Samantha Jackson
CSCI4406
10-30-2024

CLab 7 - Address Resolution Protocol (ARP)

start _here.ipynb

Observations/Issues: None. Shows the FABlib Config table which contains information about the Credential Manager, Orchestrator, Project ID, Token File, Bastion File, Username, Private Key, Slice Private and Public Key Files, Log Files and Levels, SSH Command Line, Version, Data Dir, Core API, and Bastion SSH Config File.

Configure environment

[1]:
```
from fabrictestbed_extensions.fablib.fablib import FablibManager as fablib_manager
fablib = fablib_manager()
conf = fablib.show_config()
```

FABlib Config

Credential Manager	cm.fabric-testbed.net
Orchestrator	orchestrator.fabric-testbed.net
Project ID	a70de2f5-9e12-4b6b-b412-0ae1a2c553b0
Token File	/home/fabric/.tokens.json
Bastion Host	bastion.fabric-testbed.net
Bastion Username	sjack012_0000240143
Bastion Private Key File	/home/fabric/work/fabric_config/fabric_bastion_key
Slice Private Key File	/home/fabric/work/fabric_config/slice_key
Slice Public Key File	/home/fabric/work/fabric_config/slice_key.pub
Log File	/tmp/fablib/fablib.log
Log Level	INFO
Sites to avoid	
SSH Command Line	ssh -i {{ _self_.private_ssh_key_file }} -F /home/fabric/work/fabric_config/ssh_config {{ _self_.username }}@{{ _self_.management_ip }}
Version	1.7.3
Data directory	/tmp/fablib
Core API	uis.fabric-testbed.net
Bastion SSH Config File	/home/fabric/work/fabric_config/ssh_config

Observations/Issues: None. Python script for checking and reserving resources. Try-except blocks for exception handling.

[2]:
```
slice_name="wireshark-" + fablib.get_bastion_username()

node_conf = [
 {'name': "romeo",   'cores': 2, 'ram': 4, 'disk': 10, 'image': 'default_ubuntu_22', 'packages': []},
 {'name': "juliet",  'cores': 2, 'ram': 4, 'disk': 10, 'image': 'default_ubuntu_22', 'packages': []},
 {'name': "hamlet",  'cores': 2, 'ram': 4, 'disk': 10, 'image': 'default_ubuntu_22', 'packages': []},
]
net_conf = [
 {"name": "net0", "subnet": "10.0.0.0/24", "nodes": [{"name": "romeo",   "addr": "10.0.0.100"}, {"name": "juliet", "addr": "10.0.0.101"}, {"name": "hamlet", "addr": "10.0.0.102"}]}
]
route_conf = [ ]
exp_conf = {'cores': sum([ n['cores'] for n in node_conf]), 'nic': sum([len(n['nodes']) for n in net_conf]) }
```

[3]:
```
try:
    slice = fablib.get_slice(slice_name)
    print("You already have a slice by this name!")
    print("If you previously reserved resources, skip to the 'log in to resources' section.")
except:
    print("You don't have a slice named %s yet." % slice_name)
    print("Continue to the next step to make one.")
    slice = fablib.new_slice(name=slice_name)
```

Observations/Issues (Reserving Resources): None. We'll reserve our resources on EDUKY changing this code. The Site "EDUCKY" table shows information about the location "301 Hilltop Avenue,Lexington, KY 40506". Detailed statistics about site resources are also included with availability, capacity, and allocations of the Cores, RAM, and Disk as well as information about the network.

[6]:
```
# while True:
#     site_name = fablib.get_random_site()
#     if ( (fablib.resources.get_core_available(site_name) > 1.2*exp_conf['cores']) and
#         (fablib.resources.get_component_available(site_name, 'SharedNIC-ConnectX-6') > 1.2**exp_conf['nic']) ):
#         break

site_name = "EDUKY"
print(site_name)
fablib.show_site(site_name)
```

```
EDUKY
```

Site

Name	EDUKY
State	Active
Address	301 Hilltop Avenue,Lexington, KY 40506
Location	(38.0325, -84.502801)
PTP Capable	True
Hosts	18
CPUs	36
Cores Available	73596
Cores Capacity	73728
Cores Allocated	132
Ram Available	8228
Ram Capacity	8604
Ram Allocated	376
Disk Available	133252
Disk Capacity	134082
Disk Allocated	830
Basic NIC Available	4327
Basic NIC Capacity	4445
Basic NIC Allocated	118
P4-Switch Available	0
P4-Switch Capacity	0
P4-Switch Allocated	0
ConnectX-6 Available	0
ConnectX-6 Capacity	0

ConnectX-6 Allocated	0
ConnectX-5 Available	0
ConnectX-5 Capacity	0
ConnectX-5 Allocated	0
NVMe Available	0
NVMe Capacity	0
NVMe Allocated	0
Tesla T4 Available	0
Tesla T4 Capacity	0
Tesla T4 Allocated	0
RTX6000 Available	0
RTX6000 Capacity	0
RTX6000 Allocated	0
A30 Available	0
A30 Capacity	0
A30 Allocated	0
A40 Available	0
A40 Capacity	0
A40 Allocated	0
U280 Available	0
U280 Capacity	0
U280 Allocated	0

[6]: '<pandas.io.formats.style.Styler object at 0x7f6a4468bf90>'

Observations/Issues (Resources Hosts and Network Segments): None.

[7]:
```
# this cell sets up the nodes
for n in node_conf:
    slice.add_node(name=n['name'], site=site_name,
                   cores=n['cores'],
                   ram=n['ram'],
                   disk=n['disk'],
                   image=n['image'])
```

[8]:
```
# this cell sets up the network segments
for n in net_conf:
    ifaces = [slice.get_node(node["name"]).add_component(model="NIC_Basic",
                                                         name=n["name"]).get_interfaces()[0] for node in n['nodes'] ]
    slice.add_l2network(name=n["name"], type='L2Bridge', interfaces=ifaces)
```

[9]:
```
slice.submit()
```

```
Retry: 10, Time: 250 sec
```

Slice

ID	3b20d99a-7acc-4a77-836c-19d3bddcdde1
Name	wireshark-sjack012_0000240143
Lease Expiration (UTC)	2024-10-29 19:20:09 +0000
Lease Start (UTC)	2024-10-28 19:20:09 +0000
Project ID	a70de2f5-9e12-4b6b-b412-0ae1a2c553b0
State	StableOK

Nodes

ID	Name	Cores	RAM	Disk	Image	Image Type	Host	Site	Username	Management IP	State	Error	SSH Command	Public SSH Key File	Private SSH Key File
344e3af6-cd27-4794-8bc3-c259c093fe19	hamlet	2	4	10	default_ubuntu_22	qcow2	eduky-w15.fabric-testbed.net	EDUKY	ubuntu	2610:1e0:1700:206:f816:3efffece:688	Active		ssh -i /home/fabric/work/fabric_config/slice_key -F /home/fabric/work/fabric_config/ssh_config ubuntu@2610:1e0:1700:206:f816:3efffece:688	/home/fabric/work/fabric_config/slice_key.pub	/home/fabric/work/fabric_config/slice_key
ac639069-9c68-4805-b231-4b3145cdd5c4	juliet	2	4	10	default_ubuntu_22	qcow2	eduky-w14.fabric-testbed.net	EDUKY	ubuntu	2610:1e0:1700:206:f816:3efffeb3:8cc0	Active		ssh -i /home/fabric/work/fabric_config/slice_key -F /home/fabric/work/fabric_config/ssh_config ubuntu@2610:1e0:1700:206:f816:3efffeb3:8cc0	/home/fabric/work/fabric_config/slice_key.pub	/home/fabric/work/fabric_config/slice_key
7a9cfc79-29de-4439-a730-906918d0f75d	romeo	2	4	10	default_ubuntu_22	qcow2	eduky-w12.fabric-testbed.net	EDUKY	ubuntu	2610:1e0:1700:206:f816:3efffe56:753e	Active		ssh -i /home/fabric/work/fabric_config/slice_key -F /home/fabric/work/fabric_config/ssh_config ubuntu@2610:1e0:1700:206:f816:3efffe56:753e	/home/fabric/work/fabric_config/slice_key.pub	/home/fabric/work/fabric_config/slice_key

Networks

ID	Name	Layer	Type	Site	Subnet	Gateway	State	Error
85f68d32-a382-4c0a-b1bd-014fc8f923df	net0	L2	L2Bridge	EDUKY	None	None	Active	

Interfaces

Name	Short Name	Node	Network	Bandwidth	Mode	VLAN	MAC	Physical Device	Device	IP Address	Numa Node	Switch Port
romeo-net0-p1	p1	romeo	net0	100	config		0A:4A:A6:5E:29:6A	enp7s0	enp7s0	fe80::84ara6fffe5e:296a	1	HundredGigE0/0/0/30
juliet-net0-p1	p1	juliet	net0	100	config		02:D4:3A:85:9E:BF	enp7s0	enp7s0	fe80::d4:3afffe85:9ebf	1	HundredGigE0/0/0/14
hamlet-net0-p1	p1	hamlet	net0	100	config		0E:E1:E8:88:B5:98	enp7s0	enp7s0	fe80::ce1:e8fffe88:b59b	1	HundredGigE0/0/0/33

```
Time to print interfaces 264 seconds
```

[9]:
```
'3b20d99a-7acc-4a77-836c-19d3bddcdde1'
```


[10]:
```
slice.get_state()
slice.wait_ssh(progress=True)
```

```
Waiting for slice . Slice state: StableOK
Waiting for ssh in slice . ssh successful
```

[10]:
```
True
```

Observations/Issues (Configuring Resources): None. Shows the “host” file that has the names and addresses of the nodes, and enables IPv4 forwarding.

[11]:
```
slice = fablib.get_slice(name=slice_name)
```

[12]:
```
# install packages
# this will take a while and will run in background while you do other steps
for n in node_conf:
    if len(n['packages']):
        node = slice.get_node(n['name'])
        pkg = " ".join(n['packages'])
        node.execute_thread("sudo apt update; sudo apt -y install %s" % pkg)
```

[13]:
```
# bring interfaces up and either assign an address (if there is one) or flush address
from ipaddress import ip_address, IPv4Address, IPv4Network

for net in net_conf:
    for n in net['nodes']:
        if_name = n['name'] + '-' + net['name'] + '-p1'
        iface = slice.get_interface(if_name)
        iface.ip_link_up()
        if n['addr']:
            iface.ip_addr_add(addr=n['addr'], subnet=IPv4Network(net['subnet']))
        else:
            iface.get_node().execute("sudo ip addr flush dev %s"  % iface.get_device_name())
```

[14]:
```
# prepare a "hosts" file that has names and addresses of every node
hosts_txt = [ "%s\t%s" % ( n['addr'], n['name'] ) for net in net_conf  for n in net['nodes'] if type(n) :
for n in slice.get_nodes():
    for h in hosts_txt:
        n.execute("echo %s | sudo tee -a /etc/hosts" % h)
```

```
10.0.0.100 romeo
10.0.0.101 juliet
10.0.0.102 hamlet
10.0.0.100 romeo
10.0.0.101 juliet
10.0.0.102 hamlet
10.0.0.100 romeo
10.0.0.101 juliet
10.0.0.102 hamlet
```

[15]:
```
# enable IPv4 forwarding on all nodes
for n in slice.get_nodes():
    n.execute("sudo sysctl -w net.ipv4.ip_forward=1")
```

```
net.ipv4.ip_forward = 1
net.ipv4.ip_forward = 1
net.ipv4.ip_forward = 1
```

[16]:
```
# set up static routes
for rt in route_conf:
    for n in rt['nodes']:
        slice.get_node(name=n).ip_route_add(subnet=IPv4Network(rt['addr']), gateway=rt['gw'])
```

[17]:
```
# turn off segmentation offload on interfaces
for iface in slice.get_interfaces():
    iface_name = iface.get_device_name()
    n = iface.get_node()
    offloads = ["gro", "lro", "gso", "tso"]
    for offload in offloads:
        n.execute("sudo ethtool -K %s %s off" % (iface_name, offload))
```

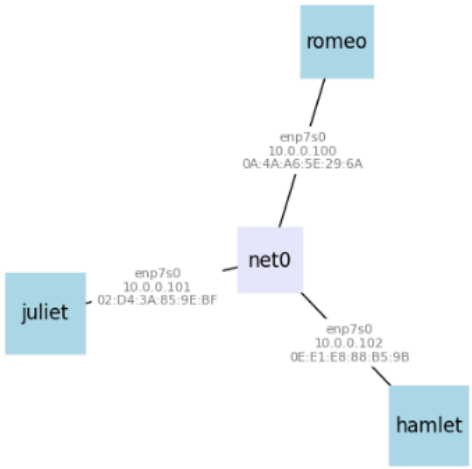
Observations/Issues (Drawing the Network Topology): None. Displays network visualization code with the networkx Python library. Shows nodes romeo, juliet, hamlet, net0.

Draw the network topology

The following cell will draw the network topology, for your reference. The interface name and addresses of each experiment interface will be shown on the drawing.

[18]:
```
l2_nets = [(n.get_name(), {'color': 'lavender'}) for n in slice.get_l2networks() ]
l3_nets = [(n.get_name(), {'color': 'pink'}) for n in slice.get_l3networks() ]
hosts   =   [(n.get_name(), {'color': 'lightblue'}) for n in slice.get_nodes()]
nodes = l2_nets + l3_nets + hosts
ifaces = [iface.toDict() for iface in slice.get_interfaces()]
edges = [(iface['network'], iface['node'],
          {'label': iface['physical_dev'] + '\n' + iface['ip_addr'] + '\n' + iface['mac']}) for iface in ifaces]
```

[19]:
```
import networkx as nx
import matplotlib.pyplot as plt
plt.figure(figsize=(len(nodes),len(nodes)))
G = nx.Graph()
G.add_nodes_from(nodes)
G.add_edges_from(edges)
pos = nx.spring_layout(G)
nx.draw(G, pos, node_shape='s',
        node_color=[n[1]['color'] for n in nodes],
        node_size=[len(n[0])*400 for n in nodes],
        with_labels=True);
nx.draw_networkx_edge_labels(G,pos,
                             edge_labels=nx.get_edge_attributes(G,'label'),
                             font_color='gray',  font_size=8, rotate=False);
```



Observations/Issues (Log into Resources): None. Checked ping on Juliet. Not going to delete slice based on the fact that I need it for CLab8.

Log into resources

Now, we are finally ready to log in to our resources over SSH! Run the following cells, and observe the table output - you will see an SSH command for each of the resources in your topology.

[20]:
```
import pandas as pd
pd.set_option('display.max_colwidth', None)
slice_info = [{'Name': n.get_name(), 'SSH command': n.get_ssh_command()} for n in slice.get_nodes()]
pd.DataFrame(slice_info).set_index('Name')
```

[20]:

Name	SSH command
romeo	ssh -i /home/fabric/work/fabric_config/slice_key -F /home/fabric/work/fabric_config/ssh_config ubuntu@2610:1e0:1700:206:f816:3eff:fe56:753e
juliet	ssh -i /home/fabric/work/fabric_config/slice_key -F /home/fabric/work/fabric_config/ssh_config ubuntu@2610:1e0:1700:206:f816:3eff:feb3:8cc0
hamlet	ssh -i /home/fabric/work/fabric_config/slice_key -F /home/fabric/work/fabric_config/ssh_config ubuntu@2610:1e0:1700:206:f816:3eff:fece:688

```
fabric@spring:arp-18%$ ssh -i /home/fabric/work/fabric_config/slice_key -F /home/fabric/work/fabric_config/ssh_config ubuntu@2610:1e0:1700:206:f816:3eff:fe56:753e
Warning: Permanently added 'bastion.fabric-testbed.net' (ED25519) to the list of known hosts.
Warning: Permanently added '2610:1e0:1700:206:f816:3eff:fe56:753e' (ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 5.15.0-112-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Mon Oct 28 19:57:24 UTC 2024

  System load:             0.0
  Usage of /:              15.2% of 9.51GB
  Memory usage:            6%
  Swap usage:              0%
  Processes:               143
  Users logged in:         0
  IPv4 address for enp3s0: 10.30.7.142
  IPv6 address for enp3s0: 2610:1e0:1700:206:f816:3eff:fe56:753e


Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


The list of available updates is more than a week old.
To check for new updates run: sudo apt update
New release '24.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.
```

```
ubuntu@romeo:~$ ping -c 10 10.0.0.101
PING 10.0.0.101 (10.0.0.101) 56(84) bytes of data.
64 bytes from 10.0.0.101: icmp_seq=1 ttl=64 time=0.239 ms
64 bytes from 10.0.0.101: icmp_seq=2 ttl=64 time=0.093 ms
64 bytes from 10.0.0.101: icmp_seq=3 ttl=64 time=0.095 ms
64 bytes from 10.0.0.101: icmp_seq=4 ttl=64 time=0.092 ms
64 bytes from 10.0.0.101: icmp_seq=5 ttl=64 time=0.099 ms
64 bytes from 10.0.0.101: icmp_seq=6 ttl=64 time=0.098 ms
64 bytes from 10.0.0.101: icmp_seq=7 ttl=64 time=0.080 ms
64 bytes from 10.0.0.101: icmp_seq=8 ttl=64 time=0.093 ms
64 bytes from 10.0.0.101: icmp_seq=9 ttl=64 time=0.099 ms
64 bytes from 10.0.0.101: icmp_seq=10 ttl=64 time=0.098 ms

--- 10.0.0.101 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9210ms
rtt min/avg/max/mdev = 0.080/0.108/0.239/0.043 ms
```

This lab taught me about how to use the Address resolution protocol (ARP) in IPv4 networks. I learned about resource management/configuration within the FABRIC testbed. I allocated computing resources from the EDUKY site. I created a simple topology with nodes (romeo, juliet, hamlet, and net0). I also analyzed these nodes by SSH'ing and pinging neighboring nodes. Overall I garnered a better understanding of the FABRIC testbed in practice.