Ansible Setup on DebianOS + Device Configuration on CML

This will be a guide for setting up Ansible on DebianOs. It should work in theory on any UNIX, but here is a link to the official documentation for your reference. This initial doc will be to confirm reachability to a device via Ansible. Pushing configs through playbooks will come later.

First, we need to get Ansible installed on our device, crazy right? Install it using "*pip install ansible*". If you're on a fresh Debian/Linux VM, you might receive this error:

```
root@DEBIAN-DT:~# pip install ansible
-bash: pip: command not found
```

Install pip using the command below. Use the following command to verify installation.

```
root@DEBIAN-DT:/home/kadeem/debianVM/Ansible# sudo apt-get install python3-pip
root@DEBIAN-DT:/home/kadeem/debianVM/Ansible# python3 -m pip
```

Install Ansible.

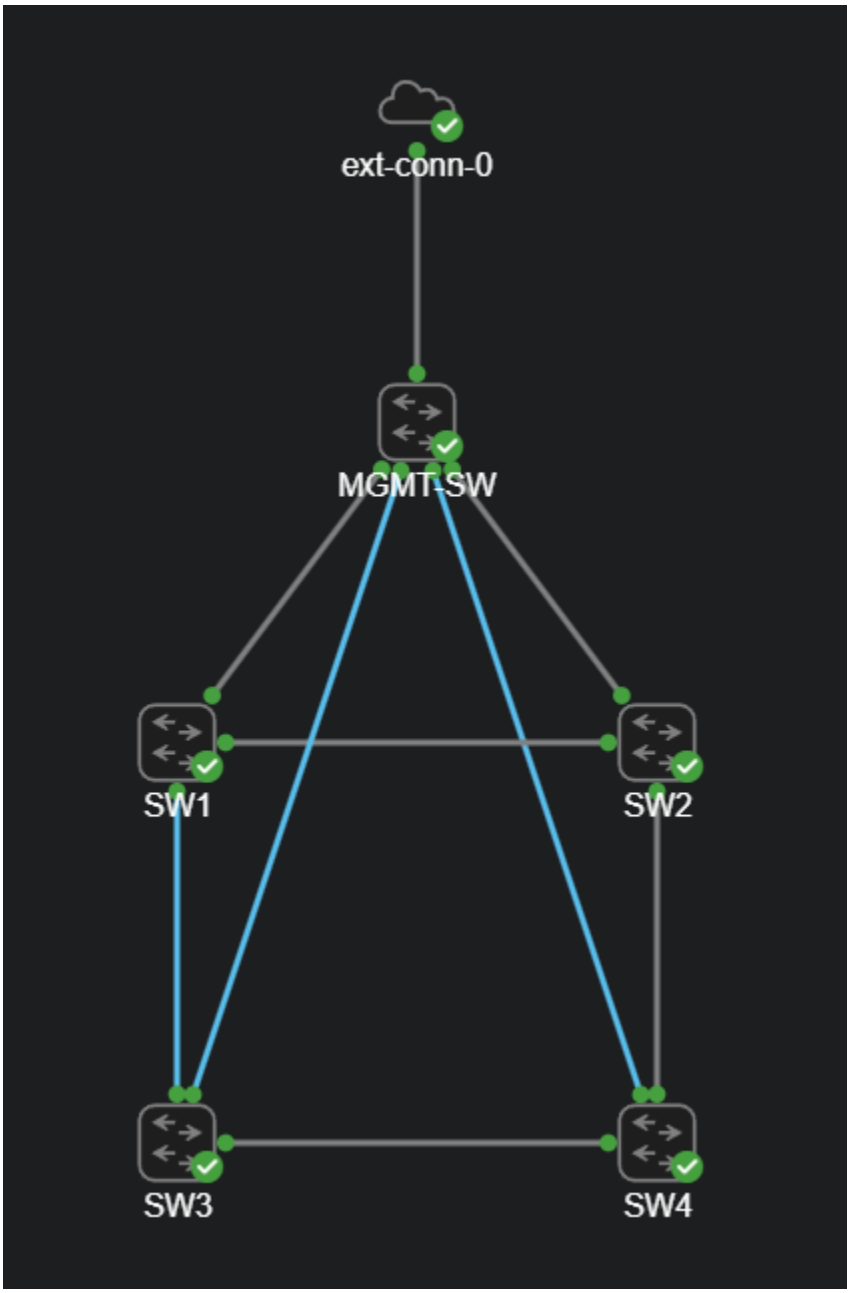
```
root@DEBIAN-DT:~# apt install ansible
```

Verify Ansible installation.

```
root@DEBIAN-DT:~# ansible --version
ansible [core 2.14.3]
  config file = None
  configured module search path = ['/root/.ansible/plugins/modules',
'/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  ansible collection location =
/root/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/bin/ansible
  python version = 3.11.2 (main, Mar 13 2023, 12:18:29) [GCC 12.2.0]
(/usr/bin/python3)
  jinja version = 3.1.2
  libyaml = True
```

Cool, we have our device running Ansible. Now we proceed to the part that we actually care about, using it to automate configuration in our CML lab. This should work on other virtual environments like EVE-NG and GNS3 as long as the hosts are reachable via SSH. Below is the lab I'm going to be working on.

My Ansible control node is hosted on a VM outside of CML.



Let's start with "MGMT-SW". Configure it so that it's pingable from the Control node. This is what mine looks like:

```
interface GigabitEthernet0/0
 no switchport
 ip address 10.0.0.100 255.255.255.0
 negotiation auto
 no cdp enable
-
 *Jun  7 18:32:41.589: ICMP: echo reply sent, src 10.0.0.100, dst 10.0.0.22, topol-
ogy BASE, dscp 0 topoid 0
*Jun  7 18:32:42.591: ICMP: echo reply sent, src 10.0.0.100, dst 10.0.0.22, topology
BASE, dscp 0 topoid 0
*Jun  7 18:32:43.593: ICMP: echo reply sent, src 10.0.0.100, dst 10.0.0.22, topology
BASE, dscp 0 topoid 0
```

Enable SSH:

Now we can proceed to configuring SSH.

```
MGMT-SW(config)#ip domain-name LAB
MGMT-SW(config)#crypto key generate rsa modulus 2048
The name for the keys will be: MGMT-SW.LAB

% The key modulus size is 2048 bits
% Generating 2048 bit RSA keys, keys will be non-exportable...
[OK] (elapsed time was 1 seconds)

MGMT-SW(config)#ip ssh ver 2
MGMT-SW(config)#username kadeem privilege 15
MGMT-SW(config)#enable secret cisco
MGMT-SW(config)#line vty 0 4
MGMT-SW(config-line)#login local
MGMT-SW(config-line)#transport input ssh
```

This is the error we get when attempting to SSH. From what I've read, its because the IOS images on CML run a dated key exchange method that's been deprecated because of security concerns. No biggie, it's a lab, so we ignore that.

```
kadeem@DEBIAN-DT:~/.ssh$ ssh 10.0.0.100
Unable to negotiate with 10.0.0.100 port 22: no matching key exchange method found.
Their offer: diffie-hellman-group-exchange-sha1,diffie-hellman-group14-sha1,diffie-
hellman-group1-sha1
```

Go to your control node and navigate to the /~/.ssh/config file. Create one if it's not already there. Add this.

```
Host <Hostname/IP>
   KexAlgorithms +<KeyExchange algorithms>
```

Mine looks like:

```
Host 10.0.0.100
   KexAlgorithms +diffie-hellman-group-exchange-sha1,diffie-hellman-group14-
sha1,diffie-hellman-group1-sha1
```

The next test produces this error:

```
Unable to negotiate with 10.0.0.100 port 22: no matching host key type found. Their
offer: ssh-rsa
```

Same thing as before, CML IOS runs a dated host key algo so we have to specify to use that for the host. Under the entry you created before add

```
HostKeyAlgorithms +ssh-rsa
```

End result:

```
Host 10.0.0.100
   KexAlgorithms +diffie-hellman-group-exchange-sha1,diffie-hellman-group14-
sha1,diffie-hellman-group1-sha1
   HostKeyAlgorithms +ssh-rsa
```

Cool, we have access.

```
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.0.100' (RSA) to the list of known hosts.

IOSv - Cisco Systems Confidential -

Supplemental End User License Restrictions

This IOSv software is provided AS-IS without warranty of any kind. Under no
circumstances may this software be used separate from the Cisco Modeling Labs
Software that this software was provided with, or deployed or used as part of a
production environment.

By using the software, you agree to abide by the terms and conditions of the Cisco
End User License Agreement at http://www.cisco.com/go/eula. Unauthorized use or
distribution of this software is expressly prohibited.
(kadeem@10.0.0.100) Password:
```

If you noticed in the config I used to enable SSH, I didn't set a password for the user. I'm going to allow authentication via public keys only, so I'll go into that next.

Authentication via public keys:

To allow access via public keys, the server(The switch), has to have the key of the client(DebianHost) that is requesting access. So, let's generate the key on the client that will be shared. This is lengthy(pause), so I'll get straight to the point.

Generate key on the client:

```
ssh-keygen -b 4096 -t rsa
```

Structure it so that it's readable by the switch:

```
fold -b -w 72 id_rsa.pub
```

Configure the switch to only accept public keys for authentication. Add the key to the switch:

```
MGMT-SW(config)#ip ssh server algorithm authentication publickey
MGMT-SW(config)#ip ssh pubkey-chain
MGMT-SW(conf-ssh-pubkey)#username kadeem
MGMT-SW(conf-ssh-pubkey-user)#key-string
MGMT-SW(conf-ssh-pubkey-data)#<Enter Key Here>
MGMT-SW(conf-ssh-pubkey-data)#exit
```

Confirm SSH Key fingerprint on the client and switch are identical.

On client:

```
kadeem@DEBIAN-DT:~/.ssh$ ssh-keygen -l -E md5 -f id_rsa.pub
4096 MD5:68:29:e6:96:fb:57:45:5a:4b:91:f7:54:1b:ba:36:11 kadeem@DEBIAN-DT (RSA)
```

On server:

```
MGMT-SW#sh run | sec pubkey
ip ssh pubkey-chain
  username kadeem
   key-hash ssh-rsa 6829E696FB57455A4B91F7541BBA3611
```

PS: Not sure if showing those keys is a security risk but its all good. I don't have anything, hack me bro.

Now if we test SSH, I should get access automatically without the need for a password.(Lab purposes)

```
kadeem@DEBIAN-DT:~/.ssh$ ssh kadeem@10.0.0.100

IOSv - Cisco Systems Confidential -

Supplemental End User License Restrictions

This IOSv software is provided AS-IS without warranty of any kind. Under no
circumstances may this software be used separate from the Cisco Modeling Labs
Software that this software was provided with, or deployed or used as part of a
production environment.

By using the software, you agree to abide by the terms and conditions of the Cisco
End User License Agreement at http://www.cisco.com/go/eula. Unauthorized use or
distribution of this software is expressly prohibited.
kadeem@10.0.0.100: Permission denied (publickey)
```

Kidding, go back to the SSH config file at /~/.ssh/config and add this line underneath the host:

```
PubKeyAcceptedAlgorithms=ssh-rsa
```

Shoutout to David, show him love here

End result:

```
kadeem@DEBIAN-DT:~/.ssh$ ssh kadeem@10.0.0.100

IOSv - Cisco Systems Confidential -

Supplemental End User License Restrictions

This IOSv software is provided AS-IS without warranty of any kind. Under no
circumstances may this software be used separate from the Cisco Modeling Labs
Software that this software was provided with, or deployed or used as part of a
production environment.

By using the software, you agree to abide by the terms and conditions of the Cisco
End User License Agreement at http://www.cisco.com/go/eula. Unauthorized use or
distribution of this software is expressly prohibited.

IOSv - Cisco Systems Confidential -
Supplemental End User License Restrictions

This IOSv software is provided AS-IS without warranty of any kind. Under no
circumstances may this software be used separate from the Cisco Modeling Labs
Software that this software was provided with, or deployed or used as part of a
production environment.

By using the software, you agree to abide by the terms and conditions of the Cisco
End User License Agreement at http://www.cisco.com/go/eula. Unauthorized use or
distribution of this software is expressly prohibited.

MGMT-SW#
```

Finally, Ansible.(I know, sorry.)

Now we actually do Ansible stuff. First, create your host file. I'm using YAML format since that's on the CCNP ENCOR and it's best practice for when you start managing larger environments. The path that Ansible points to by default is /etc/ansible/hosts so let's create it there.

This is how my inventory file looks.

```
---
LabEnvir:
  hosts:
    MGMT-SW:
      ansible_host: 10.0.0.100
...
```

To explain:

- LabEnvir: is the group where our hosts reside.
- hosts: is pretty obvious, it's where the hosts are.
    - Within this category, we can specify the names we will use to call the hosts in our Ansible commands. I used MGMT-SW
- ansible_host: is the IP address assigned to the host

For more info, reference the documentation.

To confirm Ansible is working, we'll run an Ad-hoc command and save the configuration changes we made. Result, following.

```
kadeem@DEBIAN-DT:/etc/ansible/hosts$ ansible MGMT-SW -m raw -a "write memory"
MGMT-SW | CHANGED | rc=0 >>

Building configuration...
Compressed configuration from 3427 bytes to 2001 bytes[OK]
IOSv - Cisco Systems Confidential -

Supplemental End User License Restrictions

This IOSv software is provided AS-IS without warranty of any kind. Under no
circumstances may this software be used separate from the Cisco Modeling Labs
Software that this software was provided with, or deployed or used as part of a
production environment.

By using the software, you agree to abide by the terms and conditions of the Cisco
End User License Agreement at http://www.cisco.com/go/eula. Unauthorized use or
distribution of this software is expressly prohibited.
Shared connection to 10.0.0.100 closed.
```

Playbooks coming next. Thanks for reading!