Tutorial: Take a Test Drive with the Cloud Native Network Function (CNF) Testbed
ONS Europe 2019 - Antwerp Belgium
Monday, September 23, 2019

Workstation information:
  ● SSH username: onseu19

Use case #1 - NSM single node packet filter

Connect to the correct cluster
  1. Type: source choose_k8s_cluster
  2. Type: my
  3. Type kubectl get nodes.
You should see something like:

$ kubectl get nodes
NAME                               STATUS     ROLES    AGE    VERSION
ons19-tutws1-master1     Ready   master   110d   v1.14.2
ons19-tutws1-worker1-1  Ready   <none>   110d   v1.14.2

Deploy Steps
  1. helm init --service-account tiller
  2. Install NSM service by running: helm install --namespace=nsm-system nsm/
  3. Deploy the packet filter use case by running

Use case #2 - Physical NIC GW with shared access
This is a shared use case.
● Find someone who has the same group number and worker node. Eg. group number 3 and worker node 1
● One person from the group can deploy the service chain at a time

Deploy Steps
1. Type source choose_k8s_cluster
2. Type: shared
3. Type the Worker Node number from your handout
4. Go into cnf-testbed/examples/use_case/external-packet-filtering-on-k8s-nsm-on-packet/helm
5. Deploying Packet-filtering Service Chains by using your Group Number and Worker Node
   ○ helm install ./pfchain/ --set input.id=<Group Number>,input.server=<Worker Node>
   ○ Example: helm install ./pfchain/ --set input.id=1,input.server=5
6. Wait for deployment to complete
   ○ It will take a little while (~30 seconds) for the helm command to run, and following that it might take another 30-60 seconds before all the containers are deployed.
7. Ignore any complaints from helm regarding a NetworkService already exists because the gateway Network Function was deployed ahead of time.
   ○ Safely ignore the following: Error: release roiling-seal failed: networkservices.networkservicemesh.io "packet-filtering-X" already exists
8. Checking what services are running with
   ○ helm list
   ○ kubectl get pods
9. Test connectivity from both clients towards the external endpoint
   ○ Go to use_case/external-packet-filtering-on-k8s-nsm-on-packet folder
   ○ Run ./ONS_check_ext_connectivity.sh <Group Number> <Worker Node>

If successful you should be output similar to the below:

$ ./ONS_check_ext_connectivity.sh 3 5
===== >>>>> PROCESSING simple-client-3-6f6b647cd7-72dn9       <<<< ===========
Try 1
PING 10.60.1.60 (10.60.1.60): 56 data bytes
64 bytes from 10.60.1.60: seq=1 ttl=62 time=4.051 ms
64 bytes from 10.60.1.60: seq=2 ttl=62 time=4.201 ms
64 bytes from 10.60.1.60: seq=3 ttl=62 time=4.262 ms
64 bytes from 10.60.1.60: seq=4 ttl=62 time=4.253 ms
64 bytes from 10.60.1.60: seq=5 ttl=62 time=4.219 ms
64 bytes from 10.60.1.60: seq=6 ttl=62 time=4.227 ms
64 bytes from 10.60.1.60: seq=7 ttl=62 time=4.238 ms
64 bytes from 10.60.1.60: seq=8 ttl=62 time=4.208 ms
64 bytes from 10.60.1.60: seq=9 ttl=62 time=4.228 ms
64 bytes from 10.60.1.60: seq=10 ttl=62 time=4.230 ms

--- 10.60.1.60 ping statistics ---
11 packets transmitted, 10 packets received, 9% packet loss
round-trip min/avg/max = 4.051/4.210/4.262 ms
NSC simple-client-3-6f6b647cd7-72dn9 with IP 10.52.3.5/30 pinging External Physical Server
TargetIP: 10.60.1.60 successful
All check OK. NSC simple-client-3-6f6b647cd7-72dn9 behaving as expected.

Try 1
116 bytes from 10.60.1.60: icmp_seq=1 ttl=62 time=5.4053 ms
116 bytes from 10.60.1.60: icmp_seq=2 ttl=62 time=4.3089 ms
116 bytes from 10.60.1.60: icmp_seq=3 ttl=62 time=4.3061 ms

Statistics: 3 sent, 3 received, 0% packet loss
NSC ucnf-client-3-d89b59577-gb7tf with IP 10.52.3.1/30 pinging External Physical Server
TargetIP: 10.60.1.60 successful