

LoxiLB 사용자 매뉴얼

2023년 3월 5일

안종석

james@jslab.kr



james@jslab.kr

JS Lab

I. LoxiLB 개요 ----- 3

- loxilb 도커 이미지 pull/run 실행하기
- loxilb 로드밸런서 구성하기
- loxilb 로드밸런서 모드

II. LoxiLB 명령어 ----- 15

- 로브밸런서 정보 확인 / 실행 / 구성
- Loxilb 명령어별 실행
- Session Recorder



I. LoxiLB 개요

- loxilb 도커 이미지 pull/run 실행하기
- loxilb 로드밸런서 구성하기
- loxilb 로드밸런서 모드

I. LOXILB 개요

❖ loxilb – 도커 이미지 pull 실행하기

• 도커(docker) 사용

- sudo docker pull ghcr.io/loxilb-io/loxilb:v0.8.3
- sudo docker image ls

```
jslab@jslab:/$ sudo docker pull ghcr.io/loxilb-io/loxilb:v0.8.3
v0.8.3: Pulling from loxilb-io/loxilb
7608715873ec: Already exists
a2980a311ab9: Pull complete
fcbcd64d7060: Pull complete
c439593cb8e3: Pull complete
bf63e60ddae6: Pull complete
ca383162d7c3: Pull complete
bf50c4a78ac6: Pull complete
1095be6b0624: Pull complete
e7b7aea3395c: Pull complete
8f5b94d7d162: Pull complete
4f7792f70a1f: Pull complete
be11cdfdb1f8: Pull complete
bdd1a5489187: Pull complete
abe72ab14729: Pull complete
043f0e76f091: Pull complete
bf1db7052dc7: Pull complete
9fcce4f203f4: Pull complete
8483f4b25fe7: Pull complete
Digest: sha256:36efdab945ea594cea5df5d1015b2292b7d5f83bc30a3b0c3541781e4ebb034d
Status: Downloaded newer image for ghcr.io/loxilb-io/loxilb:v0.8.3
ghcr.io/loxilb-io/loxilb:v0.8.3
jslab@jslab:/$ sudo docker image ls
REPOSITORY          TAG          IMAGE ID          CREATED          SIZE
ghcr.io/loxilb-io/loxilb  v0.8.3      f48ba19fb1ac     4 hours ago     2.31GB
ghcr.io/loxilb-io/loxilb  latest      0e468cb78095     10 hours ago    2.31GB
jslab@jslab:/$
```

Source: <https://loxilb-io.github.io/loxilbdocs/run/>



I. LOXILB 개요

❖ loxilb – 도커 이미지 run 실행하기

• 도커(docker) 사용

- `sudo docker pull ghcr.io/loxilb-io/loxilb:v0.8.3`
- `sudo docker run -u root --cap-add SYS_ADMIN --restart unless-stopped --privileged -dit -v /dev/log:/dev/log --name loxilb ghcr.io/loxilb-io/loxilb:v0.8.3`
- `sudo docker exec -it loxilb bash`

```
jslab@jslab:/$ sudo docker run -u root --cap-add SYS_ADMIN --restart unless-stopped --privileged -dit -v /dev/log:/dev/log --name loxilb ghcr.io/loxilb-io/loxilb:v0.8.3
88e7e0195fbef67e8d0e31764adbec9bd4479f74f9ee131fdc58b2c2868c8c57
jslab@jslab:/$ sudo docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS                    NAMES
88e7e0195fbe   ghcr.io/loxilb-io/loxilb:v0.8.3    "/root/loxilb-io/lox..." About a minute ago Up About a minute    11111/tcp, 22222/tcp    loxilb
jslab@jslab:/$ sudo docker exec -it loxilb bash
root@88e7e0195fbe:/#
```

Source: <https://loxilb-io.github.io/loxilbdocs/run/>



I. LOXILB 개요

❖ loxilb – 실행하기

- 도커(docker) 사용
 - loxicmd create --help

```

root@88e7e0195fbc:/# loxicmd create --help
Create a Load balance features in the LoxilB.

Usage:
  loxicmd create [flags]
  loxicmd create [command]

Available Commands:
  endpoint  Create a LB EndPoint for monitoring
  fdb       Create a FDB
  firewall  Create a Firewall
  ip        Create a IPv4Address
  lb        Create a LoadBalancer
  mirror    Create a Mirror
  neighbor  Create a Neighbors
  policy    Create a Policy
  route     Create a Route
  session   Create a Session
  sessionul Create a Session UICI
  vlan      Create a vlan
  vlanmember Create a vlanmember
  vxlan     Create a vxlan
  vxlanpeer Create a vxlan

Flags:
  -h, --help  help for create

Global Flags:
  -s, --apiserver string  Set API server IP address (default "127.0.0.1")
  -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 5)

Use "loxicmd create [command] --help" for more information about a command.
root@88e7e0195fbc:/#

```

Source: <https://loxilb-io.github.io/loxilbdocs/run/>



I. LOXILB 개요

❖ loxilb - 실행하기

• loxilb 로드밸런서 실행과 구성하기

- loxicmd create lb 1.1.1.1 --tcp=1828:1920 --endpoints=2.2.3.4:1
- loxicmd get lb
- loxicmd get lb -o wide
- loxicmd get lb -o json

```

root@88e7e0195fbe:/# loxicmd create lb 1.1.1.1 --tcp=1828:1920 --endpoints=2.2.3.4:1
ProtoPortpair: map[tcp:[1828:1920]]
Debug: response.StatusCode: 200
Success
root@88e7e0195fbe:/# loxicmd get lb
| EXTERNAL IP | PORT | PROTOCOL | BLOCK | SELECT | MODE | # OF ENDPOINTS |
|-----|-----|-----|-----|-----|-----|-----|
| 1.1.1.1 | 1828 | tcp | 0 | rr | default | 1 |
root@88e7e0195fbe:/# loxicmd get lb -o wide
| EXTERNAL IP | PORT | PROTOCOL | BLOCK | SELECT | MODE | ENDPOINT IP | TARGET PORT | WEIGHT | STATE |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 1.1.1.1 | 1828 | tcp | 0 | rr | default | 2.2.3.4 | 1920 | 1 | active |
root@88e7e0195fbe:/# loxicmd get lb -o json

```

Source: <https://loxilb-io.github.io/loxilbdocs/cmd/>



I. LOXILB 개요

❖ loxilb 의 로브밸런서 기본값 round-robin

• NAT66 (round-robin) load-balancer

- loxicmd create lb 2001::1 --tcp=2020:8080 --endpoints=4ffe::1:1,5ffe::1:1,6ffe::1:1

• NAT64 (round-robin) load-balancer

- loxicmd create lb 2001::1 --tcp=2020:8080 --endpoints=31.31.31.1:1,32.32.32.1:1,33.33.33.1:1

• WRR (Weighted round-robin) load-balancer

- Divide traffic in 40%, 40% and 20% ratio among end-points
- loxicmd create lb 20.20.20.1 --select=priority --tcp=2020:8080 --endpoints=31.31.31.1:40,32.32.32.1:40,33.33.33.1:20

```
root@88e7e0195fbc:/# loxicmd create lb 2001::1 --tcp=2020:8080 --endpoints=4ffe::1:1,5ffe::1:1,6ffe::1:1
ProtoPortpair: map[tcp:[2020:8080]]
Debug: response.StatusCode: 200
Success
root@88e7e0195fbc:/# loxicmd create lb 2001::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
root@88e7e0195fbc:/# loxicmd create lb 20.20.20.1 --select=priority --tcp=2020:8080 --endpoints=31.31.31.1:40,32.32.32.1:40,33.33.33.1:20
ProtoPortpair: map[tcp:[2020:8080]]
Debug: response.StatusCode: 200
Success
```

Source: <https://loxilb-io.github.io/loxilbdocs/cmd/>



I. LOXILB 개요

❖ loxilb 의 로브밸런서 기본값 round-robin

- NAT66 (round-robin) load-balancer
 - loxicmd get lb -o wide

```

root@88e7e0195fbe:/# loxicmd get lb
| EXTERNAL IP | PORT | PROTOCOL | BLOCK | SELECT | MODE | # OF ENDPOINTS |
|-----|-----|-----|-----|-----|-----|-----|
| 1.1.1.1 | 1828 | tcp | 0 | rr | default | 1 |
| 2001::1 | 2020 | tcp | 0 | rr | default | 3 |
| 20.20.20.1 | 2020 | tcp | 0 | priority | default | 3 |
root@88e7e0195fbe:/#
root@88e7e0195fbe:/# loxicmd get lb -o wide
| EXTERNAL IP | PORT | PROTOCOL | BLOCK | SELECT | MODE | ENDPOINT IP | TARGET PORT | WEIGHT | STATE |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 1.1.1.1 | 1828 | tcp | 0 | rr | default | 2.2.3.4 | 1920 | 1 | active |
| 2001::1 | 2020 | tcp | 0 | rr | default | 31.31.31.1 | 8080 | 1 | active |
| | | | | | | 32.32.32.1 | 8080 | 1 | active |
| | | | | | | 33.33.33.1 | 8080 | 1 | active |
| 20.20.20.1 | 2020 | tcp | 0 | priority | default | 31.31.31.1 | 8080 | 40 | active |
| | | | | | | 32.32.32.1 | 8080 | 40 | active |
| | | | | | | 33.33.33.1 | 8080 | 20 | active |
root@88e7e0195fbe:/#

```

Source: <https://loxilb-io.github.io/loxilbdocs/cmd/>



I. LOXILB 개요

❖ loxilb 의 로브밸런서 기본값 round-robin

- **Sticky end-point selection load-balancer (select end-points based on traffic hash)**
 - `loxicmd create lb 20.20.20.1 --select=hash --tcp=2020:8080 --endpoints=31.31.31.1:1,32.32.32.1:1,33.33.33.1:1`
- **Load-balancer with forceful tcp-reset session timeout after inactivity of 30s**
 - `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 --in TIMEOUT=30`
- **Load-balancer with one-arm mode**
 - `loxicmd create lb 20.20.20.1 --tcp=2020:8080 --endpoints=100.100.100.2:1,100.100.100.3:1,100.100.100.4:1 --mode=onearm`

```

root@88e7e0195fbc:/# loxicmd create lb 20.20.20.1 --select=hash --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
root@88e7e0195fbc:/# 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 --in TIMEOUT=30
ProtoPortpair: map[tcp:[2020:8080]]
Debug: response.StatusCode: 200
Success
root@88e7e0195fbc:/# loxicmd create lb 20.20.20.1 --tcp=2020:8080 --endpoints=100.100.100.2:1,100.100.100.3:1,100.100.100.4:1 --mode=onearm
ProtoPortpair: map[tcp:[2020:8080]]
Debug: response.StatusCode: 200
Success

```

Source: <https://loxilb-io.github.io/loxilbdocs/cmd/>



I. LOXILB 개요

❖ loxilb 의 로브밸런서 기본값 round-robin

- NAT66 (round-robin) load-balancer
 - loxicmd get lb -o wide

```

root@88e7e0195fbe:/# loxicmd get lb -o wide
| EXTERNAL IP | PORT | PROTOCOL | BLOCK | SELECT | MODE | ENDPOINT IP | TARGET PORT | WEIGHT | STATE |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 1.1.1.1 | 1828 | tcp | 0 | rr | default | 2.2.3.4 | 1920 | 1 | active |
| 2001::1 | 2020 | tcp | 0 | rr | default | 31.31.31.1 | 8080 | 1 | active |
| | | | | | | 32.32.32.1 | 8080 | 1 | active |
| | | | | | | 33.33.33.1 | 8080 | 1 | active |
| 20.20.20.1 | 2020 | tcp | 0 | rr | onearm | 100.100.100.2 | 8080 | 1 | active |
| | | | | | | 100.100.100.3 | 8080 | 1 | active |
| | | | | | | 100.100.100.4 | 8080 | 1 | active |
root@88e7e0195fbe:/#
root@88e7e0195fbe:/# loxicmd get lb
| EXTERNAL IP | PORT | PROTOCOL | BLOCK | SELECT | MODE | # OF ENDPOINTS |
|-----|-----|-----|-----|-----|-----|-----|
| 1.1.1.1 | 1828 | tcp | 0 | rr | default | 1 |
| 2001::1 | 2020 | tcp | 0 | rr | default | 3 |
| 20.20.20.1 | 2020 | tcp | 0 | rr | onearm | 3 |
root@88e7e0195fbe:/#

```

Source: <https://loxilb-io.github.io/loxilbdocs/cmd/>



I. LOXILB 개요

❖ loxilb 의 로브밸런서 fullnat mode / DSR(direct-server return) mode

• Load-balancer with fullnat mode

- loxicmd create lb 88.88.88.1 --sctp=38412:38412 --endpoints=192.168.70.3:1 --mode=fullnat

• Load-balancer config in DSR(direct-server return) mode

- loxicmd create lb 20.20.20.1 --tcp=2020:8080 --endpoints=31.31.31.1:1,32.32.32.1:1 --mode=dsr

• Delete a load-balancer rule

- loxicmd delete lb 1.1.1.1 --tcp=1828

```
root@88e7e0195fbe:/# loxicmd create lb 88.88.88.1 --sctp=38412:38412 --endpoints=192.168.70.3:1 --mode=fullnat
ProtoPortpair: map[sctp:[38412:38412]]
Debug: response.StatusCode: 200
Success
root@88e7e0195fbe:/# loxicmd create lb 20.20.20.1 --tcp=2020:8080 --endpoints=31.31.31.1:1,32.32.32.1:1 --mode=dsr
ProtoPortpair: map[tcp:[2020:8080]]
Error: No port-translation in dsr mode
root@88e7e0195fbe:/#
```

Source: <https://loxilb-io.github.io/loxilbdocs/cmd/>



I. LOXILB 개요

❖ loxilb 의 로브밸런서 정보 확인

• 로브밸런서 정보 확인

- loxicmd get lb -o wide
- loxicmd get lb

```

root@88e7e0195fbe:/# loxicmd get lb -o wide
| EXTERNAL IP | PORT | PROTOCOL | BLOCK | SELECT | MODE | ENDPOINT IP | TARGET PORT | WEIGHT | STATE |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 1.1.1.1 | 1828 | tcp | 0 | rr | default | 2.2.3.4 | 1920 | 1 | active |
| 2001::1 | 2020 | tcp | 0 | rr | default | 31.31.31.1 | 8080 | 1 | active |
| | | | | | | 32.32.32.1 | 8080 | 1 | active |
| | | | | | | 33.33.33.1 | 8080 | 1 | active |
| 20.20.20.1 | 2020 | tcp | 0 | rr | onearm | 100.100.100.2 | 8080 | 1 | active |
| | | | | | | 100.100.100.3 | 8080 | 1 | active |
| | | | | | | 100.100.100.4 | 8080 | 1 | active |
| 88.88.88.1 | 38412 | sctp | 0 | rr | fullnat | 192.168.70.3 | 38412 | 1 | inactive |
root@88e7e0195fbe:/#
root@88e7e0195fbe:/# loxicmd get lb
| EXTERNAL IP | PORT | PROTOCOL | BLOCK | SELECT | MODE | # OF ENDPOINTS |
|-----|-----|-----|-----|-----|-----|-----|
| 1.1.1.1 | 1828 | tcp | 0 | rr | default | 1 |
| 2001::1 | 2020 | tcp | 0 | rr | default | 3 |
| 20.20.20.1 | 2020 | tcp | 0 | rr | onearm | 3 |
| 88.88.88.1 | 38412 | sctp | 0 | rr | fullnat | 1 |
root@88e7e0195fbe:/#

```

Source: <https://loxilb-io.github.io/loxilbdocs/cmd/>



I. LOXILB 개요

❖ loxilb – 실행하기

• loxilb 로드밸런서 실행과 구성하기 (mac-vlan / bridge)

- # Create a mac-vlan (on an underlying interface e.g. ens33)
- docker network create -d macvlan -o parent=ens33 --subnet 172.30.1.0/24 --gateway 172.30.1.254 --aux-address 'host=172.30.1.193' llbnet
- # Run loxilb docker with the created macvlan
- docker run -u root --cap-add SYS_ADMIN --restart unless-stopped --privileged -dit -v /dev/log:/dev/log --net=llbnet --ip=172.30.1.195 --name loxilb ghcr.io/loxilb-io/loxilb:latest
- # If we still want to connect loxilb docker additionally to docker's default "bridge" network or more macvlan networks
- docker network connect bridge loxilb
- docker network connect llbnet2 loxilb --ip=172.30.2.195

```

jslab@jslab:/$ sudo docker run -u root --cap-add SYS_ADMIN --restart unless-stopped --privileged -dit -v /dev/log:/dev/log --name
loxilb ghcr.io/loxilb-io/loxilb:v0.8.3
88e7e0195fbef67e8d0e31764adbec9bd4479f74f9ee131fdc58b2c2868c8c57
jslab@jslab:/$ sudo docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS                NAMES
88e7e0195fbe   ghcr.io/loxilb-io/loxilb:v0.8.3    "/root/loxilb-io/lox...  About a minute ago   Up About a minute   11111/tcp, 22222/tcp   loxilb
jslab@jslab:/$ sudo docker exec -it loxilb bash
root@88e7e0195fbe:/#

```

Source: <https://loxilb-io.github.io/loxilbdocs/run/>



II. LoxiLB 명령어

- 로브밸런서 정보 확인 / 실행 / 구성
- Loxilb 명령어별 실행
- Session Recorder



II. LOXILB 명령어

❖ 로브밸런서 정보 확인

- **Load Balancer: 로브밸런서 정보 확인**
 - loxicmd get lb ##### Get basic information
 - loxicmd get lb -o wide ##### Get detailed information
 - loxicmd get lb -o json ##### Get info in json



II. LOXILB 명령어

❖ Loxilb 실행과 구성

- **Configure load-balancer rule: Simple NAT44 tcp (round-robin) load-balancer**
 - `loxicmd create lb 1.1.1.1 --tcp=1828:1920 --endpoints=2.2.3.4:1`
- **NAT66 (round-robin) load-balancer**
 - `loxicmd create lb 2001::1 --tcp=2020:8080 --endpoints=4ffe::1:1,5ffe::1:1,6ffe::1:1`
- **NAT64 (round-robin) load-balancer**
 - `loxicmd create lb 2001::1 --tcp=2020:8080 --endpoints=31.31.31.1:1,32.32.32.1:1,33.33.33.1:1`
- **WRR (Weighted round-robin) load-balancer (Divide traffic in 40%, 40% and 20% ratio among end-points)**
 - `loxicmd create lb 20.20.20.1 --select=priority --tcp=2020:8080 --endpoints=31.31.31.1:40,32.32.32.1:40,33.33.33.1:20`
- **Sticky end-point selection load-balancer (select end-points based on traffic hash)**
 - `loxicmd create lb 20.20.20.1 --select=hash --tcp=2020:8080 --endpoints=31.31.31.1:1,32.32.32.1:1,33.33.33.1:1`
- **Load-balancer with forceful tcp-reset session timeout after inactivity of 30s**
 - `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 --inatimeout=30`
- **Load-balancer with one-arm mode**
 - `loxicmd create lb 20.20.20.1 --tcp=2020:8080 --endpoints=100.100.100.2:1,100.100.100.3:1,100.100.100.4:1 --mode=onearm`
- **Load-balancer with fullnat mode**
 - `loxicmd create lb 88.88.88.1 --sctp=38412:38412 --endpoints=192.168.70.3:1 --mode=fullnat`
- **Load-balancer config in DSR(direct-server return) mode**
 - `loxicmd create lb 20.20.20.1 --tcp=2020:8080 --endpoints=31.31.31.1:1,32.32.32.1:1 --mode=dsr`



II. LOXILB 명령어

❖ Loxilb 실행과 구성

- Load-balancer config in DSR(direct-server return) mode

- loxicmd create lb 20.20.20.1 --tcp=2020:8080 --endpoints=31.31.31.1:1,32.32.32.1:1 --mode=dsr

- Delete a load-balancer rule

- loxicmd delete lb 1.1.1.1 --tcp=1828

Load-balancer yaml example

```
apiVersion: netlox/v1
kind: Loadbalancer
metadata:
  name: test
spec:
  serviceArguments:
    externalIP: 1.2.3.1
    port: 80
    protocol: tcp
    sel: 0
  endpoints:
    - endpointIP: 4.3.2.1
      weight: 1
      targetPort: 8080
    - endpointIP: 4.3.2.2
      weight: 1
      targetPort: 8080
    - endpointIP: 4.3.2.3
      weight: 1
      targetPort: 8080
```



II. LOXILB 명령어

❖ Endpoint

- **Get load-balancer end-point health information**

- loxicmd get ep

- **Create end-point for health probing**

- # loxicmd create endpoint IP [--desc=<desc>] [--probetype=<probetype>] [--probereq=<probereq>] [--proberesp=<proberesp>] [--probeport=<port>] [--period=<period>] [--retries=<retries>]
- loxicmd create endpoint 32.32.32.1 --desc=zone1host --probetype=http --probeport=8080 --period=60 --retries=2

- **Create end-point with https probing information**

- # loxicmd create endpoint IP [--desc=<desc>] [--probetype=<probetype>] [--probereq=<probereq>] [--proberesp=<proberesp>] [--probeport=<port>] [--period=<period>] [--retries=<retries>]
- loxicmd create endpoint 32.32.32.1 --desc=zone1host --probetype=https --probeport=8080 --probereq="health" --proberesp="OK" --period=60 --retries=2

- **Delete end-point information**

- loxicmd delete endpoint 31.31.31.31

Note: loxilb requires CA certificate for TLS connection and private certificate and private key for mTLS connection. Admin can keep a common(default) CA certificate for all the endpoints at "/opt/loxilb/cert/rootCA.crt" or per-endpoint certificates can be kept as "/opt/loxilb/cert/<IP>/rootCA.crt", private key must be kept at "/opt/loxilb/cert/server.key" and private certificate at "/opt/loxilb/cert/server.crt". Please see [Minica](#) or [Certstrap](#) or [this](#) CICD test case to know how to generate certificates.

Endpoint yaml example

```
apiVersion: netlox/v1
kind: Endpoint
metadata:
  name: test
spec:
  hostName: "Test"
  description: string
  inactiveRetries: 0
  probeType: string
  probeReqUrl: string
  probeDuration: 0
  probePort: 0
```



II. LOXILB 명령어

❖ Session

- **Get Session information**

- loxicmd get session

- **Create Session information**

- #loxicmd create session <userID> <sessionIP> --accessNetworkTunnel=<TeID>:<TunnelIP> --coreNetworkTunnel=<TeID>:<TunnelIP>
- loxicmd create session user1 192.168.20.1 --accessNetworkTunnel=1:1.232.16.1 coreNetworkTunnel=1:1.233.16.1

- **Delete Session information**

- loxicmd delete session user1

userID(string): User Ident\ sessionIP(string): Session IPaddress\ accessNetworkTunnel(string): accessNetworkTunnel has pairs that can be specified as 'TeID:IP'\ coreNetworkTunnel(string): coreNetworkTunnel has pairs that can be specified as 'TeID:IP'

Session yaml example

```
apiVersion: netlox/v1
kind: Session
metadata:
  name: test
spec:
  ident: user1
  sessionIP: 88.88.88.88
  accessNetworkTunnel:
    TeID: 1
    tunnelIP: 11.11.11.11
  coreNetworkTunnel:
    TeID: 1
    tunnelIP: 22.22.22.22
```



II. LOXILB 명령어

❖ SessionUICI

- **Get SessionUICI information**
 - loxicmd get sessionulcl
- **Create SessionUICI information**
 - #loxicmd create sessionulcl <userID> --ulclArgs=<QFI>:<ulclIP>,...
 - loxicmd create sessionulcl user1 --ulclArgs=16:192.33.125.1
- **Delete SessionUICI information**
 - loxicmd delete sessionulcl --ulclArgs=192.33.125.1

userID(string): User Ident\ ulclArgs(string): Port pairs can be specified as 'QFI:UICIIP'

ulclArgs(string): UICI IP address can be specified as 'UICIIP'. It don't need QFI.

SessionUICI yaml example

```
apiVersion: netlox/v1
kind: SessionULCL
metadata:
  name: test
spec:
  ulclIdent: user1
  ulclArgument:
    qfi: 11
    ulclIP: 8.8.8.8
```



II. LOXILB 명령어

22

❖ IPaddress

- **Get IPaddress information**
 - loxicmd get ip
- **Create IPaddress information**
 - #loxicmd create ip <DeviceIPNet> <device>
 - loxicmd create ip 192.168.0.1/24 eno7
- **Delete IPaddress information**
 - #loxicmd delete ip <DeviceIPNet> <device>
 - loxicmd delete ip 192.168.0.1/24 eno7

*DeviceIPNet(string): Actual IP address with mask \ device(string): name of the related device *

IPaddress yaml example

```
apiVersion: netlox/v1
kind: IPaddress
metadata:
  name: test
spec:
  dev: eno8
  ipAddress: 192.168.23.1/32
```



II. LOXILB 명령어

23

❖ FDB

- **Get FDB information**
 - loxicmd get fdb
- **Create FDB information**
 - #loxicmd create fdb <MacAddress> <DeviceName>
 - loxicmd create fdb aa:aa:aa:aa:bb:bb eno7
- **Delete FDB information**
 - #loxicmd delete fdb <MacAddress> <DeviceName>
 - loxicmd delete fdb aa:aa:aa:aa:bb:bb eno7

*MacAddress(string): mac address \ DeviceName(string): name of the related device *

FDB yaml example

```
apiVersion: netlox/v1
kind: FDB
metadata:
  name: test
spec:
  dev: eno8
  macAddress: aa:aa:aa:aa:aa:aa
```



II. LOXILB 명령어

❖ Route

- **Get Route information**
 - loxicmd get route
- **Create Route information**
 - #loxicmd create route <DestinationIPNet> <gateway>
 - loxicmd create route 192.168.212.0/24 172.17.0.254
- **Delete Route information**
 - #loxicmd delete route <DestinationIPNet>
 - loxicmd delete route 192.168.212.0/24

DestinationIPNet(string): Actual IP address route with mask\ gateway(string): gateway information if any

Route yaml example

```
apiVersion: netlox/v1
kind: Route
metadata:
  name: test
spec:
  destinationIPNet: 192.168.30.0/24
  gateway: 172.17.0.1
```



II. LOXILB 명령어

❖ Neighbor

- **Get Neighbor information**

- loxicmd get neighbor

- **Create Neighbor information**

- #loxicmd create neighbor <DeviceIP> <DeviceName> [--macAddress=aa:aa:aa:aa:aa:aa]

- loxicmd create neighbor 192.168.0.1 eno7 --macAddress=aa:aa:aa:aa:aa:aa

- **Delete Neighbor information**

- #loxicmd delete neighbor <DeviceIP> <device>

- loxicmd delete neighbor 192.168.0.1 eno7

*DeviceIP(string): The IP address\ DeviceName(string): name of the related device\
macAddress(string): resolved hardware address if any*

Neighbor yaml example

```
apiVersion: netlox/v1
kind: Neighbor
metadata:
  name: test
spec:
  dev: eno8
  macAddress: aa:aa:aa:aa:aa:aa
  ipAddress: 192.168.23.21
```



II. LOXILB 명령어

❖ Vlan

- **Get Vlan and Vlan Member information**

- loxicmd get vlan
- loxicmd get vlanmember

- **Create Vlan and Vlan Member information**

- #loxicmd create vlan <Vid>
- loxicmd create vlan 100
- #loxicmd create vlanmember <Vid> <DeviceName> --tagged=<Tagged>
- loxicmd create vlanmember 100 eno7 --tagged=true
- loxicmd create vlanmember 100 eno7

- **Delete Vlan and Vlan Member information**

- #loxicmd delete vlan <Vid>
- loxicmd delete vlan 100
- #loxicmd delete vlanmember <Vid> <DeviceName> --tagged=<Tagged>
- loxicmd delete vlanmember 100 eno7 --tagged=true
- loxicmd delete vlanmember 100 eno7

Vid(int): vlan identifier

Vid(int): vlan identifier \ **DeviceName(string):** name of the related device \ **tagged(boolean):** tagged or not (default is false) \

Vlan yaml example

```
apiVersion: netlox/v1
kind: Vlan
metadata:
  name: test
spec:
  vid: 100
```

Vlan Member yaml example

```
apiVersion: netlox/v1
kind: VlanMember
metadata:
  name: test
  vid: 100
spec:
  dev: eno8
  Tagged: true
```



II. LOXILB 명령어

❖ Vxlan

- **Get Vxlan and Vxlan Peer information**
 - loxicmd get vxlan
 - loxicmd get vxlanpeer
- **Create Vxlan and Vxlan Peer information**
 - #loxicmd create vxlan <VxlanID> <EndpointDeviceName>
 - loxicmd create vxlan 100 eno7
 - #loxicmd create vxlanpeer <VxlanID> <PeerIP>
 - loxicmd create vxlan-peer 100 30.1.3.1
- **Delete Vxlan and Vxlan Peer information**
 - #loxicmd delete vxlan <VxlanID>
 - loxicmd delete vxlan 100
 - #loxicmd delete vxlanpeer <VxlanID> <PeerIP>
 - loxicmd delete vxlan-peer 100 30.1.3.1

VxlanID(int): Vxlan Identifier\ EndpointDeviceName(string): VTEP Device name(It must have own IP address for peering.)\

VxlanID(int): Vxlan Identifier\ PeerIP(string): Vxlan peer device IP address\

Vxlan yaml example

```
apiVersion: netlox/v1
kind: Vxlan
metadata:
  name: test
spec:
  epIntf: eno8
  vxlanID: 100
```

Vxlan Peer yaml example

```
apiVersion: netlox/v1
kind: VxlanPeer
metadata:
  name: test
  vxlanID: 100
spec:
  peerIP: 21.21.21.1
```



II. LOXILB 명령어

❖ Firewall

• Get Firewall information

- loxicmd get firewall

• Create Firewall information

- #loxicmd create firewall --firewallRule=<ruleKey>:<ruleValue>, [--allow] [--drop] [--trap] [--redirect=<PortName>] [--setmark=<FwMark>]
- loxicmd create firewall --firewallRule="sourceIP:1.2.3.2/32,destinationIP:2.3.1.2/32,preference:200" --allow
- loxicmd create firewall --firewallRule="sourceIP:1.2.3.2/32,destinationIP:2.3.1.2/32,preference:200" --allow --setmark=10
- loxicmd create firewall --firewallRule="sourceIP:1.2.3.2/32,destinationIP:2.3.1.2/32,preference:200" --drop
- loxicmd create firewall --firewallRule="sourceIP:1.2.3.2/32,destinationIP:2.3.1.2/32,preference:200" --trap
- loxicmd create firewall --firewallRule="sourceIP:1.2.3.2/32,destinationIP:2.3.1.2/32,preference:200" --redirect=hs1

• Delete Firewall information

- #loxicmd delete firewall --firewallRule=<ruleKey>:<ruleValue>
- loxicmd delete firewall --firewallRule="sourceIP:1.2.3.2/32,destinationIP:2.3.1.2/32,preference:200"

firewallRule sourceIP(string) - Source IP in CIDR notation\ **destinationIP(string)** - Destination IP in CIDR notation\ **minSourcePort(int)** - Minimum source port range\ **maxSourcePort(int)** - Maximum source port range\ **minDestinationPort(int)** - Minimum destination port range\ **maxDestinationPort(int)** - Maximum destination port range\ **protocol(int)** - the protocol\ **portName(string)** - the incoming port\ **preference(int)** - User preference for ordering\

Firewall yaml example

```
apiVersion: netlox/v1
kind: Firewall
metadata:
  name: test
spec:
  ruleArguments:
    sourceIP: 192.169.1.2/24
    destinationIP: 192.169.2.1/24
    preference: 200
  opts:
    allow: true
```



II. LOXILB 명령어

❖ Mirror

- **Get Mirror information**

- loxicmd get mirror

- **Create Mirror information**

- #loxicmd create mirror <mirrorIdent> --mirrorInfo=<InfoOption>:<InfoValue>,... --targetObject=attachement:<port1,rule2>,mirrObjName:<ObjectName>
- loxicmd create mirror mirr-1 --mirrorInfo="type:0,port:hs0" --targetObject="attachement:1,mirrObjName:hs1"

- **Delete Mirror information**

- #loxicmd delete mirror <mirrorIdent>
- loxicmd delete mirror mirr-1

mirrorIdent(string): Mirror identifier\ **type(int)** : Mirroring type as like 0 == SPAN, 1 == RSPAN, 2 == ERSPAN\
port(string) : The port where mirrored traffic needs to be sent\
vlan(int) : for RSPAN we may need to send tagged mirror traffic\
remoteIP(string) : For ERSPAN we may need to send tunnelled mirror traffic\
sourceIP(string) : For ERSPAN we may need to send tunnelled mirror traffic\
tunnelID(int) : For ERSPAN we may need to send tunnelled mirror traffic\

Mirror yaml example

```
apiVersion: netlox/v1
kind: Mirror
metadata:
  name: test
spec:
  mirrorIdent: mirr-1
  mirrorInfo:
    type: 0
    port: eno1
  targetObject:
    attachment: 1
    mirrObjName: eno2
```



II. LOXILB 명령어

❖ Policy

- **Get Policy information**

- loxicmd get policy

- **Create Policy information**

- #loxicmd create policy IDENT --rate=<Peak>:<Committed> --target=<ObjectName>:<Attachment> [--block-size=<Excess>:<Committed>] [--color] [--pol-type=<policy type>]
- loxicmd create policy pol-hs0 --rate=100:100 --target=hs0:1
- loxicmd create policy pol-hs1 --rate=100:100 --target=hs0:1 --block-size=12000:6000
- loxicmd create policy pol-hs1 --rate=100:100 --target=hs0:1 --color
- loxicmd create policy pol-hs1 --rate=100:100 --target=hs0:1 --color --pol-type 0

- **Delete Policy information**

- #loxicmd delete policy <Polident>
- loxicmd delete policy pol-hs1

rate(string): Rate pairs can be specified as 'Peak:Committed'. rate unit : Mbps\ **block-size(string):** Block Size pairs can be specified as 'Excess:Committed'. block-size unit : bps\ **target(string):** Target Interface pairs can be specified as 'ObjectName:Attachment'\ **color(booleant):** Policy color enable or not\ **pol-type(int):** Policy traffic control type. 0 : TrTCM, 1 : SrTCM\

Policy yaml example

```
apiVersion: netlox/v1
kind: Policy
metadata:
  name: test
spec:
  policyIdent: pol-eno8
  policyInfo:
    type: 0
    colorAware: false
    committedInfoRate: 100
    peakInfoRate: 100
  targetObject:
    attachment: 1
    polObjName: eno8
```



II. LOXILB 명령어

❖ Session Recorder

- **Set n-tuple policy for recording**
 - `loxicmd create firewall --firewallRule="destinationIP:31.31.31.0/24,preference:200" --allow --record`
- **Check or record with tcpdump**
 - # Any valid tcpdump option can be used including saving to a pcap file
 - `tcpdump -i llb0 -n`
- **Get live connection-track information**
 - `loxicmd get contrack`
- **Get port-dump information**
 - `loxicmd get port`
- **Save all loxilb's operational information in DBStore**
 - # ** This will ensure that whenever loxilb restarts, it will start with last saved state from DBStore
 - `loxicmd save -a`



II. LOXILB 명령어

❖ Configure loxicmd with yaml (Beta)

• Command

- #loxicmd apply -f <file.yaml>
- #loxicmd delete -f <file.yaml>
- loxicmd apply -f lb.yaml
- loxicmd delete -f lb.yaml

The loxicmd support yaml based configuration. The format is same as Kubernetes. This beta version support only one configuraion per one file. That means "Do not use --- in yaml file." . It will be supported at next release.

Policy yaml example

```
apiVersion: netlox/v1
kind: Loadbalancer
metadata:
  name: load
spec:
  serviceArguments:
    externalIP: 123.123.123.1
    port: 80
    protocol: tcp
    sel: 0
  endpoints:
  - endpointIP: 4.3.2.1
    weight: 1
    targetPort: 8080
  - endpointIP: 4.3.2.2
    weight: 1
    targetPort: 8080
  - endpointIP: 4.3.2.3
    weight: 1
    targetPort: 8080
```





**THANK
YOU**