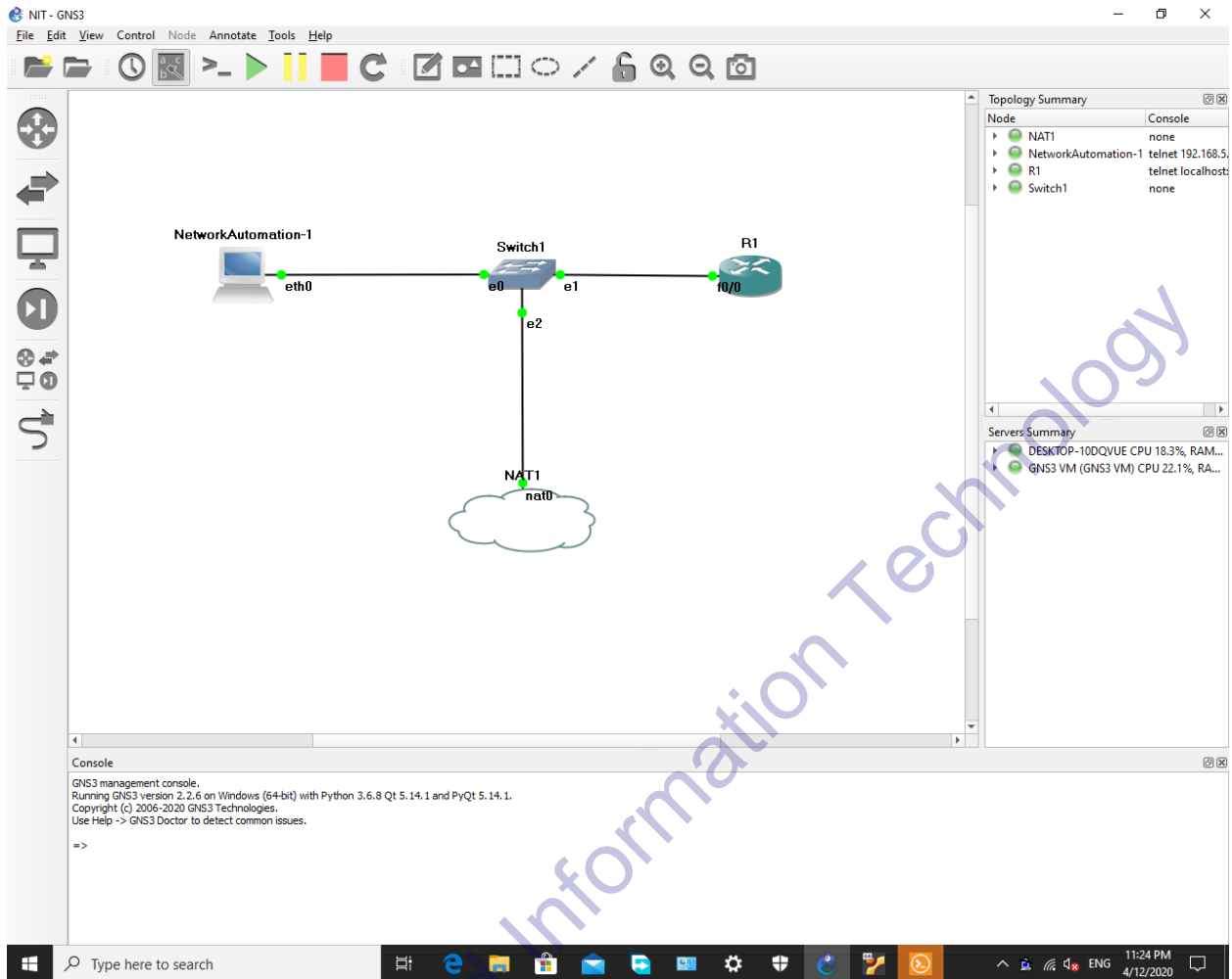


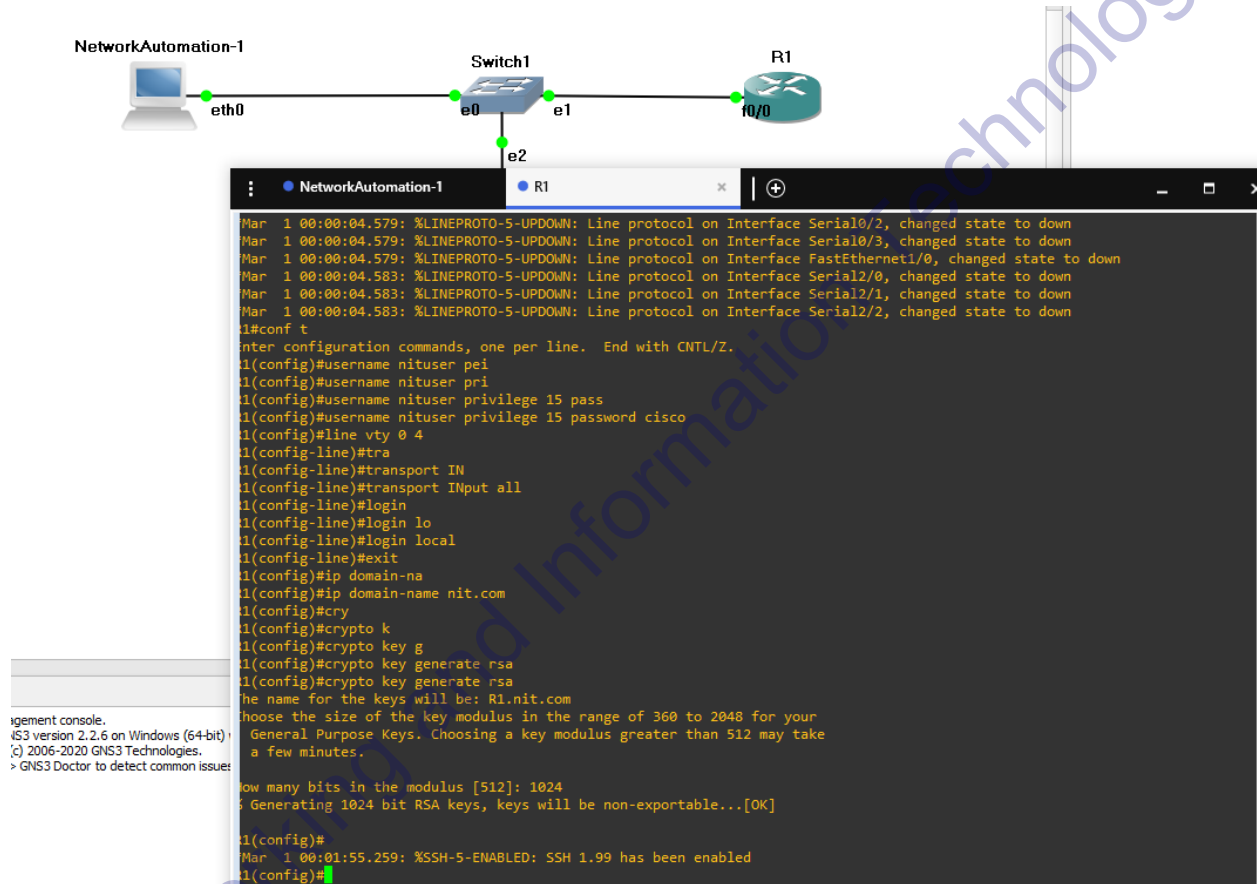
ကျနော်တို့ ansible ရဲ့ အရေးပါပုံတွေနဲ့ အသုံးပြုပုံတွေကို pageမှာတင်ခဲ့ပြီးဖြစ်ပါတယ်။ အခုတခေါက်ကတော့ ကျနော်တို့ ansible ကိုအသုံးပြုပြီး basic lab လေးတခုပြုလုပ်သွားမှာဖြစ်ပါတယ်။ basic lab အတွက် network emulator ကတော့ GNS3 နဲ့ GNS3 VM ကိုအသုံးပြုသွားမှာဖြစ်ပြီး ansible အကြောင်း ကျနော်တ page မှာရေးထားတာကိုဖတ်တာပဲဖြစ်ဖြစ်၊ တခြား youtube ကဖြစ်ဖြစ် website တခုခုကဖြစ်ဖြစ် ansible အကြောင်းကိုဖတ်ထားသိထားပြီးမှာ ဒီlab လေးကိုစမ်းကျိပါလို့အကြံပြုပါရစေ။ GNS3 ကတော့ တော်တော်များများသိထားပြီးသားမို့ ကျနော် GNS3 နဲ့ GNS3 VM install လုပ်တာနဲ့ configuration ပိုင်းတွေကိုထည့်မပြောတော့ပါဘူး။

ကျနော်တို့ ansible lab တခုစတော့မယ်ဆိုရင် အဓိကအားဖြင့် cisco router series ကိုတော့ 3700 series သုံးဖို့အကြံပေးပါရစေ၊ တခြား series တွေသုံးလို့ရလားဆိုတော့ ရတော့ရပါတယ်၊ တချို့ router တွေကျတော့ ssh configure လုပ်ပြီး ansible ကနေလှမ်းဝင်တာမှာ error တက်တာမျိုးတွေ ကြုံတတ်လို့ပါ။ အဓိကကတော့ GNS3 Lab မှာဆိုရင် ကျနော်တို့ အနည်းဆုံး router တလုံးရှိရမယ်၊ network automation control node အဖြစ်သုံးမယ့် linux OS run ထားတဲ့ PC တစ်လုံးရှိရမယ်၊ နောက်ပြီး PCနဲ့ router ကိုချိတ်ပေးမယ့် switch ရှိရမယ်၊ နောက်ပြီး internet access ရှိရမယ်ပေါ့ဗျာ။ စစ်ချင်းကျနော်တို့ ရဲ့ lab topology ကတော့ ဒီလိုဖြစ်နေပါမယ်။

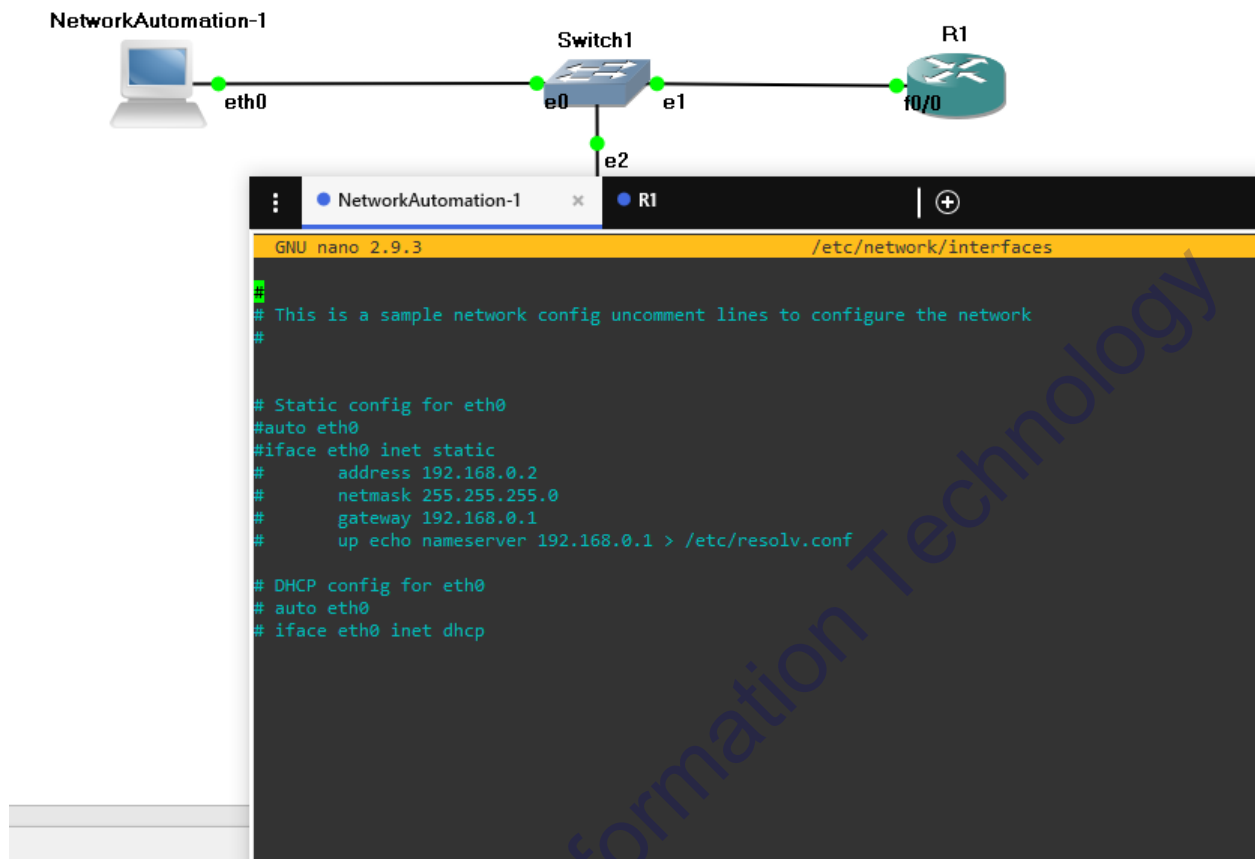


ကျနော်တို့ control node PC အတွက် မိမိစက်မှာ Ubuntu VM တင်ပြီး အသုံးပြုလိုရသလို ကျနော်တို့ GNS3 မှာ ready-made ပါတဲ့ network automation appliance ကိုအသုံးပြုလည်းရပါတယ်။ Network automation appliance ကိုတော့ GNS3 ရဲ့ official website ဖြစ်တဲ့ <https://www.gns3.com/marketplace/appliance/network-automation> မှာ download လုပ်နိုင်ပါတယ်။ ဒီ appliance မှာတော့ ကျနော်တို့ ansible နဲ့ python ကို install လုပ်စရာမလိုပဲ တခါတည်း ready-made ပါလာပြီးသားမို့ အခုမှစမယ့် beginner တွေအတွက် တော်တော်လေးအသုံးဝင်ပါတယ်။ ဒါမှမဟုတ်ပဲ ကျနော်တို့ VM သက်သက်တင်ပြီးသုံးမယ်ဆိုလည်း ရပါတယ်၊ နည်းနည်းတော့ အလုပ်ရှုပ်မှာပေါ့ဗျာ။ နောက်ပြီးတော့ စောစောက network automation appliance ကို အများအားဖြင့် GNS3 VM မှာပဲ import လုပ်ဖို့အကြံပေးပါရစေ၊ GNS3 ကိုယ်တိုင်ကလည်း GNS3 VM ပေါ်တင်ဖို့ကို recommend ပေးထားပါတယ်။ တကယ်လို့ local PC မှာပဲတင်မယ်ဆိုလည်းရပါတယ်၊ အများအားဖြင့် local PC ကျတော့ Firewall ပိတ်ရတာနက်ပြီးတော့ RAM မနိုင်လို့နက် error တွေအများကြီး ကျနော်ကြုံဖူးပါတယ်။ အခုကြောင့် အခုမှစမယ့် beginner ဆိုရင်တော့ GNS3 VM ပေါ်မှာပဲတင်ဖို့အကြံပြုပါရစေ။

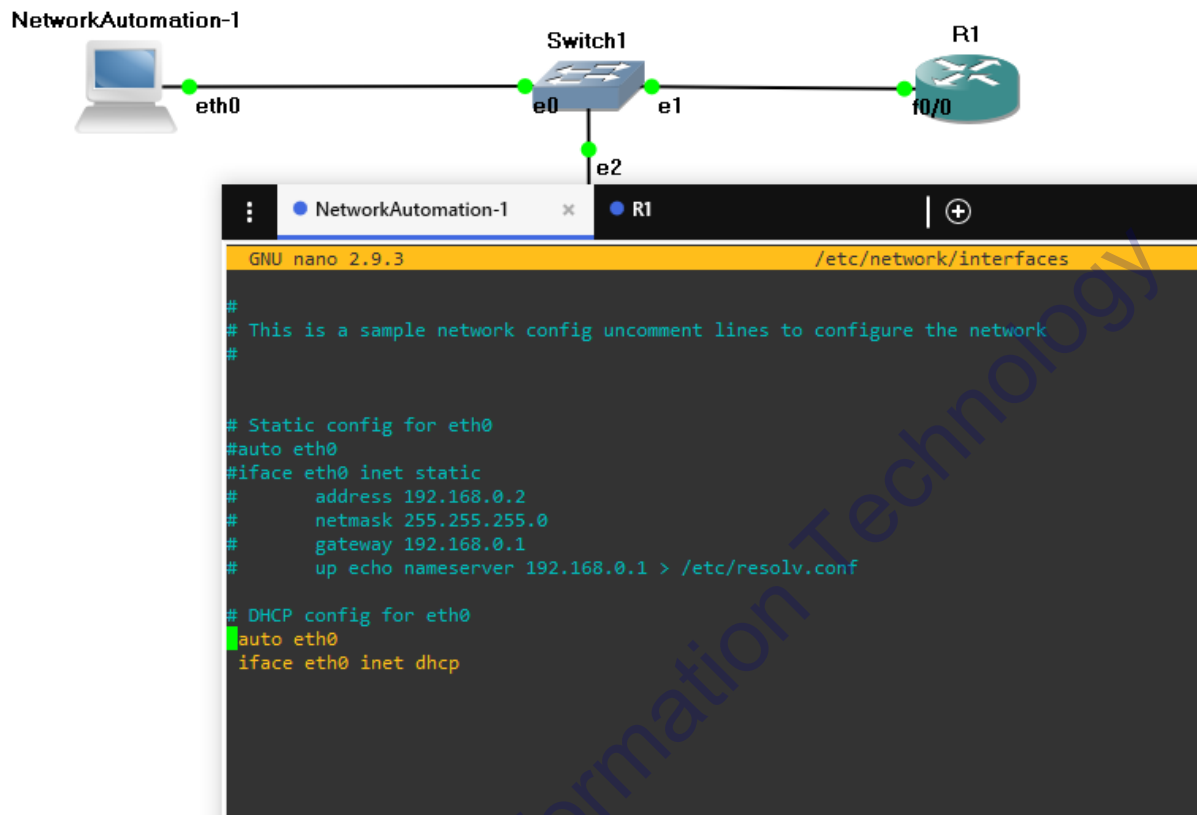
အခုကျနော်တို့ topology တည်ဆောက်ပြီးပြီဆိုတော့ device တွေကို power on ပေးပြီး console ထဲကိုဝင်ပါ။ ဒီနေရာမှာတခုပြောချင်တာက ကျနော်တို့ network automation စတော့မယ်ဆိုရင် ပထမဆုံး မိမိလှမ်းဝင်မယ့် router တွေထဲမှာ ssh configure လုပ်ထားဖို့လိုပါမယ်။ ပြီးတော့ control node PC အတွက် internet access ရဖို့ NAT ဆီက DHCP IP ရယူရပါမယ်။ အခုလိုရဖို့အတွက် control node PC ရဲ့ console ကိုဝင်ပြီး ကျနော်တို့ /etc/network directory ထဲက interfaces file ကို ပြင်ရပါမယ်။ သုံးမယ့် command ကတော့ nano /etc/network/interfaces ဖြစ်ပါတယ်။ ပထမဆုံး router ကို ssh configure လုပ်ပါမယ်။



နောက်ပြီးတော့ control node PC ကို Dynamic IP ပြောင်းပါမယ်။



ဒီနေရာမှာ ကျနော်တို့ '#' လေးတွေက အသုံးပြုလို့မရအောင် ပိတ်ထားတာဖြစ်ပါတယ်။ အခုထဲကမှ auto eth0 နဲ့ iface eth0 inet dhcp တို့ကိုဖွင့်ပေးရပါမယ်။ auto eth0 ကတော့ Control node PC ရဲ့ ethernet port ကို static မဟုတ်ပဲ dynamic ip ပြောင်းတာဖြစ်ပြီး iface eth0 inet dhcp ကတော့ စောစောက ethernet port ကနေ dhcp request လုပ်တာဖြစ်ပါတယ်။



နောက်ပြီးတော့ router ရဲ့ management interface ကို ip ပေးပါမယ်။



NetworkAutomation-1



eth0

Switch1

e0

e1

R1

fa0/0

```
NetworkAutomation-1 R1
R1(config)#username nituser privilege 15 password cisco
R1(config)#line vty 0 4
R1(config-line)#transport IN
R1(config-line)#transport INput all
R1(config-line)#login
R1(config-line)#login lo
R1(config-line)#login local
R1(config-line)#exit
R1(config)#ip domain-na
R1(config)#ip domain-name nit.com
R1(config)#crypto
R1(config)#crypto k
R1(config)#crypto key g
R1(config)#crypto key generate rsa
R1(config)#crypto key generate rsa
The name for the keys will be: R1.nit.com
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
% Generating 1024 bit RSA keys, keys will be non-exportable...[OK]

R1(config)#
*Mar 1 00:01:55.259: %SSH-5-ENABLED: SSH 1.99 has been enabled
R1(config)#int fa0/0
R1(config-if)#no shu
R1(config-if)#no shutdown
R1(config-if)#ip add
R1(config-if)#ip address 192
*Mar 1 00:07:44.375: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
*Mar 1 00:07:45.375: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
R1(config-if)#ip address 192.168.122.10 255.255.255.0
R1(config-if)#end
R1#
*Mar 1 00:08:01.259: %SYS-5-CONFIG_I: Configured from console by console
R1#
```

ent console.
ersion 2.2.6 on Windows (64-bit)
06-2020 GNS3 Technologies.
i3 Doctor to detect common issues

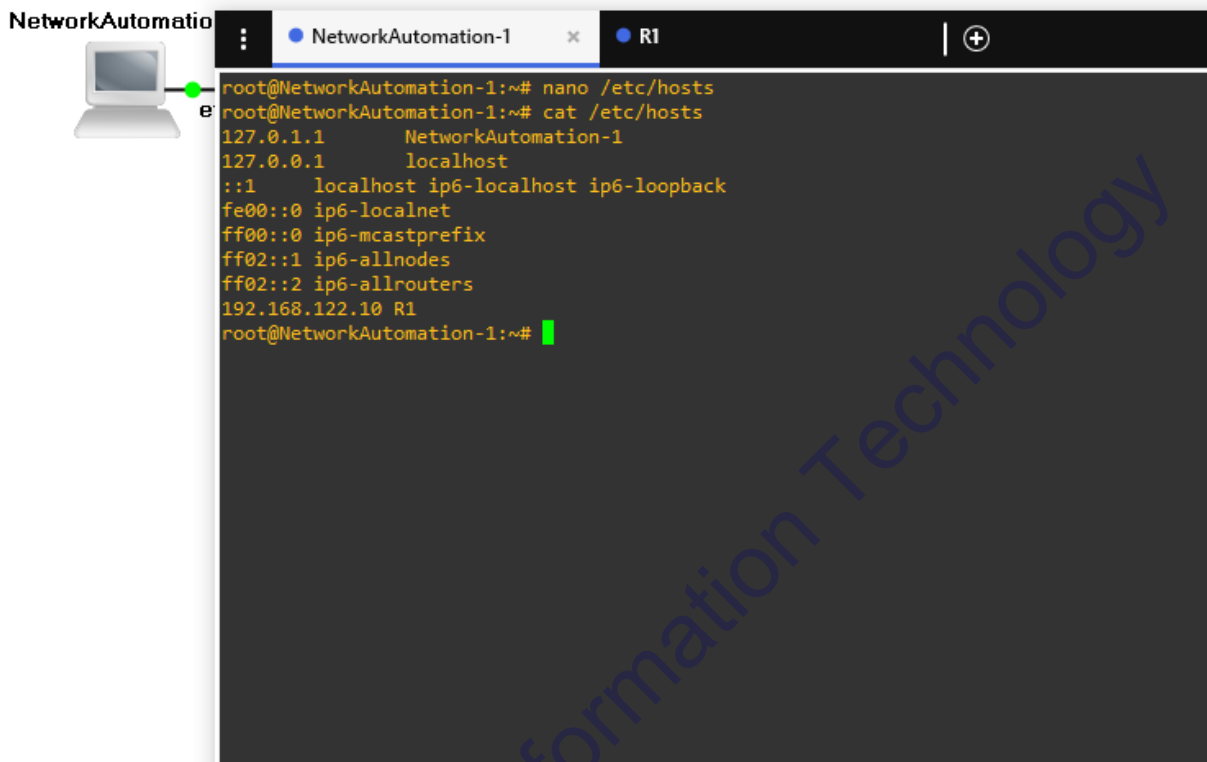
```
*Mar 1 00:08:01.259: %SYS-5-CONFIG_I: Configured from console by console
R1#sh ip int br
Interface IP-Address OK? Method Status Protocol
FastEthernet0/0 192.168.122.10 YES manual up up
Serial0/0 unassigned YES unset administratively down down
FastEthernet0/1 unassigned YES unset administratively down down
Serial0/1 unassigned YES unset administratively down down
Serial0/2 unassigned YES unset administratively down down
Serial0/3 unassigned YES unset administratively down down
FastEthernet1/0 unassigned YES unset administratively down down
Serial2/0 unassigned YES unset administratively down down
Serial2/1 unassigned YES unset administratively down down
Serial2/2 unassigned YES unset administratively down down
Serial2/3 unassigned YES unset administratively down down
R1#
```

အခုဆိုရင်တော့ control node PC ကလည်း internet access ရသွားသလို router နဲ့လည်း network မိသွားပြီဖြစ်ပါတယ်။ ဒီနေရာမှာတခုပြောချင်တာက ကျနော်တို့ အရင်ဆုံး PC မှာ IP ကျလာပြီဆိုရင် အခု ip

address နဲ့ router ရဲ့ interface မှာပေးမယ့် interface က subnet တူရပါမယ်။ အခုမှ network မိမာဖြစ်ပါတယ်။

နောက်တခုကျနော်တို့ လုပ်ရမှာက router ကို name resolution ပေးရပါမယ်။ တကယ်လို့ မပေးပဲနက် ip address နဲ့ သွားမယ်ဆိုလည်းရပါတယ်။ တကယ့် real world မှာဆိုရင်တော့ ကျနော်တို့ department တခုချင်းစီအလိုက် ansible host groups တွေဖွဲ့ထားမယ်။ department တခုချင်းစီက router တွေ L3 Switch တွေကိုလည်း ဥပမာ HR ဆိုရင် HR_SW1, HR_R1 စသဖြင့်ပေါ့ဗျာ။ စောစောက IP address နဲ့ပဲသွားတာထက်စာရင်တော့ ပိုပြီးရိုးရှင်းပီးတော့ တကယ့် တကယ် troubleshoot လုပ်ရမယ်ဆိုရင်လည်း တိုင်မပတ်တော့ဘူးပေါ့ဗျာ။

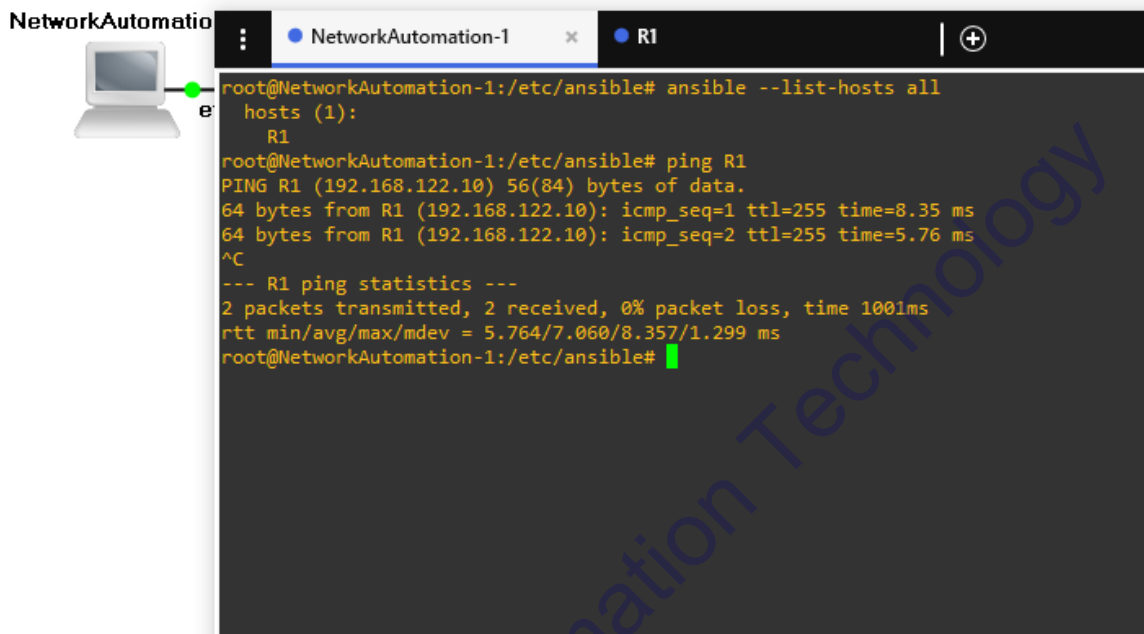
အခုကျနော်တို့ topology ထဲမှာ ရှိတဲ့ router ကို name resolution ပေးပါမယ်။ တကယ်လို့ router hostname ကိုသွားပြောင်းချင်တယ်ဆိုရင်တော့ hostname <name> command နဲ့ router ရဲ့ global config mode မှာသွားပြီးချိန်းလို့ရပါတယ်။ name resolution ပေးဖို့ ကျနော်တို့ root အောက်က etc အောက်က hosts file ကို text editor နဲ့ပြင်ရပါမယ်။ Linux ရဲ့ original text editor command ဂုဏ်တော့ Vim နဲ့ nano ဖြစ်ပါတယ်။ အခု GNS3 မှာတော့ vim မရှိတဲ့အတွက် nano နဲ့ text တွေကို read-only နဲ့ပြန်ကျိတ် cat command တွေကိုအသုံးပြုသွားပါမယ်။ အခု name resolution ပေးဖို့အတွက် သုံးမယ့် command ကတော့ nano /etc/hosts ဖြစ်ပြီး ဆိုလိုတာကတော့ etc အောက်က hosts file ကိုပြင်မယ်လို့ပြောတာဖြစ်ပါတယ်။ ကျနော်က router ကို R1 လို့ hostname ပေးလိုက်တဲ့အတွက် router ip address ကို R1 နဲ့ name resolute လုပ်သွားမှာဖြစ်ပါတယ်။



The image shows a terminal window titled "NetworkAutomation-1" with a tab for "R1". The terminal output is as follows:

```
root@NetworkAutomation-1:~# nano /etc/hosts
root@NetworkAutomation-1:~# cat /etc/hosts
127.0.1.1    NetworkAutomation-1
127.0.0.1    localhost
::1         localhost ip6-localhost ip6-loopback
fe00::0     ip6-localnet
ff00::0     ip6-mcastprefix
ff02::1     ip6-allnodes
ff02::2     ip6-allrouters
192.168.122.10 R1
root@NetworkAutomation-1:~#
```

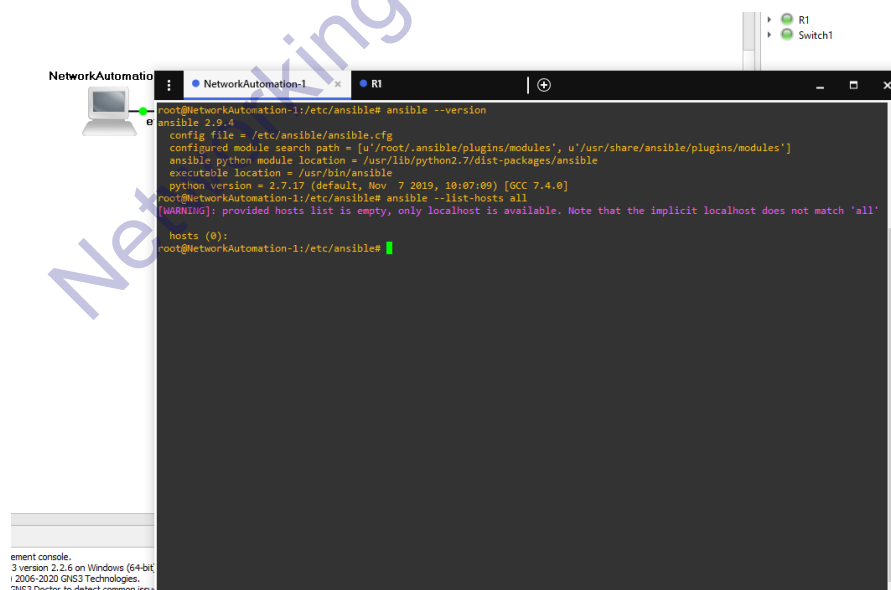
အခုဆိုကျနော်တို့ PC ကနေ router ဆီကို ping မယ်ဆိုရင် ip address ရှိက်စရာမလိုတော့ပဲနက် R1ဆိုပီးရှိက်ရုံနဲ့ ရပါတယ်။ name resolution ကအဓိက ဘာလုပ်ပေးလဲဆိုတော့ ip address ကို မိမိပေးချင်တဲ့ name ပေးရုံနဲ့ PC ထဲက DNS Resolver cache ထဲမှာသွားသိမ်းပြီး နောက်ထပ်ကျနော်တို့ ip address အစား name နဲ့ပဲ အသုံးပြုရတာမျိုးဖြစ်ပါတယ်။ ထိုနည်းလည်းကောင်းပဲ ကျနော်တို့ ssh သုံးမယ်ဆိုလည်း name နဲ့ပဲ အသုံးပြုလို့ရမှာဖြစ်ပါတယ်။



```
root@NetworkAutomation-1:/etc/ansible# ansible --list-hosts all
hosts (1):
  R1

root@NetworkAutomation-1:/etc/ansible# ping R1
PING R1 (192.168.122.10) 56(84) bytes of data:
64 bytes from R1 (192.168.122.10): icmp_seq=1 ttl=255 time=8.35 ms
64 bytes from R1 (192.168.122.10): icmp_seq=2 ttl=255 time=5.76 ms
^C
--- R1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 5.764/7.060/8.357/1.299 ms
root@NetworkAutomation-1:/etc/ansible#
```

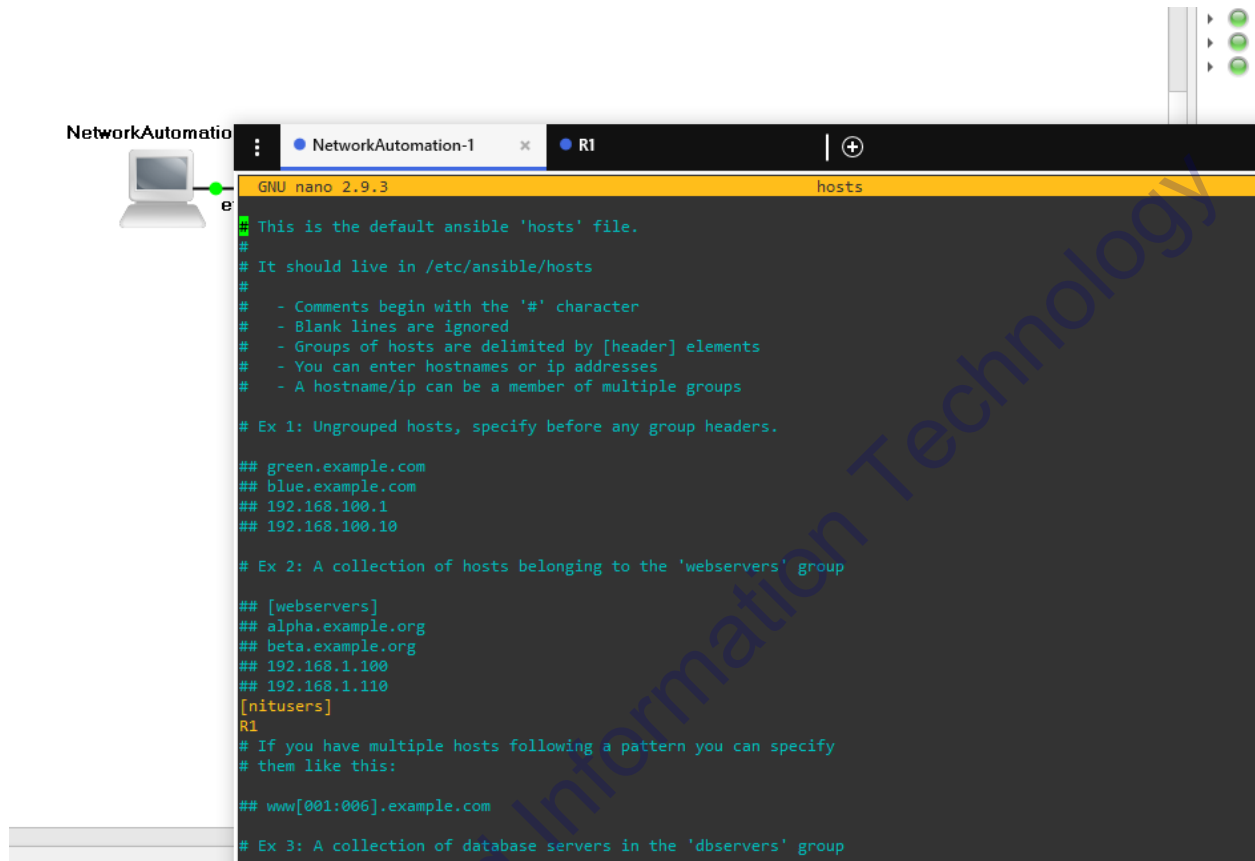
နောက်ထပ်ကျနော်တို့ ansible စတင် configure လုပ်ဖို့အတွက် ansible hosts တွေထည့်ပေးဖို့လိုအပ်ပါတယ်။ ဆိုလိုတာက ဒီ network ထဲမှာ ansible host file ထဲမှာ မှတ်ထားတဲ့ network device တွေရဲ့ ip ဖြစ်စေ name ဖြစ်စေ အာ name တွေအတိုင်း ansible အလုပ်လုပ်လို့ရမှာဖြစ်ပါတယ်။ hostfile ထဲမှာ မရှိတဲ့ ip သို့မဟုတ် name တခုကို ansible နဲ့ run လိုက်ရင် အခုလို error message ပြပါလိမ့်မယ်။ ansible host တွေကျိချင်ရင်တော့ command က ansible -list-hosts all ဖြစ်ပါတယ်။



```
root@NetworkAutomation-1:/etc/ansible# ansible --version
ansible 2.9.4
  config file = /etc/ansible/ansible.cfg
  configured module search path = [u'/root/.ansible/plugins/modules', u'/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python2.7/dist-packages/ansible
  executable location = /usr/bin/ansible
  python version = 2.7.17 (default, Nov 7 2019, 10:07:09) [GCC 7.4.0]
root@NetworkAutomation-1:/etc/ansible# ansible --list-hosts all
[WARNING]: provided hosts list is empty, only localhost is available. Note that the implicit localhost does not match 'all'

hosts (0):
root@NetworkAutomation-1:/etc/ansible#
```

အခုတော့ ကျနော်တို့က စောစောက R1 ကို /etc/hosts ထဲသွားထည့်ပေးရပါမယ်။ nano /etc/hosts လို့ရိုက်ပြီး အခုလိုပြင်လိုက်ပါ။



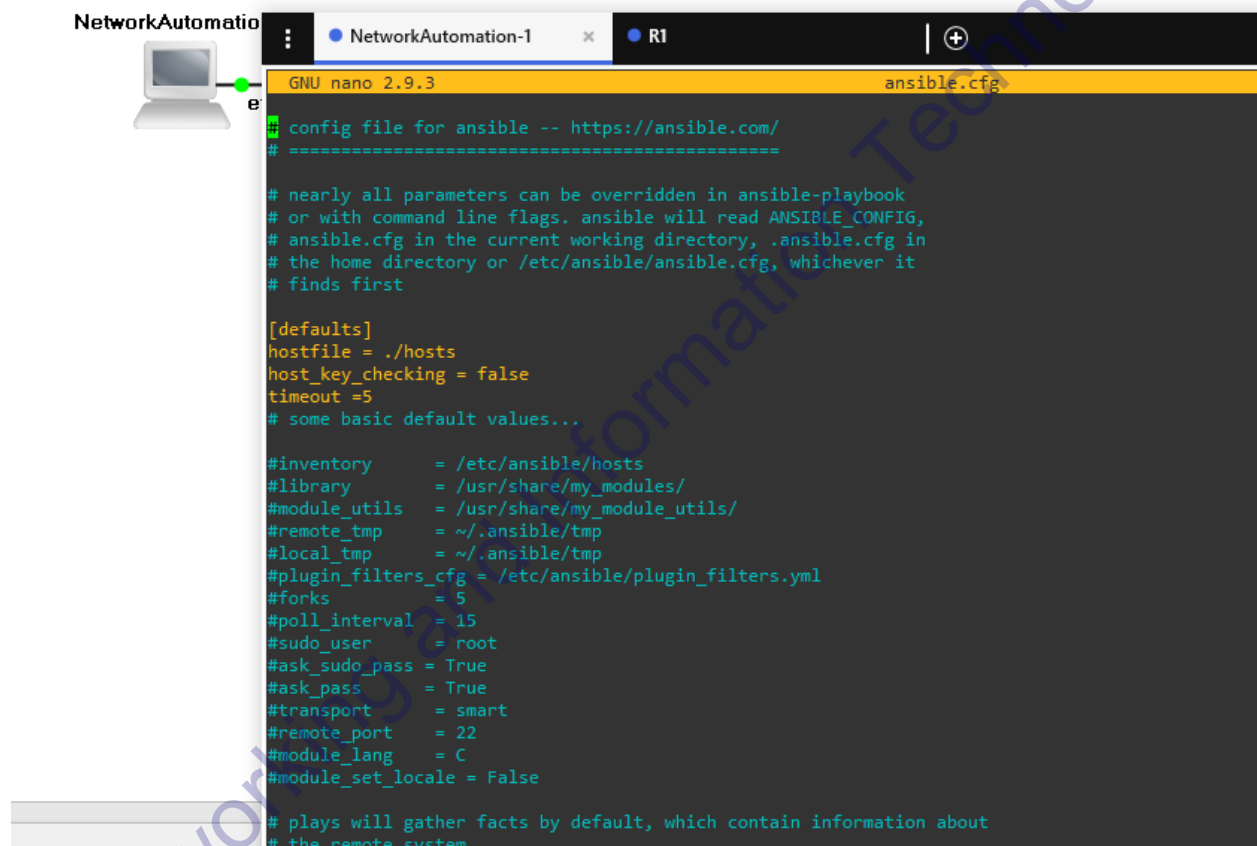
The screenshot shows a terminal window titled 'NetworkAutomation-1' with a sub-window 'R1'. The terminal is running the 'nano' text editor to edit the '/etc/hosts' file. The file content is as follows:

```
GNU nano 2.9.3 hosts
# This is the default ansible 'hosts' file.
#
# It should live in /etc/ansible/hosts
#
# - Comments begin with the '#' character
# - Blank lines are ignored
# - Groups of hosts are delimited by [header] elements
# - You can enter hostnames or ip addresses
# - A hostname/ip can be a member of multiple groups
#
# Ex 1: Ungrouped hosts, specify before any group headers.
## green.example.com
## blue.example.com
## 192.168.100.1
## 192.168.100.10
#
# Ex 2: A collection of hosts belonging to the 'webserver' group
## [webserver]
## alpha.example.org
## beta.example.org
## 192.168.1.100
## 192.168.1.110
[nitusers]
R1
# If you have multiple hosts following a pattern you can specify
# them like this:
## www[001:006].example.com
#
# Ex 3: A collection of database servers in the 'dbserver' group
```

တခြား command တွေကတော့ # sign နဲ့ ပိတ်ထားတာတွေရပါမယ်။ ကျနော်တို့ ansible host တွေသတ်မှတ်ရာမှာ grouped host နဲ့ ungrouped host ဆိုပြီးရှိပါတယ်။ ကျနော်တို့ server တွေ firewall deviceတွေဆိုရင်တော့ များသောအားဖြင့် ungrouped host နဲ့သတ်မှတ်ကြပါတယ်။ ungrouped host ကတော့ အတိုချုံးပြောရရင် server တလုံးချင်းစီတို့ domain name တခုချင်းစီတို့ သတ်မှတ်ရာတွင်အသုံးပြုပါတယ်။ grouped host ကတော့ ကျနော်တို့ဥပမာအားဖြင့် department တခုချင်းစီပေါ့။ HR ဆို HR Group ထဲမှာ သူနက်သက်ဆိုင်တဲ့ name တွေ ip တွေထည့်မယ်။ အခုလိုမျိုးဆိုကျနော်တို့က ansible configuration လုပ်လို့ host ကိုရွေးတော့မယ်ဆို group name လေးထည့်လိုက်တာနဲ့ အခု group ထဲက name တွေ ip တွေအားလုံးကို သက်ရောက်သွားမှာဖြစ်ပါတယ်။

Ansible configuration ကတော့ စတင်ရင်း Ad hoc command နဲ့အရင်စပါမယ်။ command ကတော့ ansible <host-name> -m raw -a "show version | I IOS" -u <ssh username> -k ဖြစ်ပါတယ်။ Ansible က ssh နဲ့ အလုပ်လုပ်တာဖြစ်လို့ run မယ်ဆိုရင် အရင်ဆုံး စောစောက router မှာပေးခဲ့တဲ့ ssh username နဲ့ password ကိုပေးပြီး အသုံးပြုရမှာဖြစ်ပါတယ်။

အခုတစ်ကြိမ်ရင် ကျနော်တို့ /etc/ansible/ directory ထဲက ansible.cfg file လေးကိုသွားပြင်ပါမယ်။ ပြင်တဲ့ နေရာမှာကတော့ ကျနော်တို့ ထိုးစံတိုင်း nano နဲ့ သွားပြင်ပါမယ်။



```
GNU nano 2.9.3 ansible.cfg
# config file for ansible -- https://ansible.com/
# =====

# nearly all parameters can be overridden in ansible-playbook
# or with command line flags. ansible will read ANSIBLE_CONFIG,
# ansible.cfg in the current working directory, .ansible.cfg in
# the home directory or /etc/ansible/ansible.cfg, whichever it
# finds first

[defaults]
hostfile = ./hosts
host_key_checking = false
timeout = 5
# some basic default values...

#inventory      = /etc/ansible/hosts
#library        = /usr/share/my_modules/
#module_utils   = /usr/share/my_module_utils/
#remote_tmp     = ~/.ansible/tmp
#local_tmp      = ~/.ansible/tmp
#plugin_filters_cfg = /etc/ansible/plugin_filters.yml
#forks          = 5
#poll_interval  = 15
#sudo_user      = root
#ask_sudo_pass  = True
#ask_pass       = True
#transport      = smart
#remote_port    = 22
#module_lang    = C
#module_set_locale = False

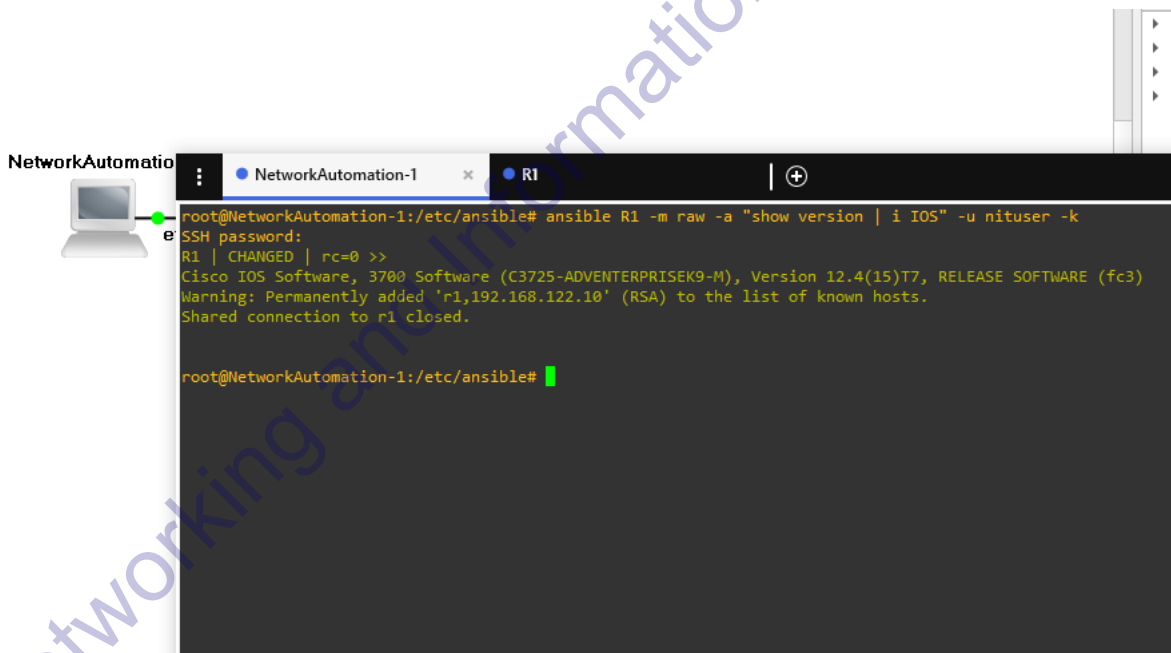
# plays will gather facts by default, which contain information about
# the remote system.
```

ဒီနေရာမှာ ထည့်လိုက်တဲ့ command လေးတောကြောင်းတချက်ရှင်းပြချင်ပါတယ်။ ကျနော်တို့ defaults ဆိုတာကတော့ group name ဖြစ်ပါတယ်။ ဆိုလိုတာကတော့ ကျနော်တို့ ansible playbook file တွေrun မယ်ဆိုရင် ဘယ် playbookကိုပဲ run run။ ဒီ defaults ထဲက အချက်တွေနဲ့ ကိုက်ညီရမယ်ပေါ့ဗျာ။ အောက်မှာတော့ hostfile ဆိုပြီး ကျနော်တို့ စောစောက /etc/ansible/hosts file ကို ညွှန်းပြထားတာတွေရပါမယ်။ ဆိုလိုတာကတော့ ကျနော်တို့ ansible နဲ့ network automation လုပ်တော့မယ်ဆိုရင် ဘယ် playbook ပဲဖြစ်ဖြစ် သူ့ရဲ့ သက်ဆိုင်ရာ hostname ရှိတဲ့ file က ဒီ file ဖြစ်ပါတယ်ဆိုပြီး ပြောလိုက်တာပါပဲ။ အဓိကကတော့ host list မပါရင် ဒါမှမဟုတ် hostfile ကိုပေးတဲ့ location မှားရင်ကျနော်တို့ အပေါ်ကလိုမျိုး host file နေရာမှာ 0ဆိုပြီး error တက်တာနဲ့ ကြုံရမှာပါ။

နောက်တစ်ခုကတော့ hostkeychecking ပါ။ သူကတော့ ကျနော်တို့ router နဲ့ Control node ကြားဆက်သွယ်တဲ့ ssh key တွေကို checking လုပ်တာပါ။ ဒီနေရာမှာ ကျနော်တို့ ဒီ value ကို false လုပ်ထားရင်၊ ကျနော်တို့ ansible နဲ့ router ကို configure လုပ်ရင် ssh key undefined ဆိုပြီး error တက်တာနဲ့ကြုံရမှာပါ။ ဘာကြောင့်လဲဆိုတော့ hostkeychecking ကကျနော်တို့ putty ကနေ ssh နဲ့လှမ်းဝင်တဲ့ အခါ private key၊ public key check လုပ်တာမျိုးပဲဖြစ်ပါတယ်။ အခုကြောင့် ကျနော်တို့ ဒီ host_key_checking = false ဆိုတာကို ရေးပေးရပါတယ်။ သဘောကတော့ host key တွေကို မစစ်ဆေးတော့ဘူးပေါ့ဗျာ။

နောက်တစ်ခုကတော့ timeout ပါ။ သူကတော့ အဓိကက ကျနော်တို့ ခါတိုင်း ssh သုံးသလို အချိန်အကြာကြီး မထားပဲ 5 seconds ဆိုရင် session exit လုပ်လိုက်တော့ လို့ ဆိုလိုတာပါ။ အဓိကက တော့ ansible က configure လုပ်တာသက်သက်ဖြစ်တဲ့အတွက် ကျနော်တို့ အရင်ကလို session time ကိုအကြာကြီး ထားလို့မရပါဘူး။

အခုဆိုရင်ကျနော်တို့ Ad-hoc command mode နဲ့ configure လုပ်ပါမယ်။ လုပ်ဖို့အတွက် /etc/ansible location ထဲမှာ ansible R1 -m raw -a "show version | IOS" -u nituser -k ဆိုပြီး command လေးရိုက်ထည့်လိုက်ပါ။ မိမိရိုက်ထည့်လိုက်တဲ့ command အတိုင်း ansible ကအလုပ်လုပ်သွားမှာဖြစ်ပါတယ်။



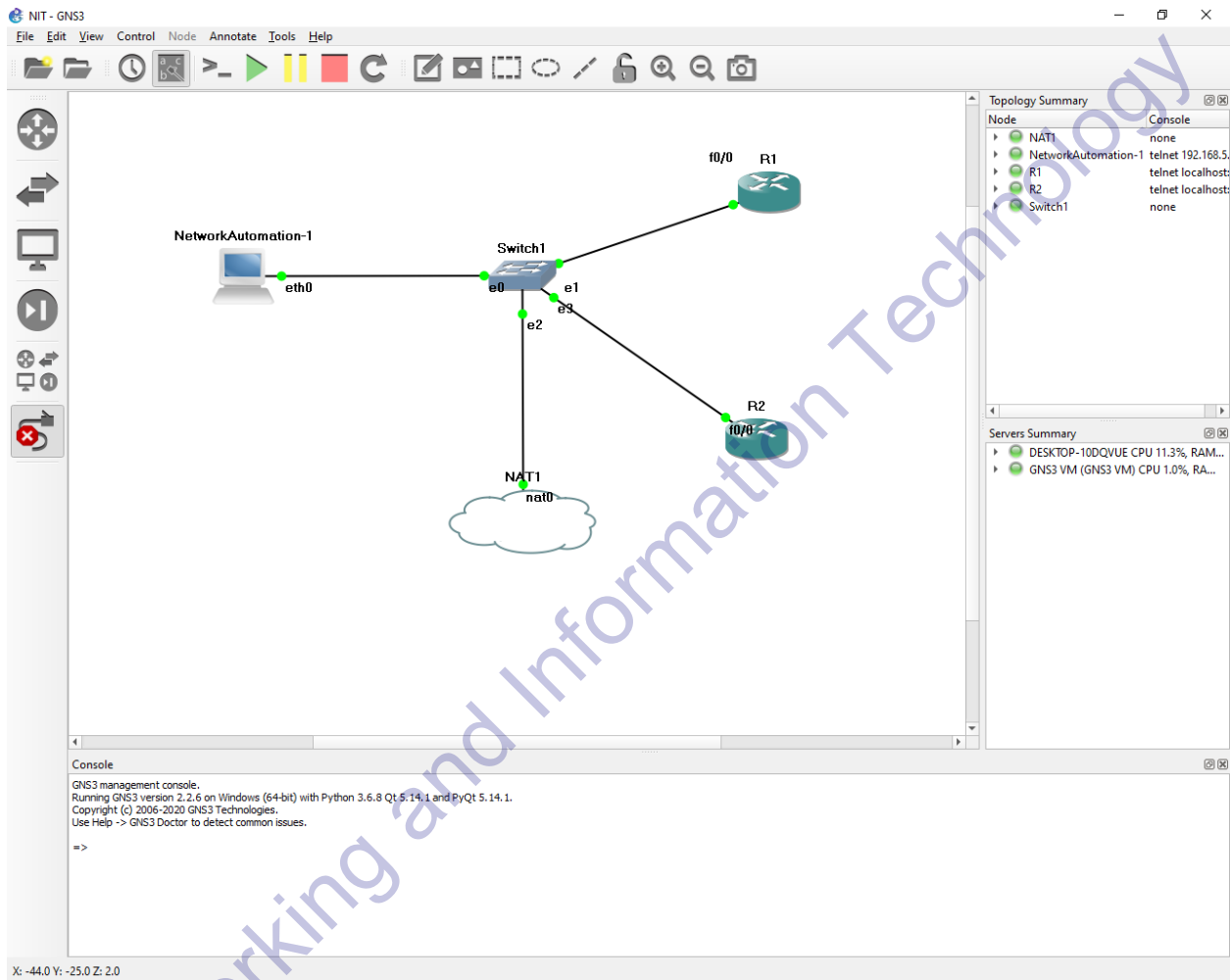
```
root@NetworkAutomation-1:/etc/ansible# ansible R1 -m raw -a "show version | i IOS" -u nituser -k
SSH password:
R1 | CHANGED | rc=0 >>
Cisco IOS Software, 3700 Software (C3725-ADVENTERPRISEK9-M), Version 12.4(15)T7, RELEASE SOFTWARE (fc3)
Warning: Permanently added 'r1,192.168.122.10' (RSA) to the list of known hosts.
Shared connection to r1 closed.

root@NetworkAutomation-1:/etc/ansible#
```

အခုတစ်ခုကျနော်တို့ command လေးအကြောင်းရှင်းပြပါမယ်။ ansible R1 ကတော့ ကျနော်တို့ ansible configure လုပ်ဖို့အတွက် R1 ဆိုတဲ့ host လေးကို show version | i IOS ဆိုတဲ့ cisco command လေး run ပေးလိုက်ပါလို့ ဆိုလိုတာပါ။ ဒီနေရာမှာ -mဆိုတာကတော့ module ကိုပြောတာဖြစ်ပြီး raw ဆိုတာကတော့ module type ဖြစ်ပါတယ်။ -a ဆိုတာကတော့ command လို့ဆိုလိုပါတယ်။ အခုတော့ အားလုံးချုပ်ရရင် R1 ဆိုတဲ့ host လေးကို raw module လေးတဲ့ show version | i IOS ဆိုတဲ့ command လေး ကို username nituser နဲ့ password တောင်းပါလို့ ဆိုလိုပါတယ်။ -u ကတော့ username ကိုပြောတာဖြစ်ပြီး -k ကတော့ password ကိုပြောတာဖြစ်ပါတယ်။ ကျနော်တို့ ဘာ error

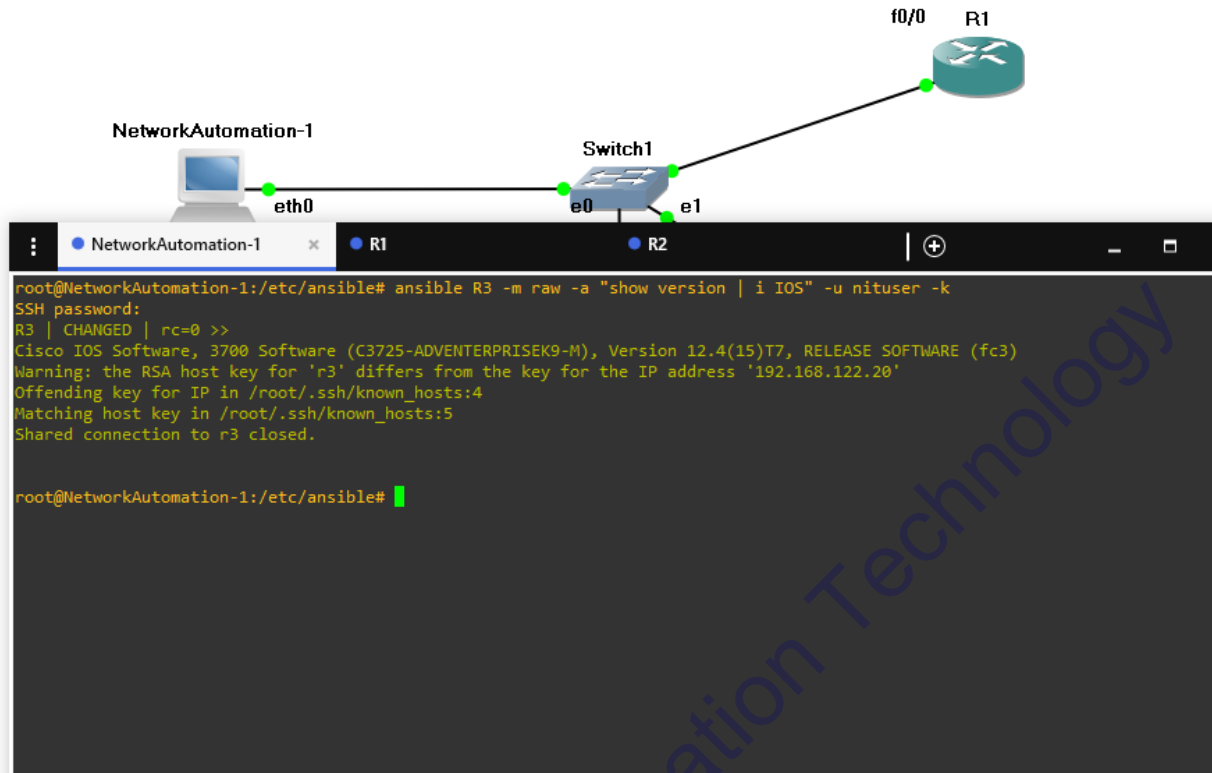
မှတက်ဘူးဆိုရင်တော့ ပုံမှန်ပြထားတဲ့အတိုင်း မြင်ရမှာဖြစ်ပါတယ်။ error တက်တယ်ဆိုရင်တော့ စောစောကကျနော်ပြောသလို ansible.cfg နဲ့ host file ကိုတချက်စစ်ပေးပါလို့ပြောချင်ပါတယ်။

အခုကျနော်တို့ router လေးနောက်တစ်လုံးလောက်ထည့်ပြီး ထပ်စမ်းကြည့်တာပေါ့ဗျာ။ ဒီ router ကိုတော့ ထုံးစံတိုင်း ssh configure လုပ်ပြီး hostname သတ်မှတ်ပေးပါ။ ကျနော်ကတော့ R3 ဆိုပြီး ပေးထားပါတယ်။

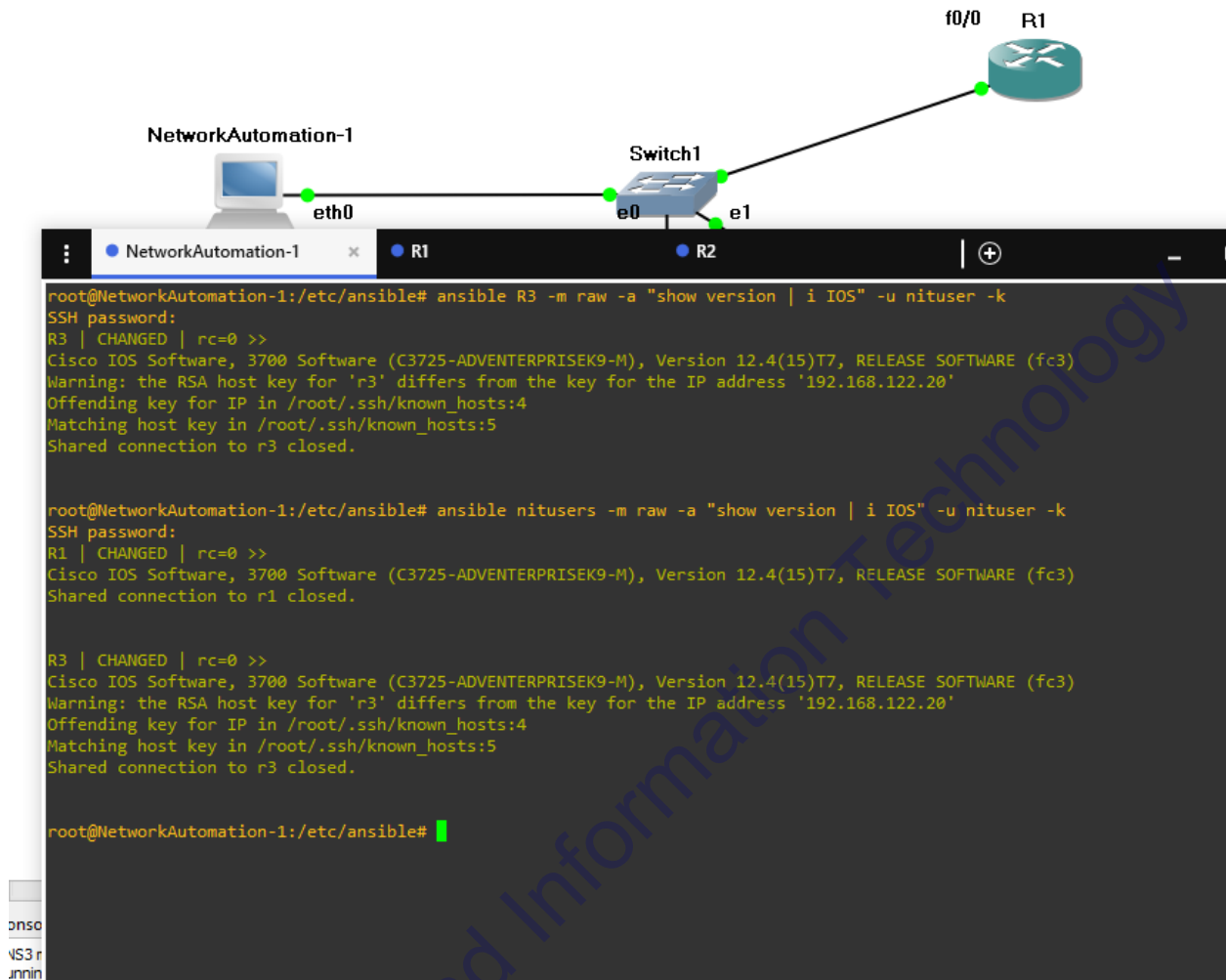


ပြီးရင်ကျနော်တို့ R3 ကို name resolution ပေးဖို့ /etc/hosts ထဲမှာသွားပြီးရေးပါမယ်။ ပြီးရင် ansible host file ထဲက nitusers group အောက်ထဲကို သွားထည့်ပေးရပါမယ်။ ဒီအဆင့်တွေကတော့ အပေါ်မှာလည်း လုပ်ခဲ့ပြီးတဲ့အတွက် ကျနော်အထူးတလည်မရှင်းပြတော့ပါဘူး။

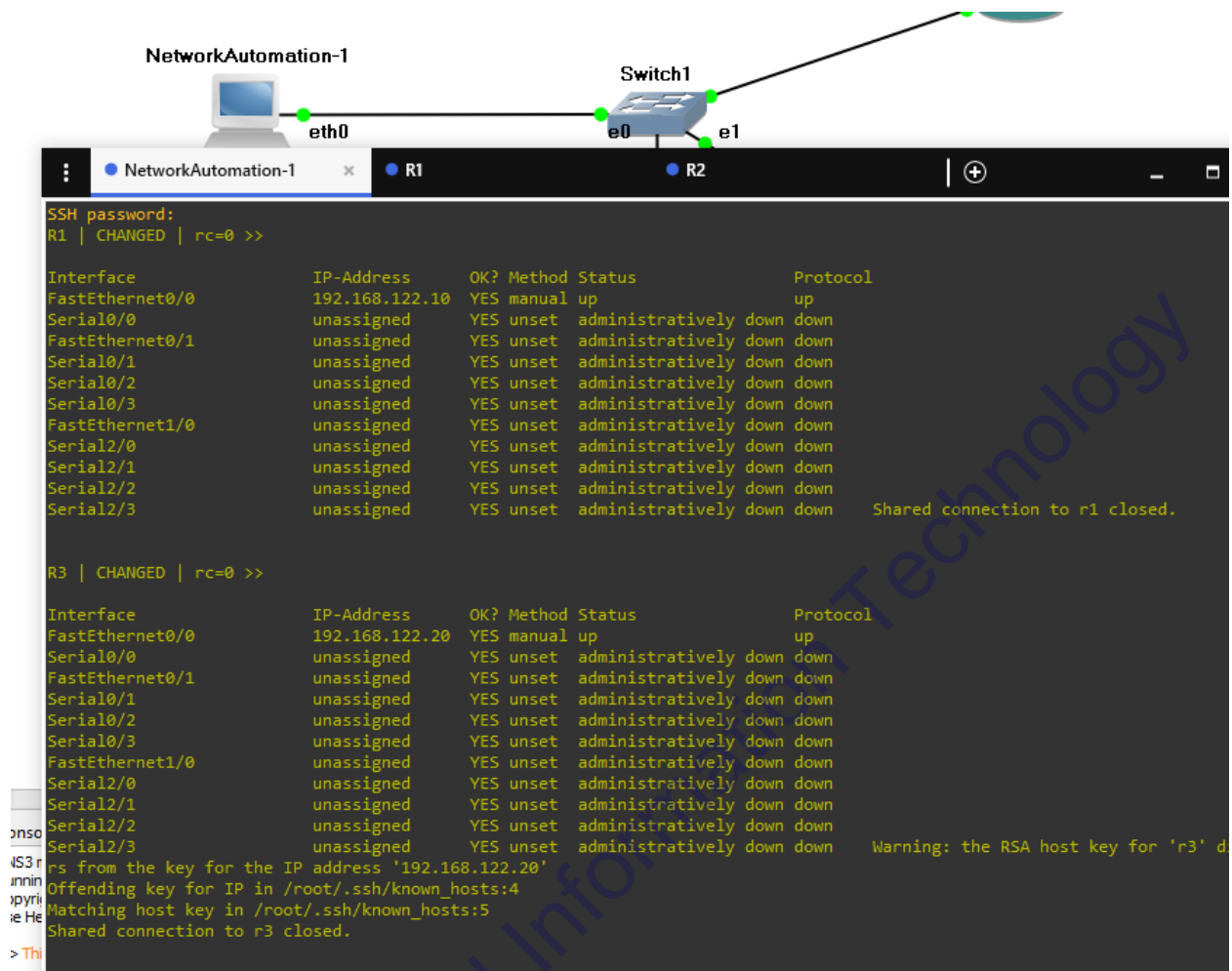
ပြီးရင်ကျနော်တို့ စောစောက R1 တုန်းကလိုပဲ R3 ကို show version ကြည့်ဖို့ စောစောက command လေးရဲ့ R1နေရာမှာ R3 ကိုအစားထိုးပြီး ရေးကြည့်ပါ။ Error မရှိဘူးဆိုရင်တော့ အခုလိုမျိုးမြင်ရပါလိမ့်မယ်။



နောက်တခု ကျနော်တို့ဆီမှာ R1 ရော R3 ရော ရှိပြီဆိုရင်ကျနော်တို့ router 2 ခုလုံးကို configure ချတော့မယ်ဆို စောစောကလို hostname တခုစီနဲ့ မသွားပဲ အကုန်လုံးကိုတပြင်တည်း configure လုပ်ချင်တယ်ဆိုရင် သင်ခန်းစာအစမှာကျနော်ပြောခဲ့တဲ့ group ဖွဲ့တာက ဒီမှာအသုံးဝင်လာပါပြီ။ အခုကျနော်တို့ ပေးထားတဲ့ group name (ကျနော်ကတော့ nitusers ပေါ့ဗျာ) လေးကို စောစောက R3 နေရာမှာအစားသွင်းပြီး run ကြည့်ပါ။ Error မရှိဘူးဆိုရင်တော့ အခုလိုတွေ့ရပါလို့မယ်။



ဒီနေရာမှာ ကျနော်တို့ တခြား ios command တွေနဲ့လည်း စမ်းနိုင်ပါတယ်။ show ip int br တို့ show ip route တို့ show running-config ထဲကမှ မိမိကြည့်ချင်တဲ့အပိုင်းကို | i ခံပြီးလည်းစမ်းသပ်နိုင်ပါတယ်။ ad-hoc command နဲ့ raw module ကတော့ အဓိကက router ထဲက information တွေကို ကြည့်ရုံသက်သက်ပဲဖြစ်တဲ့အတွက် ကျနော်တို့ OSPF configuration တို့ တခြား configure လုပ်တဲ့ အပိုင်းတွေကိုတော့ စမ်းသပ်လို့မရပါဘူး။ Ad-hoc command တွေကများသောအားဖြင့် network ထဲက information တွေကို report ထုတ်ပြတာမျိုးတွေမှာအသုံးပြုပါတယ်။ file အနေနဲ့ save ချင်ရင် စောစောက command လေးနောက်မှာ > filename.txt ဆိုပြီးရိုက်လိုက်ရင် ကျနော်တို့ စမ်းလိုက်တဲ့ directory ထဲမှာ text file တခုအဖြစ် save သွားမှာဖြစ်ပါတယ်။



ကျနော်တို့ ad-hoc command လေးနဲ့ စမ်းသပ်ပြီးတဲ့အခါမှာ ကျနော်တို့ playbook သွားပါမယ်။ playbook ကတော့ ကျနော်တို့ တကြောင်းထက်ပိုတဲ့ multiple command တွေနဲ့ network device တွေကို configure လုပ်တဲ့နေရာမှာအသုံးပြုပါတယ်။ playbook မှာအဓိကသုံးတဲ့ ansible module ခုနစ်ပါးတယ်။ တခြား module တွေလည်းရှိပါသေးတယ်။ အသေးစိတ်သိချင်ရင်တော့

https://docs.ansible.com/ansible/latest/modules/list_of_network_modules.html

မှာဝင်ရောက်လေ့လာနိုင်ပါတယ်။ နောက်တချက်က ansible က cisco အတွက်သီးသန့်မဟုတ်ပါဘူး တခြား vendor တေဖြစ်တဲ့ Juniper, ZTE, Huawei, Fortinet နဲ့ တခြား vendor ပေါင်းများစွာအတွက် အလွယ်တကူအသုံးပြုလို့ရတဲ့ open source platform ဖြစ်ပါတယ်။

ios_command = ad hoc command မှာလိုချိုး information တွေကိုချင်ရင် အသုံးပြုပါတယ်။

ios_config = OSPF တို့ EIGRP တို့ switch မှာဆို VLAN တို့ VTP တို့ configure လုပ်တဲ့နေရာမှာအသုံးပြုပါတယ်။

raw module = ad hoc လိုပါပဲ command တကြောင်းထဲ အသုံးပြုရတာပါ ဘာမှမထူးပါဘူး။

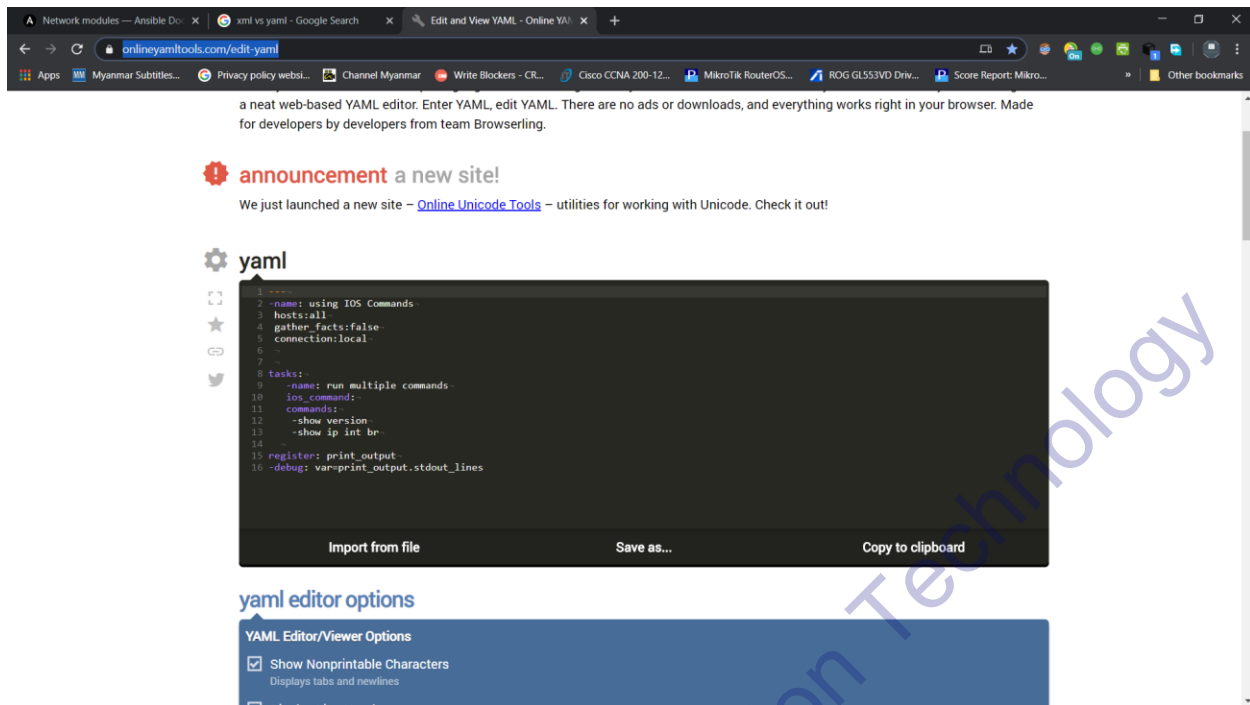
အခုကျနော်တို့ အဓိကလုပ်သွားမှာကတော့ ios_command module ကိုစမ်းပြသွားမှာဖြစ်ပါတယ်။

Ansible playbook ရေးတဲ့နေရာမှာ YAML Script ကိုအသုံးပြုပါတယ်။ တနည်းအားဖြင့် programming language ပေါ့ဗျာ။ XMLနဲ့လည်းအသုံးပြုလို့ရပါတယ်။ ဒါပေမယ့် YAML ကပို လွယ်ပြီး human readable ဖြစ်တဲ့အတွက် ကျနော် YAML နဲ့ပဲ ရေးပြပါမယ်။

XML	JSON	YAML
<pre><Servers> <Server> <name>Server1</name> <owner>John</owner> <created>123456</created> <status>active</status> </Server> </Servers></pre>	<pre>{ Servers: [{ name: Server1, owner: John, created: 123456, status: active }] }</pre>	<pre>Servers: - name: Server1 owner: John created: 123456 status: active</pre>

ကျနော့်အမြင်မှာတော့ YAML က JSON ထက်ပိုပြီးတော့တောင် ရေးရတာလွယ်ကူတယ်လို့ထင်ပါတယ်။ ဒါပေမယ့် Web development သမားတွေနဲ့ တခြား html ရေးဖူးတဲ့ လူတွေကတော့ XML ကိုသိပ်ပြီး စိမ်းမှာမဟုတ်ပါဘူး။

ကျနော်ဒီမှာတော့ network automation ကိုလေ့လာတာဖြစ်တဲ့အတွက် YAML ရဲ့အသေးစိတ်ကိုတော့ ထည့်ပြီး ရှင်းလင်းသွားမှာမဟုတ်ပါဘူး။ ဒါပေမယ့် နောင်လာမယ့် ကျနော်ရဲ့ နောက်လာမယ့်networking age 3 နဲ့ 4 စာစုတွေမှာ အလျဉ်းသင့်သလို ရှင်းလင်းသွားမှာဖြစ်ပါတယ်။ ios_config module သုံးပြီး ospf configuration လုပ်တာကိုလည်း နောင်တွင် video file နဲ့ရှင်းပြသွားမှာဖြစ်ပါတယ်။ အခုကျနော် ios_command module သုံးပြီး ကျနော်တို့ router 2 လုံးရဲ့ version, interface နဲ့ ssh information တွေကိုသိရမယ့် playbook လေးတခုရေးထားပါတယ်။ Codes တွေကို ကျနော် google drive မှာ ဒီ pdf နဲ့အတူတခါတည်းတင်ပေးမှာဖြစ်ပါတယ်။ ဒါပေမယ့် တကယ်ရေးတတ်သုံးတတ်ချင်ရင်တော့ ကိုယ်တိုင် ရေးကျိပါလို့suggestion ပေးပါရစေ။ နောက်ပြီး YAML က space လေးတချက်လွဲတာကအစ sensitive ဖြစ်ပြီး error တက်တဲ့ အတွက် ကျနော်တို့ online YAML editor website မှာအရင် စမ်းရေးကျိပြီးမှ copy ကူးပြီး playbook ပြန်ရေးဖို့ အကြံပြုပါရစေ။ <https://onlineyamlttools.com/edit-yaml> ဒါကတော့ YAML Editor website ဖြစ်ပါတယ်။



```

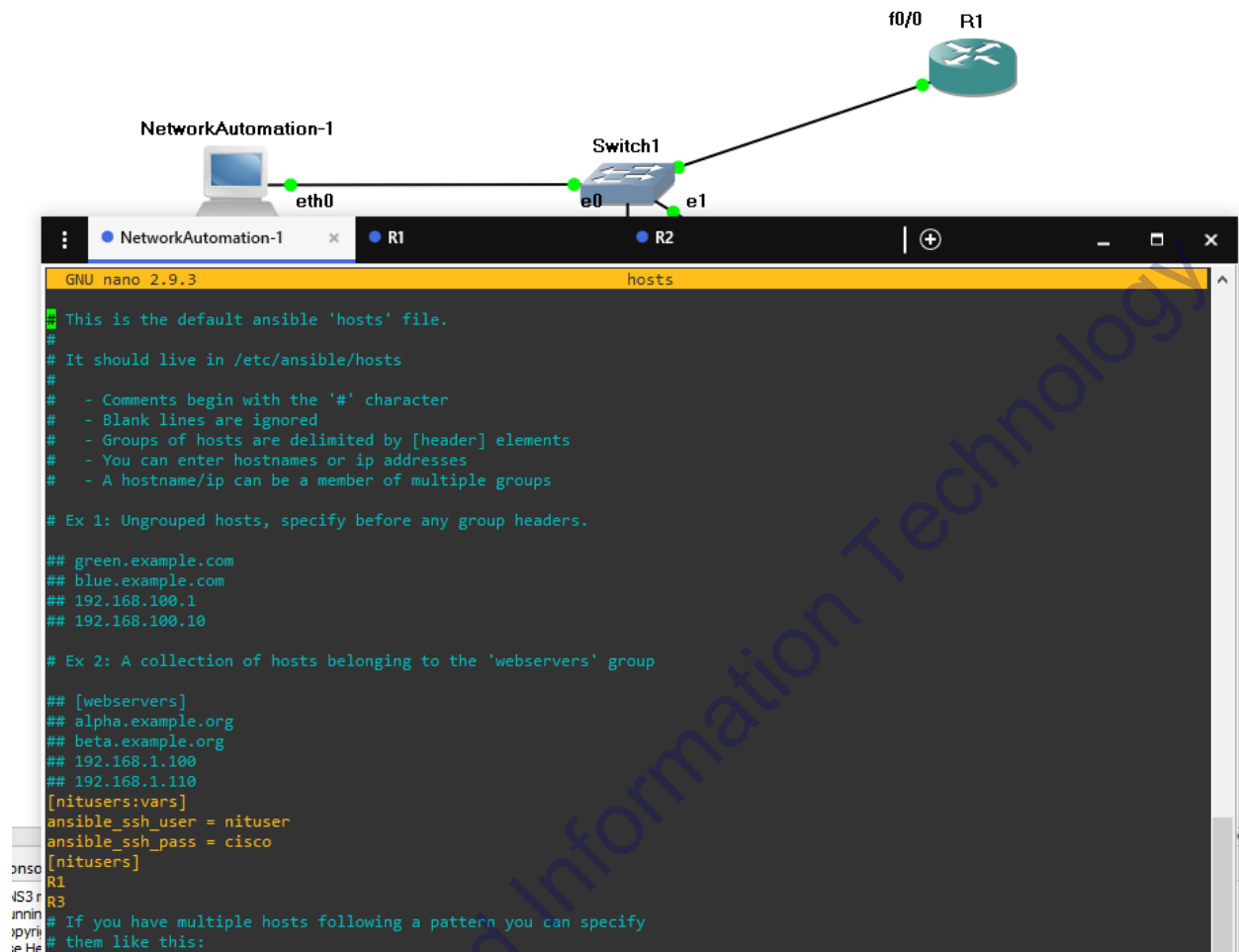
root@NetworkAutomation-1:/etc/ansible# cat getinfo.yml
---
- name: Backup show run (enable mode commands)
  hosts: all
  gather_facts: false
  connection: local

  tasks:
    - name: run enable level commands
      ios_command:
        authorize: yes
        commands:
          - show version | i IOS
          - show ip int br
          - show ip ssh
      register: print_output

    - debug: var=print_output.stdout_lines

```

ဒီ code လေးကို nano getinfo.yml ဆိုပြီး file location တခုခုမှာ save လိုက်ပါ။ ပြီးရင် ကျနော်တို့ အရင် ad hoc command တုန်းကလို username password တွေထည့်စရာမလိုတော့အောင် မိမိ router ရဲ့ ssh username နဲ့ password ကို ကျနော်တို့ /etc/ansible/hosts file မှာသွားပြီးထည့်လိုက်ပါ။



ပြီးရင်တော့ ansible-playbook getinfo.ymlဆိုပြီး playbook file ကို run လိုက်ပါ။ error မရှိဘူးဆိုရင်တော့ အခုလိုမျိုးပြပါလိမ့်မယ်။

```
root@NetworkAutomation-1:/etc/ansible# ansible-playbook getinfo.yml
```

```
[WARNING]: Skipping plugin (/usr/lib/python2.7/dist-packages/ansible/plugins/connection/napalm.py)
as it seems to be
```

```
invalid: invalid syntax (cisco_base_connection.py, line 143)
```

```
PLAY [Backup show run (enable mode commands)]
```

```
*****
```

```
TASK [run enable level commands]
```

```
*****
```

```
*****
```

```
[DEPRECATION WARNING]: Distribution Ubuntu 18.04 on host R3 should use /usr/bin/python3, but is
using /usr/bin/python for
```

```
backward compatibility with prior Ansible releases. A future Ansible release will default to using the
discovered platform
```

```
python for this host. See
```

```
https://docs.ansible.com/ansible/2.9/reference\_appendices/interpreter\_discovery.html for more
```

```
information. This feature will be removed in version 2.12. Deprecation warnings can be disabled by
setting
```

```
deprecation_warnings=False in ansible.cfg.
```

```
ok: [R3]
```

```
[DEPRECATION WARNING]: Distribution Ubuntu 18.04 on host R1 should use /usr/bin/python3, but is
using /usr/bin/python for
```

```
backward compatibility with prior Ansible releases. A future Ansible release will default to using the
discovered platform
```

```
python for this host. See
```

```
https://docs.ansible.com/ansible/2.9/reference\_appendices/interpreter\_discovery.html for more
```

```
information. This feature will be removed in version 2.12. Deprecation warnings can be disabled by
setting
```

```
deprecation_warnings=False in ansible.cfg.
```

```
ok: [R1]
```

```
TASK [debug]
```

```
*****
```

```
*****
```

```

ok: [R1] => {
  "print_output.stdout_lines": [
    [
      "Cisco IOS Software, 3700 Software (C3725-ADVENTERPRISEK9-M), Version 12.4(15)T7, RELEASE
      SOFTWARE (fc3)"
    ],
    [
      "Interface      IP-Address    OK? Method Status      Protocol",
      "FastEthernet0/0    192.168.122.10 YES manual up          up",
      "Serial0/0           unassigned   YES unset  administratively down down",
      "FastEthernet0/1     unassigned   YES unset  administratively down down",
      "Serial0/1           unassigned   YES unset  administratively down down",
      "Serial0/2           unassigned   YES unset  administratively down down",
      "Serial0/3           unassigned   YES unset  administratively down down",
      "FastEthernet1/0     unassigned   YES unset  administratively down down",
      "Serial2/0           unassigned   YES unset  administratively down down",
      "Serial2/1           unassigned   YES unset  administratively down down",
      "Serial2/2           unassigned   YES unset  administratively down down",
      "Serial2/3           unassigned   YES unset  administratively down down"
    ],
    [
      "SSH Enabled - version 2.0",
      "Authentication timeout: 120 secs; Authentication retries: 3"
    ]
  ]
}

ok: [R3] => {
  "print_output.stdout_lines": [
    [

```

"Cisco IOS Software, 3700 Software (C3725-ADVENTERPRISEK9-M), Version 12.4(15)T7, RELEASE SOFTWARE (fc3)"

```
],
[
  "Interface      IP-Address  OK? Method Status      Protocol",
  "FastEthernet0/0    192.168.122.20 YES manual up          up      ",
  "Serial0/0          unassigned  YES unset  administratively down down    ",
  "FastEthernet0/1     unassigned  YES unset  administratively down down    ",
  "Serial0/1           unassigned  YES unset  administratively down down    ",
  "Serial0/2           unassigned  YES unset  administratively down down    ",
  "Serial0/3           unassigned  YES unset  administratively down down    ",
  "FastEthernet1/0     unassigned  YES unset  administratively down down    ",
  "Serial2/0           unassigned  YES unset  administratively down down    ",
  "Serial2/1           unassigned  YES unset  administratively down down    ",
  "Serial2/2           unassigned  YES unset  administratively down down    ",
  "Serial2/3           unassigned  YES unset  administratively down down    "
],
[
  "SSH Enabled - version 2.0",
  "Authentication timeout: 120 secs; Authentication retries: 3"
]
]
}
```

PLAY RECAP

```
*****
*****
```

R1 : ok=2 changed=0 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

R3 : ok=2 changed=0 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

ကျနော်တို့ စာလည်းအရမ်းများနေပြီဆိုတော့ ဒီ code အလုပ်လုပ်ပုံကို ကျနော်နောက်မှ post တင်ပြီး ပြန်ရှင်းပြပါမယ်။ အဓိကလုပ်ရမှာကတော့ ဒီ lab လေးကို ပြန်စမ်းကြည့်ပါ ၊ basic lab လေးဖြစ်တဲ့အတွက် နောက်ထပ်ပြီး လည်း code အသစ်တွေ command အသစ်တွေ သက်ဆိုင်ရာ website တွေမှာ လေ့လာနိုင်ပါတယ်။ ပြီးတော့ မိတ်ဆွေတို့အနေနဲ့ error တစ်ခုတည်း တက်တာမျိုးဖြစ်ဖြစ် တခြား Network programmability အကြောင်းသိချင်လို့ဖြစ်ဖြစ် ကျနော်တို့ page ရဲ့ mailbox မှာလာရောက်မေးမြန်းနိုင်ပါတယ်ခင်ဗျာ။

ကျေးဇူးတင်လျက်

Networking and information technology

Networking and Information Technology