

LoxiLB 5G SCTP Load Balancing

2023년 7월

안종석

james@jslab.kr



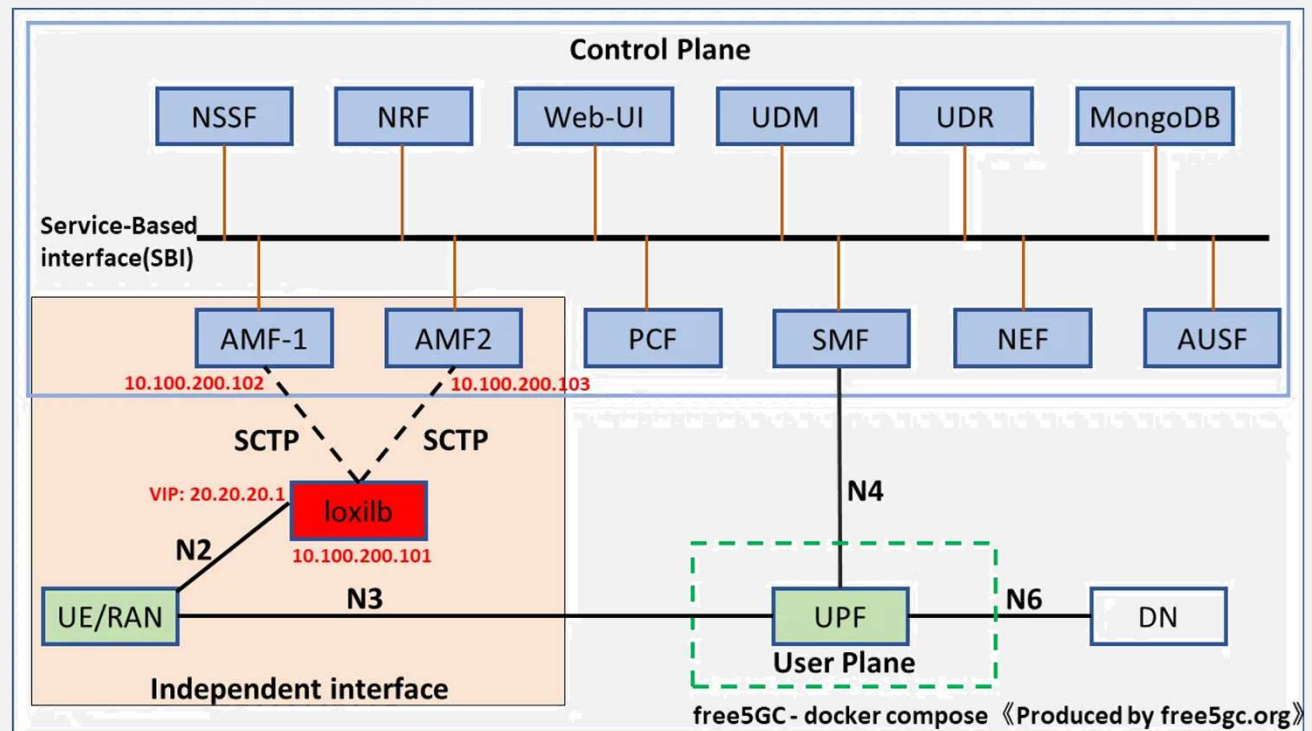
james@jslab.kr

JS Lab

5G SCTP LOADBALANCER

❖ sctplb: LoxiLB Sctp Load Balancing

- Reference for free5GC: 5G SCTP LoadBalancer Using LoxiLB Applying



Source: <https://medium.com/@ben0978327139/5g-sctp-loadbalancer-using-loxilb-applying-on-free5gc-b5c05bb723f0>



PREREQUISITES

❖ sctplb: LoxiLB SCTP Load Balancing

• Prerequisites

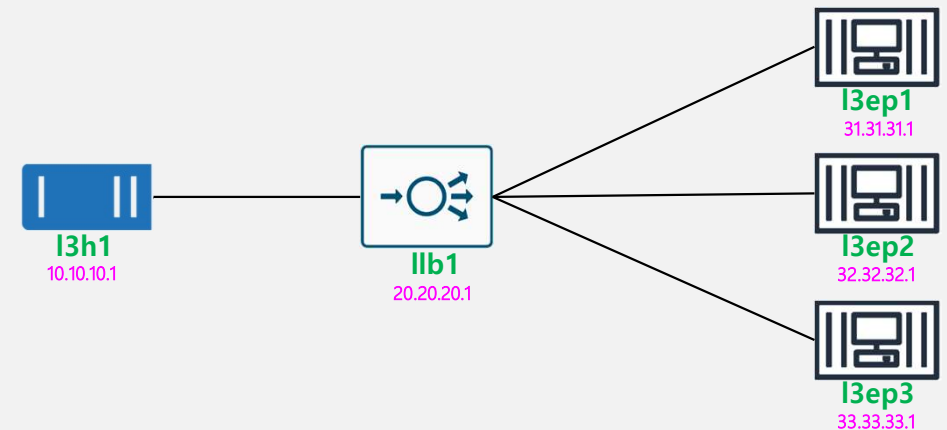
- `sudo git clone https://github.com/loxilb-io/loxilb.git`
- `sudo apt install net-tools docker.io nodejs libsctp1`
- `sudo su - root` ### ('control' + d for exit)
- `cd /home/jslab/loxilb/cicd/sctplb`
- `chmod +x file_name`

• LoxiLB Configuration (구성)

- `./rmconfig.sh` ### Check remove configuration
- `./config.sh` ### Check configuration
- `./validation.sh` ### Check validation source

• Operations

```
jslab@jslab:~/loxilb/cicd/sctplb$ sudo su - root
root@jslab:/home# cd /home/jslab/loxilb/cicd/sctplb
root@jslab:/home/jslab/loxilb/cicd/sctplb# dir
config.sh rmconfig.sh validation.sh
root@jslab:/home/jslab/loxilb/cicd/sctplb# chmod +x config.sh
root@jslab:/home/jslab/loxilb/cicd/sctplb# ./config.sh
```



Source: <https://github.com/loxilb-io/loxilb/tree/main/cicd/sctplb>

Reference: <https://killercoda.com/netlox/scenario/loxilb-sctplb>



LOXILB CONFIGURATION

❖ sctplb: LoxiLB SCTP Load Balancing

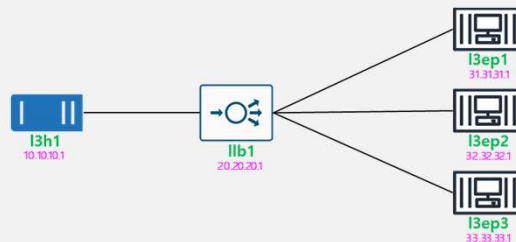
• Prerequisites

- `sudo git clone https://github.com/loxilb-io/loxilb.git`
- `sudo apt install net-tools docker.io nodejs libsctp1`
- `sudo su - root ### ('control' + d for exit)`
- `cd /home/jslab/loxilb/cicd/sctplb`
- `chmod +x file_name`

• LoxiLB Configuration (구성)

- `./rmconfig.sh ### Check remove configuration`
- `./config.sh ### Check configuration`
- `./validation.sh ### Check validation source`

• Operations



```

root@jslab: /home/jslab/loxilb/cicd/sctplb# ./config.sh
#####
Spawning all hosts
#####
Spawning l1b1 (loxilb)
645197ff1f21b3c7a43549f4393b49d6dff77948f82942c3539161d371606a2f
12782
Spawning l3h1 (host)
fc8a5b89738d945610226c1dedac639e4bf703add8c57aaa3d8c896e6174b72
12953
Spawning l3ep1 (host)
56630b95311f8ffe3313549d699ae5bd673628a79e48971773422e0b7b94cb89
13067
Spawning l3ep2 (host)
00aa0efc8fff429831c8341907ce2204544b947f9e7be4b5f13ed0b1f4efb4e9
13320
Spawning l3ep3 (host)
f02d2b14f4efb8c471c62605fdd21d7de601d85d5e9227bc1603910012511171
13436
#####
Connecting and configuring hosts
#####
l1b1: loxicmd create lb 20.20.20.1 --sctp=2020:8080 --endpoints=31.31.31.1:1,32.32.32.1:1,33.33.33.1:1
ProtoPortpair: map[sctp:[2020:8080]]
Debug: response.StatusCode: 200
Success
l1b1: loxicmd create lb 20.20.20.1 --tcp=2020:8080 --endpoints=31.31.31.1:1,32.32.32.1:1,33.33.33.1:1
ProtoPortpair: map[tcp:[2020:8080]]
Debug: response.StatusCode: 200
Success
Debug: response.StatusCode: 200
Success
Debug: response.StatusCode: 200
Success
Debug: response.StatusCode: 200
Success
Debug: response.StatusCode: 200
Success
Debug: response.StatusCode: 200
Success
  
```

Source: <https://github.com/loxilb-io/loxilb/blob/main/cicd/sctplb/config.sh>



LOXILB CONFIGURATION

❖ sctplb: LoxiLB SCTP Load Balancing

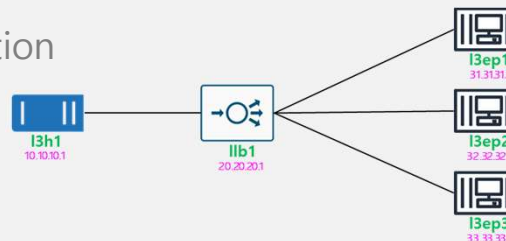
• Prerequisites

- `sudo git clone https://github.com/loxilb-io/loxilb.git`
- `sudo apt install net-tools docker.io nodejs libsctp1`
- `sudo su - root ### ('control' + d for exit)`
- `cd /home/jslab/loxilb/cicd/sctplb`
- `chmod +x file_name`

• LoxiLB Configuration (구성)

- `./rmconfig.sh ### Check remove configuration`
- `./config.sh ### Check configuration`
- `./validation.sh ### Check validation source`

• Operations



Bash: "Bourne Again SHell"의 줄임말이며 유닉스 및 유사한 운영 체제에서 일반적으로 사용되는 명령줄 셸 및 스크립팅 언어이다. 명령줄 셸인 Bash는 운영 체제와 상호 작용할 수 있는 텍스트 인터페이스를 제공하며, 셸에 명령을 입력하면 Bash가 해당 명령을 해석하고 실행한다. 명령 기록, 탭 완성 및 스크립팅 기능과 같은 기능을 제공하여 작업을 자동화하고 시스템을 관리하는 데 도구이다.

Source: <https://github.com/loxilb-io/loxilb/blob/main/cicd/sctplb/config.sh>

```

#!/bin/bash
source ../common.sh

echo "#####"
echo "Spawning all hosts"
echo "#####"

spawn_docker_host --dock-type loxilb --dock-name llb1
spawn_docker_host --dock-type host --dock-name l3h1
spawn_docker_host --dock-type host --dock-name l3ep1
spawn_docker_host --dock-type host --dock-name l3ep2
spawn_docker_host --dock-type host --dock-name l3ep3

echo "#####"
echo "Connecting and configuring hosts"
echo "#####"

connect_docker_hosts l3h1 llb1
connect_docker_hosts l3ep1 llb1
connect_docker_hosts l3ep2 llb1
connect_docker_hosts l3ep3 llb1

#L3 config
config_docker_host --host1 l3h1 --host2 llb1 --ptype phy --addr 10.10.10.1/24 --gw 10.10.10.254
config_docker_host --host1 l3ep1 --host2 llb1 --ptype phy --addr 31.31.31.1/24 --gw 31.31.31.254
config_docker_host --host1 l3ep2 --host2 llb1 --ptype phy --addr 32.32.32.1/24 --gw 32.32.32.254
config_docker_host --host1 l3ep3 --host2 llb1 --ptype phy --addr 33.33.33.1/24 --gw 33.33.33.254
config_docker_host --host1 llb1 --host2 l3h1 --ptype phy --addr 10.10.10.254/24
config_docker_host --host1 llb1 --host2 l3ep1 --ptype phy --addr 31.31.31.254/24
config_docker_host --host1 llb1 --host2 l3ep2 --ptype phy --addr 32.32.32.254/24
config_docker_host --host1 llb1 --host2 l3ep3 --ptype phy --addr 33.33.33.254/24
sleep 5
create_lb_rule llb1 20.20.20.1 --sctp=2020:8080 --
endpoints=31.31.31.1,32.32.32.1,33.33.33.1
create_lb_rule llb1 20.20.20.1 --tcp=2020:8080 --
endpoints=31.31.31.1,32.32.32.1,33.33.33.1

#We use this test case with iperf too, and iperf doesn't work well with probing, so
changing it to ping
$dexec llb1 loxicmd create ep 31.31.31.1 --name=31.31.31.1_tcp_8080 --probttype=ping
$dexec llb1 loxicmd create ep 32.32.32.1 --name=32.32.32.1_tcp_8080 --probttype=ping
$dexec llb1 loxicmd create ep 33.33.33.1 --name=31.31.31.1_tcp_8080 --probttype=ping

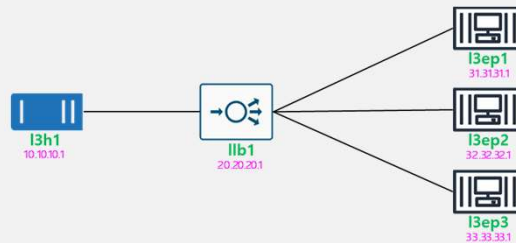
$dexec llb1 loxicmd create ep 31.31.31.1 --name=31.31.31.1_sctp_8080 --probttype=ping
$dexec llb1 loxicmd create ep 32.32.32.1 --name=32.32.32.1_sctp_8080 --probttype=ping
$dexec llb1 loxicmd create ep 33.33.33.1 --name=31.31.31.1_sctp_8080 --probttype=ping
  
```



VALIDATION

❖ sctplb: LoxiLB SCTP Load Balancing

- Prerequisites
- LoxiLB Configuration (구성)
 - `./rmconfig.sh` ### Check remove configuration
 - `./config.sh` ### Check remove configuration
 - `./validation.sh` ### Check validation source
- Operations



```

root@jslab:/home/jslab/loxilb/cicd/sctplb# ./validation.sh
SCENARIO-sctplb
server1 UP
server2 UP
server3 UP
server1
server2
server3
server1
server2
server3
server1
server2
server3
server1
server2
server3
SCENARIO-sctplb [OK]
root@jslab:/home/jslab/loxilb/cicd/sctplb#
  
```

Source: <https://github.com/loxilb-io/loxilb/blob/main/cicd/sctplb/validation.sh>



OPERATIONS

❖ sctplb: LoxiLB SCTP Load Balancing

- Prerequisites
- LoxiLB Configuration (구성)
 - ./rmconfig.sh ### check remove configuration
 - ./config.sh
 - ./validation.sh ### check validation source
- Operations
 - **docker exec -it llb1 loxicmd help**
 - ip netns exec llb1 ping 31.31.31.1
 - ip netns exec llb1 ping 32.32.32.1
 - ip netns exec llb1 ping 33.33.33.1
 - ip netns exec llb1 ping 10.10.10.1
 - ip netns exec llb1 route -n
 - **docker exec -it llb1 bash**
 - **loxicmd get lb -o wide**
 - ./validation.sh ### check validation source

```
root@jslab:/home/jslab/loxiLB/cicd/sctplb# docker exec -it llb1 loxicmd help
loxicmd is the command-line tool for loxiLB. It is equivalent of "kubectl" for loxiLB. loxicmd provides
the following (currently) :
  - Create/Delete/Get - Service type external load-balancer, Vlan, Vxlan, Qos Policies, Endpoint
  client, FDB, IPaddress, Neighbor, Route, Firewall, Mirror, Session, UICI
  - Get Port(interface) dump used by loxiLB or its docker
  - Get Connection track (TCP/UDP/ICMP/SCTP) information
loxicmd aim to provide all of the configuration for the loxiLB.

Usage:
  loxicmd [command]

Available Commands:
  apply      Apply configuration
  completion Generate completion script
  create     Create a Load balance features in the LoxiLB.
  delete     Delete a Load balance features in the LoxiLB.
  get        Get a Load balance features from LoxiLB.
  help       Help about any command
  save       saves current configuration
  set        Set configurations
  version    Get a version

Flags:
  -s, --apiserver string  Set API server IP address (default "127.0.0.1")
  -h, --help               help for loxicmd
  -o, --output string      Set output layer (ex.) wide, json
  -p, --port int16         Set API server port number (default 11111)
  --protocol string        Set API server http/https (default "http")
  -t, --timeout int16      Set timeout (default 10)

Use "loxicmd [command] --help" for more information about a command.
root@jslab:/home/jslab/loxiLB/cicd/sctplb#
```

Source: <https://github.com/loxiLB-io/loxiLB/tree/main/cicd/sctplb> Reference: root@8b74b5ddc4d2:/# loxicmd create lb 20.20.20.1 --sctp=2020:8080 --endpoints=31.31.31.1:1,32.32.32.1:1,33.33.33.1:1



IP OPERATIONS

❖ sctplb: LoxiLB SCTP Load Balancing

- Prerequisites
- LoxiLB Configuration (구성)
- Operations
 - `docker exec -it llb1 loxicmd help`
 - `ip netns exec llb1 ping 31.31.31.1`
 - `ip netns exec llb1 ping 32.32.32.1`
 - `ip netns exec llb1 ping 33.33.33.1`
 - `ip netns exec llb1 ping 10.10.10.1`
 - `ip netns exec llb1 route -n`
 - `docker exec -it llb1 bash`
 - `loxicmd get lb -o wide`
 - `./validation.sh ### check validation source`

```
root@jslab:/home/jslab/loxilb/cicd/sctplb# ip netns exec llb1 ping 31.31.31.1
PING 31.31.31.1 (31.31.31.1) 56(84) bytes of data.
64 bytes from 31.31.31.1: icmp_seq=1 ttl=64 time=0.033 ms
64 bytes from 31.31.31.1: icmp_seq=2 ttl=64 time=0.096 ms
64 bytes from 31.31.31.1: icmp_seq=3 ttl=64 time=0.119 ms
^C
--- 31.31.31.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2056ms
rtt min/avg/max/mdev = 0.033/0.082/0.119/0.036 ms
root@jslab:/home/jslab/loxilb/cicd/sctplb# ip netns exec llb1 ping 32.32.32.1
PING 32.32.32.1 (32.32.32.1) 56(84) bytes of data.
64 bytes from 32.32.32.1: icmp_seq=1 ttl=64 time=0.028 ms
64 bytes from 32.32.32.1: icmp_seq=2 ttl=64 time=0.162 ms
64 bytes from 32.32.32.1: icmp_seq=3 ttl=64 time=0.252 ms
64 bytes from 32.32.32.1: icmp_seq=4 ttl=64 time=0.041 ms
^C
--- 32.32.32.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3038ms
rtt min/avg/max/mdev = 0.028/0.120/0.252/0.092 ms
root@jslab:/home/jslab/loxilb/cicd/sctplb# ip netns exec llb1 ping 33.33.33.1
PING 33.33.33.1 (33.33.33.1) 56(84) bytes of data.
64 bytes from 33.33.33.1: icmp_seq=1 ttl=64 time=0.055 ms
64 bytes from 33.33.33.1: icmp_seq=2 ttl=64 time=0.100 ms
64 bytes from 33.33.33.1: icmp_seq=3 ttl=64 time=0.096 ms
^C
--- 33.33.33.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2034ms
rtt min/avg/max/mdev = 0.055/0.083/0.100/0.020 ms
root@jslab:/home/jslab/loxilb/cicd/sctplb# ip netns exec llb1 ping 10.10.10.1
PING 10.10.10.1 (10.10.10.1) 56(84) bytes of data.
64 bytes from 10.10.10.1: icmp_seq=1 ttl=64 time=0.031 ms
64 bytes from 10.10.10.1: icmp_seq=2 ttl=64 time=0.096 ms
64 bytes from 10.10.10.1: icmp_seq=3 ttl=64 time=0.101 ms
^C
--- 10.10.10.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2029ms
rtt min/avg/max/mdev = 0.031/0.076/0.101/0.
```

Source: <https://github.com/loxilb-io/loxilb/tree/main/cicd/sctplb> Reference: `root@8b74b5ddc4d2:~# loxicmd create lb 20.20.20.1 --sctp=2020:8080 --endpoints=31.31.31.1:1,32.32.32.1:1,33.33.33.1:1`



IP OPERATIONS

❖ sctplb: LoxiLB SCTP Load Balancing

- Prerequisites
- LoxiLB Configuration (구성)
- Operations
 - `docker exec -it llb1 loxicmd help`
 - `ip netns exec llb1 ping 31.31.31.1`
 - `ip netns exec llb1 ping 32.32.32.1`
 - `ip netns exec llb1 ping 33.33.33.1`
 - `ip netns exec llb1 ping 10.10.10.1`
 - `ip netns exec llb1 route -n`
 - `docker exec -it llb1 bash`
 - `loxicmd get lb -o wide`
 - `./validation.sh ### check validation source`

```
root@jslab:/home/jslab/loxilb/cicd/sctplb# ip netns exec llb1 route -n
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
0.0.0.0 172.17.0.1 0.0.0.0 UG 0 0 0 eth0
10.10.10.0 0.0.0.0 255.255.255.0 U 0 0 0 e11b113h1
31.31.31.0 0.0.0.0 255.255.255.0 U 0 0 0 e11b113ep1
32.32.32.0 0.0.0.0 255.255.255.0 U 0 0 0 e11b113ep2
33.33.33.0 0.0.0.0 255.255.255.0 U 0 0 0 e11b113ep3
172.17.0.0 0.0.0.0 255.255.0.0 U 0 0 0 eth0
root@jslab:/home/jslab/loxilb/cicd/sctplb#
```

Source: <https://github.com/loxilb-io/loxilb/tree/main/cicd/sctplb> Reference: `root@8b74b5ddc4d2:/# loxicmd create lb 20.20.20.1 --sctp=2020:8080 --endpoints=31.31.31.1:1,32.32.32.1:1,33.33.33.1:1`



LOXICMD OPERATIONS

❖ sctplb: LoxiLB SCTP Load Balancing

- Prerequisites
- LoxiLB Configuration (구성)
- Operations
 - `docker exec -it llb1 loxicmd help`
 - `ip netns exec llb1 ping 31.31.31.1`
 - `ip netns exec llb1 ping 32.32.32.1`
 - `ip netns exec llb1 ping 33.33.33.1`
 - `ip netns exec llb1 ping 10.10.10.1`
 - `ip netns exec llb1 route -n`
 - `docker exec -it llb1 bash`
 - `loxicmd get lb -o wide`
 - `./validation.sh ### check validation source`

```

root@jslab:/home/jslab/loxilb/cicd/sctplb# docker exec -it llb1 loxicmd help
loxicmd is the command-line tool for loxilb. It is equivalent of "kubect1" for loxilb. loxicmd provides the following (currently) :
- Create/Delete/Get - Service type external load-balancer, Vlan, Vxlan, Qos Policies, Endpoint client, FDB, IPaddress, Neighbor,
Route, Firewall, Mirror, Session, UICI
- Get Port(interface) dump used by loxilb or its docker
- Get Connection track (TCP/UDP/ICMP/SCTP) information
loxicmd aim to provide all of the configuration for the loxilb.

Usage:
  loxicmd [command]

Available Commands:
  apply      Apply configuration
  completion Generate completion script
  create     Create a Load balance features in the LoxiLB.
  delete     Delete a Load balance features in the LoxiLB.
  get       Get a Load balance features from LoxiLB.
  help      Help about any command
  save      saves current configuration
  set       Set configurations
  version   Get a version

Flags:
  -s, --apiserver string  Set API server IP address (default "127.0.0.1")
  -h, --help              help for loxicmd
  -o, --output string     Set output layer (ex.) wide, json
  -p, --port int16        Set API server port number (default 11111)
  --protocol string       Set API server http/https (default "http")
  -t, --timeout int16     Set timeout (default 10)

Use "loxicmd [command] --help" for more information about a command.

```

```

root@jslab:/home/jslab/loxilb/cicd/sctplb# docker exec -it llb1 bash
root@645197ff1f21:/# loxicmd get lb -o wide

```

EXTERNAL IP	SECONDARY IPS	PORT	PROTOCOL	BLOCK	SELECT	MODE	ENDPOINT IP	TARGET PORT	WEIGHT	STATE
20.20.20.1		2020	sctp	0	rr	default	31.31.31.1	8080	1	-
							32.32.32.1	8080	1	-
							33.33.33.1	8080	1	-
20.20.20.1		2020	tcp	0	rr	default	31.31.31.1	8080	1	-
							32.32.32.1	8080	1	-
							33.33.33.1	8080	1	-

```

root@645197ff1f21:/#

```

Source: <https://github.com/loxilb-io/loxilb/tree/main/cicd/sctplb> Reference: root@8b74b5ddc4d2:/# loxicmd create lb 20.20.20.1 --sctp=2020:8080 --endpoints=31.31.31.1:1,32.32.32.1:1,33.33.33.1:1



./VALIDATION.SH

❖ sctplb: LoxiLB SCTP Load Balancing

- Prerequisites
- LoxiLB Configuration (구성)
- Operations
 - `docker exec -it llb1 loxicmd help`
 - `ip netns exec llb1 ping 31.31.31.1`
 - `ip netns exec llb1 ping 32.32.32.1`
 - `ip netns exec llb1 ping 33.33.33.1`
 - `ip netns exec llb1 ping 10.10.10.1`
 - `ip netns exec llb1 route -n`
 - `docker exec -it llb1 bash`
 - `loxicmd get lb -o wide`
 - `./validation.sh ### check validation source`

Bash: "Bourne Again SHell"의 줄임말이며 유닉스 및 유사한 운영 체제에서 일반적으로 사용되는 명령줄 셸 및 스크립팅 언어이다. 명령줄 셸인 Bash는 운영 체제와 상호 작용할 수 있는 텍스트 인터페이스를 제공하며, 셸에 명령을 입력하면 Bash가 해당 명령을 해석하고 실행한다. 명령 기록, 탭 완성 및 스크립팅 기능과 같은 기능을 제공하여 작업을 자동화하고 시스템을 관리하는 데 도구이다.

Source: <https://github.com/loxi-lb-io/loxi-lb/blob/main/cicd/sctplb/validation.sh>

```
#!/bin/bash
source ../common.sh
echo SCENARIO-sctplb

servArr=( "server1" "server2" "server3" )
ep=( "31.31.31.1" "32.32.32.1" "33.33.33.1" )

$hexec l3ep1 ../common/sctp_server $[ep[0]] 8080 server1 >/dev/null 2>&1 &
$hexec l3ep2 ../common/sctp_server $[ep[1]] 8080 server2 >/dev/null 2>&1 &
$hexec l3ep3 ../common/sctp_server $[ep[2]] 8080 server3 >/dev/null 2>&1 &

sleep 5
code=0
j=0
waitCount=0
while [ $j -le 2 ]
do
  res=$( $hexec l3h1 timeout 10 ../common/sctp_client 10.10.10.1 0 $[ep[j]] 8080 )
  #echo $res
  if [[ $res == "${servArr[j]}" ]]
  then
    echo "$res UP"
    j=$(( j + 1 ))
  else
    echo "Waiting for ${servArr[j]} ($[ep[j]])"
    waitCount=$(( waitCount + 1 ))
    if [[ $waitCount == 10 ]];
    then
      echo "All Servers are not UP"
      echo SCENARIO-sctplb [FAILED]
      sudo pkill sctp_server >/dev/null 2>&1
      exit 1
    fi
  fi
  sleep 1
done

for i in {1..4}
do
  for j in {0..2}
  do
    res=$( $hexec l3h1 timeout 10 ../common/sctp_client 10.10.10.1 0 20.20.20.1 2020 )
    echo -e $res
    if [[ $res != "${servArr[j]}" ]]
    then
      code=1
    fi
  done
  sleep 1
done
done
sudo pkill sctp_server >/dev/null 2>&1
if [[ $code == 0 ]]
then
  echo SCENARIO-sctplb [OK]
else
  echo SCENARIO-sctplb [FAILED]
fi
exit $code
```

