\n no ip address\n shutdown\n duplex half\n!\ninterface FastEthernet3/0\n no ip address\n shutdown\n dupl
ex auto\n speed auto\n!\ninterface FastEthernet3/1\n no ip address\n duplex auto\n speed auto\n!\ninterface GigabitEthernet4/0\n
no ip address\n negotiation auto\n!\ninterface Serial5/0\n no ip address\n shutdown\n serial restart-delay 0\n!\ninterface Serial
5/1\n no ip address\n shutdown\n serial restart-delay 0\n!\ninterface Serial5/2\n no ip address\n shutdown\n serial restart-delay
 0\n!\ninterface Serial5/3\n no ip address\n shutdown\n serial restart-delay 0\n!\ninterface Serial6/0\n no ip address\n shutdown
\n serial restart-delay 0\n!\ninterface Serial6/1\n no ip address\n shutdown\n serial restart-delay 0\n!\ninterface Serial6/2\n n
o ip address\n shutdown\n serial restart-delay 0\n!\ninterface Serial6/3\n no ip address\n shutdown\n serial restart-delay 0\n!\n
interface Serial6/4\n no ip address\n shutdown\n serial restart-delay 0\n!\ninterface Serial6/5\n no ip address\n shutdown\n seri
al restart-delay 0\n!\ninterface Serial6/6\n no ip address\n shutdown\n serial restart-delay 0\n!\ninterface Serial6/7\n no ip ad
dress\n shutdown\n serial restart-delay 0\n!\n!\nip forward-protocol nd\n!\nno ip http server\nno ip http secure-server\n!\n!\nno
 cdp log mismatch duplex\n!\n!\n!\ncontrol-plane\n!\n!\n!\n!\n!\n!\ngatekeeper\n shutdown\n!\nbanner motd ^C\nUnauthorized Person
els are Prohibited!\n^C\n!\nline con 0\n exec-timeout 0 0\n privilege level 15\n password cisco\n logging synchronous\n login\n s
topbits 1\nline aux 0\n exec-timeout 0 0\n privilege level 15\n logging synchronous\n stopbits 1\nline vty 0 4\n login local\n tr
ansport input all\n!\n!\nend"
        ],
        "stdout_lines": [
            [
                "Building configuration...",
                "",
                "Current configuration : 2224 bytes",
                "!",
                "version 12.4",
                "service timestamps debug datetime msec",
                "service timestamps log datetime msec",
                "no service password-encryption",
                "!",
                "hostname Router1",
                "!",
                "boot-start-marker",
                "boot-end-marker",
                "!",
                "enable secret 5 $1$u7m9$cvQ4cqycyXfSccQxu0uuf/",
                "!",
                "no aaa new-model",
                "no ip icmp rate-limit unreachable",
                "!",
                "!",
                "ip cef",
```



```
"no ip domain lookup",
"ip domain name netacad.com",
"!",
"!",
"!",
"!",
"!",
"!",
"!",
"!",
"!",
"!",
"!",
"!",
"!",
"!",
"!",
"!",
"username cisco privilege 15 secret 5 $1$5MFl$qkASMTIVLJg35mZQmQcP5/",
"!",
"!",
"ip tcp synwait-time 5",
"ip ssh version 2",
"! ",
"!",
"!",
"!",
"!",
"interface FastEthernet0/0",
" ip address 192.168.254.20 255.255.255.0",
" duplex half",
"!",
"interface ATM1/0",
" no ip address",
" shutdown",
" no atm ilmi-keepalive",
"!",
"interface FastEthernet2/0",
" no ip address",
" shutdown",
" duplex half",
"!",
"interface FastEthernet3/0",
" no ip address",
" shutdown",
" duplex auto",
" speed auto",
"!",
"interface FastEthernet3/1",
" no ip address",
" duplex auto",
" speed auto",
"!",
"interface GigabitEthernet4/0",
" no ip address",
" negotiation auto",
"!",
"interface Serial5/0",
" no ip address",
" shutdown",
" serial restart-delay 0",
"!",
"interface Serial5/1",
" no ip address",
" shutdown",
" serial restart-delay 0",
"!",
"interface Serial5/2",
" no ip address",
" shutdown",
" serial restart-delay 0",
"!",
"interface Serial5/3",
" no ip address",
" shutdown",
" serial restart-delay 0",
"!",
"interface Serial6/0",
```



```
" no ip address",
" shutdown",
" serial restart-delay 0",
"!",
"interface Serial6/1",
" no ip address",
" shutdown",
" serial restart-delay 0",
"!",
"interface Serial6/2",
" no ip address",
" shutdown",
" serial restart-delay 0",
"!",
"interface Serial6/3",
" no ip address",
" shutdown",
" serial restart-delay 0",
"!",
"interface Serial6/4",
" no ip address",
" shutdown",
" serial restart-delay 0",
"!",
"interface Serial6/5",
" no ip address",
" shutdown",
" serial restart-delay 0",
"!",
"interface Serial6/6",
" no ip address",
" shutdown",
" serial restart-delay 0",
"!",
"interface Serial6/7",
" no ip address",
" shutdown",
" serial restart-delay 0",
"!",
"!",
"ip forward-protocol nd",
"!",
"no ip http server",
"no ip http secure-server",
"!",
"!",
"no cdp log mismatch duplex",
"!",
"!",
"!",
"control-plane",
"!",
"!",
"!",
"!",
"!",
"!",
"gatekeeper",
" shutdown",
"!",
"banner motd ^C",
"Unauthorized Personels are Prohibited!",
"^C",
"!",
"line con 0",
" exec-timeout 0 0",
" privilege level 15",
" password cisco",
" logging synchronous",
" login",
" stopbits 1",
"line aux 0",
" exec-timeout 0 0",
" privilege level 15",
" logging synchronous",
" stopbits 1",
"line vty 0 4",
" login local",
```

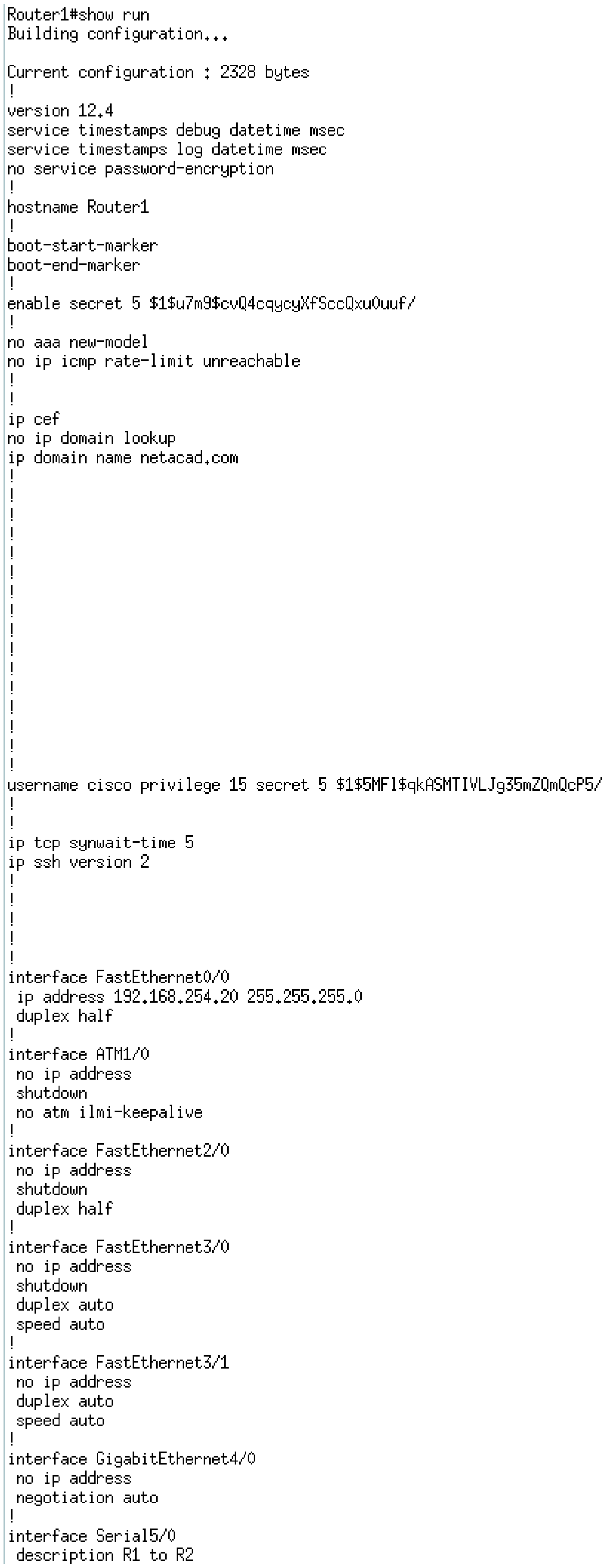


```
                " transport input all",
                "!",
                "!",
                "end"
            ]
        ]
    }
}

PLAY [routers[0]] **************************************************************************

TASK [c7200 : configuring ip address in serial5/0] ****************************************
changed: [192.168.254.20]

TASK [c7200 : configuring ip address in serial5/1] ****************************************
changed: [192.168.254.20]

TASK [c7200 : checking hostname] ***********************************************************
ok: [192.168.254.20]

TASK [c7200 : debug] ***********************************************************************
ok: [192.168.254.20] => {
    "msg": {
        "changed": false,
        "failed": false,
        "stdout": [
            "Interface                  IP-Address      OK? Method Status                Protocol\nFastEthernet0/0           192
.168.254.20  YES manual up                    up      \nATM1/0                     unassigned      YES unset  administratively do
wn down    \nFastEthernet2/0            unassigned      YES unset  administratively down down   \nFastEthernet3/0            una
ssigned      YES unset  administratively down down    \nFastEthernet3/1            unassigned      YES unset  up
   up      \nGigabitEthernet4/0         unassigned      YES unset  up                    up      \nSerial5/0                   10.
0.1.1        YES manual up                    up      \nSerial5/1                   10.0.2.1        YES manual up
   down    \nSerial5/2                  unassigned      YES unset  administratively down down   \nSerial5/3                   una
ssigned      YES unset  administratively down down    \nSerial6/0                   unassigned      YES unset  administratively do
wn down    \nSerial6/1                  unassigned      YES unset  administratively down down   \nSerial6/2                   una
ssigned      YES unset  administratively down down    \nSerial6/3                   unassigned      YES unset  administratively do
wn down    \nSerial6/4                  unassigned      YES unset  administratively down down   \nSerial6/5                   una
ssigned      YES unset  administratively down down    \nSerial6/6                   unassigned      YES unset  administratively do
wn down    \nSerial6/7                  unassigned      YES unset  administratively down down"
        ],
        "stdout_lines": [
            [
                "Interface                  IP-Address      OK? Method Status                Protocol",
                "FastEthernet0/0            192.168.254.20  YES manual up                    up      ",
                "ATM1/0                     unassigned      YES unset  administratively down down    ",
                "FastEthernet2/0            unassigned      YES unset  administratively down down    ",
                "FastEthernet3/0            unassigned      YES unset  administratively down down    ",
                "FastEthernet3/1            unassigned      YES unset  up                    up      ",
                "GigabitEthernet4/0         unassigned      YES unset  up                    up      ",
                "Serial5/0                  10.0.1.1        YES manual up                    up      ",
                "Serial5/1                  10.0.2.1        YES manual up                    down    ",
                "Serial5/2                  unassigned      YES unset  administratively down down    ",
                "Serial5/3                  unassigned      YES unset  administratively down down    ",
                "Serial6/0                  unassigned      YES unset  administratively down down    ",
                "Serial6/1                  unassigned      YES unset  administratively down down    ",
                "Serial6/2                  unassigned      YES unset  administratively down down    ",
                "Serial6/3                  unassigned      YES unset  administratively down down    ",
                "Serial6/4                  unassigned      YES unset  administratively down down    ",
                "Serial6/5                  unassigned      YES unset  administratively down down    ",
                "Serial6/6                  unassigned      YES unset  administratively down down    ",
                "Serial6/7                  unassigned      YES unset  administratively down down"
            ]
        ]
    }
}

PLAY RECAP *********************************************************************************
192.168.254.20             : ok=12   changed=5    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

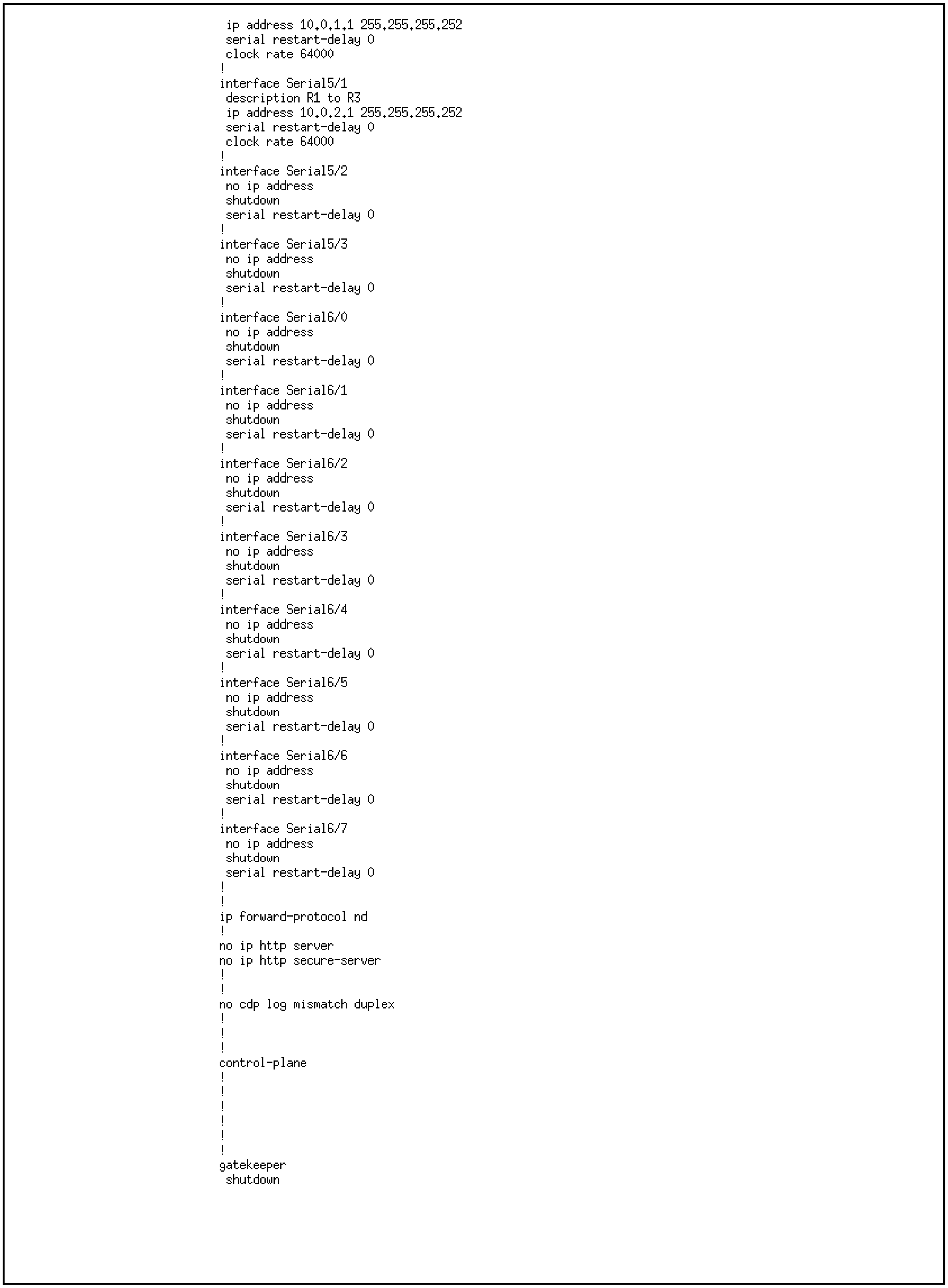
Figure 28. The screenshot above shows the result after running the playbook.

14. Verify the configuration done on the Cisco router(manually check the configuration on the Cisco

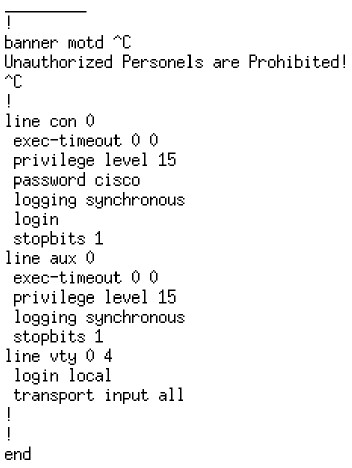
router VM).



```
Router1#show run
Building configuration...

Current configuration : 2328 bytes
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Router1
!
boot-start-marker
boot-end-marker
!
enable secret 5 $1$u7m9$cvQ4cqycyXfSccQxu0uuf/
!
no aaa new-model
no ip icmp rate-limit unreachable
!
!
ip cef
no ip domain lookup
ip domain name netacad.com
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
username cisco privilege 15 secret 5 $1$5MFl$qkASMTIVLJg35mZQmQcP5/
!
!
ip tcp synwait-time 5
ip ssh version 2
!
!
!
!
!
interface FastEthernet0/0
 ip address 192.168.254.20 255.255.255.0
 duplex half
!
interface ATM1/0
 no ip address
 shutdown
 no atm ilmi-keepalive
!
interface FastEthernet2/0
 no ip address
 shutdown
 duplex half
!
interface FastEthernet3/0
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface FastEthernet3/1
 no ip address
 duplex auto
 speed auto
!
interface GigabitEthernet4/0
 no ip address
 negotiation auto
!
interface Serial5/0
 description R1 to R2
```

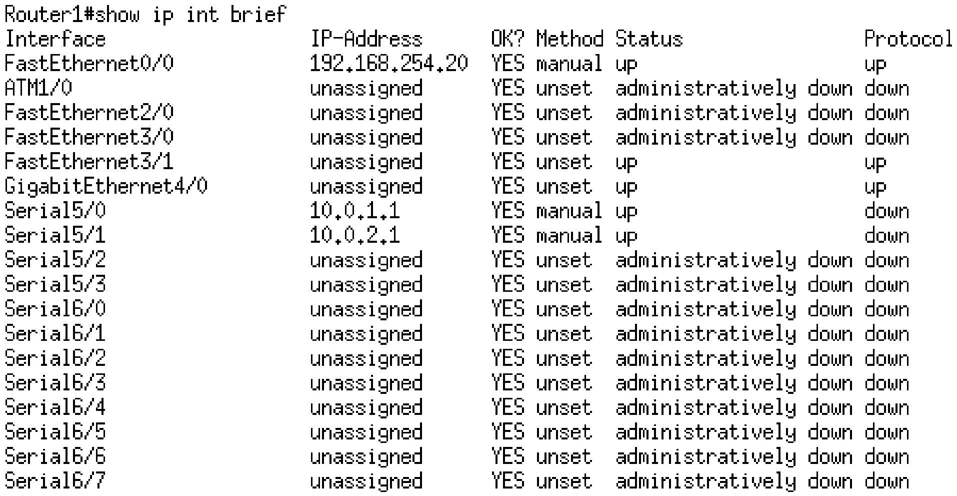


```
 ip address 10.0.1.1 255.255.255.252
 serial restart-delay 0
 clock rate 64000
!
interface Serial5/1
 description R1 to R3
 ip address 10.0.2.1 255.255.255.252
 serial restart-delay 0
 clock rate 64000
!
interface Serial5/2
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial5/3
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial6/0
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial6/1
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial6/2
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial6/3
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial6/4
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial6/5
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial6/6
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial6/7
 no ip address
 shutdown
 serial restart-delay 0
!
!
ip forward-protocol nd
!
no ip http server
no ip http secure-server
!
!
no cdp log mismatch duplex
!
!
!
control-plane
!
!
!
!
!
!
gatekeeper
 shutdown
```



```
!
banner motd ^C
Unauthorized Personels are Prohibited!
^C
!
line con 0
 exec-timeout 0 0
 privilege level 15
 password cisco
 logging synchronous
 login
 stopbits 1
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
 stopbits 1
line vty 0 4
 login local
 transport input all
!
!
end
```

Figure 29. The screenshot above shows the verification of configuration inside of the router device using the “show running-config” command.



```
Router1#show ip int brief
Interface                  IP-Address      OK? Method Status                Protocol
FastEthernet0/0            192.168.254.20  YES manual up                    up
ATM1/0                     unassigned      YES unset  administratively down down
FastEthernet2/0            unassigned      YES unset  administratively down down
FastEthernet3/0            unassigned      YES unset  administratively down down
FastEthernet3/1            unassigned      YES unset  up                    up
GigabitEthernet4/0         unassigned      YES unset  up                    up
Serial5/0                  10.0.1.1        YES manual up                    down
Serial5/1                  10.0.2.1        YES manual up                    down
Serial5/2                  unassigned      YES unset  administratively down down
Serial5/3                  unassigned      YES unset  administratively down down
Serial6/0                  unassigned      YES unset  administratively down down
Serial6/1                  unassigned      YES unset  administratively down down
Serial6/2                  unassigned      YES unset  administratively down down
Serial6/3                  unassigned      YES unset  administratively down down
Serial6/4                  unassigned      YES unset  administratively down down
Serial6/5                  unassigned      YES unset  administratively down down
Serial6/6                  unassigned      YES unset  administratively down down
Serial6/7                  unassigned      YES unset  administratively down down
```

Figure 30. The screenshot above shows the verification of configuration inside of the router device using “show ip int brief” command.

Github Link:

https://github.com/piolotorrecampo/CPE-243_HOA_10.1.git

Conclusion:

In conclusion, creating a Cisco Router virtual machine on VirtualBox and automating its configuration using Ansible is a useful skill for network administrators and system administrators. This approach allows for easier management of network infrastructure and improves efficiency by automating repetitive tasks. VirtualBox provides a platform for creating virtual machines, and Ansible can be used to automate the configuration process. With the help of Ansible, administrators can quickly configure multiple routers with minimal effort, and this can save a significant amount of time and resources. Overall, this approach can be an effective way to improve network management and streamline operations.